Sustainable Cloud Computing

by

Zahra Abbasi

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved September 2014 by the
Graduate Supervisory Committee:

Sandeep K. S. Gupta, Chair
Chaitali Chakrabarti
Aviral Shrivastava
Carole-Jean Wu

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

Energy consumption of the data centers worldwide is rapidly growing fueled by ever-increasing demand for Cloud computing applications ranging from social networking to e-commerce. Understandably, ensuring energy-efficiency and sustainability of Cloud data centers without compromising performance is important for both economic and environmental reasons. In order to achieve these objectives, this dissertation develops a cyber-physical multi-tier server and workload management architecture which operates at the local and the global (geo-distributed) data center level. This dissertation devises optimization frameworks for each tier to optimize energy consumption, energy cost and carbon footprint of the data centers. The proposed solutions are aware of various energy management tradeoffs that manifest due to the cyber-physical interactions in data centers, while providing provable guarantee on the solutions' computation efficiency and energy/cost efficiency. The local data center level energy management takes into account the impact of server consolidation on the cooling energy, avoids cooling-computing power tradeoff, and optimizes the total energy (computing and cooling energy) considering the data centers' technology trends (servers' power proportionality and the cooling power efficiency). The global data center level cost management explores the diversity of the data centers to minimize the utility cost while satisfying the carbon cap requirement of the Cloud and while dealing with the adversity of the prediction error on the data center parameters. Finally, the synergy of the local and the global data center energy and cost optimization is shown to help towards achieving carbon neutrality (net-zero) in a cost efficient manner.

To my parents and family.

ACKNOWLEDGMENTS

I sincerely thank my advisor, Dr. Sandeep K. S. Gupta, for his valuable guidance, persistent encouragement, and support during my PhD study at Arizona State University. This thesis wouldn't have been completed without his expertise to guide me through the whole dissertation process. His foresight and thoughtful feedback has greatly contributed to the development of ideas in this thesis. His constant attempts to motivate me to work hard has helped tremendously in improving my research skills, technical writing and presentation skills. I specially thank him for the countless hours of revisions and advice on my thesis and publications.

I thank the other committee members, Dr. Chaitali Chakrabarti, Dr. Aviral Shrivastava, and Dr. Carole-Jean Wu for their advice on my research, and their time and effort to help me fulfill the degree requirements.

I thank Dr. Georgios Varsamopoulos who has been a great guide for the most part of my doctoral study. He is definitely someone I found it a pleasure to work with. He was always there for me when I needed advice on my research and help for Linux and LaTeX. I especially appreciate his constructive criticism on my research, and his support on improving the technical writing of my thesis.

I would also like to thank all the past and the present members of the Impact Lab Tridib Mukherjee, Ayan Banerjee, Madhurima Pore, Priyanka Bagade, Joseph Milazzo, Joshua Ferguson, Sayan Kole, Koosha Sadeghi and Babak Esmaeili for making my PhD experience a memorable one. In particular, I thank Ayan Banerjee who was always available with positive encouraging advice and guidance.

I also thank all my friends, Moeed Haghnevis, Yasaman khodadadegan, Shahrzad Shirazipourazad, Zahra Derakhashandeh, Mitra Eftekhar, Bahareh Aazarnoush, Mahdi Hamzeh, Tahoora Nazer, and Reza Zafarani for their encouragement, support and

making me feel at home while I was away.

I thank the National Science Foundation (through grants CRI-0855527, CNS-0834797, CNS-0855277, and CNS-1218505), and Intel Corp. for funding this research.

Finally, I sincerely thank my husband Mohammad Ali Abbasi, my daughter and my parents, for their love, understanding, encouragement, support and much more. I would also like to thank my uncle Esmaeil Abbasi whose support and encouragement helped me to pursue my studies.

Thank you all!

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

This dissertation addresses the problem of energy usage and energy cost minimization in single (local data center level) and multiple geographically distributed Internet data centers (global data center level) towards achieving carbon neutrality. The dissertation designs software-based workload management schemes integrated with server power control and energy buffering solutions for data centers. The focus is on Internet services such as those provided publicly by Internet providers (e.g., Amazon and Microsoft or enterprise private services provided by the clouds). This chapter motivates the research problems, gives an overview of the research challenges, the solutions and the contributions.

1.1 Motivation

Internet Data Centers (IDCs) are rapidly expanding with hundreds of thousands of servers to accommodate the enormous growth of Internet and cloud services. Large data centers require huge energy to power and cool their servers. It has been shown that the electricity used by data centers worldwide is increased by about 56% from 2005 to 2010 [63] (see Fig. 1.1(a)). This value is about 36% in USA [63] (see Fig. 1.1(b)). Also a recent survey of 300 North American corporations from Digital Reality Trust reports that the data centers' energy consumption is increased by 24% from 2011-2013 [68]. Further, the total electricity used in data centers is accounted between 1.7-2.2% of the total electricity use in USA, as shown in Fig. 1.1(b) [63].

This enormous energy consumption translates into huge monthly operational energy cost (utility bill, $) and large carbon footprints (the amount of greenhouse gases

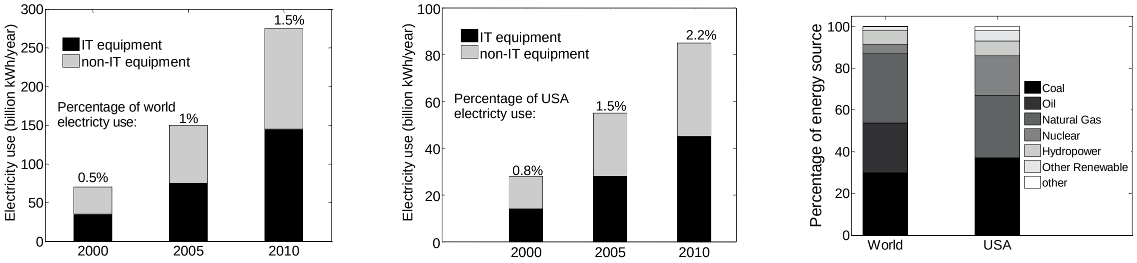
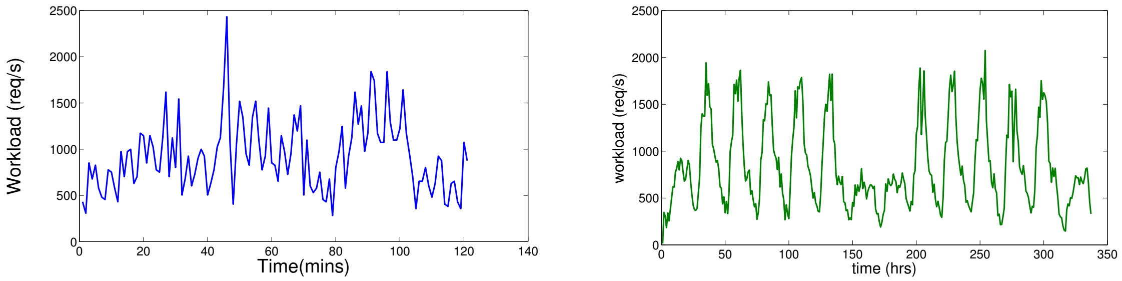


Figure 1.1: Electricity Use Growth of Data Centers Over Years in **(a)** World, and **(b)** USA (data from [62]). **(c)** Energy Source in World and USA in 2012 (data from http://www.eia.gov/). According to the figures the energy consumption of data centers is growing and so is their carbon footprint, further non-IT equipment (e.g., cooling systems) are significant contributors in the data center energy consumption.

produced by the utilities which is typically given in tons of CO_2 per year[1]). The reason is that most of the electricity produced in USA and in most other countries comes from non-renewable energy sources (see Fig. 1.1(c)). According to a report by Intel and Microsoft, the energy cost accounts for over 10% of the total cost of ownership of a data center [64]. Further, McKinsey & Company study estimates Carbon dioxide emissions from data centers will quadruple to exceed emissions from the airline industry by 2020 [114].

Therefore, there has been increasing pressures on data center operators to decrease their data center energy consumption, cost and carbon footprints. Environmental activists, in particular, have asked data center operators to give priority to renewable energy as their energy sources [34]. In addition, governments and organizations around the worlds such as European Union Emission Trading System (EU ETS) impose carbon footprint capping policies and provide economic incentives for decreasing

[1] Greenhouse gases consists of carbon dioxide (CO_2) and other gases such as water vapor (H_2O), methane (CH_4), and nitrous oxide (N_2O). For simplicity, however, carbon footprint is often expressed in terms of the amount of carbon dioxide or its equivalent of other greenhouse gases emitted.



(a) (b)

Figure 1.2: Workload Trace from NASA 1995 Trace [1]: **(a)** Minute Variation of Workload, and **(b)** Hourly Variation of Workload.

carbon footprints and increasing renewable energy use in data centers.

In response, this dissertation seeks software based **server and workload management** techniques towards achieving energy and cost sustainability in data centers and clouds. Energy sustainability requires carbon neutrality (net-zero), whereas cost sustainability requires reduced (affordable) energy cost. The keys to achieving these targets are to increase the data centers' energy efficiency and to match the energy consumption to the available green and low-cost energy sources. To this end, this research seeks a holistic energy management solution which accounts for (i) increasing the energy efficiency of data centers through thermal aware server consolidation (i.e., adjusting the active server set to the input workload and the cooling power demand) and workload management (i.e., workload distribution among active servers to reduce data center energy consumption), and (i) increasing the matching of the energy consumption to the available green and low-cost energy sources through global workload management (i.e., intelligently distributing the workload across data centers according to their electricity price ($/J) and carbon emission factor (CO_2 g/J) at a given time) and energy buffering. The effectiveness of the first method is based on (a) low average utilization of data centers [24]; this is because of the workload

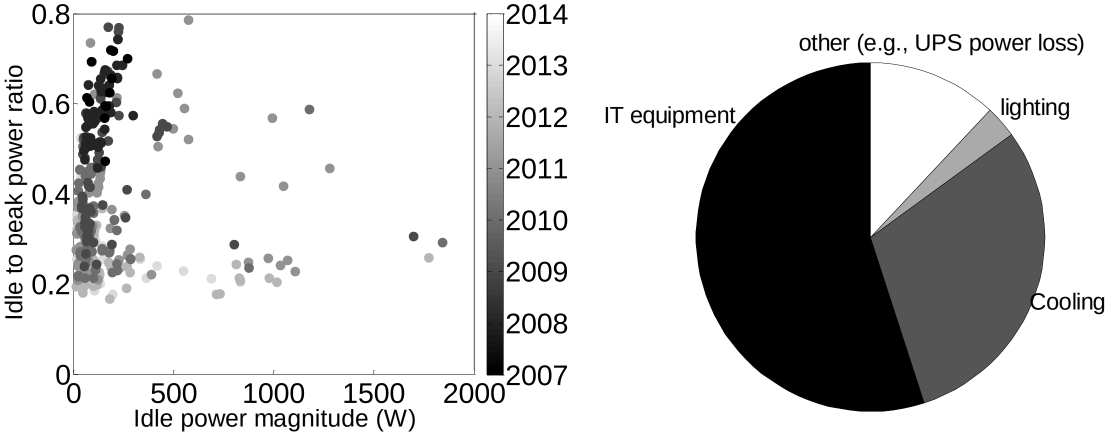
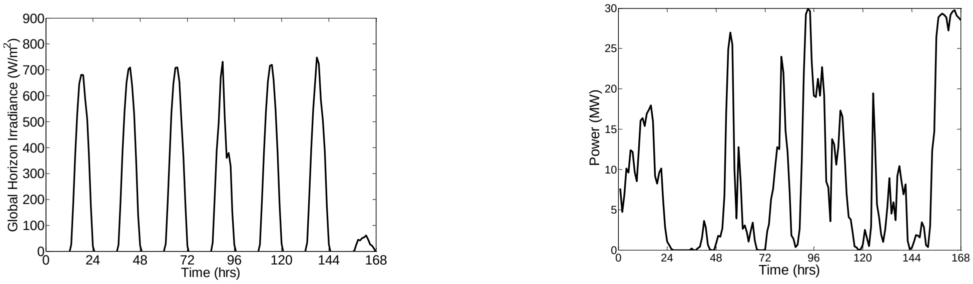


Figure 1.3: **(a)** Idle to Peak Power Ratio versus Idle Power Magnitude of Computing Systems, color-coded by release date, based on SPECPower ssj2008's benchmark data (http://www. spec.org/power_ssj2008/results/); idle power of recent systems is around 20%-60% of their peak power consumption, and **(b)** Power Consumption Break Down of Data Centers (Source: EPA 2007 report to congress on server and data center energy efficiency [42]); cooling power is a significant portion of data centers total power consumption.

intensity difference between low periods and workload peaks which is normally about two to three times as intense (see Figs. 1.2), (b) current computing systems consume significant amount of power when idle compared to being turned off or throttled down as shown in Fig. 1.3(a), and (c) cooling power is a significant portion of a data center total power consumption (see Fig. 1.3(b)) and that the total energy consumption of data centers can be reduced when considering the impact of server consolidation on the cooling power. The effectiveness of the second method is based on the spatio and temporal variation of the available renewable energy sources, the electricity prices and the carbon emission factors across different locations as shown in Figs 1.4 and 1.5, respectively.

However, the above solutions should be designed by taking into consideration sev-



(a)

(b)

Figure 1.4: Hourly Variation of Wind and Solar Renewable Energy Traces for Two Centers in USA [2, 112]: **(a)** Solar Energy Trace, and **(b)** Wind Energy Trace.

eral energy and cost management tradeoffs, each of which introduce complexity in the solution in terms of modeling, computation complexity, and curse of dimensionality (solution challenges due to the high state space of the problem) as described in the section below.

1.2 Overview of Requirements and Challenges

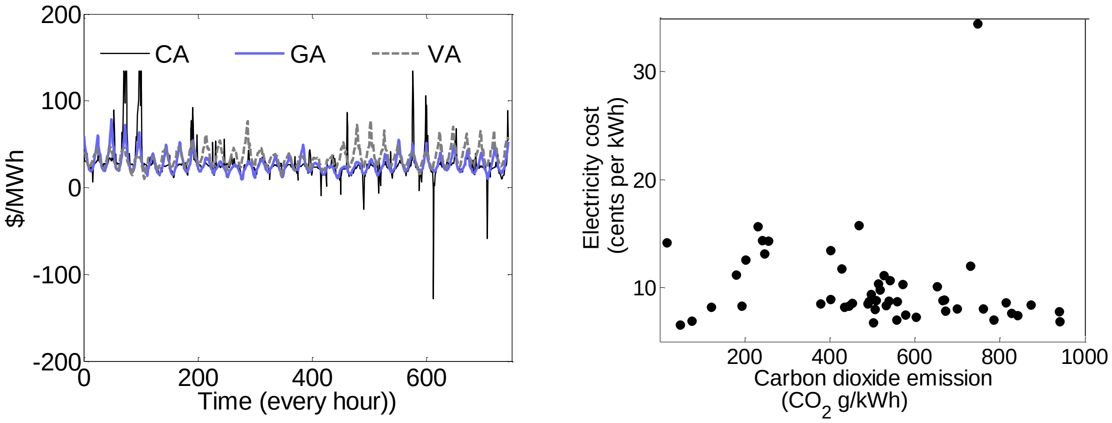
This section gives an overview on the data center circumstances and the requirements which this dissertation takes into consideration to design energy and cost efficient solutions.

Cooling and computing power tradeoff at the local data center level: A data center power consumption consists of power to operate servers (computing power), cooling system (cooling power) and other accessories such as lighting which are usually insignificant. The cooling system removes the heat dissipated by the servers and its efficiency partially depends on the data center room layout. The cooling power efficiency of a data center can be evaluated using *Power Usage Effectiveness* (PUE), which is defined as the total power over the computing power [99]. A large PUE is a strong indication of large cooling power, since the cooling system is the biggest consumer of the non-computing power in a data center (followed by power conversion and other losses) as shown in Fig. 1.3(b). According to the recent Uptime Institute report [106], the average PUE of data centers is around 1.65 which means that $0.65/1.65 \simeq 40\%$ of the power is consumed in cooling the data center.

For data centers with high PUE, server and workload consolidation may not always be *effective* or *sufficient* in reducing energy consumption because of the *cooling–computing power tradeoff*: consolidating the workload on fewer servers tends to decrease the computing power (since servers are not energy-proportional) but also may create hot spots which typically demand greater cooling power [44]. Moreover, in some cases, this cooling power increase may outweigh the computing power decrease.

It is possible to design thermal aware server and workload management schemes to avoid the cooling-computing power tradeoff [16, 44, 117]. In general, the active server set selection affects the total power of the data center due to the non-uniform temperature distribution in the room (because servers do not equally impact the temperature in the room, nor are the airflow patterns symmetric) and the servers' heterogeneity in terms of their power and computing performance. Further, It is necessary to characterize the conditions where a non thermal aware server management scheme causes cooling computing power tradeoff. The reason is that under such a condition, a non thermal aware server consolidation scheme not only may yield reduced energy savings, it may well increase the energy consumption instead.

Energy cost optimization at the global data center level: Cloud computing paradigm offers a large pool of computation and network resources from geo-distributed data centers to serve various applications. For large scale Internet applications, the cloud geo-distributed infrastructure is utilized to ensure high reliability, scalability and low access delay. The cloud's overall energy cost can be lowered by leveraging the spatio and temporal variation of electricity prices and energy efficiencies



(a)

(b)

Figure 1.5: **(a)** Spatio and Temporal Variation of Hourly Electricity Price across some States in USA in August, 2012., and **(b)** Average Carbon Dioxide Emission versus Average Electricity Cost across USA States in 2012, (each circle represents a state and data are taken from http://www.eia.gov/).

of the participating data centers in the cloud. In this scenario, workload distribution policies of the proxies and the workload distribution units (e,g, web front-ends) are designed not only to maintain the quality of service but also to reduce the electricity cost across the cloud. However, such a global workload management solution needs to be aware of the data centers' cost efficiency metrics, the live migration cost for the stateful applications, and the applications' performance requirements. The challenge of this problem is twofolds: (i) modeling the live migration overhead, and (ii) finding a computation-efficient solution given the problem's NP-hardness. Cost aware server and workload management is proposed in some recent work [96, 97]. However, a cost-optimal solution to the workload and server management across data centers has not been proposed, nor has a study on the approximation ratio of the polynomial-time heuristics has been performed.

Joint energy cost and Carbon footprint optimization at the global data

center level: To commit their responsibility to the environment, and to operate under carbon capping policies mandated by the governments, data centers seek cost efficient solutions to achieve carbon neutrality. Examples include big players such as Google and Microsoft who have already taken measures for achieving carbon neutrality (a.k.a. net-zero) [3]. Carbon neutrality can be achieved by capping the carbon footprint of data centers and purchasing carbon credits (e.g., Renewable Energy Certificates that data centers can purchase to contribute in the growth of renewable energy industry) for the remaining offset. Data centers get their primary power from the grid, and their carbon emission factor in unit of CO_2 g/J mainly depends on the grid fuel types. Various parameters such as the availability of fuel type, the market, and the environment affect both the carbon emission factor and the electricity price such that both vary over time and location (e.g., see Fig. 1.5(a)). However, there is not always positive correlation between them in different locations, e.g., across USA as shown in Fig. 1.5(b). This means that, optimizing the energy cost does not necessarily results optimizing of the data centers' carbon footprint. Further, due to the intermittent nature of the available renewable energy as well as the time-varying nature of the workload, the carbon footprint cap is typically defined for a long term operation of a data center (e.g., a year). Therefore, a global workload management solution that can both optimally manage the electricity cost and satisfy the carbon footprint capping target of a cloud can only be found offline. However, it is natural that an offline solution is impractical due to the "curse of dimensionality", and that it is based on the availability of the data center input information (e.g., workload) for a long term. In partciular, the traditional approaches to construct optimal policies to manage energy cost and carbon footprint dynamics involve the use of Markov Decision Theory and Dynamic Programming [26, 116]. It is well known that these techniques suffer from the curse of dimensionality where the optimal strategy computing complexity exponentially grows with the system size [105]. This is because the problem state space depends not only on the system size (e.g., number of variables) but also on the possible values that the system parameters get depending on the system sate. Hence, a challenging task is to design an online solution, which does not have access to the entire future information, and yet competitively minimizes the cost under the cloud's carbon cap requirement with respect to the offline solution. Some of the related work solely focus on either cost minimization or carbon footprint capping [96, 97], and some others design heuristic solutions which manage the cloud's carbon cap in a best-effort manner [46, 69].

Energy buffering tradeoffs to shave peak power demand at the global data center level: Data centers spend 10 to 25 dollars per watt in provisioning their power infrastructure, regardless of the watts actually consumed [52]. Since peak power needs arise rarely, provisioning power infrastructure for them can be expensive. Further, some utilities penalize data centers for their peak power in addition to the energy they consumed. Energy buffering management using existing UPS devices in data centers or other types of energy storage devices (ESD) has been shown to be promising to either shift the peak demand away from the high tariff periods, or to shave the demand, allowing aggressive under-provisioning of the power infrastructure [22, 121]. Global workload management, can also be of significant aid to shave the peak power draw without requiring large-scale energy storage devices. Although energy buffering and global workload management have been throughly studied in the literature and in this work, the solutions designed so far are piecemeal in the sense that each of which addresses some aspects of the problem. In particular, prior work has independently considered aspects of (i) energy cost and carbon footprint reduction through an intelligent global workload distribution for geo-distributed data centers, and (i) peak power cost reduction through energy storage devices. These types of managements for energy cost reduction, peak power cost reduction and carbon capping have been investigated separately. We argue that there is a need for a holistic approach that combines all the available leverages. Accordingly, we propose a new holistic global workload management for large-scale Internet services running in geo-distributed data centers. Such a holistic management, however, introduces new challenges to the solution of the global workload management. First, peak power cost minimization, energy buffering management and carbon capping, all introduce time coupling in the solution of the global workload management. Prior online algorithms are designed to manage each (or two) of the aforementioned coupling factors separately, disregarding their implications on each other. Particularly, window based predictive scheme, efficient for online management of peak power shaving [22], fails to competitively manage carbon capping with respect to the offline solution. This is because adjusting the carbon cap for each prediction window is difficult considering the intermittent nature of the available renewable energy. We propose to use a combination of window based predictive scheme and T-slot Lyapunov optimization to jointly manage the cost (electricity cost and peak power cost) and the carbon footprint. The idea is to leverage the variability of data center parameters within the time frame T (e.g., a day) in order to smoothen the peak power draw, and utilize the technique of Lyapunov optimization to adjust the desired carbon footprint for each time frame over the entire budgeting period (e.g., a year).

Nevertheless, the efficiency of the previously discussed solution heavily depends on the prediction accuracy of data center parameters over T, those being input workload, electricity prices, the available renewables energy and the grids' carbon footprints. In particular, the prediction error has a very harmful impact on the peak power cost as observed by the related work [13]. The reason is that the optimal approach is to utilize the data centers with low electricity cost as much as possible without increasing their peak power in that time frame T. Under any under prediction of those data centers' workload, for instance, their peak power most likely increases, resulting in increase in the peak power cost. As a result, the cost efficiency of the solution decreases since the peak power cost contributes a significant portion of the data centers' operational energy cost. We seek solutions to mitigate/remove such a harmful impact of the prediction error in increasing the peak power cost. Previous prediction based schemes of peak power minimization are performed without considering the impact of the prediction error [43, 121].

Practical considerations: In addition to the aforementioned requirements, a server and workload management scheme should be designed with taking into consideration the quality of service requirements (e.g., response time/delay for Internet workload) and the nature of data center parameters. Internet applications, being the focus of this paper, are delay sensitive. In order to provide a high quality of service, the delay of end users should not exceed a threshold. This is often challenging considering the workload variability and the spikes. Similarly, data center parameters, e.g., electricity cost, workload, and on-site renewable energy have different characteristics in terms of the predictability and the stochastic nature, where the solutions should be designed accordingly. Further, in practice a distributed implementation of the global workload management is desired. Accordingly, each data center and workload distribution units (proxy or front-ends) should independently decide on the cost and workload management with the least possible information exchange in order to preserve the confidentiality and the scalability.

1.3 Overview of Results and Contributions

Motivated by the aforementioned requirements, *this research proposes workload and server management schemes to optimize the energy consumption, energy cost,* *and carbon footprint in a cloud.* The research takes the fundamental keys to achieve energy and cost sustainability altogether into consideration, i.e., energy efficiency, cost efficiency, and carbon emission efficiency, manages their associated tradeoffs and leverages their synergistic impact to achieve sustainability in a cloud.

Common themes and assumptions of the solutions: The solutions are designed through framing optimization problems of energy and cost management at both the local and the global data center levels. The optimization frameworks are made up of models characterizing the power demand and the power supply of data centers. The power demand model of data centers is derived from the workload distribution model according to their performance requirements, and the power consumption model of servers and the cooling system. The power supply model consists of models to describe the power drawn from the grid, batteries and the on-site renewable power sources of data centers. The solutions periodically and dynamically monitor data center parameters (e.g., input workload), and decide on energy and cost management polices. Due to the management overhead, the decision time interval are chosen to be relatively large (e.g., an hour). Then, the steady state variation of the data center parameters are considered over the intervals (e.g., average available renewable energy sources over an interval). The optimization frameworks at the local and the global data center level range from linear programming, nonlinear binary programing and complex offline stochastic programming depending on the different aspects of energy management that they address. The appropriate solutions, therefore, are designed corresponding to the nature of the optimization problems at each data center level to address their computation efficiency and their cost and energy efficiency. The optimization problems at the global data center level, as shown in Fig. 1.7, are incrementally extended to address various aspects of energy and cost management across data centers. Further, the underlying assumptions of the global

Table 1.1: Summary of Optimization Frameworks and Solutions.

level	problem	optimization problem	solution
Local	Problem: TACOMA (Ch. 4), Thermal aware server (TASP) and workload management (TAWD):		
	TASP, heterogen. datacenters	nonlinear binary prog.	heuristic ($O(n^3)$)
	TASP, homogen. datacenters	linear binary prog.	Greedy with proven performance ($O(n^3)$)
	TAWD	linear programming	heuristic ($O(n^2)$)
Global	Problem: Electricity cost minimization, DAHM (Ch. 5):		
	zero migration cost	integer linear prog.	linear prog. with proven performance
	non-zero migration cost	offline integer linear prog.	linear prog. (heuristic)
Global	Problem: Energy cost and carbon footprint optimization (Chs. 6 and 7):		
	zero peak power cost	offline linear prog.	one-slot Lyap. opt. with proven performance
	non-zero peak power cost	offline linear prog.	T-slot Lyap. opt. and stochastic prog.

and the local data center management solutions are consistent such that they can be integrated to optimize energy consumption, energy cost and carbon footprints all together at the local and the global data center level as discussed in Chapter 6. Table 1.1 gives an overview on the optimization problems and their solutions.

Application of the solutions: The flowchart in Fig. 1.6 gives an overview on the circumstances where the aforementioned energy and cost management tradeoffs manifest depending on the utility cost model, the spatio-temporal variation of electricity prices and carbon footprints, and the data centers' physical layout. These tradeoffs introduce challenges in designing server and workload management solutions as depicted in the figure, which are addressed throughout this dissertation. The

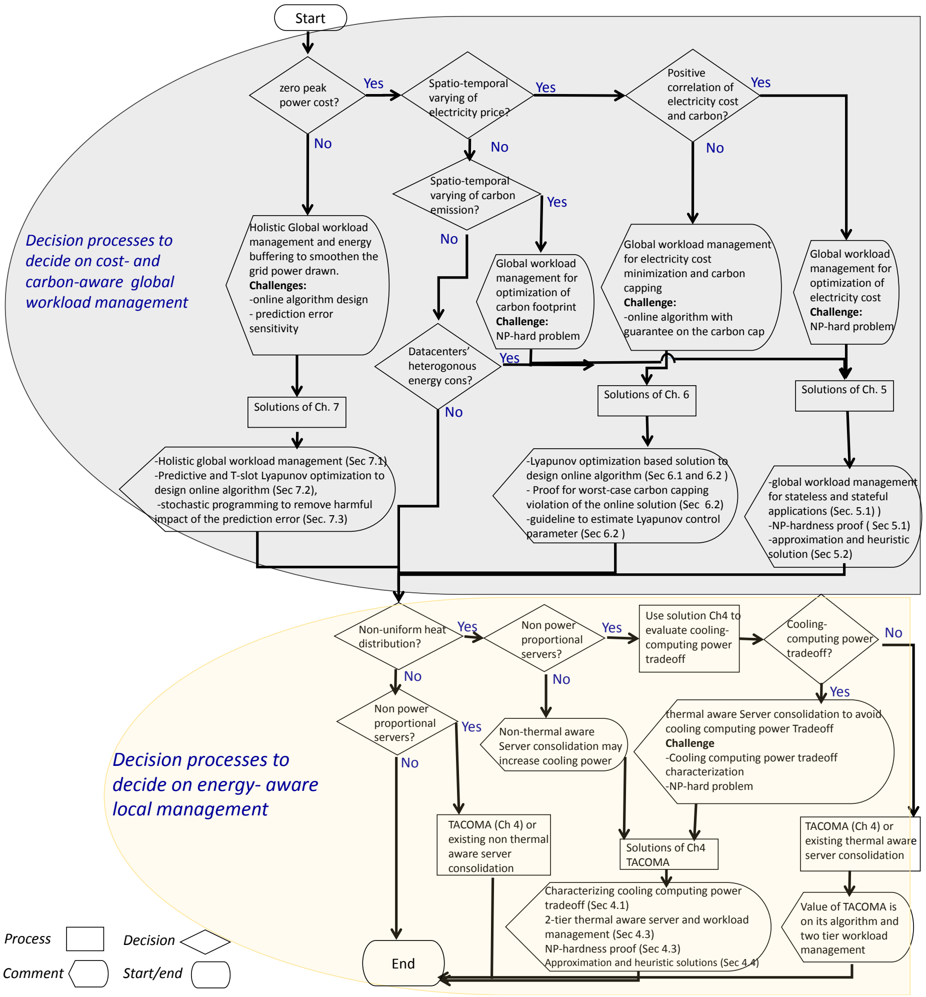

Start
zero peak power cost?
Yes
No
Spatio-temporal varying of electricity price?
Yes
No
Positive correlation of electricity cost and carbon?
Yes
No
Spatio-temporal varying of carbon emission?
Yes
No
Decision processes to decide on cost- and carbon-aware global workload management
Holistic Global workload management and energy buffering to smoothen the grid power drawn.
Challenges:
-online algorithm design
- prediction error sensitivity
Global workload management for optimization of carbon footprint
Challenge:
NP-hard problem
Global workload management for electricity cost minimization and carbon capping
Challenge:
-online algorithm with guarantee on the carbon cap
Global workload management for optimization of electricity cost
Challenge:
NP-hard problem
Datacenters' heterogonous energy cons?
Yes
No
Solutions of Ch. 7
Solutions of Ch. 6
Solutions of Ch. 5
-Holistic global workload management (Sec 7.1)
-Predictive and T-slot Lyapunov optimization to design online algorithm (Sec 7.2),
-stochastic programming to remove harmful impact of the prediction error (Sec. 7.3)
-Lyapunov optimization based solution to design online algorithm (Sec 6.1 and 6.2)
- Proof for worst-case carbon capping violation of the online solution (Sec 6.2)
-guideline to estimate Lyapunov control parameter (Sec 6.2)
-global workload management for stateless and stateful applications (Sec. 5.1))
-NP-hardness proof (Sec 5.1)
-approximation and heuristic solution (Sec 5.2)
Non-uniform heat distribution?
Yes
No
Non power proportional servers?
Yes
No
Use solution Ch4 to evaluate cooling-computing power tradeoff
Cooling-computing power tradeoff?
No
Yes
Non power proportional servers?
Yes
No
Non-thermal aware Server consolidation may increase cooling power
thermal aware Server consolidation to avoid cooling computing power Tradeoff
Challenge
-Cooling computing power tradeoff characterization
-NP-hard problem
Decision processes to decide on energy- aware local management
TACOMA (Ch 4) or existing non thermal aware server consolidation
Solutions of Ch4 TACOMA
TACOMA (Ch 4) or existing thermal aware server consolidation
Characterizing cooling computing power tradeoff (Sec 4.1)
2-tier thermal aware server and workload management (Sec 4.3)
NP-hardness proof (Sec 4.3)
Approximation and heuristic solutions (Sec 4.4)
Value of TACOMA is on its algorithm and two tier workload management
End
Process
Decision
Comment
Start/end

Figure 1.6: Decision Process for Energy-aware and Cost-aware Global and Local Data Center Management.

flowchart also summarizes the developed solutions, the challenges they addressed and their practical applications depending on the cloud and data center parameters. It starts with the decision process for the type of global workload management solution required depending on the cloud parameters. The flowchart, then, explores the appropriate local data center management solution depending on the physical layout of the data centers and servers' power proportionality.

Overview of the results: As shown in Fig. 1.7, the dissertation devices thermal

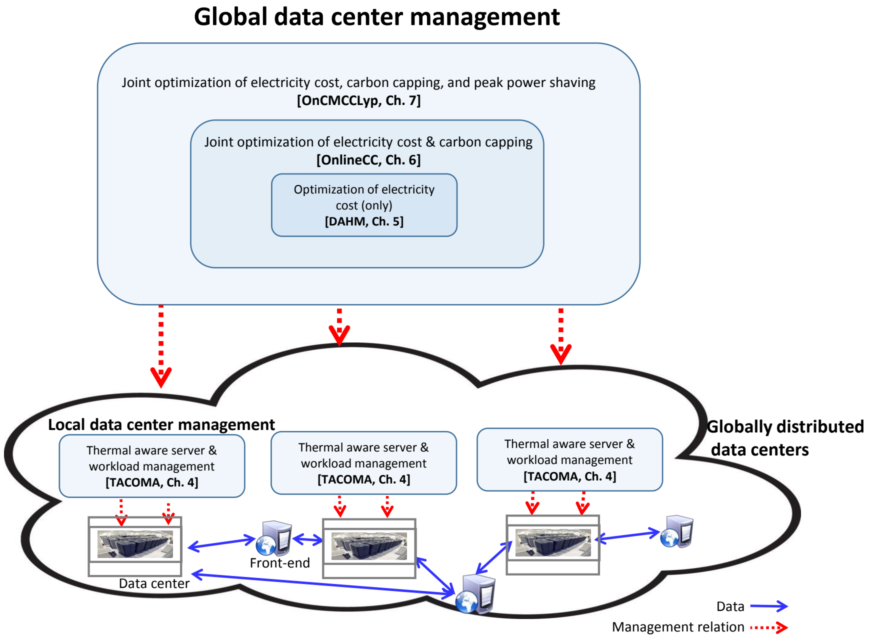

Global data center management
Joint optimization of electricity cost, carbon capping, and peak power shaving
[OnCMCCLyp, Ch. 7]
Joint optimization of electricity cost & carbon capping
[OnlineCC, Ch. 6]
Optimization of electricity cost (only)
[DAHM, Ch. 5]
Local data center management
Thermal aware server & workload management
[TACOMA, Ch. 4]
Thermal aware server & workload management
[TACOMA, Ch. 4]
Thermal aware server & workload management
[TACOMA, Ch. 4]
Globally distributed data centers
Front-end
Data center
Data
Management relation

Figure 1.7: Overview of the Proposed Solutions.

aware energy management solutions at the local data center level and global workload management solutions at the global data center level. The synergistic outcome of these two level cost energy and cost optimization is shown to help towards achieving carbon neutrality (net-zero) in a cost efficient way. To increase the data centers energy efficiency, we develop *TACOMA* (Chapter 4), a thermal aware server consolidation and workload distribution solution towards achieving power proportional data centers. TACOMA accounts for both the cooling and the computing energy in data centers, performs management in the local data center level and is shown (in a simulation study) to outperform the state of the art solutions by increasing the energy efficiency of data centers (up to 20%). Next, the dissertation studies the energy cost minimization at the global data center level which determines workload distribution policies for a set of geo distributed data centers, i.e., workload distribution between web front-ends and data centers as shown in Fig. 1.7. This problem is studied for three cases of data center energy cost models as depicted in Fig. 1.7. First, we focus on solely minimizing the energy cost, develop a solution namely, DAHM (Chapter 5), and study its effectiveness for the various cases of applications requirements using the developed greedy solutions, with analytically proven performance. The analytical results are further utilized to design linear-programming based solutions for more complex cost models. Then, we study the workload management to jointly optimize the energy cost and the carbon footprint of data centers. We develop an online solution, OnlineCC (Chapter 6), which is proven to achieve near optimal offline energy cost, while bounding the potential violation from the target carbon footprint cap. Simulation results show that OnlineCC reduces cost by more than 18% compared to a prediction-based online solution while resulting in equal or smaller carbon footprint. We also study the effectiveness of OnlineCC, when integrated with TACOMA. In this case, OnlineCC, is shown to provide a holistic solution to increase the data center energy efficiency, reduce the energy cost and the carbon footprint. Third, we extend OnlineCC to leverage the predictability of data center parameters within a time frame to efficiently shave the peak power demand, while optimizing the energy cost and the carbon footprints (Chapter 7). The solution, however, is very sensitive to the prediction error. In particular, the efficient peak power cost minimization fails in the presence of the parameters' prediction error. We use stochastic programming approach to solve it, taking into account the randomness of the parameters (Section 7.3). This solution is shown to remove up to 66% of the harmful impact of the prediction error in increasing the cost (sum of electricity cost and the peak power cost). Finally, we adapt the Alternating Direction Method of Multipliers (ADMM) to design distributed algorithms with linearly convergence for OnlineCC and its variants (Chapter 7). All of the simulation results are performed using real-world traces. Further, some small scale experiments are performed to complement the analysis.

Overview of the related work and the novelty of this dissertation: This research distinguishes itself from existing research as it analytically explores the energy management tradeoffs which manifest depending on the data center physical circumstances and depending on the server and workload management polices, models optimization problems to design server and workload management solutions which achieve the desired tradeoffs, and seeks computation-efficient solutions.

In particular, non-thermal-aware server and workload consolidation schemes may cause cooling computing power tradeoff which is also witnessed by the related work [44]. However, the existing studies are performed for a particular setting of data centers i.e., performing simulation or empirical studies for a given data center thermal profile and a given servers power efficiency. Given a wide range of power efficiency for data centers, we study the questions of under what circumstances of data centers power efficiency, a non-thermal-aware server consolidation solution causes cooling computing power tradeoff and how to avoid such a tradeoff. We devise abstract models to describe the power efficiency of non-thermal-aware server consolidation solutions and provide worst-case analysis of the impact of a non-thermal-aware server consolidation solution on the cooling and on the computing energy. Such a technique helps data center operators to decide on their server consolidation policy. We further study thermal-aware server and workload consolidation solutions to optimize the data centers' total energy consumption and ensure avoiding cooling computing power tradeoff. The existing thermal-aware scheduling algorithms for Internet data centers are heuristic in the sense that they are either based on simulation studies or do not provide guarantee on their optimality and avoiding cooling-computing power tradeoff [44, 94, 104].

Cost-aware workload and server management solutions are also studied in the related work. Existing research addresses energy cost optimization [77, 96, 97, 125], and joint optimization of energy cost and carbon footprint of data centers [39, 46, 69, 78, 100, 127]. The existing cost- aware workload management solutions lack a mathematical analysis for the optimality of the solutions. We prove that the problem is NP-hard and derive the approximation ratio of the solution. Further, the existing cost and carbon-aware workload management solutions adopt standard steps of Lyapunov optimization in order to design an online algorithm for minimizing cost without significantly violating the carbon capping requirement of data centers [78, 127]. In this regard, the worst case carbon capping violation of the online solution and the right adjustment of the Lyapunov control parameter depends on the estimation of the offline optimal solution which is not easy to obtain. We leverage the structure of the Lyapunov optimization model of the problem and derive the worst case carbon capping violation of the solution through the cloud parameters (e.g., electricity prices and carbon emission factors of data centers). The solution is also used to adjust the Lyapunov control parameter, a key parameter to control the proximity of the online solution to the optimal solution. Further, the existing work lacks a holistic solution which addresses the cost (sum of electricity cost and peak power cost) and carbon footprint reduction tradeoff all-together. We devise an online solution based on T-slot Lyapunov optimization to optimize both the electricity cost, and the peak power cost without significantly violating the carbon capping requirement of the data centers. Finally, while the existing work on data center peak power optimization rely on predictability of cloud parameters [47, 50, 51, 52, 61, 75, 116, 121], we show that the prediction error of cloud parameters such as workload has a very harmful impact on the peak power optimization, and adopt stochastic programming to remove such an impact. The following sections give a detailed overview on the contributions.

In increasing energy efficiency at the local data center level (Chapter 4): We develop Two-tier Architecture for Cooling cOmputing energy Management Architecture (TACOMA) with Thermal Aware Server Provisioning (TASP) to run at "epochs" (∼1 hr) and Workload Distribution (TAWD) to run at "slots" (∼5 sec) that matches the long term and the short term fluctuation of the web traffic. It runs online algorithms to manage servers (i.e. on/off control) and distribute workloads among active servers. In designing TACOMA we make the following contributions:

- *Formalizing conditions for the existence of cooling-computing power tradeoff.* We prove some lemmata to identify the parameters that affect the cooling–computing power tradeoff, namely energy proportionality, energy efficiency of the data center (in terms of PUE), and the size of the active server set with respect to the available servers (Chapter4, Section 4.1.2). The lemmata provide an easy-to-solve analytical method to test the occurrence of cooling-computing power tradeoff due to workload consolidation.

- *Formalizing TASP and TAWD* for both homogeneous and heterogeneous data centers. In the first tier, TASP adjusts the number of active servers to the incoming workload and chooses thermal and power efficient servers as active servers to avoid cooling-computing power tradeoff and save energy.

- *Fast heuristic solutions to TASP and TAWD problems.* Due to the NP-hardness of TASP problems (mainly because of thermal awareness and computing power awareness), heuristic solutions, namely TASP Least Recirculated Heat (TASP-LRH) and Computing Power LRH hybrid (TASP-CPLRH) are devised (Chapter4, Section 4.4). TASP-LRH, an approximation solution, runs a rank-and-sort technique using the LRH metric and TASP-CPLRH, a heuristic solution, sorts the equipment according to their computing power efficiency and then applies LRH ranking in each group of servers with the same efficiency. TAWD also employs LRH and CPLRH metrics to quickly decide on the workload distribution of the active servers.

We also perform a comprehensive simulation study to evaluate the developed schemes.

In energy cost minimization at the global data center level (Chapter 5): To perform cost efficient computing across data centers we design *Dynamic Application Hosting Management (DAHM)* which performs server and workload management across data centers. To develop DAHM we make the following contributions:

- *In formulating the DAHM problem:* DAHM is formulated as a *Mixed Integer Programming (MIP)*, and is proven to be NP-hard (Chapter5, Lemma 5.1.1) for both cases of stateful (non-zero migration overhead) and stateless applications (zero migration overhead). In the case of stateless applications, DAHM is shown to be a specific type of MIP that is Fixed-Charge Min-Cost Flow (FCMCF) problem.

- *In designing solutions to the problem*: Optimal solutions for DAHM in both stateless and stateful applications, are provided by use of branch-and-bound, which has exponential time complexity with respect to the product of the number of front-ends into the number of data centers, in the worst case. Further, polynomial-time greedy algorithms are developed that dynamically decide on the number of active servers and the workload share (Chapter5, Section 5.2). The analytical results show that the greedy solution at most increases the cost by the cost of an active idle server per data center with respect to the optimal solution (which is negligible considering the large number of active servers in data centers)

In joint optimization of electricity costs and carbon footprints at the global data center level (Chapter 6): We make use of Lyapunov optimization to devise **OnlineCC**, an online workload and server management algorithm to minimize the electricity cost while satisfying the carbon cap requirement of a set of geo-distributed data centers using only one hour ahead future information (Section 6.2). We show that **OnlineCC can get time averaged cost within O(1/V) of the offline optimal solution** (see Theorem 6.2.3), where V is Lyapunov control parameter. More importantly, we further extend the Lyapunov optimization technique to find **the maximum carbon cap violation that OnlineCC yields in the worst case**, which is within $O(V)$ (see Theorem 6.2.3). A salient feature of this bound is that it gives an estimation of the worst case carbon violation of OnlineCC without the need to solve the optimal offline solution. For data centers with non-stationary input parameters we design OnlineCC-T that leverages the predictability of data center parameters within the time frame T (Section 6.2.2). Analytical results (Theorem 6.2.4) show that OnlineCC and OnlineCC-T has a very similar performance compared to Optimal solutions. The real-world trace based simulation study shows a slightly better performance of OnlineCC-T, compared to OnlineCC, suggesting the use of this algorithm depending on the nature of data center parameters (Section 6.4.4).

We further extend and evaluate OnlineCC when integrated with TACOMA. We device Thermal-aware OnlineCC which uses a convex cost model to account for data center cooling energy as a result of applying TACOMA's active server set selection algorithm at each individual data center. The model is evaluated when using energy consumption model of an actual data center, i.e., BlueCenter (a small testbed data center at ASU). The results show that OnlineCC increases cost saving around 10% when using TACOMA's thermal aware server section algorithm (TASP-LRH) as opposed to the reference non-thermal aware server selection algorithm (Section 6.4.5).

Finally, we perform a small scale experimental study to show the effectiveness of OnlineCC in optimizing cost and carbon footprint with satisfactory performance (Section 6.5).

In peak power shaving at the global data center level (Chapter 7): We frame the holistic global workload management, energy buffering and peak power shaving problem as a linear programming (Section 7.1). The linear programming model, disregards the nonlinear constraint posed by physical characteristics of energy storage devices, yet is shown to always give a feasible solution in terms of the missing nonlinear constraint (Lemma 7.1.1). We design an online solution (which is an extension of OnlineCC), namely Cost Minimization and Carbon Footprint Capping based on Lyapunov optimization (OnCMCCLyp) which leverages T slots ahead information to smoothen the peak power draw, and Lyapunov optimization to manage the dynamics of the cloud's carbon footprint (Section 7.2). OnCMCCLyp is shown to achieve near optimal solution performance (through analytical, Theorem 7.2.1 and simulation study), when T is sufficiently large and that the information over T is accurately available. Prediction error, however, downgrades the performance of OnCMCCLyp by increasing the cost (sum of electricity cost and peak power cost) up to 45% compared to the offline optimal solution. Our stochastic programming solution (Section 7.3) is shown to remove up to 66% of such a harmful impact of the prediction error. We also design 2-block ADMM based algorithm to solve OnCMCCLyp in a distributed way which is shown to converge linearly (Section 7.4).

1.4 Dissertation Structure

In the next chapter, we give an overview on data centers' power infrastructure and applications under which we perform the study. Next, we review the related work and the way we contribute in the area. In the next four chapters (Chapters 4 to 7), we give a detailed description of the aforementioned solutions, and analyze their usefulness and drawbacks. Each of the Chapters 4, 5, 6, and 7, are self-contained, that is, they can be read in any order. In particular, Chapter 4 presents TACOMA, an energy efficient solution at the local data center level. Chapter 5 presents our energy cost minimization solution (DAHM) at the global data center level. Chapter 6 presents OnlineCC for joint optimization of energy cost and carbon footprint at the global data center level. Chapter 7 further extends OnlineCC for joint optimization of energy cost, carbon footprint, energy buffering and peak power shaving at the global data center level. Finally, we conclude the research and discuss open research problems in the domain of cloud sustainability in Chapter 8. Chapter 8, also accounts for the relationships between all of the proposed solutions and the way that all of them together result to a holistic solution toward achieving sustainability in a cloud.

Chapter 2

BACKGROUND

This chapter gives a brief overview of data centers, their power infrastructure and Internet applications under which we perform the study. Further, the chapter gives an introduction of the research system model, and the problem formulation.

2.1 Data Centers

A *data center* is a facility built to house information technology infrastructure including servers, storage systems and network equipment. Internet service providers, and enterprises use this facility in order to provide secure and reliable information technology services such as storing, processing, and managing the data as well as providing high-speed network based applications (see Fig. 2.1). Information technology operations are a crucial aspect of most organizational operations around the world. Therefore, data centers are designed in such a way that their long term uninterrupted operation is guaranteed. They employ various redundant or backup techniques in both software and hardware level to ensure their reliability. Further, they employ several air-conditioning controls and security solutions to ensure their thermal safety and security, respectively.

Data centers come in different sizes depending on their design objectives and functionalities. They range from small facilities hosting a few computers without sophisticated power and cooling system infrastructure to massive facilities hosting hundreds of thousands of servers and offering a variety of cloud services. Small data centers are employed locally by small enterprises such as universities, whereas large scale data centers such as those provided by Google, Amazon, and Facebook offer

Figure 2.1: BlueCenter, a Small Testbed Data Center at ASU, (a) Front View of Racks, and (b) Back Side of Racks.

worldwide online and cloud services. Further, there are also some data centers in between, i.e., medium data centers, such as those offering hosting services. This research focuses on medium and large data centers that require sophisticated power and cooling system infrastructure. This section gives an overview on the physical layout, the power infrastructure and the power efficiency trend of such data centers.

2.1.1 Physical Layout

In contemporary data centers, computing servers are organized in rows of racks of blade systems organized in chassis. The equipment is arranged so that, in each aisle between two rows, either front panels or back panels are facing each other; this is called the hot aisle/cold aisle arrangement. Most of data centers use air cooling technology, where the equipment is placed on a raised floor in the hot aisle/cold aisle layout (see Fig. 2.2). The raised floor in the cold aisles features perforations which allow cool air to enter the room; perforations or other contraptions above the hot aisles gather the hot air, which is passed to the computer room air conditioner (CRAC). The supplied temperature of the cooling system (CRAC) should be low enough so that the temperature of the computing nodes does not go beyond the red

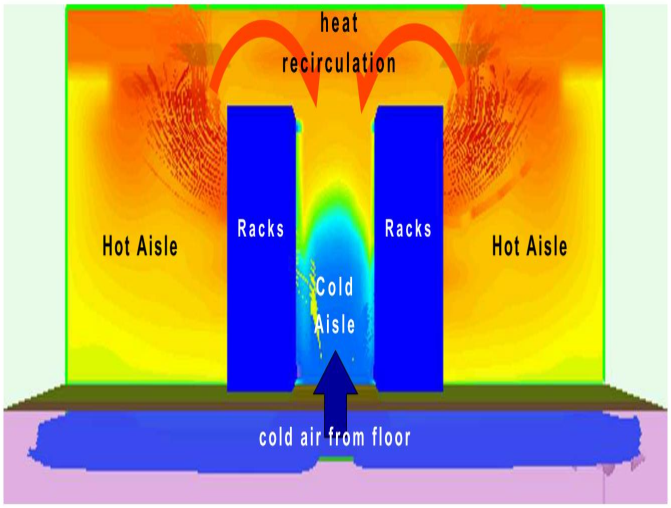


Figure 2.2: Hot Aisle Cold Aisle Data Center Layout with Heat Recirculation (figure from [110]).

line temperature which is specified by the manufacturers. In the ideal case, all the hot air should directly go back to the CRAC; but, in practice, some of the hot air recirculates back to the computing servers (see Fig. 2.2). The heat recirculation in the room is non-uniform as servers contribute or receive the heat recirculation depending on their physical location. The solutions of this dissertation account for data centers with heat recirculation.

2.1.2 Power Infrastructure

Data centers get their primary power source from grid. As shown in Fig. 2.3, power enters the data center through a utility substation which serves as its primary power source. Other power sources act as the power source backup. In particular, a Diesel Generator unit (DG) is usually used as a secondary backup power source upon a utility failure. An Automatic Transfer Switch (ATS) is employed to automatically select/switch between these two sources. DGs have a startup time of around 10-20

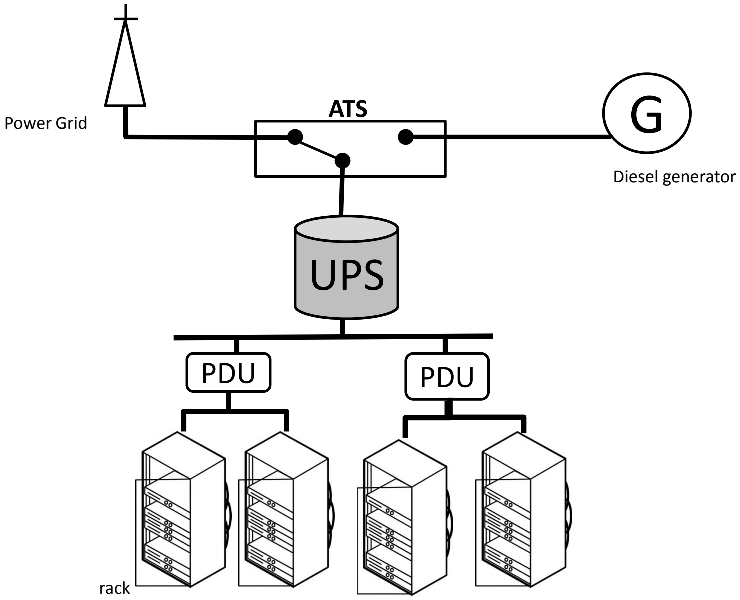


Figure 2.3: A Typical Data Center Power Infrastructure.

seconds, a time duration for DGs to get activated in order to supply power. To bridge this time gap, data centers employ Uninterrupted Power Supply (UPS) units which store energy during power availability. UPS typically can power the data center for about 10-15 minutes. UPSes are primarily installed centrally in data centers, where their power is available to all the IT equipment [61]. As shown in Fig. 2.3, in this configuration power from the UPS units is fed to several Power Distribution Units (PDUs). The PDUs have transformers that step down the voltage and route power to several racks.

Some modern data centers employ distributed UPSes for groups of chassis or for each server [61]. In the first case, the UPS will deliver power to the PDU. In the second case, the UPS directly supplies current to the server. The UPS in this case does not need the inverter and thus have lower power loss due to distortion. However, this calls for greater expenses in infrastructure.

The problems studied in this research focus on the centrally connected UPS configuration. The other configurations do not change the nature of the problems, their solutions and the results' trend, but affect the number of decision variables involved and the numbers in the results. There are some sources of power inefficiencies when utilizing UPSes, e.g., power loss due to AC-DC conversion and the power inefficiencies for frequent charging and discharging of UPSes. In this study we do not contribute on the power infrastructure design of data centers, hence we do not consider the power inefficiencies due to the type of deployment of UPS. We study the energy cost minimization problems where UPSes can be used to shave the peak power demand and reduce the electricity cost in addition to serving as the power source backup during the power outage. Therefore, we consider the UPS power inefficiencies which come from their charging/discharging.

2.1.3 Power Consumption

The power consumption of data centers depends on several factors including the power consumption characteristics of computing equipment, the cooling system, and the size of data centers in terms of number of racks, and number of servers per rack. Many other factors, such as power loss (e.g., due to AC-DC conversion), and lighting also contribute to the power consumption of data centers. This dissertation only considers the power consumption of the servers and the cooling systems, as the other contributors to the power consumption are either insignificant or irrelevant to the designed workload management schemes. The power efficiency of thermal aware server consolidation schemes, being the focus of this research, depend on how power-proportional the servers are and how significant the cooling power is. Hence, this dissertation designs the solutions considering the power proportionality trend of the servers as well the cooling power consumption trend of data centers.

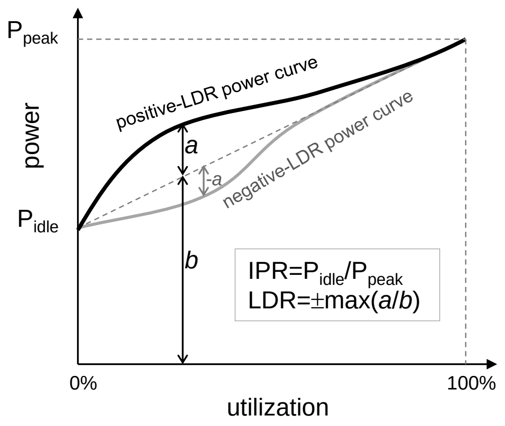


Figure 2.4: Pictorial Definition of Idle to Peak Power Ratio (IPR) and Linear Deviation Ratio (LDR).

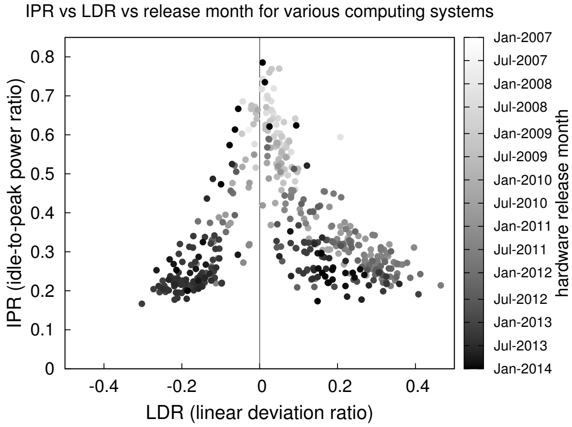


Figure 2.5: Scatter-plot of IPR vs LDR of over 290 Computing Systems, color-coded by release date, based on SPEC Power 2008 public benchmark data (http://www.spec.org/power_ssj2008/results/).

Power Proportionality of Servers

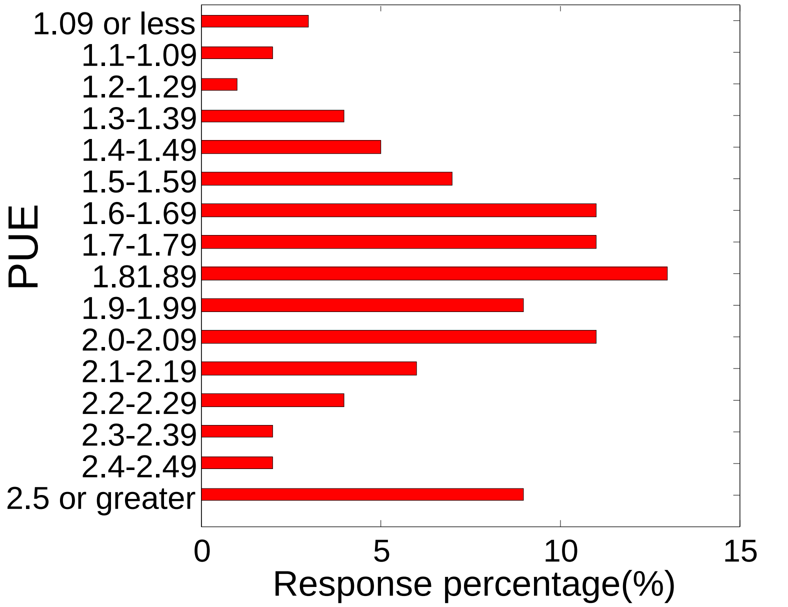
Power proportionality of a server depends on how its power consumption varies with respect to its utilization. An ideal power-proportional server has a zero idle power and a linear power-utilization curve. Current servers are far from being ideally power-proportional. One characteristic of computing systems, until recently, has been the high energy consumption when they are idle. Variable performance technologies, such as dynamic duty cycling, considerably dampen the power consumption of recent computing systems at near-idle utilization levels [24]. However, the power-utilization curve is not as linear. Varsamopoulos and Gupta observe this behavior and introduce two metrics to measure how power-proportional a system is (see Fig. 2.4): (i) Idle to Peak Ratio (IPR) measures how close to zero the idle power is (consider a server with linear power consumption of the form: $p = p^{util}u + p^{idle}$, where $0 \leqslant u \leqslant 1$

denotes the utilization, p denotes the power consumption, and p^{idle} and p^{util} denote the idle power and the power gradient with respect to utilization, respectively, then IPR=$\frac{p^{idle}}{p^{idle}+p^{util}}$.), and (ii) LDR measures the linearity of the power curve (i.e., how close the power curve is to hypothetical linear curve connecting idle power to peak power) [118]. *Ideal power-proportional* servers have zero IPR and LDR.

Recent trends, as shown in Fig. 2.5, indicate that existing computing systems cover a large area of the IPR-LDR spectrum (see Fig. 2.5). The scatter plot using SPEC Power 2008 public benchmark data released in July 2014 shows that (i) as systems lower their IPR, their LDR gets larger, and (ii) there are no systems, at least among the ones tested, with an IPR less than 0.2 (see Fig. 2.5). Chapter 4 uses IPR and LDR metrics to study how the energy proportionality of systems affect the performance of energy aware workload and server management schemes.

Cooling Power Efficiency

To evaluate the cooling energy trend of data centers, one can use PUE (the ratio of the total power used by a data center to the power used by its IT equipment). PUE is a widely used metric developed by the Green Grid consortium to measure the power efficiency of non-computing equipment in data centers [99]. Large PUE is an indication of large cooling power. Ideally, PUE should be equal to one. However, the PUE value of data centers range from over 2.5 down to around 1.1 according to the 2012 survey reports by Uptime institute [107] (see Fig. 2.6). In recent years, large data center operators, such as Google and Facebook, have improved the PUE of their modern data centers (PUE is reported 1.18 for a Google modern data center and 1.08 for a Facebook data center). However, as shown in Fig. 2.6, only 6% of data centers report a PUE of less than 1.3. Therefore, it is important to design and evaluate the workload management schemes taking into account high PUE of data centers.



(a)

Figure 2.6: PUE of Data Centers according to Uptime Institute Survey Report, 2012 . The survey reports a wide range of PUE from its 1,100 respondents; the average PUE is around 1.8-1.89, and only six percent of respondents claim a PUE of less than 1.3 (data from Uptime Institute).

Further, “watts per square foot” is typically used to measure the power density of data centers where the square foot is usually calculated per the room square foot (it may also refer to the rack sqaure foot, or the production area square foot i.e., the actual room space used for equipment). The current typical data centers have 35 upto 100 watts per square foot depending on the server density. Modern data centers, however, due to the technology trend towards high density computing, specifically blade chassis environments, have higher power density e.g., 150 watt - 300 watt per square foot. Dense deployment of servers demands high cooling power.

2.2 Renewable Energy

Renewable energy are usually very expensive to implement, depend on the surrounding weather conditions, intermittently available, and require a big land area to implement in many cases. Despite drawbacks, data centers have already started to deploying them in various ways, not only to make their commitments for sustainability, but also to mitigate any steep raise in the electricity price in future. Google, Apple, FaceBook and many other industry leaders already have made investments to partially or totally power their data centers from renewable energy sources, primarily using solar and wind energy [4, 85, 119]. The number of these data centers will likely to grow as (i) the installation costs of the renewable energy technologies tends to decrease [5], (ii) the on-site renewable energy helps to mitigate the electricity price raise in future[120], and (iii) governments increasingly provide incentives to generate and utilize green power [6].

Due to the limitations of the on-site renewable energy sources, i.e., geographical location or land, many companies do not have opportunities to install on-site renewable sources or directly use renewable utility power. There are other solutions such as Power Purchasing Agreement (PPA) and Renewable Energy Certificates (RECs) that data centers can purchase to contribute in the growth of the renewable energy industry.

By purchasing PPA, as done by Google, the data centers invest and sign with several renewable energy plants such that the generated renewable energy will be directly fed into the local electricity grid and then used to offset the brown energy usage of the data centers [4].

By purchasing RECs, data centers support renewable energy producers by committing to buying their energy for long-term, but use brown energy on sites [4]. This dissertation studies how on-site renewable energy sources can be managed through global workload management to reduce the carbon footprints, and the operational energy cost of data centers, while accounting for their intermittent nature and predictability.

2.3 Energy Cost

We study the operational energy cost of data centers, which consists of the electricity cost, and the cost per peak power: in addition to the electricity cost, some utility providers also penalize the excess power draw by imposing additional fee if the peak power draw exceeds the stipulated power in a certain time window [22, 121]. We study the cost minimization problem under both of the above models as well as the spatio-temporal variation of the electricity price as explained below.

There are typically three types of electricity plans offered through the various providers of electricity [43]: (i) fixed pricing, (ii) time of use pricing, and (iii) dynamic pricing. Under the fixed pricing, the electricity price is constant over time. Under the time of use pricing, the electricity pricing has a constant daily pattern, e.g., there might be two prices, one for day-time and the other for night-time depending on the periods of peak and low power demands. Under the dynamic pricing, managed by the wholesale electricity market, the electricity pricing is dynamic, significantly varies over time and has seasonal daily, and monthly pattern. Big power consumers often use dynamic pricing to leverage the electricity price variation by scheduling their electricity consumption intelligently and saving money[96].

2.3.1 Wholesale Electricity Market

The price of electricity in the wholesale market depends on a number of factors including the fuel type, the supply-demand variation, and the market.

Electricity is produced from a variety of sources including coal, natural gas , nuclear power, and hydroelectric generation. Different regions use different sources depending on the availability of sources and their expenditure. For example in USA the total generation output in 2012 shows that coal dominates (37%), followed by natural gas (39%), nuclear (19%), hydro (7%), and renewables (5%) generation [8].

The key limitation of the electricity comes from the fact that it currently cannot be stored in a scalable and cost-efficient way. A sophisticated control is needed to ensure a close match between the supply and the demand. Any mismatch between the two can induce a high cost as power producers may need to add or remove the generation plants or load both of which are costly. To mitigate such problems, system operators, known as balance authority, closely monitor the system to ensure capacity reliability. The system operators consisting of utilities, federal agencies and Independent System Operators (ISO) or Regional Transmission Organization (RTO)s, forecast demand in the day-ahead market, schedule power generation, reserve the transmission, adjust schedule as hours get closer, correct imbalances in real time, restore systems if disturbance occur, and sometimes plan for long-term capacity and transmission upgrade.

System operators in many regions of north America are ISO/RTOs which manage the grid and the wholesale electricity market. The pricing in the wholesale market can be day-ahead, hourly basis or real-time.

The system price in the day-ahead market is determined by auctioning mechanism for the producers and the customers at each node to develop a classic supply and demand equilibrium price, usually on an hourly interval, and is calculated separately for subregions in the grid [96, 122].

RTOs set the Locational Marginal Price (LMP) (e.g., for hourly interval or real-time) for different nodes in the grid which consists of three components: (i) System

Table 2.1: Carbon Emission Factor of well-known Electricity Fuels (CO_2 g/kWh).

coal	PL	NG	Nuclear energy	Wind	solar
986	890	440	15	22.5	18

Energy Price (SEP): system clearing price if no congestion exists (always same at all locations), (ii) Marginal Lost Cost (MLC): Cost of marginal losses along transmission into specific node, and (iii) Marginal Congestion Cost (MCC): If congestion is positive, cost is incurred by expensive energy delivered to the destination. Whereas negative congestion indicates that the electricity generated is more than its demand. The cost is then calculated for each less MW that destination nodes consume compared to what is generated at the source nodes in the grid [122].

In this study we assume data centers use hourly-basis dynamic pricing where an hour ahead price can be predicted with reasonable accuracy.

2.4 Carbon Footprint Capping and Carbon Neutrality

Carbon emission factor of a power plant is the carbon emitted for a given amount of energy consumed which is calculated in CO_2 g/J. As shown in Table 2.1, brown energy sources such as coal, Petroleum Liquids, and Natural Gas has very high carbon intensities. *Carbon neutrality*, or having a net zero carbon footprint, refers to achieving net zero carbon emissions by balancing a measured amount of carbon released with an equivalent amount of offset (e.g., planting trees), or buying enough carbon credits to make up the difference.

A carbon neutral or net-zero data center is a data center with net-zero carbon emissions, e.g., the total amount of energy used by the data center on an annual basis is roughly equal to the amount of renewable energy directly (on-site) or indirectly (e.g., REC) generated by the data center, and purchasing carbon credits for the remaining carbon footprint offset.

Although currently there is little financial motivation to use green or clean energy sources (green energy sources are more expensive than the brown energy sources), data centers will soon be required to cap their carbon footprint towards achieving carbon neutrality. This is due to either the mandatory carbon capping policies by governments, policies of utility companies, voluntarily purposes, or pressures from non-profit environmental organizations [69]. First, some countries such as UK and Australia have already developed and regulated carbon capping policies, and some others are taking steps towards regulating those polices (e.g., USA), with cap-and-trade and carbon-tax being the most popular ones [49]. A *carbon tax* imposes a tax on each unit of greenhouse gas emissions and gives companies an incentive to reduce pollution whenever doing so would cost less than paying the tax. A *cap-and-trade* system sets a maximum level of pollution, a cap, and distributes emissions permits among companies that produce emissions. Companies must have a permit to cover each unit of pollution they produce, and they can obtain these permits either through an initial allocation or auction, or through trading with other companies. Data centers, as big power consumers, should operate under the carbon capping policies. In UK businesses consuming more than 6 GWh per year should participate, which includes relatively small data centers with 700 KW power consumption. Second, for some utility companies, it is desired to cap the maximum power draw due its cost efficiency. Third, some data centers desire to take volunteer steps toward achieving carbon neutrality in order to benefit from favorable accreditation, and/or business promotion [4]. Finally, environmental activists have started pushing data centers towards achieving sustainability [53].

Recently, several companies such as Google, and Microsoft have set carbon neutral-

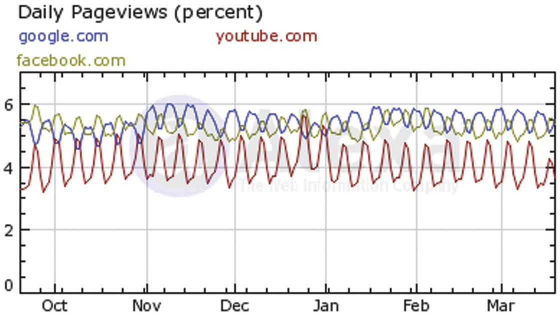


Figure 2.7: Demonstration of the Variation and Cyclic Behavior of Web Traffic for Three Popular Web Sites (source: www.alexa.com).

ity as their long-term strategic goals [4, 38]. Despite being desirable, achieving carbon neutrality is challenging. In particular, it needs to cap data centers' brown electricity usage over a long time such that the entire future brown energy consumption can be completely offset by the limited and intermittent available renewable energy sources and the carbon credits. Further, the capping brown energy consumption should be performed without compromising the quality of service and considerably increasing the energy cost. The magnitude of the brown energy cap is decided by data centers by taking into consideration the cost benefit analysis of the existing carbon capping policies, carbon credits such as REC, their on-site and off-site renewable energy plan and the electricity cost. The solution designed in this dissertation, OnlineCC, works under such a carbon capping policies of data centers (see Chapter 6). Suppose data centers uses carbon capping and carbon credits to achieve carbon neutrality. Denote by Σ, the carbon cap, then the enough carbon credits to remove Σ has to be purchased in order to achieve carbon neutrality.

2.5 Internet Services

Data centers generally host heterogeneous applications ranging from web services to batch and highly computationally intensive jobs. This study accounts for Internet-type applications which are offered online using large server farms in data centers. The solutions of this research are consistent with the performance requirement of these applications, their workload nature, and their data requirement.

First, Internet applications are delay sensitive such that for a high Quality of Service (QoS) the end users' delay should not exceed a reference delay. Therefore, a crucial aspect of the energy management solutions is managing the quality of service. Data center and cloud providers offer their service based on a Service Level Agreement (SLA), where any quality of service violation from SLA costs them a punishment fee. In addition to such a punishment fee, frequent delay violation increases the risk of losing the customers and increases the tendency of customers to leave the provider [60]. Service providers such as Google, and Amazon, are reluctant to trade QoS violation in any profit (e.g., energy saving). The reason is that any service QoS degradation, may decrease their service revenue dramatically (e.g., decreasing the number of people visiting and shopping in Amazon). This dissertation aims to design energy management solutions without compromising the QoS requirement of the applications.

Second, the solutions in this research is based on the variability assumptions of the workload (e.g., daily and weekly variation). Intensity variation in the web traffic has been witnessed by several researches [24, 31]. The variation originates from the size variability of files communicated, users' thinking times (e.g., the time interval between each click) which form a short-term variation, fluctuation in time scale of a few seconds [23] (see Fig. 1.2(a) as an example) and number of online users which forms a long-term, i.e., hourly and daily, cyclic variation (see Fig. 2.7).

Finally, throughout most of this research, we assume that data requirements of the applications are fully replicated such that they can run on any server at any data center. Although data replications is prevalent especially for large-scale Internet applications (e.g., Google search), in practice, the applications tend to be only partially replicated, e.g., data is replicated at multiple clusters of data centers, but not at all clusters [96]. This means that a request may be serviced by more than one cluster of servers, but not by any server.

The following sections give an overview on the cloud service models and cloud infrastructure of Internet services.

2.5.1 Cloud Services

Cloud computing provides several service models including: Infrastructure as a Service (IaaS), Software as a service (Sass), and Platform as a Service (PaaS) [83]. In IaaS, the cloud provides physical or virtual machines and other sources to run the clients' applications. In PaaS, in addition to the hardware and Virtual Machines (VMs), the cloud also provides the operating system and other resources for the clients to develop and run their applications. In SaaS, the cloud installs the software, such that the users can access the software from the cloud (such as email service). In addition to the above, there are other form of services, where one or both of the computation and data of the clients' applications can be fully or partially outsourced to the cloud [109].

The algorithms developed in this dissertation need to access: (i) the cloud infrastructure parameters such as electricity price, thermal profile of the data centers and the physical location of applications, and (ii) the web application parameters such as workload, and the performance model. Theoretically, the algorithms can be deployed in all of the cloud service models and the deployment can be performed either by the cloud owner (for the case where cloud offers PaaS and IaaS) or the application owner (for the case where the cloud IaaS is used). However, this is only true, when the cloud infrastructure related parameters can be exposed to the algorithm. In practice, however, the access and control permission to the cloud infrastructure related parameters are very limited for the security and confidentiality reasons. Therefore, given that the cloud has ability/willing to expose the infrastructure related parameters to the PaaS and SaaS platforms, the algorithms can be deployed in a cloud that offer PaaS and SaaS where the two aforementioned group of parameters can be exposed to the algorithm.

2.5.2 Geo-distributed Data Centers

Current large-scale Internet services tend to be replicated over several data centers around the world. For example in USA, Akamai, a Content Delivery Network (CDN) provider, spreads its servers across hundreds of locations [96]. Similarly, Google provide its service from several geographically distributed data centers as shown in Fig. 2.8. Also a recent data center survey by Uptime Institute reports that 82% of respondents mange more than one data center. Geographically distribution of data centers is primarily performed for fault tolerance and quality of service purposes. Recently, the research community proposes to leverage the spatio-temporal variability of energy cost and carbon footprint across data centers' locations using global workload management [71, 73, 77, 96, 97, 123]. Such geographically distributed data centers are expected to form the infrastructure of the clouds and a design model for the future cloud infrastructure, since such a model increases the reliability and provides many energy and quality of service management possibilities. The solutions of this dissertation i.e., DAHM (Chapter 5), OnlineCC (Chapter 6), and OnCMCCLyp (Chapter 7) account for global workload management for geo-distributed data centers.

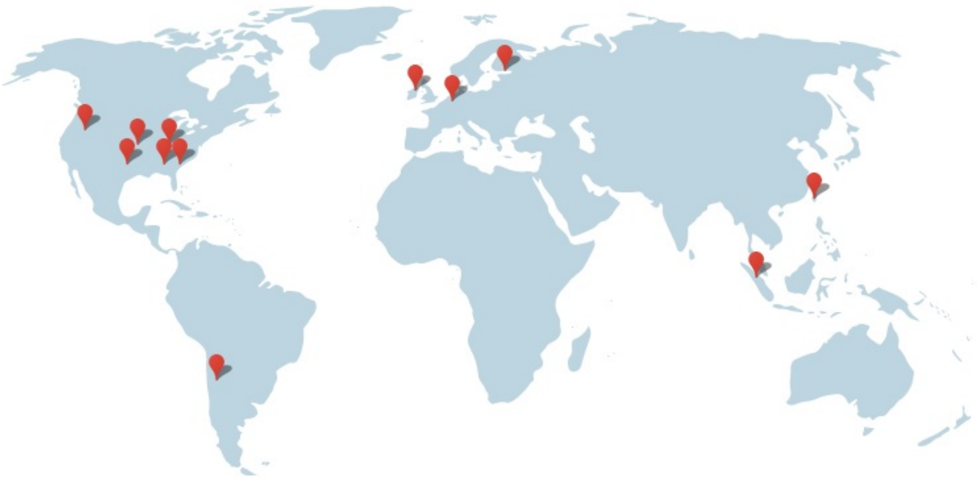


Figure 2.8: Locations of Google Data Centers (source:www.google.com).

2.6 Implementations Aspects of Energy Management Solutions

energy aware server and workload management solutions require the underlying infrastructure allow dynamic workload balancing across data centers and server consolidation. The following sections describe the current state of the data centers to implement energy management solutions.

2.6.1 Workload Distribution Across Data Centers

Global workload management, being a design goal of this research, is based on the assumption that the underlying infrastructure (cloud or geographically distributed data centers) allows request redirection mechanism in order to distribute requests across data centers based on some policies. Request redirection solutions are already in use to enable replication over Internet and CDNs [33, 95]. In this regard DNS based request-routing techniques are common due to the ubiquity of the DNS system. In DNS based request-routing techniques, a specialized DNS server is inserted in the DNS resolution process. The server is capable of returning a different set of records based on user defined policies, metrics, or a combination of both. There are also other techniques such as HTTP redirection using persistent HTTP proxies to tunnel requests, which are currently employed for selecting data centers.

2.6.2 Server Consolidation

This dissertation designs various energy-aware and cost-aware dynamic server provisioning schemes. The idea of dynamic server provisioning is to adjust the number of active servers in a server farm to the offered workload at a given time. In addition to the modeling and algorithmic challenges which have been addressed in this research, server provisioning has some implementation challenges. Especially suspending servers costs energy as servers consume energy to be turned back on, there is a switching delay, availability of service may be violated because of suspending servers which are still in service, and there is an increase in wear and tear of the server components. Our approach avoids the above repercussions by performing infrequent and proactive switching. Dynamic server provisioning requires to selectively suspend individual servers; modern data centers are increasingly likely to support this functionality. Server consolidation is currently used by data centers to overcome increasingly growing of the demand, and to save energy [107].

2.7 System Model

In brief, we perform this research in the context of delay-sensitive and large-scale Internet services, such as those provided from eBay, Amazon, and Google, or large scale hosting centers (e.g., GoDaddy). As depicted in Fig. 2.9, we assume these services are supported by multiple data centers. The data centers sit behind front-end devices that inspect each client's request and forward it to one of the data centers according to a request distribution policy.

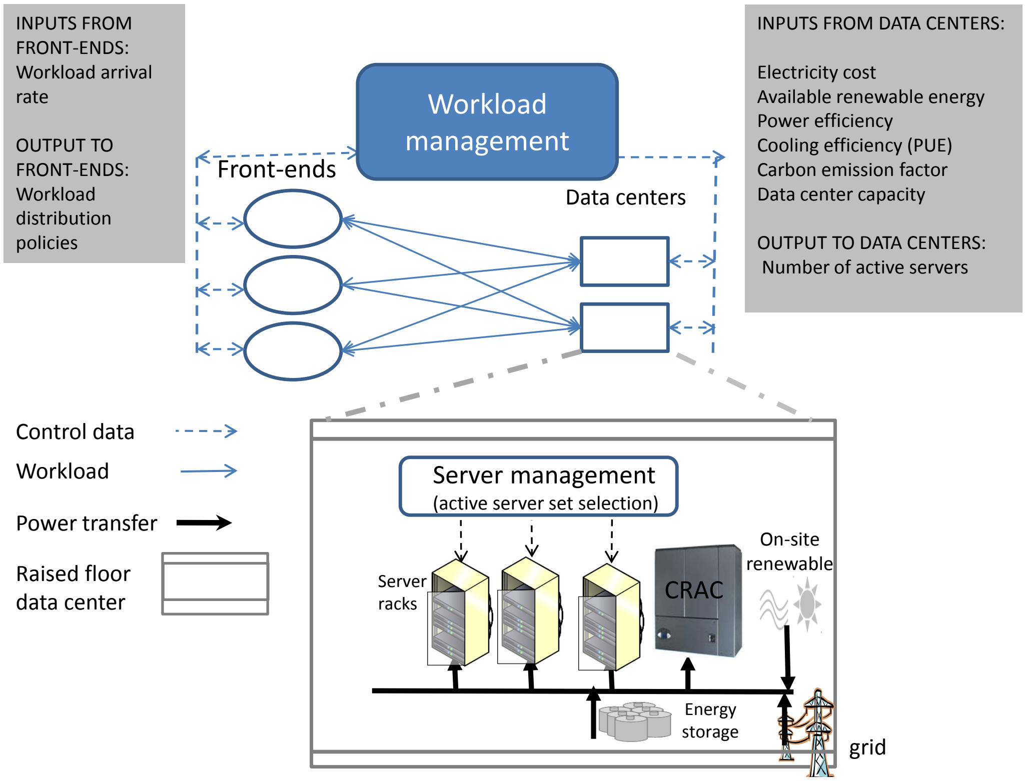


Figure 2.9: System Model for Local and Global Data Center Workload Management.

We assume the most common cooling technology for data centers, i.e., air-cooling technology. Further, we assume the heat created by the servers is partially distributed in the room, a common phenomenon in contemporary data centers with air-cooling technology. Also, the heat distribution in data centers is not uniform (i.e., servers have different contribution on the heat recirculation depending on their physical location).

Finally, as shown in Fig. 2.9, we assume data centers power their servers and the associated accessories from a mix of grid, energy storage devices and on-site renewable energy sources.

In order to design dynamic server provisioning and workload management schemes, we consider a discrete-time model by dividing the entire budgeting period (e.g., typically a year) into S time slots each of which has a duration that is short enough to capture the variation of data center input parameters (e.g., workload, and electricity cost) yet long enough to prevent the solutions' overhead (e.g., computation and network overhead, server switching overhead). We frame the energy and cost management problems, as optimization problems each of which are defined to reconcile a number of competing objectives, e.g., reducing both the cooling and the computing energy, reducing the electricity cost, maintaining requests' delay requirement and the cloud carbon footprint cap in order to decide on the dynamic server provisioning and workload distribution.

We base the management schemes at the local data center level on a system model which consists of assumptions and modeling of data center physical layout as given in Chapter 4.

The solutions at the global data center level all are designed based on the system model given in Fig. 2.9. In particular, the system model assumes a cloud consisting of N geo-distributed data centers, each containing at most Y_i servers. We perform work management at long time intervals (e.g., hourly) and control servers' *on* and *off* power states with zero power consumption in the off state. Other server and CPU power state management (e.g., DVFS) which are typically performed in short intervals (e.g., seconds) [81] can be considered as complement to the proposed solutions.

End users' requests first arrive at one of the M front-end proxy servers. The proxy servers then decide how to distribute the requests to data centers according to the policies dictated by our workload management schemes. All the management schemes are designed to optimally decide on (i) the workload division among the data centers, denoted by $\lambda_{i,j}(t)$, i.e., the workload arrival rate from front-end j to data center i, and (ii) the number of active servers at each data center i, denoted by $y_i(t)$ (the remaining servers i.e., $Y_i - y_i(t)$ are set to inactive to save the unnecessary idle power). The solutions, however, are distinguished based on the specific aspect of energy cost

Table 2.2: Symbols and Definitions.

Sym.	Definition	Sym.	Definition
t	slot index	g	power drawn from grid
S	total # of slots	p^{tot}	total power cons.
j	frontend index	r	renewable harvesting
i	data center index	ε^g	grid carbon emission
N	# of data centers	ε^r	renew. carbon emission
μ	service rate	b	total carbon emission
λ	workload arrival rate	Y	total # of servers
d^{ref}	reference delay	y	# of active servers
d'^{ref}	service reference delay	y^{slack}	percentage of reserved active servers
d''	network delay	α	electricity price
β	Peak power cost		

minimization problem that they address. These aspects consist of solutions' optimality and computation efficiency analysis, carbon capping and peak power shaving in data centers, each of which represents practical cases and are associated with some specific problem formulation details which is described in the corresponding chapter.

The optimization problems at both the local and the global data center levels are made up of models characterizing the power demands and the power supply of data centers. The power demand model of data centers is derived from the workload distribution model according to their performance requirements, and the power consumption model of servers and the cooling system. The power supply model consists of models to describe the power draw from the grid, batteries and the on-site renewable power of data centers. The models are the basis of the optimization frameworks of Chapters 4, 5, 6, 7, in order to design dynamic cloud energy/cost management

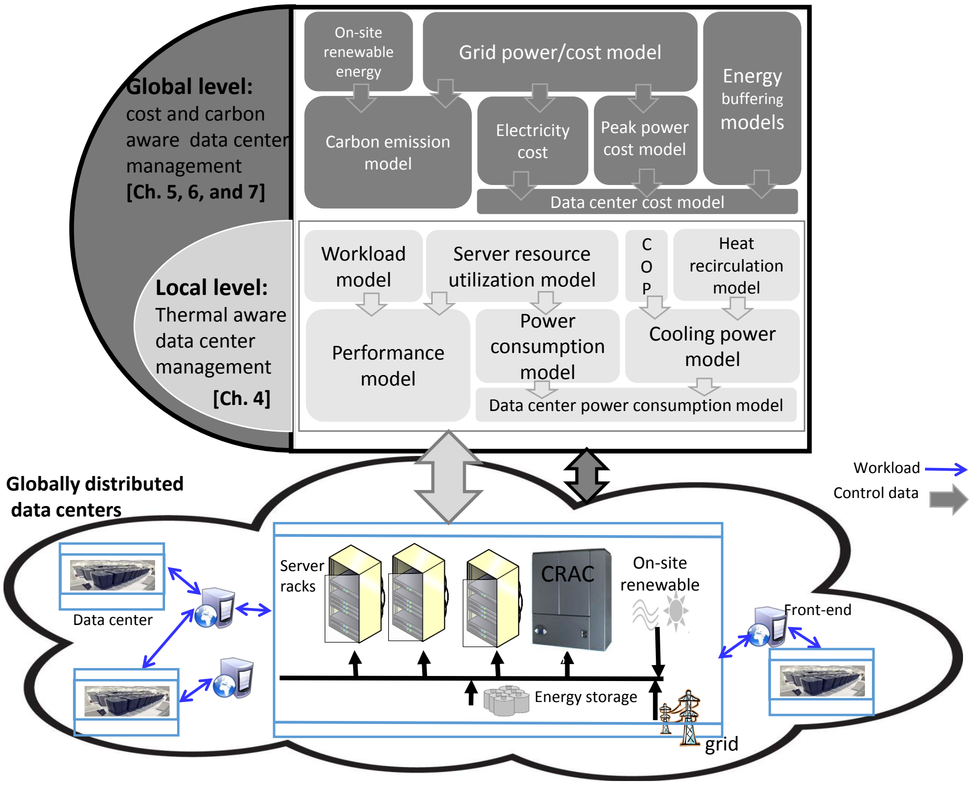


Figure 2.10: Models for Local and Global Data Center Management.

schemes. Fig. 2.10 gives an overview on the models and the way they are derived and fed in to the mathematical frameworks for optimization of data centers' cost and energy consumption. The formulation and models build on some existing data centers models which are described in the following sections. A summary of notations used to design global data center level management schemes are given in Table 2.2.

2.7.1 Power Demand Modeling

The power demand model describes the total power consumption of data centers at a given time which depends on the input workload, the resource assignment (e.g., number of servers) and the power consumption models of servers and the cooling system. The resource assignment should be performed in such a way that the performance requirements are maintained.

Workload Modeling

Throughout this research we assume short, interactive, and delay sensitive Internet workload. The workload are originated from online Internet users. Users generate Internet requests (e.g., by clicking a URL or submitting an online bank transaction) which need to be completed in a fraction of a second. The workload can be modeled as the request rate, i.e., number of requests per second which can be directly calculated from the web servers traces. We denote the workload arrival rate by λ, which is allowed to vary over time and location, consistent with the nature of Internet workload as explained in Section 2.5. When designing management schemes for the local data center level, we account for the time variation of workload offered to an individual data center. Whereas to design the management schemes for the global data center level, we account for both the spatio and the temporal variation of workload entered to the different geo-distributed front-ends (recall front-end refers to the entry point of the Internet workload from which the workload is distributed across data centers). This is natural, considering that users across different locations around the world contribute to the cloud's workload according to their local time zone.

Performance Modeling

Cloud performance from end users' perspective is primarily about their response time. The delay experienced by the end users (i.e., response time) should not exceed form a reference value. In this context, the Service Level Agreement (SLA), an agreement between service providers and the users, statistically bounds the response time:
Prob[*response_time* ¿ response_threshold$_{\text{SLA}}$] ¡ probability_threshold$_{\text{SLA}}$.

As such, the delay requirement metric is often defined as the percentile of requests which should receive a delay within a reference delay (e.g., 99-th percentile delay). Hence, given a workload arrival rate to a data center, the performance model of a server provisioning scheme calculates the number of active servers needed to statically bound the response time experienced by the end users. We adapt the following models form the related work, some of which we independently evaluate in some experimental studies: Performance models based on queuing theory and models based on CPU utilization threshold. The following section first gives an overview on the models followed by a brief overview on the way we use the models in the research.

Performance models based on queuing theory The average response time of a server can be modeled using queuing theory. If the server happens to be modeled by an M/M/1 processor-sharing queue, then given that all requests are queued, the average response time, denoted by d, can be written as $d = \frac{1}{\mu} + \frac{1}{\mu - \lambda}$, where the service rate of the server is μ. The first term in the equation gives the service delay and the second term gives the queuing delay. The $M/M/1$ queuing model is based on the exponential distribution for the average service time and the inter-arrival time between requests. There exist some other queuing theory models based on different distribution assumptions for the service time and the inter-arrival time of the requests. In particular, $G/G/1$ do not assume any distribution for the service time or the inter-arrival time of the requests.

Similarly, the average response time of a data center with n servers can be modeled by $M/M/n$ or $G/G/n$ queuing models, each of which provides a mathematical model to express the average response time of the data center. In $M/M/n$ model given that all requests are queued, the average response time is given as follows:

$$d = \frac{1}{\mu} + \frac{1}{n\mu - \lambda}. \tag{2.1}$$

The average response time for a $G/G/n$ system can be captured using the following approximation model [20]:

$$d = \frac{1}{\mu} + \frac{u'_n}{\mu(1-u)} \frac{C_A^2 + C_B^2}{2n}, \tag{2.2}$$

where C_A and C_B denote the Coefficient of Variation (COV), i.e., standard deviation divided by the mean, of request inter arrival time and service time respectively, and u' is as follows:

$$u' = u^{\frac{n+1}{2}} \text{ for } u \leqslant 0.7 \text{ and } u'_n = \frac{u^n + u}{2}\text{otherwise.}$$

The parameter u denotes the average utilization of a server and can be written as $\frac{\lambda}{n\mu}$.

Given the average workload arrival rate to a data center and the service rate of the servers, the above models can be used to calculate the number of required active servers to achieve the desired average response time. However, as mentioned earlier, the performance (QoS) of Internet data centers is typically defined as some metric on delay, e.g., 95th -percentile delay. One widely used solution to this is to calculate the number of active servers for the peak workload arrival rate at a given time instead of the average workload arrival rate. Alternatively, minimum number of active servers can be calculated as the number of required servers to maintain the average response time plus an addition slack to deal with spikes at the peak traffic time [44, 70, 115].

Performance Model based on CPU Utilization Threshold Although web traffic is not CPU-intensive, related research has identified that the CPU utilization level is strongly correlated to the QoS; specifically, the SLA is violated beyond a CPU utilization point [30].

The aforementioned correlation is observed in the following experiment as well. We configured one computer as the web server and another computer as the client generating TCP-based requests on files with size distribution ranging from 0.3KB to 90KB, in accordance to a study on the file size distribution of web image content [101].

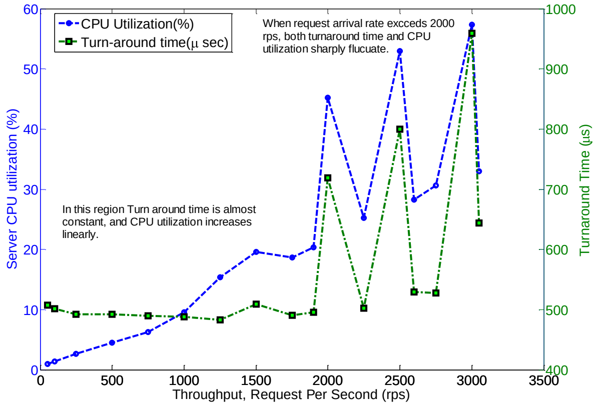


Figure 2.11: Turnaround Time and CPU Utilization versus Throughput.

Both the web server and the client are dual-CPU dual-core E7520-chipset "Sossaman" Xeon LV systems. The results as shown in Fig. 2.11 depicts the average turnaround time and the web servers' CPU utilization over the input workload (measured as arrival rate). It can be seen that, the turnaround time is constant until the utilization reaches to around 20% (or the arrival rate reaches to 2000 requests per second) and then it goes up and even fluctuates. This experiment shows that the quality of service of Internet requests in terms of delay can be guaranteed if a server is not utilized up to a threshold point. The amount of threshold point depends on the hardware capacity of servers and the type of requests. Therefore, it can be considered that *by posing a bound to the CPU utilization, (i.e. preventing overloading of a server such that its CPU utilization does not go beyond a threshold value), one automatically pose a bound to the SLA violation rate*. This is an important observation as CPU utilization levels are easier to track than response time.

Discussion on Performance Modeling Observe that each of the aforementioned performance models require different input data to model the delay. Based on the availability of the data and the monitoring tool available for the data center power management solution either of them can be used in practice. The models, however, are not exactly equivalent. Each of them can be utilized in a way to provide overestimation number of active server, a requirement for server provisioning schemes to deal with workload spikes.

Both $M/M/n$ and $G/G/n$ models are frequently used in the literature in order to model Internet data centers' average response time. For Internet-like workload, with proven heavy tail workload, $G/G/n$ (representing a n-server queuing system serving requests with generalized arrival and service time distribution) can capture each data centers' service delay. We use this model to design the solutions at the local data level (see Chapter4). Similar to the related work [77, 97] we use $M/M/n$ model, however, to design solutions at the global data center level. This is because, first, using $M/M/n$ model, one can calculate the average number of required servers as a linear function of the workload, favoring computation efficient solutions for dynamic server provisioning. Second, for CDN and Internet data centers, having large number of servers, $M/M/n$ gives an overestimation of the average delay. In particular, the average queuing delay of $G/G/n$ depends on the factor of $\frac{C_A^2+C_B^2}{2n}$. As long as this element is evaluated to a number less than one, the $M/M/n$ delay model overestimates the delay of $G/G/n$, which is true for CDNs and Internet data centers with large number of n and COV of around 4 [21, 25, 96, 101]. We evaluate this model using experiments in Section 6.5.

Both $M/M/n$ queuing model and CPU threshold utilization based model can be interchangeably used in all of the designed solutions without changing the nature of the problems, because both of them model the average delay as a linear function of the number of required servers. In practice, however, CPU threshold utilization model gives an upper bound on the number of required active servers compared to $M/M/n$ model due to overlooking the stochastic aggregation of workload on the large number of servers of a data center.

Total Delay The two previous delay modelings account for the data center delay. However the delay $d_{i,j}(t)$ experienced by a user of front-end j, receiving service from data center i at slot t consists of the service delay $d'_i(t)$, i.e., data center delay, and the network delay $d''_{i,j}(t)$, i.e., the delay between a front-end i and a data center j; the total delay becomes $d_i(t) = d'_i(t){+}d''_{i,j}(t)$. Global workload management schemes needs to account for such a total delay. Network delay between front-ends and data centers vary over time depending on the network congestion. We denote the total delay as $d_{i,j}(t)$ to mark the dependence on data center i, front-end j and slot t.

The optimization frameworks of global workload management schemes (e.g., Chapter 5) formulate the delay requirements as constraints. However, instead of bounding the total delay, they are designed to bound both the service delay (to do not exceed the reference service delay), and sum of the reference service delay and the network delay (to do not exceed the reference delay): $d'_i(t){\le}d'^{ref}$, and $d''_{i,j}(t){+}d'^{ref} \le d^{ref}$. In this way, we linearly model the delay requirements. Given that service delay is in the range of 6 ms, and the network delay is in the range of 100 ms, this simplification has a negligible effect on the performance of the solutions.

Computing Power Modeling

The power consumption model of a server specifies how its power consumption changes with respect to its utilization, which is typically expressed as a linear model: $p = p^{idle}{+}u(t)p^{util}$, where u, $0 \le u(t){\le}1$ denotes the utilization, and p^{idle} and p^{util} denote the idle power and the additional power at full utilization with respect to the idle power, respectively. The utilization, refers to the overall utilization of a server for a given input workload. Mathematically, given workload arrival rate of $\lambda(t)$ at slot t the utilization can be expressed as follows: $u(t) = \frac{\lambda(t)}{\mu}$. We assume, μ, p^{idle}, and p^{util} are known for the Internet workload which can be obtained by profiling. Further, we consider the sum of the power consumption of servers as the total computing power of the data center.

Cooling Power Modeling

Cooling power model specifies the power consumption of the cooling system for a given computing power of a data center over time. We use two cooling power models based on (i) heat recirculation model of a data center, and (ii) PUE of a data center. The first one allows to study the impact of active server set on the temperature distributing within a data center room, and on its cooling power. Heat recirculation in a data center room refers to the amount of heat that flows from one server to another. We base our heat circulation model on the model proposed by Tang et al. [110] which is explained in detail in Chapter 4 and is used to design TACOMA. The second cooling power model is used to estimate the cooling power for a given computing power and a given active server set selection algorithm of a data center. In this model, the total power consumption of a data center is calculated as multiplication of its total computing power and PUE. The later cooling power model, when PUE is given independent of the active server set, simplifies data centers' total power consumption model and is used to derive analytical results for global workload management solutions. We also study the impact of the heat recirculation on PUE and on the global workload management solution in Chapter 6.

2.7.2 Power Supply Modeling

The power supply model, describes the power type, its cost and carbon footprint. Most of data centers get their primary power from the grid. Further, some data centers partially or totally power their data centers from on-site renewable energy sources. Furthermore, some data centers utilize energy storage devices to smoothen power draw from the grid and the on-site renewable energy sources.

Energy storage devices are associated with physical characteristics and limitation in terms of maximum charging and discharging rate, total number of charging/discharging cycles and energy inefficiencies per charging and discharging which are considered in their modeling as described in detail in Chapter 7.

Energy Cost Modeling

The energy cost model specifies the monetary fee that the data center operator should pay to the utility provider for the total energy consumed during a time interval. Denote by $p^{tot}(t)$, the total one-slot energy consumption of a data center, and by $\alpha(t)$, the unit energy price at time t, then $p^{tot}(t)\alpha(t)$ gives the total electricity cost of the data center at slot t. Further, consider a time frame T consists of multiple slots, and denote by p_0 the stipulated power from which if the data center's power exceeds, the peak power cost is incurred. Also denote by β the violation fee, then $\max_{t \in T}(p^{tot}(t) - p_0)^{+}\beta$ gives the peak power cost, where the "+" indicates that only when the difference is positive the cost is considered, otherwise there is no cost.

Renewable Energy Modeling

Solutions in Chapters 6 and 7 take into account data centers that integrate on-sire renewable energy generation. Given $r^{tot}(t)$, the average one-slot renewable energy generated at a data center, the solutions are designed to increase the available renewable energy utilization. Renewable energy utilization in those techniques are studied from the following perspectives: (i) percentage of renewable energy that is actually utilized in data centers out of the total renewable energy generated: $\frac{r(t) \times 100}{r^{tot}(t)}$, where r is the amount of renewable energy that is actually utilized by the load, and (ii) percentage of energy consumption in data centers that is of type renewable: $\frac{r(t) \times 100}{p^{tot}(t)}$.

The above metrics help to evaluate the performance of the solutions in utilizing renewable energy, and the renewable energy infrastructure capacity to achieve energy sustainability.

Carbon Footprint Modeling

The carbon footprint model specifies the total CO_2 generated over time. Denote by $\varepsilon^g(t)$ and $\varepsilon^r(t)$, the average CO_2 per unit of energy from the grid power and the available renewable power, respectively, then $b(t)$ the total carbon footprint of data center i is calculated as follows:

$$b_i(t) = g_i(t)\varepsilon_i^g(t) + r_i(t)\varepsilon_i^r(t), \tag{2.3}$$

where $g_i(t)$ and $r_i(t)$ denote energy consumed from grid and the available renewable energy of data center i at slot t.

Chapter 3

RELATED WORK

This chapter gives an overview on the related work in the area of software based workload and server management at the local and the global data center level. The chapter first reviews the related work in each of the domains of thermal ware workload management, cost optimization, carbon footprint capping and peak power reduction at data centers. Then it closes by concluding and summarizing the literature review.

3.1 Thermal-aware Scheduling

Server consolidation, i.e., dynamically adjusting the number of active servers and suspending unnecessary servers, has been proposed to reduce both computing and cooling power for IDCs [30, 31, 55, 65, 67, 74, 81, 82], since (i) the idle power is comparable to the maximum power for the current computing systems [24, 118], and (ii) web servers have periods of low utilization due to the periodic nature of the workload [24, 30, 31]. However, analytical formalizations of server consolidation schemes focus mainly on the tradeoff between the quality of service (QoS) and the minimum number of active servers [30, 31, 65], and the tradeoff between removing idle power and server switching cost [74, 82]. The proposed solutions in the literature are designed in order to reduce the computing power and, expectedly, the associated cooling power. However, these solutions may not always be effective, due to the cooling computing power tradeoff [44].

Thermal-aware scheduling at both the chip level and the data center level has been proposed in some works [88, 89, 111, 126]. The idea is to decrease the heat generated by microchips and servers, respectively, through workload scheduling. Moore et al., and Bash and Forman showed that thermal-aware workload placement can save energy in data centers [45, 87, 88]. Tang et al., and Mukherjee et al., modeled the heat that, inefficiently, is recirculated among the servers; using this model, they proposed spatio-temporal thermal-aware job scheduling algorithms for high performance computing data centers [89, 110]. Thermal-aware scheduling for IDCs is also studied in some papers. Sharma et al., introduced thermal load balancing and showed that dynamic thermal management based upon asymmetric workload placement can promote uniform temperature distribution and reduce cooling energy [104]. Parolini et al., provided analytical formulation to manage the workload distribution among servers which relies on the expected value of the traffic over time [94]. Thermal-aware server provisioning for IDCs is also studied by Faraz and Vijaykumar. The authors proposed *PowerTrade-d*, a dynamic thermal-aware server provisioning which trades the idle power and the cooling power for each other [44]. They showed that reducing the active server set size may not always reduce the total power, as it may increase the cooling power. PowerTrade-d manages the trade-off through a dynamic refinement process such that whenever a change in the size of active server set is required, extra servers are activated or deactivated one by one to ensure the desired balance between the cooling power and the idle power.

Discussion on the related work: This thesis focuses on thermal-aware scheduling at the data center level which can be seen as complementary to thermal-aware workload scheduling at the chip level. Further, the study is performed based on the existing data center heat recirculation model proposed by [110]. Using this model, the thesis develops new thermal-aware workload and server management solutions which are proven for their energy and computation efficiency. Similar to PowerTrade-d [44], the proposed solutions are aware of both the cooling and the computing power and are targeted to minimize the total power consumption of data centers. However, the refinement process to minimize the total power is analytically formulated, eliminating the need for dynamic refinement. The thesis implements PowerTrade-d and uses it as a reference algorithm to evaluate the proposed approaches. Furthermore, the thesis performs a comprehensive analytical and simulation study to investigate under what circumstances of servers' power proportionality and data centers' cooling efficiency, thermal-aware server consolidation is necessary in order to save energy in data centers. servers.

3.2 Electricity Cost Optimization

Virtualization and the spatio-temporal variation of electricity price offer leveraging opportunities to perform cost-efficient workload placement across data centers [12, 28, 70, 76, 77, 96, 97, 125]. The result of the current literature highlights that workload management across data centers can significantly reduce the electricity bill [19, 43, 70, 96, 97], and can potentially be a significant aid in reducing the carbon footprint of data centers without requiring large-scale energy storage devices [19, 70, 70, 76, 108, 125]. The existing work address the cost efficiency of global workload management problem in practice [96], framed the problem as an optimization framework [70] and found computation efficient solutions for various cases (e.g., interactive and batch jobs) of the problem [28, 77, 97, 98].

Qureshi et al., performed the very first work in the area of cost-aware workload management across data centers to prove the concept and show effective parameters in the cost efficiency of the problem [96]. The authors used heuristics to quantify the potential economic gain of considering electricity price in the location of computation. Through simulation using historical electricity prices for twenty nine locations in the US, and network traffic data collected on Akamai CDN, they report that judicious location of computation load may save millions of dollars on the total operation cost of data centers. They also showed that the magnitude of cost savings depends on how power-proportional the servers are and whether there is a constraint on the network bandwidth. They found that the cost saving is the highest when servers are ideally power-proportional and when the available network bandwidth is unconstrained.

Le et al., framed the cost-aware workload management problem as an optimization problem [70]. The problem is modeled as a nonlinear optimization problem and it is solved using Simulated Annealing. Their simulation results showed that by leveraging the electricity price, significant cost can be saved when servers are ideally power-proportional, and the cost saving decreases when servers have greater-than-zero idle power.

Rao et al., designed an efficient algorithm for the problem of load distribution across data centers with the objective of minimizing electricity cost subject to delay constraints [97]. The energy cost considered accounted for the average energy cost of active servers (i.e., active servers are assumed to operate at an average utilization and frequency). The authors used linear programming techniques and min-cost flow model to find a near optimal solution. Rao et al., fuhrer, extended their scheme above by developing a joint optimization of server management (i.e., resizing the active server set) and power management (i.e., CPU dynamic voltage and frequency scaling) across data centers using General Benders Decomposition [98].

Buchbinder et al., studied the problem for stateful jobs which incur significant migration overhead when migrated across data centers [28]. The authors designed an algorithm to solve the problem and proved a competitive bound of $\log(n)$ for their proposed algorithm, where n is the total number of servers across the cloud. However, due to the complexity of the algorithm, a heuristic easy-to-implement online algorithm is proposed which is evaluated through simulations using real electricity pricing and job workload data. The assumptions, under which the analytical bound is derived, are more suited to batch jobs.

Liu et al., tackled the management overhead of global workload management by developing two distributed algorithms [77]. The authors developed a convex cost model which accounts for per active server energy cost, and delay cost. The delay cost is incurred due to overloading servers' or network propagation delay. They designed decentralized algorithms which allow each data center and front-ends to optimize based on partial information. The authors also provided theories to guarantee the convergence of algorithms solution to the optimal solution.

The related work also highlight that geographical workload management can help to efficiently utilize renewable energy. Liu et al., proposed a convex-optimization framework to study the economic and environmental benefits of renewable energy when using geographical load balancing[77]. The authors, also performed a trace-based simulation study and showed that workload management across data centers can reduce the required size of energy storage devices to maximally utilize renewables [76] . Finally, Akoush et al., proposed to maximize the use of renewable energy by workload migration [19] .

Discussion on the Related Work: The existing solutions are based on simplification assumptions on the power consumption model of servers (e.g., average power consumption of an active server) or number of servers (i.e., using a continuous variable to model number of servers in data centers) without considering their impact on the optimality of the solution. This thesis uses the existing models (e.g. delay model), studies the problem under non-power-proportional servers, and designs approximation solution for the problem with proven approximation ratio against the optimal solution. Further, the thesis shows the effectiveness of cost-aware global workload management in reducing the electricity cost for both the stateful and the stateless Internet applications.

3.3 Carbon Capping

Carbon capping in data centers has recently received attention both in industry [3] and in literature [39, 46, 69, 78, 100, 127]. Large data center operators took initiative to utilize renewable energy in data centers. Further related literature devised workload scheduling mechanisms for data centers to operate under carbon capping policies. The existing work range from “systems” work focusing on implementation aspects in practice [37, 39, 46, 69] to “analytical” work focusing on developing algorithms with provable guarantees [78, 100, 127].

Le et al., devised a heuristic online global workload management to dynamically solve green and brown energy mix of data centers in a cloud in order to minimize the electricity cost while operating under carbon cap-and-trade policy [69]. The online solution divides the given carbon cap (typically for a year) into chunks, i.e. one chunk per week, which is weighted by the service load predicted for the corresponding week. The workload management is then solved based on the predicted service load and the chunk for the following week. Using trace based simulation and experimental studies, the authors showed that their solution allows a service to trade off brown energy consumption and energy cost (e.g., reducing brown energy consumption by 24% for 10% increase in cost).

Exploiting the daily variation of the workload, Deng et al., developed a heuristic scheme to dynamically adjust the carbon offset of a set of geo-distributed green data centers to increase their total profit [37]. The authors used empirical traces to show that how many carbon offsets a host should provide to maximize its profit. The authors also showed that their solution utilizing the diurnal fluctuations and bursty surges of workload, triples the profit of certain hosts compared to a fixed approach used in practice.

Xian et al., devised a request routing scheme for content distribution networks to minimize the weighted sum of the energy cost, the carbon footprint, and the service delay violation cost [46]. The weights need to be carefully adjusted in order to achieve the desired optimization of the electricity cost and the carbon footprint, which is typically a very difficult task. The authors, used real-world traces and showed that carbon taxes or credits are impractical in incentivizing carbon output reduction by providers of large-scale Internet applications (since the the carbon tax incentives are low compared to the electricity cost). A cloud workload management solution to handle the three-way tradeoff between latency, carbon footprint, and electricity costs, is further studied in [39]. The authors utilized Voronoi partitions to determine how to balance the workload across data centers based on the cloud operator's priorities on minimizing the network delay, the electricity cost, and the carbon footprint [39]. Their trace based simulation study suggests a cloud can be operated in such a manner to lower carbon emissions and operational cost, which comes at the penalty in terms of average service request time.

There are also some recent works which used Lyapunov optimization to jointly optimize the electricity cost and the carbon footprint in data centers [78, 100, 127]. Ren et al., and Mahmud et al., focused on designing an online electricity cost-aware workload management to achieve carbon neutrality for a single data center [78, 100]. The cost efficiency and carbon neutrality of the online solution is analytically proven compared to the offline solution with T future lookahead information. The authors showed that their online solution outperforms against an online predictive scheme in minimizing cost, while yielding lower carbon footprint. Finally, Zhou et al., leveraged Lyapunov optimization to design a carbon-aware geographical load balancing, where each data center in a cloud is associated with a carbon cap [127]. The proposed optimization framework solves for workload distribution across data centers, number of active servers for each data center and their CPU frequency. The authors showed that their solution, performing dynamic workload migration across data centers, achieves carbon capping for data centers without excessively increasing the total electricity cost.

Discussion on the related work: The solutions proposed in [37, 39, 46, 69] successfully framed the optimization framework of carbon-aware global workload management problem and identified the efficiency of such an approach in practice. The solutions, however, are heuristics in the sense that there is no guarantee on the total cost and the carbon emission gap with respect to the offline optimal solution. This thesis uses a predictive solution inspired by the solution of [69] as a reference solution. Similar to [78, 100, 127], the solution in Chapter 6, i.e., OnlineCC, utilizes Lyapunov optimization to dynamically and jointly optimize the electricity cost and the carbon footprint. OnlineCC, further, extends the Lyapunov optimization and the related work results to prove the upper bound of online solutions' carbon capping violation compared to the optimal offline solution with entire future information. The salient feature of the proven bound is that it can be estimated according to the data center parameters (e.g., carbon footprint variation over time) without the need to solve the offline problem.

3.4 Peak Power Reduction and Energy Buffering

There are also some recent works which explored the use of ESDs to reduce both the energy cost and the peak power cost within [47, 50, 51, 52, 61, 75, 116, 121] and across [43] data centers. The idea is to store energy in UPS batteries during “valley” periods of lower demand, which can be drained during “peak” periods of higher demand. The related work designed online algorithms for energy buffering mnagement[116], studied the feasibility of utilizing existing UPS devices for energy buffering management [50], addressed the data center power infrastructure design to utilize the set of heterogeneous ESD [121], and performed a set of experimental and simulation studies to show the cost and the power efficiency of energy buffering management using ESDs [52, 61]. In the following we give an overview on the above work.

Urgaonkar et al., developed an online control algorithm using Lyapunov optimization to exploit UPS devices for energy cost minimization [116]. Through analytical studies, the authors showed that their proposed solution achieves near one competitive ratio in optimizing the electricity cost depending on the battery capacity and the magnitude of the Lyapunov control parameter.

Govindan et al., performed a comprehensive study on the feasibility of utilizing UPS to store low-cost energy. The authors developed the constraints (e.g., charging discharging periods depending on life-cycle of batteries) and a Markovian based solution for cost-aware energy buffering management at data centers [50]. Their results showed that the existing UPSes are indeed effective to be used for cost-aware energy buffering.

Wang et al., devised a scheme which help data centers to leverage the existing huge set of heterogeneous Energy Storage Devices (ESDs) [121]. The proposed solution determines how heterogeneous set of ESDs can be placed in different levels of data centers power hierarchy (i.e., data center, rack, and server levels) in a cost efficient way i.e., energy cost saving and peak power shaving. The authors also developed useful cost models to study the cost-benefit of various ESDs for using in data centers’ power hierarchy.

Kontorinis et al., presented an energy buffering management policies for distributed per-server UPSs [61]. The objective of management policies is to leverage the distributed nature of the UPS batteries, store energy during low activity periods, use this energy during power spikes, and prolong the usage duration of UPSes. Finally, Govardian et al., proposed aggressive data center power provisioning with batteries at local data center level. The authors presented several heuristics to maximize use of UPS batteries for peak power shaving while ensuring data center availability [52]. Through the experimental studies, the authors showed that such an aggressive power provisioning is indeed effective in practice to reduce the peak power with no or little violations of the quality of service.

Tinski et al., framed an optimization framework to leverage cost reduction through both reducing energy consumption and strategically avoiding periods or data centers with the highest electricity costs using batteries [43]. The authors identified which strategies are most efficient under under a broad spectrum of design parameters and conditions, such as battery capacity, service level requirements, and common electricity pricing models.

Discussion on the related work: The related works showed the feasibility and the efficiency of cost-aware energy buffering in data centers using existing UPSes or any other type of ESDs. However, the studies are performed without considering the carbon capping requirements of data centers. The related work, thereby, lacks a holistic solution to jointly manage energy cost, peak power cost and carbon capping, a requirements for today's data centers to operate under carbon capping policies. This is important since the joint management of carbon capping, energy and peak power cost introduces new challenges which need to be addressed. In particular, such a holistic management favors an offline solution, due to the time coupling to manage energy storage devices, carbon capping and peak power cost minimization. Yet the existing online algorithms are often designed to address each of the aforementioned coupling factors separately, disregarding their management implications on each other. For instance, [116] designed an online algorithm using Lyapunov technique to exploit batteries in data centers for energy cost minimization. The solution and its performance is based on restricting the maximum value that the Lyapunov control parameter can get, and the minimum required energy storage capacity which is relatively a large value. However, first, we seek a practical solution without requiring large scale ESDs to avoid their space and financial overhead. Second, the proposed solution only accounts for energy cost. However, ESDs can be best utilized to shave the peak power draw, where its online management is shown to be effective when using a window-based predictive approach [22]. Third, using Lyapunov optimization for online management of both the carbon footprint and the ESD dynamics becomes a tedious task (if possible at all) since it requires one Lyapunov control parameter adjustment to optimally manage the two. In other words, the optimal control of each of carbon footprint and ESD dynamics requires a particular adjustment of Lyapunov control parameter, and it may not be possible to choose a value for Lyapunov control parameter that optimally manages the two altogether (e.g., [116] designed a solution that restricts the maximum value that Lyapunov control parameter can get to control energy buffering dynamics).

Further, the existing solutions on data center peak power shaving relied on the predictability of data centers' parameters over a window of time [22, 43, 52, 121], but they lack an analysis/solution to overcome the harmful impact of the prediction error on the peak power shaving. We use stochastic programming, a well-known solution for optimization problems with uncertainties (see Chapter 7). Stochastic programming has been successfully applied in many applications, particularly, in grid power management and optimization of energy use from renewables [54, 93]. However, we are the first (to our knowledge) to apply it for data center energy and power cost optimization.

3.5 Conclusions from the Literature Review

Tables 3.1 and 3.2 give a big picture of the state of the art schemes along with our contribution for energy management at local and global data center level, respectively.

As shown in Table 3.1, the proposed solution in this thesis, TACOMA (see Chapter 4) extends, and enhances the state of the art solutions as follows: First, existing non thermal-aware server consolidation schemes (from Table 3.1 non thermal-aware server provisioning solutions) which disregard the impact of cooling power on the data center energy consumption are not guaranteed to reduce the data center energy consumption, due to the cooling-computing power tradeoff. TACOMA addresses this problem through accounting for thermal awareness and the cooling computing power tradeoff. Further, the existing thermal-aware solutions (from Table 3.1 see thermal aware solutions) do not answer to the question that what circumstances (e.g., energy proportionality trend of servers) do necessitate thermal awareness for the server and workload management in order to save energy. TACOMA presents exhaustive analytical and experimental studies to identify which strategies (thermal-aware and non thermal-aware server management) are most effective in reducing data center energy consumption and under what conditions. Furthermore, the proposed thermal-aware algorithms are heuristic solutions. TACOMA solves the proven NP-hard server management problem using a greedy solution with a known approximation ratio with respect to the optimal solution. Finally, the existing solutions account either solely for the long term variation of Internet workload, or solely for the short term variation

Table 3.1: Summary of Server and Workload Management Schemes for Internet Applications at the Local Data Center Level.

Articles & solutions	Switch cost	Cool-comp power tradeoff	Tech. trend	Alg. optimal.	Energy buffering	Renew.	Long term	Short term
Non thermal-aware server provisioning								
[30, 31]	✗	✗	✗	✓	✗	✗	✓	✗
[65, 67, 74]	✓	✗	✗	✓	✗	✗	✓	✗
[55, 65, 81, 82]	✓	✗	✗	✓	✗	✗	✗	✓
Thermal-aware server provisioning								
[94, 104]	✗	✗	✗	✗	✗	✗	✓	✓
[44]	✗	✓	✗	✗	✗	✗	✓	✓
Energy buffering [50, 61, 92, 116, 121]	✗	✗	✗	✓	✓	✗	✓	✓
Green solutions [47, 48, 72, 78, 100, 103]	✗	✗	✗	✗	✓	✓	✗	✓
Our solution: TACOMA [16, 17, 117]	✗	✓	✓	✓	✗	✗	✓	✓

of the workload. TACOMA is a multi-tier energy management scheme considering the long and short term aviation of the workload, and it is shown to reap the energy usage benefits from the inherent variabilities of workload at both long and short term. TACOMA performs proactive server management on hourly basis, where the server switching cost becomes negligible.

As shown in Table 3.2, our global workload and server management solutions contribute against the state of the art solutions as follows. First, the existing cost

Table 3.2: Summary of Electricity Cost-aware Server and Workload Management Schemes for Internet Applications at the Global Data Center Level.

Articles and solutions	Energy buffering	Pred. error	Carbon capping	Peak power shaving	Algorithm optimality	Opt. distributed alg.
Cost optimization [71, 73, 77, 96, 97, 123]	✗	✗	✗	✗	✓	✓
Cost and renewable usage optimization [19, 76, 77, 77, 108, 125]	✓	✗	✗	✗	✓	✓
Cost optimization and carbon capping [39, 46, 69, 127]	✗	✗	✓	✗	✗	✗
Cost optimization energy buffering [43]	✓	✗	✗	✓	✗	✗
Our solutions [11, 12, 14, 15], DAHM,OnlineCC,& OnCMCCLyp	✓	✓	✓	✓	✓	✓

optimization solutions (from Table 3.2 see the corresponding row) simplified the problem by assuming zero migration overhead for the workload and an integer value for the number of active servers, both of which are addressed in our DAHM solution (see Chapter 4). Second, the existing carbon capping solutions (from Table 3.2 see cost optimization and carbon capping solutions) are heuristics in the sense that they manage the carbon cap in a best effort manner. Our OnlineCC solution provides a bound on the maximum carbon capping violation that the solution yields in the worst case (see Chapter 6). The salient feature of this bound is that it can be estimated without the need to solve the offline solution. Third, the existing solutions are piecemeal in the sense that each of them addressed either of peak power shaving, energy cost minimization, energy buffering and carbon capping, but not all of them together. We argue that there is a need for a holistic approach that combines all the available leverages. Such a holistic management, however, introduces new challenges in terms of simultaneously handling all of the coupling factors to mange the dynamics of batteries, carbon capping and peak power shaving. The existing online algorithms are often designed to address each of the aforementioned coupling factors separately, disregarding their management implications on each other and the practical considerations. These problems are addressed in OnCMCCLyp solution (see Chapter 7). Finally, there exists work on peak power shaving of data centers (from Table 3.1 see energy buffering solutions and from Table 3.2 see cost optimization and energy buffering solution) which rely on the predictability of data center parameters over a window of future time, disregarding the impact of the prediction error on the cost efficiency of the solution. However, peak power cost is very sensitive to the prediction error which is studied in this work. The thesis also proposed a stochastic programming based solution to remove the harmful impact of the prediction error in increasing the peak power cost (see Chapter 7).

Chapter 4

THERMAL-AWARE ENERGY MANAGEMENT AT LOCAL DATA CENTER LEVEL

This chapter presents TACOMA, a two-tier Internet data center management scheme, TACOMA, with *thermal-aware server provisioning* (TASP) in one tier, and *thermal-aware workload distribution* (TAWD) in the other. TASP and TAWD co-ordinate to maximize the energy savings by leveraging the workload dynamics, at coarse and fine time scale, respectively. TACOMA is aware of the QoS constraints, the energy proportionality of servers, and the potential tradeoff between cooling and computing power. The obtained energy savings are a combination of suspending idle servers, using servers at their peak efficiency, and avoiding heat recirculation.

TACOMA accounts for energy proportionality of servers by using the IPR and LDR metrics defined in Chapter 2. This chapter first gives an overview on the thermal aware scheduling and provide an analytical study on the occurrence of cooling-computing power tradeoff due to server consolidation (Section 4.1). Then the chapter formally describes TACOMA and gives an overview of the performance and power consumption modeling of a data center (Section 4.2). The chapter then formally defines TASP and TAWD problems in Section 4.3. Further, Section 4.4 presents the analysis of these problems and introduces the optimal and heuristic solutions for TASP and TAWD. Furthermore, Section 4.6 presents the simulation-based evaluation of TASP and TAWD under different energy proportionality levels of servers and different PUEs of data centers and discuses the results. Finally, the chapter concludes and summarizes the results in Section 4.7.

Table 4.1: Symbols to Model TASP and TAWD

Symbol	Definition
Y	total number of computing nodes
$y(t)$	minimum required number of active servers
L	number of slots in a given epoch
i	index of nodes
t	index of epochs
k	index of slots
f	function to map utilization to power
p_i^{idle}	idle power of node i
p_i^{util}	power gradient of node i w.r.t. utilization
h_{ij}	heat dissipated from node j to node i
c_i	computing capacity of node i
C^A	coefficient of variation of workload arrival rate
C^B	coefficient of variation of request size
λ_i^{thres}	max. affordable workload of server i
λ^{peak}	peak request arrival rate
μ^{ref}	reference service rate
T^{sup}	CRAC supplied temperature
T^{red}	Servers' redline temperature
T^{in}	Servers' inlet temperature
A	set of all servers
$A'(t)$	active server set
$\mathbf{w}$	vector $p_i^{idle}{}_{\{Y\}}$
$\mathbf{a}$	vector $p_i^{util}{}_{\{Y\}}$
$\mathbf{u}(t)$	utilization vector $u_i(t)_{\{Y\}}$
$\mathbf{p}(t)$	computing power vector $p_i(t)_{\{Y\}}$
$\mathbf{rk}$	servers' ranking vector
$\mathbf{H}$	heat recirculation matrix $\{h_{ji}\}_{(Y \times Y)}$
P^{comp}	data center total computing power
P^{AC}	data center cooling power
E^{total}	data center total energy consumption
E^{AC}	total cooling energy
TASP	Thermal Aware Server Provisioning
TAWD	Thermal Aware Workload Distribution
CPSP	Computing Power aware Server Provisioning
NoSP	No Server Provisioning
LRH	Least Recirculated Heat
CPLRH	Computing Power LRH hybrid

Notations: The data center models used in this chapter are consistent with the aforementioned definitions and models in Chapter 2. Nevertheless, this chapter uses some additional notations to describe the heat recirculation and the computing nodes' heterogeneity (e.g., in terms of service rate) and the two tier server and workload management solution which are given in Table 4.1.

4.1 Characterizing Cooling-computing Power Tradeoff

This section studies the effect of energy proportionality and thermal impact of servers on energy (i.e., cooling and computing energy) savings of the workload and server management in data centers. The section focuses on the systems with power consumption of linearly increase over utilization(i.e., zero LDR) and the power model of the form: $p = p^{util}u + p^{idle}$, where IPR=$\frac{p^{idle}}{p^{util}+p^{idle}}$, and $0 \leqslant u \leqslant 1$ denotes the utilization. Using the data center heat recirculation model and the servers' power consumption model, the section characterizes the conditions where server provisioning causes cooling- computing power tradeoff. We remove the index time t throughout this section for brevity.

4.1.1 Data Center Heat Recirculation Modeling

This section describes the heat recirculation model and gives an overview on how the cooling energy is affected by the workload and server management. The study is performed for the physical layout of the contemporary air-cooled data centers with non-uniform heat recirculation as described in Chapter 2. It is assumed that, the temperature of the supplied cooled air, denoted as T^{sup}, should be low enough so that the inlet temperature of the computing nodes does not go beyond the red line temperature (T^{red}) which is specified by the manufactures. This thesis bases the data center heat recirculation according to the model in [110]. Tang et al. model the heat recirculation coefficients for all pairs of nodes in a data center, considering the data center layout and thermodynamic conditions: $\mathbf{H}$=$\{h_{ij}\}_{Y\times Y}$, where Y is the total number of nodes, and each h_{ij} denotes the fraction of heat that flows from node j to node i [110]. Assume $\mathbf{p}$ is a vector denoting the computing power consumption of servers, then T^{in}, the inlet temperature vector of servers can be written as:

$$T^{in} = T^{sup} + H\mathbf{p}. \tag{4.1}$$

Eq. 4.1 shows that the inlet temperatures of servers depend on the power consumption of nodes, and consequently on the amount of workload they are assigned. Indeed, workload assignment is hidden in the computing power vector ($\mathbf{p}$). On the other hand, cooling energy depends on T^{sup}. Cooling energy of the CRAC can be modeled by its *coefficient of performance* (CoP), which is the ratio of the heat removed (i.e., computing energy) over the work required to remove that heat (i.e., cooling energy). A higher CoP means more efficient cooling, and usually the higher the required supplied temperatures (T^{sup}), the better the CoP is. In other words, CoP is usually monotonically increasing function of the supplied temperature, e.g., for an HP data center a polynomial approximation of CoP for positive temperature reported as $CoP(T^{sup}) = 0.0068T^{sup2} + 0.0008T^{sup} + 0.458$ [88]. However, according to Eq. 4.1, the highest CRAC output temperature is limited by the servers' *redline temperature*. Therefore, T^{sup} can be *at most* equal to:

$$T^{sup} = T^{red} - \max(H\mathbf{p}^{comp}), \tag{4.2}$$

where the function max ensures that the supplied temperature of CRAC does not exceed the redline temperature of the hottest equipment. Respectively, the cooling power, denoted by P^{AC}, can be written as a function of the CoP of the supplied temperature:

$$P^{AC} = \frac{P^{comp}}{CoP(T^{red} - \max(H\mathbf{p}^{comp}))}, \tag{4.3}$$

where P^{comp} denotes the total computing power. Eq. 4.3 suggests that for *a given load*, the cooling power can be potentially improved by efficient workload distribution. Intuitively, this is possible if the workload distribution is thermally balanced among the servers, meaning that a higher portion of the workload is assigned to the servers that have the least contribution on the heat recirculation. This section refers to those servers as *thermally efficient* servers. Since, the main contributors of the total power in a data center are the computing and the cooling power, a high PUE implies the high heat recirculation in a data center.

Cooling power is also affected by server consolidation. The computing power,

p, of a server is a function of the idle power and the peak power, and the server's utilization level, such that by using a server consolidation scheme, e.g., CPSP, the total idle power decreases. However, due to workload consolidation effect of the scheme, the computing power of active servers also increases, which may increase the value of $\max(H\mathbf{p})$ in Eq. 4.3, i.e., hot spots may be created which increase the required cooling power. In other words, under a server consolidation scheme such as CPSP, a tradeoff between cooling and computing (idle) power can occur [44]. The following section provides analysis to illustrate the conditions for this tradeoff to occur.

4.1.2 Analytical Study of Cooling-computing Power Tradeoff

This section presents a formal definition of the cooling-computing power tradeoff and prove some lemmata to: (i) clarify the conditions when the tradeoff occurs, and (ii) provide an easy-to-solve analytical method to examine the occurrence of the tradeoff. The section also provides some discussion on the tradeoff's implication on the energy saving of the workload and server consolidation schemes. The lemmata are given for the special case of server/workload consolidation, namely balanced workload consolidation as defined below[1].

Definition 1. *In a homogeneous data center, any workload consolidation from Y servers with balanced utilization of u down to $y < Y$ servers with a balanced utilization of $u' > u$ such that the total utilization remains unchanged, i.e., $Yu = yu'$, is called* Balanced Workload Consolidation, BWC(Y, u, y, u'). *This definition also assumes that the remaining $Y - y$ servers, being relieved of the workload, are switched-off.*

Definition 2. *For a fixed workload if a consolidation in a data center decreases the computing power from P^{comp} to $P^{comp\prime}$, and the cooling power increases from P^{AC} to*

[1] In this section,“$'$” denotes parameter values after consolidation, “$''$” and “$*$” denote the parameter values of a particular consolidation scheme.

$P^{AC'}$*, then a* cooling-computing power tradeoff *is said to exist.*

Definition 3. *In a matrix* $H = \{h_{ij}\}_{Y\times Y}$*, a row* i *in which* $\exists j,k:\ h_{ik} \neq h_{ij}$*, is called a* non-uniform *row. A matrix with only non-uniform rows, is called a* strictly row-wise non-uniform *matrix.*

Definition 4. *A row-wise uniform matrix has no non-uniform rows.*

Lemma 4.1.1. *Consider a homogeneous data center* $\mathcal{H}$ *with cooling whose power adheres to Eq. 4.3, strictly row-wise non-uniform and positive heat recirculation matrix of* $H = \{h_{ij}\}_{Y\times Y}$*, and a CoP that is a strictly monotonically increasing function of the supplied temperature* T^{sup}*. There exist a BWC* (Y, u, y, u') *and an associated* $\varepsilon > 0$ *for* $\mathcal{H}$ *such that BWC causes cooling-computing power tradeoff for IPR* $< \varepsilon$*.*

Proof. Let $P^{comp} = Y(p^{util}u + p^{idle})$, and $P^{comp\prime} = y(p^{util}u' + p^{idle})$ be the computing power before and after the consolidation. Similarly, let P^{AC} and $P^{AC'}$ be the cooling power before and after the consolidation. According to Definition 1 $P^{comp} > P^{comp\prime}$. Hence, according to Definition 2 to show the existence of a cooling-computing power tradeoff for a consolidation, it is enough to show $P^{AC} < P^{AC'}$. A constructive proof follows.

According to Eq. 4.2 the CRAC's supplied temperature before consolidation, T^{sup}, is as follows: $T^{sup} = T^{red} - d(p^{util}u + p^{idle})$, where $h = \|H\|_\infty = \max\left\{\sum_{j=1}^{Y} h_{1j}, \ldots, \sum_{j=1}^{Y} d_{Yj}\right\}$[2]. After the consolidation BWC(Y, u, y, u') chooses the active server set $A' \subseteq \{1, \ldots, Y\}$, $|Y'| = y$, leading to $T^{sup\prime} = T^{red} - h'(p^{util}u' + p^{idle})$, where $h' = \|H\backslash A'\|_\infty$.

[2] $\|H\|_\infty$ is the *infinity norm* of a matrix H. Conventionally, the definition sums the absolutes of the matrix elements, which does not affect the usage in this chapter. Also, the notations are expanded to $\|H\backslash A'\|_\infty$,i.e., "*infinity norm of H* ***under*** A'," which equals to the submatrix of H that has only the columns that correspond to the servers in active set A', i.e.,$\|H\backslash A'\|_\infty = \max\{\sum_{j\in A'} h_{1j}, \ldots, \sum_{j\in A'} h_{1j}\}$.

Choose a BWC(Y, u, y, u'), say $\mathcal{C}$, with $h' > \frac{y}{Y}h$ and let $h' = \frac{y}{Y}h + \delta$, where $\delta > 0$. Note that such a BWC exists due to the row-wise non-uniformity of the heat recirculation matrix (Definition 3). Following shows that $\mathcal{C}$ satisfies the lemmas' conditions:

- **Hypothesis 1.** For the given $\mathcal{C}$, ($\exists\varepsilon_1 > 0$: IPR $< \varepsilon_1$) $\Leftrightarrow T^{sup} > T^{sup\prime}$. Note $T^{sup} > T^{sup\prime}$ is a necessary condition for the occurrence of the tradeoff. Hypothesis 1 can be proved as follows:

$$\begin{aligned}
T^{sup} > T^{sup\prime} &\Leftrightarrow h(p^{util}u + p^{idle}) < h'(p^{util}u' + p^{idle}) \qquad \text{[from Eq. 4.2]} \\
&\Leftrightarrow h(p^{util}u'\frac{y}{Y} + p^{idle}) < (h\frac{y}{Y} + \delta)(p^{util}u' + p^{idle}) \\
&\Leftrightarrow hp^{idle} < \frac{y}{Y}hp^{idle} + \delta(p^{util}u' + p^{idle}) \\
&\Leftrightarrow \frac{p^{idle}}{p^{util} + p^{idle}}h(1 - \frac{y}{Y}) < \frac{\delta(p^{util}u' + p^{idle})}{p^{util} + p^{idle}} \\
&\Leftrightarrow \text{IPR} < \frac{\delta(p^{util}u' + p^{idle})}{d(1 - \frac{y}{Y})(p^{util} + p^{idle})}. \qquad (4.4)
\end{aligned}$$

  Let $\varepsilon_1 \equiv \frac{\delta(p^{util}u' + p^{idle})}{h(1-\frac{y}{Y})(p^{util}+p^{idle})}$. Since $y > Y$, it follows that $\varepsilon_1 > 0$, establishing Hypothesis 1.

- **Hypothesis 2.** For the given $\mathcal{C}$, ($\exists\varepsilon_2 > 0 :$ IPR $< \varepsilon_2$) $\Leftrightarrow P^{AC} < P^{AC\prime}$. Note $P^{AC} < P^{AC\prime}$ is a sufficient condition for the occurrence of the tradeoff. Due to $Yu = yu'$, and the linearity of the power consumption model, it follows:

$P^{comp} = n(p^{util}u' + p^{idle}) + (Y-y)p^{idle}$. Hypothesis 2 can be proved as follows:

$$\begin{aligned}
P^{AC} < P^{AC'} \Leftrightarrow & \frac{y(p^{util}u' + p^{idle}) + (Y-y)p^{idle}}{CoP(T^{sup})} < \frac{y(p^{util}u' + p^{idle})}{CoP(T^{sup\prime})} \\
\Leftrightarrow & \frac{y(p^{util}u' + p^{idle})/(p^{util} + p^{idle}) + (Y-y)p^{idle}/(p^{util} + p^{idle})}{CoP(T^{sup})} \\
< & \frac{y(p^{util}u' + p^{idle})/(p^{util} + p^{idle})}{CoP(T^{sup\prime})} \\
\Leftrightarrow & \frac{y(p^{util}u' + p^{idle})}{p^{util} + p^{idle}} + (Y-y)IPR \\
< & \frac{CoP(T^{sup})y(p^{util}u' + p^{idle})/(p^{util} + p^{idle})}{CoP(T^{sup\prime})} \\
\Leftrightarrow & \text{IPR} < (\frac{CoP(T^{sup})}{CoP(T^{sup\prime})} - 1)\frac{y(p^{util}u' + p^{idle})}{(Y-y)(p^{util} + p^{idle})}. \qquad (4.5)
\end{aligned}$$

Let $\varepsilon_2 \equiv (\frac{CoP(T^{sup})}{CoP(T^{sup\prime})} - 1)\frac{y(p^{util}u' + p^{idle})}{(Y-y)(p^{util} + p^{idle})}$. According to Eq. 4.4 and monotonicity property of the CoP function, it follows that $\frac{CoP(T^{sup})}{CoP(T^{sup\prime})} > 1$. Also $Y > y$, which establishes $\varepsilon_2 > 0$.

Let $\varepsilon = \min(\varepsilon_1, \varepsilon_2)$. Since $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$, it follows that $\varepsilon > 0$. □

The following lemma shows that a necessary condition for the occurrence of cooling-computing power tradeoff is non-uniform and non-zero heat recirculation.

Lemma 4.1.2. *In a data center with row-wise uniform heat recirculation $H = \{h_{ij} = h_i\}_{Y\times Y}$, homogeneous servers of LDR=0, cooling power of Eq. 4.3, and a CoP of strictly monotonically increasing function of supplied temperature T^{sup}, a $BWC(Y, u, y, u')$ never causes cooling-computing power tradeoff.*

Proof. According to Definition 1, the cooling-computing power tradeoff occurs if $P^{comp} > P^{comp\prime}$, and:

$$P^{AC} < P^{AC'} \Leftrightarrow \frac{P^{comp}}{CoP(T^{sup})} < \frac{P^{comp\prime}}{CoP(T^{sup\prime})}.$$

Since, $P^{comp} > P^{comp\prime}$, the necessary condition for the occurrence of the tradeoff is $CoP(T^{sup}) > CoP(T^{sup\prime})$, which means that there must exist $h > 0$ such that

$T^{sup} = T^{sup\prime} + h$ (due to strictly monotonically increasing of CoP as a function of supplied temperature, $h > 0$). The lemma can be proved through contradiction as follows.

According to Eq. 4.2 the CRAC's supplied temperature before consolidation, T^{sup}, is as follows: $T^{sup} = T^{red} - h(p^{util}u + p^{idle})$, where $h = \|H\|_{\infty}$. Denote $h_* \equiv h/Y$ the maximum element in the matrix. After the consolidation BWC(Y, u, y, u'), the supplied temperature becomes $T^{sup\prime} = T^{red} - d'(p^{util}u' + p^{idle})$. Since, $Yu = yu'$, it follows that:

$$\begin{aligned}
&T^{sup} = T^{sup\prime} + h \Leftrightarrow T^{red} - Yh_*(p^{util}u + p^{idle}) = T^{red} - yh_*(p^{util}u' + p^{idle}) + h \\
&\Leftrightarrow h_*(p^{util}yu' + Yp^{idle}) = h_*y(p^{util}u' + p^{idle}) - h \\
&\Leftrightarrow h_*Yp^{idle} = h_*yp^{idle} - h \Leftrightarrow h = h_*y - h_*Y.
\end{aligned} \tag{4.6}$$

Since $y \leq Y$, $h \leq 0$, which contradicts the fact that $h > 0$. □

Theorem 4.1.3. *Consider a homogeneous data center with cooling whose power adheres to Eq. 4.3, a CoP that is a strictly monotonically increasing function of the supplied temperature T^{sup}, a strictly positive heat recirculation matrix H, there exist a BWC(Y, u, y, u') and an associated $\varepsilon > 0$ to cause cooling-computing power tradeoff when IPR $< \varepsilon$ if and only if there are y elements in one non-uniform row whose sum is greater than y times the maximum element in the uniform rows.*

Proof. Let $h = \|H\|_{\infty}$ before consolidation, and $h' = \|H \backslash A'\|_{\infty}$ after some arbitrary BWC(Y, u, y, u') chooses the active server set A'. The two directions of "if and only if" are proven in separate parts:

"if" Given that there are y elements in a row whose sum is greater than y times the maximum element in the uniform rows, it is possible to construct the BWC(Y, u, y, u') such that it yields $h' > \frac{y}{Y}h$, by setting the active server set

A' to contain the y servers that correspond to those y elements in that non-uniform row. By use of the proof steps in Lemma 4.1.1, it can be easily shown that there exists an $\varepsilon > 0$ such that if IPR $< \varepsilon$, the constructed BWC causes cooling-computing power tradeoff.

"only if" Given that there are no y elements in any non-uniform row whose sum is greater than y times the maximum element in the uniform rows, h' must be one of the uniform row sums. Therefore $h' = \frac{y}{Y}h$. By use of the proof steps in Lemma 4.1.2, it can be easily shown that the arbitrary BWC(Y, u, y, u') causes no cooling-computing tradeoff under any IPR.

□

The above theorem provides a link between the IPR of servers and the structure of the heat recirculation matrix to the *existence* of cooling-computing power tradeoff. Specifically, the inequality in Eq. 4.5 provides a sufficient condition for the existence of the cooling-computing power tradeoff. The lemma below provides a way to find a balanced workload consolidation that is close (by a bound) to the one that minimizes the supplied temperature. In general, the lower the $T^{sup'}$ of a consolidation is at a certain IPR, because of the tradeoff, the more likely that consolidation is to still cause the tradeoff if the IPR is increased.

Lemma 4.1.4. *Consider a homogeneous data center with zero-LDR servers $A = \{1, 2, \ldots, Y\}$ and positive heat recirculation coefficient matrix $H = \{h_{ij}\}_{Y\times Y}$. The lowest supplied temperature, $T^{sup*'}$, by any BWC(Y, u, y, u'), is bounded by the inequality:*

$$T^{red} - Yh''(p^{util}u' + p^{idle}) \leqslant T^{sup*'} \leqslant T^{red} - h''(p^{util}u' + p^{idle}),$$

where $h'' = \|H \backslash A''\|_\infty$*,* $A'' \subseteq A$*,* $|A''| = y$*, and* $\forall j \in A''$ *and* $\forall k \notin A''$*,* $\sum_{i=1}^{Y} h_{ij} \geqslant \sum_{i=1}^{Y} h_{ik}$*.*

Proof. The CRAC's supplied temperature when using workload consolidation, denoted by $T^{sup\prime}$, depends on the active server set. Let A'_k, for $1 \leqslant k \leqslant \binom{Y}{y}$ denote the k^{th} active server set of size y that can be chosen from Y servers. According to Eq. 4.2 there is a h'_k associated with every A'_k such that $T^{sup,k\prime} = T^{red} - h'_k(p^{util}u + p^{idle})$, where $h'_k = \|H \backslash A_k\|_\infty$. Note that A'' is one of the $\{A'_k\}$ active server sets, where $h'_k = h''$. Consider a balanced workload consolidation that chooses the active server set $A_*{}'$ which causes the lowest supplied temperature $T^{sup*\prime}$, its associated h'_* satisfies the following:

$$h'_* = \max\left\{h'_1, \ldots, h'_k, \ldots, h'_{\binom{Y}{y}}\right\} \geqslant h''. \tag{4.7}$$

According to definition of A'', it follows that $\sum_{i=1}^{Y}\sum_{j=1}^{Y} h_{ij} < \frac{Y}{y}\sum_{i=1}^{Y}\sum_{j \in A''} h_{ij}$. Also according to well-known *norm inequalities* it follows that:

$$h'_* \leqslant \sum_{i=1}^{Y}\sum_{j=1}^{Y} h_{ij} \leqslant \frac{Y}{y}\sum_{i=1}^{Y}\sum_{j \in A''} h_{ij} \leqslant \frac{Y}{y}\left(n\,\|H \backslash A''\|_\infty\right) = Yh''. \tag{4.8}$$

Eqs. 4.7 and 4.8 establish the lemma. □

In the above lemma A'' can be found through calculating the summation of columns over matrix H, sorting them in ascending order and picking y columns whose sum is maximum. This scheme is used for implementing CPSP in the evaluation section.

Based on the above, the following conclusion can be made:

1. *The lower the yielded* $T^{sup\prime}$ *of a consolidation at a certain IPR, because of the cooling-computing power tradeoff, the higher the IPR has to increase before that consolidation ceases to cause the tradeoff.*

2. *High IPR and low PUE disfavor the occurrence of cooling-computing power tradeoff, whereas low IPR and high PUE favor the occurrence of the tradeoff.* Assume[3] PUE=$1 + 1/CoP(T^{sup})$, then high PUE is indication of high heat recirculation. High heat recirculation increases the value for $\frac{CoP(T^{sup})}{CoP(T^{sup\prime})}$ where T^{sup} and $T^{sup\prime}$ denote the supplied temperature of without and with workload consolidation. According to Eq. 4.5 higher ratios of $\frac{CoP(T^{sup})}{CoP(T^{sup\prime})}$ expands the range of IPR values (i.e., one can find a higher ε) where the tradeoff may occur.

3. *Performing a balanced workload consolidation may be detrimental in improving the energy efficiency of a data center.* According to Lemma 4.1.1, there can be cases where a workload consolidation causes cooling-computing power tradeoff, then it follows that if the magnitude of cooling power increase outweigh computing power decrease, workload consolidation is detrimental in improving the energy efficiency of a data center compared to no workload consolidation. This may happen due to the high heat recirculation in data centers and low IPR of servers.

4. *A balanced workload consolidation scheme that chooses the active server set among thermal efficient servers, will mitigate the cooling-computing power tradeoff.* Such a scheme will choose servers to minimize heat recirculated to the servers. Hence, $\frac{CoP(T^{sup})}{CoP(T^{sup\prime})}$ decreases, disfavoring the occurrence of cooling-computing power tradeoff.

[3] Note that PUE characterizes the overall power efficiency of a data center with respect to its computing power whereas CoP characterizes the efficiency of CRAC unit. Specifically, PUE=$\frac{\text{Total power}}{\text{computing power}}$. However, for simplification, TACOMA ignores non-computing power other than that of the cooling system.

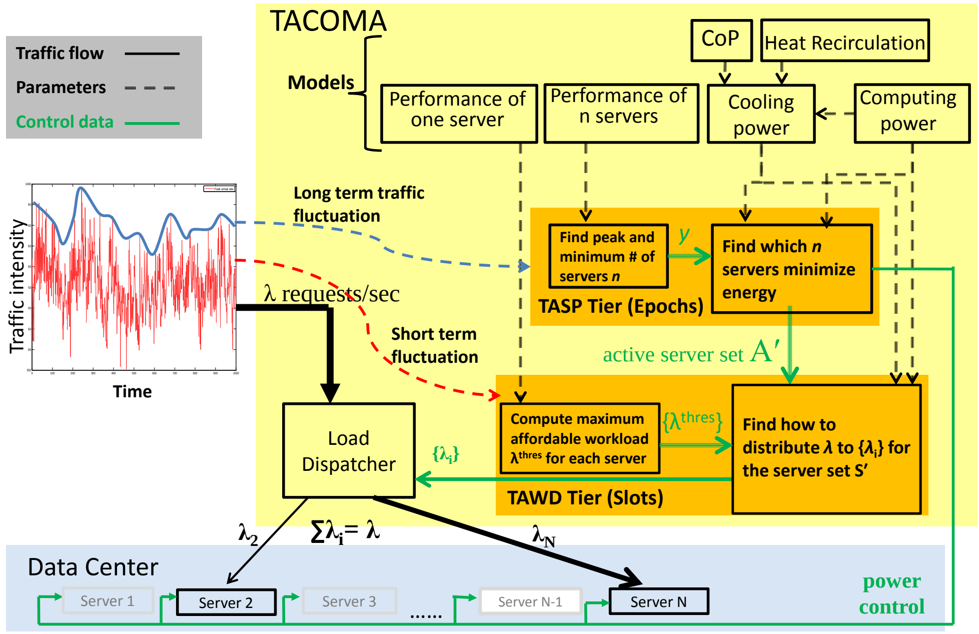


Figure 4.1: TACOMA a Two-tier Architecture for Thermal-aware Server and Workload Management of IDCs. The first tier, TASP, on each epoch determines the minimum number of active servers and chooses the active server set A' to minimize the total energy, and the second tier, TAWD, on each slot decides on the workload distribution, λ_is, of the active servers to minimize the energy consumption of active servers.

4.2 TACOMA Architecture and Models

This section develops TACOMA, a two-tier, global-view, centralized control software architecture as shown in Fig. 4.1. Tier 1 (T1), the server provisioning tier, iteratively decides the active server set in coarse-time intervals called *epochs*. Let $A=\{s_i\}$, $i=1\ldots N$ be the server set. The T1 controller at the beginning of each epoch t estimates $y(t) \leqslant Y$ minimum number of required active servers and chooses the active server set $A'(t)$, where $A'(t) \subseteq A$, $A'(t) = \{s_i, 1 \leqslant i \leqslant Y\}$, and $y(t) \leqslant |A'| \leqslant Y$. Due to the overhead of removing the servers from the active set, e.g., power control and releasing reserved computing resources, an epoch is assumed to be around half hour. Fig. 4.1 shows an example that for a given workload in an epoch, T1 controller determines that two servers are required to be active. Then T1 based on power and thermal efficiency of servers specifies active servers for which the least energy consumption is incurred and minimum server requirement is met (in this example servers s_2 and s_Y are selected and servers $s_1, s_3, s_4, \ldots s_{Y-1}$ are suspended). The controller of Tier 2 (T2), the workload distribution tier, operates at fine-time intervals called *slots* (around 1-10 seconds) and decides on the distribution (i.e., partitioning) of the workload among the active servers. Let the average request arrival rate at the k^{th} slot of epoch t be $\lambda(t,k)$; T2 controller determines $\lambda_i(t,k)$, for all $s_i \in A'(t)$ where $\sum_{s_i \in A'} \lambda_i(t,k) = \lambda(t,k)$ such that the performance requirement is met and the energy consumption is minimized. In other words, number of active servers determined by T1 is adjusted by the peak traffic during an epoch. However, in most slots during an epoch, traffic is much lower than the peak traffic. In these cases, T2 shifts traffic toward energy efficient servers and tries to utilize them at near their maximum energy efficiency level. For example consider Fig. 4.1 where T2 is going to distribute workload of a given slot to s_2 and s_Y where the intensity of traffic is lower than the total capacity of servers s_2 and s_Y. In this case, T2 gives more traffic to the energy efficient server (s_Y) and utilizes it near its maximum energy efficiency level in such a way that it is not overloaded. We assume each epoch consists of L slots. Also for notation brevity we assume slots length of one unit of time.

4.2.1 Energy Consumption Modeling

Energy consumption of a data center is the sum of its cooling and computing energy:

$$E^{total}(t) = E^{comp}(t) + E^{AC}(t). \tag{4.9}$$

Typically, there are other sources of energy consumption in a typical data center such as lighting that are: (i) irrelevant to computing energy and (ii) are insignificant. Hence, they are disregarded. To calculate computing energy, each servers' computing power is modeled as follows:

$$p_i(t) = p_i^{idle} + f(u_i(t)),$$

where p_i^{idle} denotes power consumption of server i at idle state, and f is a function that maps the utilization to the power consumption. This chapter uses two linear and nonlinear model for the function f. The linear model is the basis of the analytical study and is chosen for the sake of simplicity and driving optimal solutions. In this model the total power consumption of active servers ($\forall s_i \in A'(t)$), at epoch t, having utilization of $u_i(t)$ can be written as:

$$P^{comp}(t) = \sum\nolimits_{s_i \in A'} (p_i^{idle} + p_i^{util} u_i(t)), \tag{4.10}$$

where p_i^{util} represents extra power consumption at full utilization for each server i. Applying Eqs. 4.3 and 4.10 to Eq. 4.9, the total energy consumption at epoch t becomes:

$$E^{total}(t) = \left(1 + \frac{1}{\mathrm{CoP}\,(T^{\mathrm{red}} - \max\,(\mathbf{H}(\mathbf{w} + \mathbf{a} \odot \mathbf{u}(t))))}\right) \sum_{i=1}^{Y} (p_i^{idle} + p_i^{util} u_i(t)) L, \tag{4.11}$$

where, $\mathbf{w}$ and $\mathbf{a}$ denote vector form of the scalar computing power parameters and $\mathbf{u}(t)$ represents vector form of utilization of servers. Also the operator $\odot$ is defined such that $\mathbf{a} \odot \mathbf{u}(t)$ is vector $\langle p_i^{util} u_i(t) \rangle$.

This chapter also uses a nonlinear model of the function f in the simulation study (Section 4.6.2), which is based on the approximation of the power-utilization curve of recent systems.

4.2.2 Performance Modeling

TACOMA uses $G/G/n$ queuing model (i.e., Eq. (2.2)) to determine how many active servers to provision for each epoch as well as how much workload to assign for each active server during slots to satisfy the performance requirement of requests (i.e., response time). The model is a n-server queuing system serving requests with generalized arrival and service time distribution and has been previously used in the literature for QoS-cognizant server provisioning [44, 67, 115].

Given the average arrival rate, λ, and service rate, μ, the $G/G/n$ model is used to calculate the minimum number of required active servers to achieve the desired response time.

Consider an epoch t where the average arrival rate of requests in its L slots is as follows: $\{\lambda_1, \lambda_2, \ldots, \lambda_L\}$. To ensure that the response time is met in all the slots of e[4], the number of active servers for t should be determined according to $\lambda^{peak}(t) = \max\{\lambda_1, \lambda_2, \ldots, \lambda_L\}$. Using the existing prediction mechanisms, the average arrival rate $\lambda(t)$ of epoch t over all the slots can be predicted. It represents the smooth variation of workload over time. TACOMA uses Kalman filtering to predict $\lambda(t)$. However, the arrival rate during peak time (i.e., $\lambda^{peak}(t)$) is not easy to predict.

Analyzing some Internet traces, it is observed that there is a strong correlation between the average arrival rate λ and the peak arrival rate λ^{peak} in an epoch (Section 4.6). For this analysis, *Minitab* is used to perform a regression analysis [86] on two weeks of the web trace of 1998 FIFA World Cup [58] and two days of a web trace of Microsoft Hotmail [113] and build a regression linear model of the form: $\lambda^{peak} = \beta_1\lambda + \beta_0$. The results show that R-square predicted (i.e., R-sq(pred)) for

[4] Today's data centers performance criteria focus on the worst case or 95th-percentile response time. For that, the active server set is provisioned according to the response time at peak traffic which is the worst case during an epoch.

both the workloads are high, (77.45%, and 93.83% for Hotmail and FIFA traces, respectively), and the corresponding *P-value* for both of the *t-test* and the *test for significance of regression* (i.e., F0) are almost zero (it was smaller than 0.0001). These results indicate that $\lambda^{peak}(t)$ can be linearly predicted through $\lambda(t)$.

Service rate depends on servers' physical characteristics and the size of requests. To consider the heterogeneity of the servers and model the service rate of servers, two parameters are defined: a *reference computing capacity* and a *reference request size.* The *reference computing capacity* represents the computing capacity of the server with the minimum capacity which is normalized to one. The computing capacity c_i of server i is expressed as an integer factor of the reference computing capacity. This assumption is reasonable, since the computing capacity of modern multi-core servers are mainly affected by their core counts. The *reference request size* represents a block of data where the reference server can handle requests of that size with the rate of μ^{ref}. Therefore, the service rate for each server can be modeled as a function of c_i, μ^{ref} and the requests size. For server i, c_i, and μ^{ref} are constant, and the statistics of requests size is updated over time, such that at the beginning of each epoch the most recent statistics are used.

To calculate the maximum affordable workload by each server, a server is modeled as $G/G/1$ whose average response time d is:

$$d = \frac{1}{\mu_i} \frac{u_i}{1 - u_i} \frac{C_{Ai}^2 + C_{Bi}^2}{2}, \tag{4.12}$$

where the index i ties the queuing system variable to server i. Using this model, the maximum affordable workload arrival rate of serve i at epoch t and slot k, $\lambda_i^{\text{thres}}(t,k)$, to achieve the desired response time can be calculated by computing the $u_i^{thres}(t)$ corresponding to the desired response time and using $\lambda_i^{thres}(t,k) = \mu_i u_i^{thres}(t,k)$.

4.3 TACOMA Optimization Frameworks

The formulation of TASP and TAWD as optimization problems is based on combining the aforementioned models to express the energy consumption as a function of the active server set and the workload distribution. For both TASP and TAWD there are separate formulations for homogeneous and heterogeneous data centers. Although the homogeneous case is a sub-case of the heterogeneous one, its formulation is presented for the following reasons: (i) it can be solved by a heuristic with a known approximation bound, and (ii) that heuristic is used to build a heuristic for the heterogeneous case.

4.3.1 T1: Thermal-aware Server Provisioning (TASP) Problem

This section first formalizes the general (heterogeneous data center) case and then presents the formalization for the homogeneous data center case.

Given a data center with the server set A of Y servers, for epoch t of length L, where L is the number of one-unit-length slots in an epoch, the average arrival rate $\lambda(t)$, the peak arrival rate $\lambda^{peak}(t)$, and the minimum number of required servers $y(t)$, choose the active set $A'(t) \subseteq A$, where $y(t) \leqslant |A'(t)| \leqslant Y$, that minimizes the total energy $E^{total}(t)$.

Let the binary vector $\mathbf{x}$ denote the choice of the servers in the active set. TASP can be represented in the following optimization problem of *finding the vector* $\mathbf{x}$:

$$\text{minimize } (1+\frac{1}{\mathrm{CoP}(T^{red}-\max(\mathbf{H}(\mathbf{x}\odot(\mathbf{w}+\mathbf{a}\odot\mathbf{u}))})\sum_{i=1}^{Y} x_i(p_i^{idle}+p_i^{util}u)L,$$

subject to:

$$\begin{aligned}
&\sum_{i=1}^{Y} c_i x_i \geqslant n \qquad \text{[performance constraint]} \qquad (4.13)\\
&u=\frac{\lambda^{peak}}{\mu\sum_{i=1}^{Y} c_i x_i} \qquad \text{[load balancing constraint]}\\
&x_i\in\{0,1\},\quad \forall i=1\dots Y.
\end{aligned}$$

The above problem is nonlinear and has $Y{+}1$ variables. The vector $\mathbf{x}$ of size Y is binary and the variable u denoting the average utilization of servers is a continuous variable. Note that the load balancing constraint implies that the formulation assumes that the workload will be equally balanced among active servers. Also, note that the problem is on minimizing the total energy and not on minimizing the active server set size. It may happen that an optimal solution for one is not optimal for the other.

TASP Problem for Homogeneous Data Centers

In a homogeneous data center, where all nodes have the same computing efficiency (i.e., $c_i = c$, $\mu_i = \mu\ \forall i$), and the same computing power efficiency (i.e., $p_i^{util} = p^{util}$, and $p_i^{idle} = p^{idle}\ \forall i$), the TASP problem can be simplified as follows. Assume that $|A'(t)|$ is fixed, e.g., $|A'(t)| = y(t)$, which means that the active server set size, $|A'(t)|$, is known but the question is which servers should be chosen as the active server set. In this case, the summation part in the objective function in Eq. 4.13, representing the computing energy, becomes independent of the selection of servers and therefore a constant. The only part that depends on the server selection is $\max H(\mathbf{x}\odot(\mathbf{w}+\mathbf{a}\odot\mathbf{u}))$[5]. Hence, the

[5] Note that if $|A'(t)|$ is allowed to vary in the optimization process , then the optimization problem can not be simplified, since utilization varies (u) computing energy is no longer a constant value and

objective function in Eq. 4.13 can be simplified as follows:

$$\text{minimize} \;\; \max H(\mathbf{x}\odot(\mathbf{w}+\mathbf{a}\odot\mathbf{u})).$$

Let matrix $G = \{g_{i,j}\}_{Y\times Y}$ be such that $g_{i,j} = h_{i,j}(p_j^{idle} + p_j^{util}u)$, $\forall i, j = 1 \ldots Y$, where $u = \frac{\lambda^{peak}}{|A'(t)|\mu}$. By applying the same constraints as for the heterogeneous case, TASP can be written as the following optimization problem:

$$\begin{aligned} \text{minimize max} \quad & G\mathbf{x} \\ \text{subject to} \quad & \textstyle\sum_{s_i\in A'(t)} x_i = |A'(t)| \\ & x_i \in \{0, 1\}. \end{aligned} \tag{4.14}$$

Note that each element $z_{i,j}$ of Z represents the temperature rise of server i from server j. Also the sum of row i is the total temperature rise of server i. Although the form of the problem is simplified, it is not simplified computationally. This formulation is based on the assumption that $|A'(t)|$ is fixed, however this is not true. Section 4.4 explains how to use this problem iteratively to solve TASP for the homogeneous data center case. The following lemma provides a proof on the NP-hardness of this problem.

Lemma 4.3.1. *The TASP problem, i.e., Eq. 4.14, is NP-hard even if only two nodes are affected by heat recirculation of the data center's nodes : (i) all elements of matrix G are positive, (ii) G is an $\{2 \times Y\}$ matrix, and finally (iii) $\mathbf{x}$ is a binary vector.*

Proof. The 2-partition problem is reduced to TASP (similar to the NP-hardness proof of the min-max resource allocation problem [124]). In the 2-partition problem, a set I containing positive integers ($z_i \in Z_+$) is given and the question is: is there a subset $I' \subseteq I$ such that $\sum_{i\in I'} z_i = \sum_{i\in II'} z_i$? 2-partition problem is known to be NP-hard even if the set I contains even numbers and partitioning is subject to equal

both the computing and cooling energy should be involved in the minimization problem.

cardinalities (i.e $|I'| = |I|/2$). Given a 2-partition problem, a TASP problem with G as $\{2 \times Y\}$ matrix can be constructed as follows. let elements of A be:

$$g_{1j} = z_j$$
$$g_{2j} = \tfrac{2}{Y} \textstyle\sum_{i=1}^{Y} z_i - z_j$$

Assume two partitioned subsets are restricted to have equal cardinalities. Thus exactly $Y/2$ elements will be selected. If an optimal solution to the TASP problem, x^*, has:

$$x_i^* = 1 \quad i \in I',$$
$$x_i^* = 0 \quad \text{Otherwise},$$

and $|I'| = n/2$, then the total heat recirculated to node 1 is: $o_1 = \sum_{i \in I'} z_i$, and the total heat recirculated to node 2 is: $o_2 = Y/2(2/Y \sum_{i \in I} z_i) - \sum_{i \in I'} o_i = \sum_{i \in I/I'} o_i$. By definition $o_{TASP} = max\{o_1, o_2\}$. It is concluded that there exist a 2-partition with $|I'| = |I|/2$ if and only if TASP has an objective value $o_{TASP} = \sum_{i \in I'} z_i = \sum_{i \in I} z_i/2$. □

4.3.2 T2: Thermal-aware Workload Distribution (TAWD) Problem

Given an active server set $A'(t)$ for epoch t with length L, and the average request arrival rate of $\lambda(t,k)$ at the k^{th} slot, determine $\lambda_i(t,k)$, $\forall s_i \in A'(t)$, $\forall k = 1 \ldots L$, where $\sum_{s_i \in A'(t)} \lambda_i(t,k) = \lambda(t,k)$, such that the desired response time is achieved and the energy consumption for that slot is minimized.

Let $\lambda_i^{\text{thres}}(t,k)$ be the maximum workload arrival rate that server i at slot k can afford such that its average response time does not exceed the reference response time (see Eq. 4.12). TAWD can be represented as the following optimization problem on *finding the utilization vector of* $\mathbf{u}$:

minimize $E^{total}(t,k)$ on $A'(t)$ in Eq. 4.11,

subject to:

$$0 \leqslant u_i(t,k) \leqslant \frac{\lambda_i^{\text{thres}}(t,k)}{\mu_i}, \forall i | s_i \in A'(t) \text{ [performance constraint]} \tag{4.15}$$

$$\sum_{s_i \in A'(t)} u_i(t,k)\mu_i = \lambda(t,k) \text{ [capacity constraint].}$$

TAWD for the homogeneous case can be simplified as a min-max combinatorial optimization problem, similar to Section 4.3.1, and due to continuous variables (i.e., $u_i(t,k)$) it can be formulated as a linear program.

All problems in this section are NP-hard (except for TAWD for the homogeneous case). For this reason, some heuristics are proposed in the next section.

4.4 TASP Solutions

This section presents solutions to TASP. The solutions are devised for both heterogeneous and homogeneous data center cases. The section presents the proposed TASP solutions followed by a brief introduction to TASP reference solutions.

4.4.1 TASP Solution for Homogeneous Data Centers

Two methods, namely TASP using Mixed Integer Programming (TASP-MIP) and TASP using Least Recirculated Heat (TASP-LRH), are presented to solve TASP in the homogeneous data center case, as follows:

- TASP-MIP: The optimal active server set can be calculated by solving the mixed integer program of TASP for the homogeneous case (see Eq. 4.14) in an iterative fashion. As mentioned in Section 4.3.1, MIP modeling of the problem is only true when $|A'(t)|$ is fixed (because the variable u is then fixed, the computing energy is constant, and hence the latter can be removed from the optimization problem). Therefore, to cover all cases ($|A'(t)| = y(t) \ldots Y$), MIP is run in a loop starting from $|A'(t)| = y(t)$, i.e., the minimum number of required servers, toward $|A'(t)| = Y$, until the total energy no longer reduces (this means that increasing number of active servers no longer decrease the total energy).

- TASP-LRH: This solution is similar to TASP-MIP, except that instead of solving MIP iteratively, an *Y-approximation* algorithm [18] is used (to solve Eq. 4.14), where Y is the total number of servers (the approximation bound Y can be found similar to the proof in Lemma 4.1.4). This algorithm works as follows: (i) calculate the sum of each column in matrix G in Eq. 4.14, (ii) sort the sums in descending order, and (iii) pick $|A'(t)|$ servers (as per the iteration's number) whose corresponding sum is the lowest, and set their corresponding x_i to 1. The sum of each column is the contribution of the corresponding server to the heat recirculation. This algorithm is referred as Least Recirculated Heat (LRH) in the rest of the chapter. According to LRH, one can define the thermal efficiency rank of servers (i.e., the contribution of servers to the heat recirculation), denoted by $\mathbf{rk^{LRH}}$, as the summation of columns of matrix G which can be written as follows:

$$rk_i^{LRH} = \sum\nolimits_{j=1}^{y} h_{ji}(p_i^{util}u + p_i^{idle}), \tag{4.16}$$

  where u is calculated according to the load balancing constraint of Eq. 4.13. Using this metric to select the active servers is equivalent to the aforementioned method. LRH does not find the optimal solution because it minimizes the total heat recirculated instead of minimizing the maximum per-server heat recirculated.

Example 1: To understand how MIP and LRH work in a particular iteration, consider a simple example. Assume there are three servers and that two active servers are required. Assume the matrix G in Eq. 4.14 is as follows:

$$G = \begin{bmatrix} 0.3 & 0 & 0 \\ 0.2 & 0.1 & 0.1 \\ 0 & 0.2 & 0.3 \end{bmatrix}.$$

The LRH metrics (i.e., summation of columns) of these three servers are: $[0.5, 0.3, 0.4]$. Therefore, according to TASP-LRH, servers two and three are chosen as the active servers. Consequently the temperature rise vector is $[0, 0.2, 0.5]$ which is calculated from summing the rows of G after removing Column 1. However, the optimal active server set chosen by TASP-MIP consist of servers one and three where the temperature rise vector is $[0.3, 0.3, 0.3]$. It can be seen in this example that the maximum temperature rise for TASP-LRH is higher than that for TASP-MIP.

According to Eq. 4.16, the computation of LRH for one server is $O(Y)$. Since there are Y servers and the upper bound of iterations is $O(Y)$, the overall complexity of TASP-LRH is $O(Y^3)$.

Note both TASP-LRH and TASP-MIP are balanced workload consolidation (Definition 1) when they employ load balancing for workload distribution scheme.

4.4.2 TASP Solutions for Heterogeneous Data Centers

This section introduces solutions for the heterogeneous data center case that are TASP-MiniMax, a numerical solution, and TASP-CPLRH, a heuristic:

- TASP-MiniMax: The optimization problem of Eq. 4.13 is a nonlinear min-max optimization problem. Min-max problems can be solved numerically in MATLAB using sequential quadratic programming (SQP). SQP is polynomial but it can only find a local optimum. Since the TASP problem is a discrete min-max[6] problem (Eq. 4.13), TASP-MiniMax computes a solution in the continuous

[6] The term *min-max* refers to the class of problems, and *MiniMax* refers as the name of the

domain, and then discretizes the vector to the closest discrete feasible solution:

> **Algorithm TASP-MiniMax: 1.** In every epoch, solve the problem in Eq. 4.13 using a min-max solver such as SQP on the continuous domain; obtain vector $\mathbf{u}$. **2.** Sort $\mathbf{u}$ in descending order, then chose enough of the corresponding highest-value servers (each element in $\mathbf{u}$ corresponds to a server) as the active server set to satisfy the capacity constraint of Eq. 4.13.

The high complexity of TASP-MiniMax limits its use in the online selection of active servers—the time complexity of quadratic programming alone is $O(m^3M)$, where m is the number of variables and M is the size of input. However, due to providing a good approximation, it is used for comparison in the evaluation section.

- TASP-CPLRH: This solution is similar to TASP-LRH, except that at each iteration, instead of LRH, Computing Power LRH hybrid (CPLRH) is used, described as follows. Servers are ranked and grouped according to their computing efficiency (p_i/c_i), where each group contains servers of the same computing efficiency. Within each group servers are ranked and sorted according to LRH (i.e., Eq. 4.16).

Example 2: To understand how TASP-CPLRH works, consider again matrix G in Example 1. Further, assume that server one is the most computing efficient server and servers two and three have the same computing efficiency. Hence, if the active server set of size two is required servers one and two are chosen.

TASP-LRH and TASP-CPLRH are of low complexity and they are recommended for actual use in data centers. To evaluate the proposed TASP solutions we use the following reference solutions.

specific algorithm defined in this chapter.

CPSP: the baseline algorithm Most of the previous research only takes into account the computing efficiency in the active server selection. Therefore, the power-aware yet thermally oblivious version of CPSP is used as the baseline algorithm to evaluate the efficiency of TASP, wherein servers are ranked solely based on their computing power efficiency, $rk_i^{CPSP} = p_i/c_i$ and y lowest-ranking servers are chosen as $A'(t)$. Choosing among multiple, equally ranked servers is implementation dependent. Note that CPSP for homogeneous data centers satisfies Definition 1, i.e., it performs a balanced workload consolidation.

PT-d+: PowerTrade-d [44] is also used as a reference algorithm. PowerTrade-d avoids total energy increase due to the cooling-computing power tradeoff by dynamically and iteratively resizing the active server set by one server at a time and checking that the overall energy is not increased. The size of active servers is determined by 'SurgeGuard' which uses a $G/G/n$ model to estimate the average size and then augments it by *server-reserve*. *Server-reserve* is the number of servers that are reserved and added to the active server set prediction. The original PowerTrade-d uses the "inverse-temperature" scheme, that tries to balance the servers' inlet temperatures. This chapter, however, implements "PT-d+" using MinHR [88] instead of the inverse-temperature scheme, as its performance is better [111]. The MinHR scheme is similar to LRH with the difference being that MinHR uses experimental measurements instead of a model to rank the servers.

4.5 TAWD Solutions

Energy-efficient workload distribution, as described in Section 4.3.2, has to compensate for the energy waste resulting from using peak-optimal active servers at non peak times during each epoch. Since TAWD algorithm should run in every slot (a few seconds), it is not practical to use high-complexity solutions. Therefore, LRH and CPLRH ranking are used to shift workload to the most thermally efficient servers in the homogeneous and heterogeneous cases, respectively, and with the following goals: (i) reducing the cooling energy by utilizing thermally efficient active servers, and (ii) reducing the computing energy by operating the active server at high energy efficiency utilization level.

In *load balancing*, the workload is distributed among servers such that their utilization levels are *equalized* (i.e., *balanced*). In contrast, in TAWD servers with low LRH are more likely to be utilized close to their maximum affordable workload (while satisfying the performance constraint, Eq. 4.15) and servers with higher LRH are likely to be utilized to a lesser degree. Consider an example of 50 homogeneous servers with load balanced 30% average utilization. Also assume that there is no performance-oriented threshold on the utilization level of servers, i.e., they can reach 100% utilization. Then, applying TAWD, $\frac{50\times30\%}{100\%} = 15$ servers with the lowest LRH get 100% utilization each and the rest remain idle. Note that servers are usually more energy efficient at high utilization levels. Hence, TAWD achieves both the aforementioned goals whereas MinHR and inverse-temperature achieve only the first goal.

4.6 Evaluation

This section evaluate the proposed approaches through simulations to illustrate the cooling-computing power tradeoff and to estimate the energy saving benefits of TASP and TAWD in a typical data center. The evaluation section is organized in four parts. First, the conditions where a workload consolidation causes cooling-computing power tradeoff is illustrated and its implication on the performance of server provisioning and workload distribution approaches (TAWD, TASP-MIP, PT-d+, and

Table 4.2: Server Power Profiles.

Type	Server model	p^{idle}	p^{util}	IPR	LDR
1	Ideal	0	330	0	0
2	IBM 350M2	100	200	0.33	0
3	PowerEdge1955	242	90	0.73	0
4	PowerEdge1855	182	50	0.66	0
5	IBM dx360M3	100	230	0.3	0.3

CPSP) are evaluated under constant workload rates, variable PUE and energy proportionality values (Section 4.6.1). Section 4.6.2 and Section 4.6.2 evaluate the energy efficiency of the schemes under realistic dynamic workloads using linear power model and nonlinear power model, respectively. Finally, the possible performance violation of the schemes is evaluated in Section 4.6.3.

Data center Profiles: The data center profile consists of the thermal profile, i.e., the heat recirculation matrix, and the servers' power and performance models.

The heat recirculation matrix used in this simulation is derived from a CFD model of the ASU HPCI data center with physical dimensions 9.6 m×8.4 m×3.6 m, and having two rows of industry standard 42U racks arranged in a typical cold aisle and hot aisle layout [89, 111]. The cold air is supplied by one computer room air conditioner, with a flow rate of 8 m^3/s. The cold air rises from the raised floor plenum through vent tiles, and the exhausted hot air returns to the air conditioner through ceiling vent tiles. There are ten racks and each rack is equipped with five 7U (12.25 inch) 10-server chassis. The following CoP function $CoP(T^{sup}) = 0.0068{T^{sup}}^2 + 0.0008T^{sup} + 0.458$ [88] is used. Further, the evaluation is performed using different power profiles of servers as listed in Table 4.2. For all data center configurations, the same heat recirculation and number of servers is used. Note that, the heat recirculation model is irrelevant to the energy proportionality of servers. Power profile Type 1 in Table 4.2 is an example of an ideal energy-proportional server. The power profile characteristics of Types 2 and 5 are derived from www.spec.org/power_ssj2008/results/. Finally, power profiles of Types 3 and 4 are derived from the measurements for some I/O intensive jobs [89]. These profiles are appropriate for HTTP-like Internet requests.

To model the utilization of servers, the reference service rate μ^{ref} is set to 4000 requests per second (for a server with two cores) with a *reference request size* on the data of 512 bytes. The reference response time is set to 6 ms [32]. The simulation environment is developed using MATLAB 2009. GNU Linear Programming Kit (*GLPK*) is used to solve MIP in Eq. 4.14 and use *fminimax* solver provided by MATLAB for solving TASP-MiniMax. To do a fair comparison for power saving, PT-d+ is configured to use the same methodology to determine the minimum active server set size as in TASP (i.e., $y(t)$). SurgeGuard [44] is used to evaluate the performance violation effect of TASP (Section 4.6.3). Algorithms CPSP, PT-d+ and No Server Provisioning, namely NoSP, are used as baseline algorithms to evaluate TASP and TAWD schemes. Specifically for CPSP, when it has to choose among servers of equal power efficiency, the ones with largest LRH are chosen (i.e., using Lemma 4.1.4). This configuration choice is made to show how badly CPSP can perform compared to thermal aware server provisioning (note that a thermal aware CPSP would be equivalent to TASP-LRH and TASP-CPLRH). This section uses various combinations of server provisioning and workload distribution schemes. Whenever TAWD is used, it is denoted as a subscript, e.g., $\text{NoSP}_{\text{TAWD}}$ and $\text{TASP-LRH}_{\text{TAWD}}$, and whenever load balancing is used the subscript is empty, e.g., NoSP.

Workload model: Two web traces are used: (i) a synthetic trace based on e-commerce web benchmark suite of SPECweb2009 (`www.spec.org`) and 1998 FIFA World Cup web trace [58], and (ii) Microsoft Hotmail web trace [113] of one week. To get traces from SPECweb2009, an Apache web server is set up on a Dual-core Intel Xeon LV system and an additional system as the SPECweb traffic-generating client and run SPECweb for 24 hours. SPECweb is apt at exhibiting short term traffic fluctuation. To make it exhibit long term traffic fluctuation, the parameter *"SIMULTANEOUS_SESSION"* needs to be dynamically adjusted. For that purpose, in the 24-hour SPECweb experiment, the long term traffic fluctuation is taken from 24 hours of FIFA's web log and *SIMULTANEOUS_SESSION* is adjusted accordingly in each 30-minute epoch. The evaluation is also performed using one week of Microsoft HotMail web trace [113] is used. In both traces, the traffic intensity (i.e., λ) is scaled up to match the capacity of the simulated data center. The coefficient of variations of the workload traces for SPECweb and HotMail is 1.5, and 0.95, respectively. Hence, they are representative of high and low fluctuating workloads.

Two Kalman filters, the first for TAWD and the second for TASP, are trained for five slots and epochs, respectively. Subsequently, they respectively predict the average rates at each slots ($\lambda(t,k)$) and at each epoch ($\lambda(t)$). To estimate the peak arrival rate over an epoch (i.e., $\lambda^{peak}(t)$), this section uses $\lambda^{peak}(t) = \beta_1\lambda(t) + \beta_0$ to estimate the peak arrival rate using the average predicted rate. β_0 and β_1 are estimated using regression on the last four days of the first week for the HotMail trace and a portion of the World Cup trace. The regressed values are: for SPECweb $\lambda^{peak}(t) = 2.97\lambda(t) + 54973$, and for HotMail $\lambda^{peak}(t) = 1.522\lambda(t) + 1000000$.

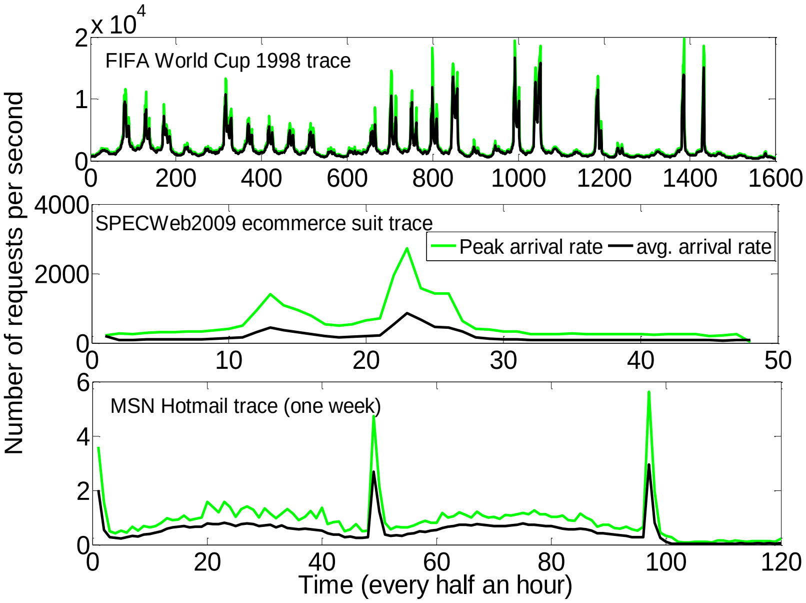


Figure 4.2: Average and Peak Arrival Rate in every Half an Hour over some Realistic Web Traces.

4.6.1 TACOMA Performance to Avoid Cooling-computing Power Tradeoff

This section investigates under what conditions of the data center utilization, the PUE and the servers' energy proportionality, cooling-computing power tradeoff occurs and what implication the tradeoff has on the energy savings of the schemes. To this end, this section simulates nine cases of homogeneous data centers, organized into three groups. The three groups use server Types 1, 2, and 3 respectively (Table 4.2). In each group, a different PUE is used (the heat recirculation matrix is scaled by half, one, and 1.5 to get different PUEs). PUE values shown in the figures reflect no server provisioning and when all the servers are utilized at 50%. In the simulation, the PUE varies from 1.2 to 2.25, a range that covers both modern and old data centers. This section runs simulations under a constant workload rate such that the utilization of the entire data center varies from 10% to 100% in 10% increment. Results are shown in Figs. 4.3-4.6.

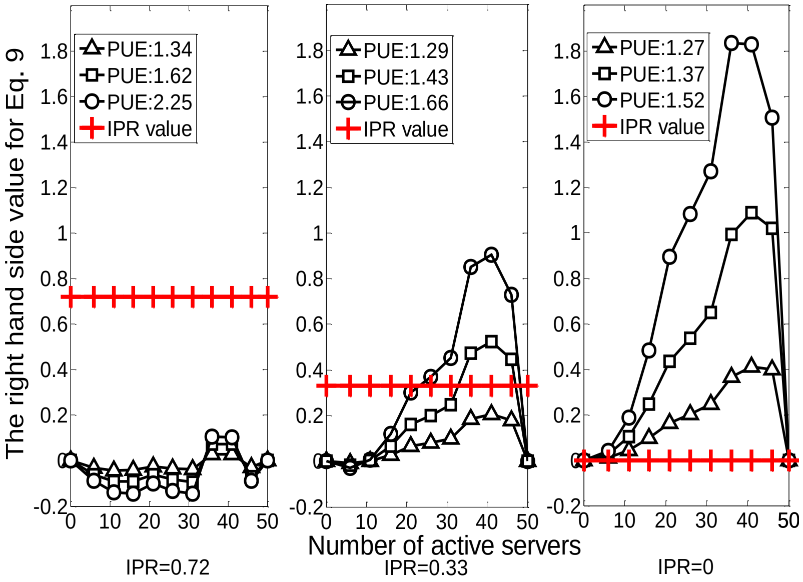


Figure 4.3: The Right Hand Side Value of Eq. 4.5 over Number of Required Active Servers (i.e., y) for Various PUE and IPRs, and $Y = 50$ (according to Lemma 4.1.1, the cooling-computing power tradeoff exists when y-axis value is greater than IPR value).

Cooling-computing Power Tradeoff

Fig. 4.3 illustrates the conditions of Lemma 4.1.1 under which there exists a balanced workload consolidation (e.g., CPSP) that causes cooling-computing power tradeoff. The right hand side value of Eq. 4.5 in Lemma 4.1.1 (i.e., ε_2) is plotted where the active server set is calculated using Lemma. 4.1.4. According to Lemma 4.1.1 the tradeoff exists if IPR $< \varepsilon_2$. The figure shows that the condition for the occurrence of the tradeoff does not hold for high IPR servers, i.e., server Type 3 (see Fig. 4.3(a)), holds when PUE is high and servers have low IPR, i.e., server Type 2 (see Fig. 4.3(b)), and always holds for ideal energy-proportional servers (see Fig. 4.3(c)). The results are compatible with arguments in Section 4.1.2 which says low IPR and high PUE favors the occurrence of cooling-computing power tradeoff. The next section shows the implication of the tradeoff occurrence, on the energy saving of various server provisioning schemes. Figs. 4.4, 4.5, and 4.7 show the energy consumption of the schemes under the data center energy efficiency configuration corresponding to Fig. 4.3(a), Fig. 4.3(b), and Fig. 4.3(c), respectively.

Energy Efficiency of the Schemes

Energy efficiency of the schemes for non energy-proportional servers (server Type 3): Fig. 4.4, demonstrating the energy consumption of schemes under the data center energy efficiency configurations corresponding to Fig. 4.3(a), shows two trends: (i) server provisioning schemes yield tremendous power savings compared to NoSP, including the thermally oblivious CPSP, due to removal of idle power (compare NoSP and $\text{NoSP}_{\text{TAWD}}$ to CPSP, PT-d+ and TASP-MIP); (ii) higher PUE values magnify the savings of thermal awareness (compare NoSP to $\text{NoSP}_{\text{TAWD}}$, and CPSP to PT-d+ and TASP-MIP). Note that absence of the cooling-computing power tradeoff (as illustrated by Fig. 4.3(a)) is sufficient condition for CPSP to always outperform NoSP. On the other hand, TASP-MIP yields more power savings than PT-d+ due to its optimality. Further, under constant workload, the performance of PT-d+ and TASP-LRH are almost the same, because both of them minimize the total heat recirculation. For plot legibility, the plot of TASP-LRH is omitted. Also observe that power savings of TASP-MIP and PT-d+ compared to CPSP are maximized within the range of 40-70% utilization; since in this range of utilization, there are more combination choices of servers, and the difference between a thermally efficient and a thermally inefficient active server set is maximized.

Energy efficiency of the schemes for modern low IPR servers (server Type 2): Fig. 4.5, demonstrating the energy consumption of schemes under the data center energy efficiency configurations corresponding to Fig. 4.3(b), shows that the power savings in this case have the same two trends of the previous case (compare Fig. 4.4 to 4.5) albeit at lower magnitude, except that for high PUE (PUE=1.66) and high utilization load (beyond 60%) CPSP consumes marginally more power than NoSP (see Fig. 4.5(c)). Under this scenario, cooling-computing power tradeoff exists as illustrated by Fig. 4.3(b). The cooling-computing power tradeoff increases the total power consumption of CPSP compared to NoSP which is consistent with arguments in Section 4.1.2. This behavior is shown in detail in Fig. 4.6 at 70% data center utilization, specifically, CPSP reduces idle power and saves 11% (20 W) compared to NoSP, however it creates hot spots which cause T^{sup} to decrease from 9.6 ℃ for NoSP to 7.2 ℃; this T^{sup} decrease causes the cooling power to increase by 17% (25 W) which translates to a net increase of total power by 2% (5 W) compared to NoSP. Under the same condition of data center load and PUE, TASP-MIP and PT-d+ save power around 30% and 18%, respectively, compared to NoSP.

Energy efficiency of the schemes for ideal energy-proportional servers (server Type 1): Fig. 4.7 shows that when servers are ideally energy proportional, server provisioning (suspending servers) yields no power savings (compare $\text{NoSP}_{\text{TAWD}}$ and PT-d+), in contrast to the trends in the previous two paragraphs. Of particular interest is CPSP, which yields higher total power than NoSP for all three PUE cases. Actually, for any PUE>1, CPSP would yield higher total power than NoSP, because it does not decrease the computing power yet creates hot spots due to workload consolidation (recall that CPSP is configured to select thermally inefficient servers). In other words, according to Lemma 4.1.1, CPSP in this case causes cooling-computing power tradeoff for all data center load and PUE conditions as illustrated by Fig. 4.3(c). In contrast to CPSP, TASP-MIP saves power for all three PUE cases, and its saving increases up to 13% with increasing PUE (Fig. 4.7(c)). The savings of TASP-MIP come from consolidating workload to the thermally efficient servers and not from suspend-

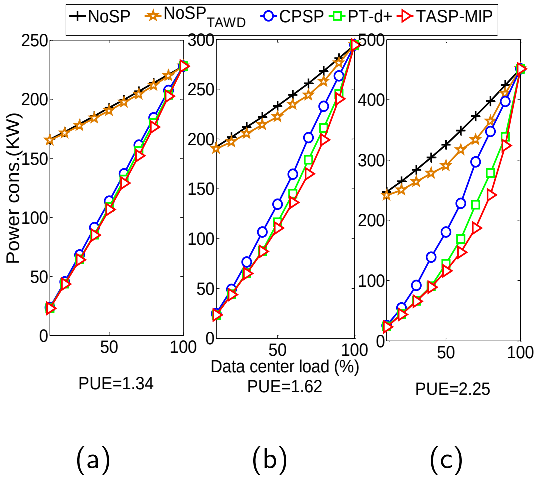


Figure 4.4: Power Consumption of Schemes for Different PUEs and under a Homogeneous Data Center with Server Type 3.

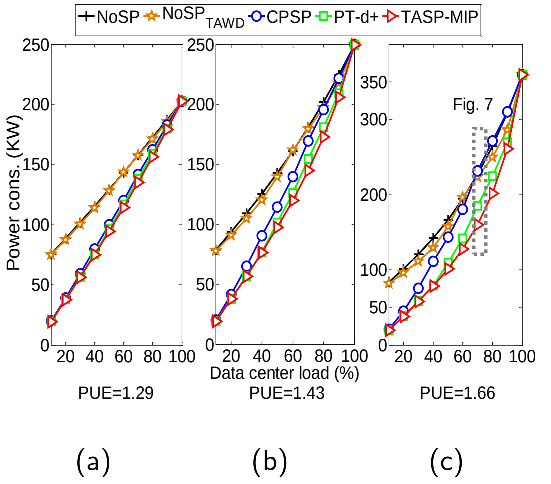


Figure 4.5: Power Consumption of Schemes for Different PUEs and under a Homogeneous Data Center with Server Type 2 (the box on the plot (c) shows the data center load condition for Fig. 4.6).

ing servers. Further, PT-d+ has marginal power savings due to its non-optimality.

4.6.2 TACOMA Energy Efficiency under Real-world Traces

To evaluate TASP and TAWD under realistic workloads, some experiments are performed with one homogeneous and one heterogeneous data center. The servers for heterogeneous data center are configured similar to ASU HPCI data center (20 chassis of server Type 4 and 30 chassis of server Type 3), and the servers for the homogeneous case are configured according to the modern server Type 2. Distinct simulations are performed for 24 hours of both SPECweb and HotMail traces. The

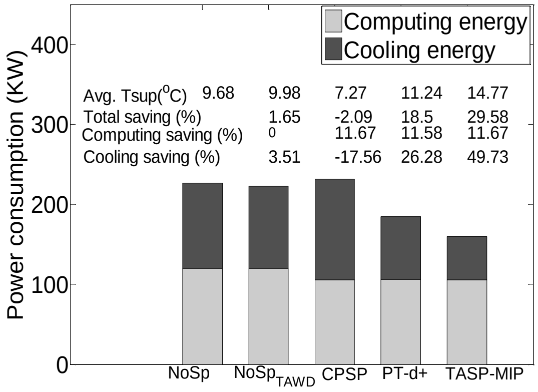


Figure 4.6: Power Saving of Schemes under a Homogeneous Data Center with Server Type 2, Load of 70%, and PUE=1.66. (see box in Fig. 4.5).

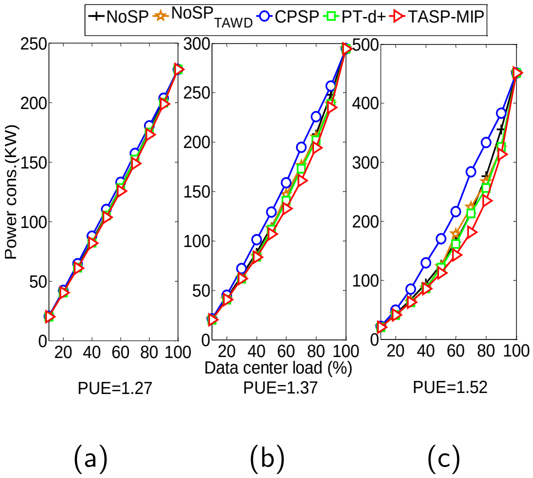


Figure 4.7: Power Consumption of Schemes for Different PUEs and under a Homogeneous Data Center with Server Type 1.

results presented in this subsection reflect the summed energy consumptions from the separate experiments on the two workload model.

Energy efficiency of TASP and TAWD for *homogeneous* data center case: Fig. 4.8, showing the percentile energy saving of various server provisioning scenarios under homogeneous data center, indicates that TASP schemes can significantly save energy compared to no server provisioning. It can be surmised from the figure that CPSP creates higher temperature hot spots compared to NoSP (since it demands cooler supplied temperatures), yet it decreases both the computing and the cooling energy compared to NoSP. Hence, it decreases the total heat produced (i.e., the total computing power). Also, in contrast to fixed workload scenarios where PT-d+ always performs better than CPSP, the total energy saving of PT-d+ is lower than

CPSP. However, PT-d+'s cooling energy saving is more than CPSP. Since, under a fluctuating workload (as opposed to fixed workload), the tradeoff may not be adequately computed by PT-d+ dynamically. Because, the variation of cooling energy can also be attributed to variation in workload and not just to adjusting of the active server set. Accordingly, the results show that under the same active server set size policy, and the same workload (HotMail) the average active server set size of Pt-d+ is 37, whereas it is 31 for CPSP and other TASP algorithms.

Except for PT-d+, all other thermal aware server provisioning schemes (i.e., TASP-LRH, TASP-MIP and TASP-LRH$_{\text{TAWD}}$) decrease the cooling demand. Interestingly, the T^{sup} of both TASPs as well as Pt-d+ are higher than T^{sup} of even NoSP. This means that thermal aware server consolidation not only can prevent creation of hot spots, but also can alleviate the existing hot spots through assigning workload to thermally efficient servers.

The optimal algorithm, i.e., TASP-MIP saves around 20% energy with respect to CPSP, and TASP-LRH saves 14% energy compared to CPSP. This result indicates that the performance of LRH is higher than what can be proved theoretically. As mentioned in Section 4.4.1, LRH is a Y-approximation algorithm (see Eq. 4.14), where Y is the number of servers. However, according to the numerical analysis, the approximation ratio of LRH is 1.18 under the aforementioned simulation setup. Notice, all extra energy savings of TASP variants compared to CPSP comes from thermal aware sever provisioning and decreasing the data centers' cooling demand. It can be seen in the figure that the supplied temperature of TASP approaches increases up to 4℃ compared to CPSP, whereas their computing energy savings is same as CPSP's. Further, TAWD under no server provisioning yields marginal energy savings compared to NoSP. Furthermore, incorporating TAWD to the TASP-LRH marginally improves its energy saving benefit.

Energy efficiency of TASP and TAWD for *heterogeneous* data center case: similar to the homogeneous data center case, CPSP saves significant energy compared to NoSP in the heterogeneous case (see Fig. 4.9). In fact, its saving is much higher than the homogeneous case, since it chooses power efficient servers. It also demands higher T^{sup} than NoSP. The reason is in the simulated data center configuration, power efficient servers accidentally were more thermal efficient. TASP schemes save up to 22% cooling energy compared to CPSP, which translates into 7% energy savings total. The results also indicates the higher performance of TASP schemes and even CPSP compared to PT-d+. However, PT-d+ causes a cooler supplied temperature than TASP-LRH and even less than TASP-LRH$_{\text{TAWD}}$. This is because PT-d+ uses thermal oriented load balancing using MinHR [88], which is more thermal efficient than TAWD. The next section shows that the performance of TAWD increases under nonlinear power curves of systems. Another important result in the figure is that heuristic TASP (i.e., TASP-CPLRH) competently saves energy compared to TASP-MiniMax solution. However, the results show that the performance of TASP schemes decrease compared to the homogeneous data center case (see Fig. 4.9). This refers to the size of the active server set. As shown in Section 4.6.1, the benefit of TASP schemes maximized when data center is half utilized (see Fig. 4.5). Since the average size of the active server set in the heterogeneous case is around 25 (out of 30 of high power efficient servers), TASP's energy savings is lower than the homogeneous data center case where its average active server set size is 34 (out of 50).

Discussion on results: Results indicate that heuristic TASP schemes can competently save energy with respect to optimal solutions. The results also show that the dynamic management of the cooling-computing power tradeoff (i.e., the PT-d+

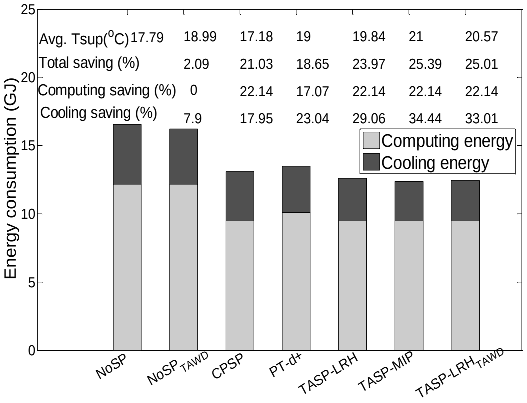


Figure 4.8: Energy Consumption of Schemes under a Homogeneous Data Center with Server Type 2. Average$(A'(t)|) = 34$ for TASP.

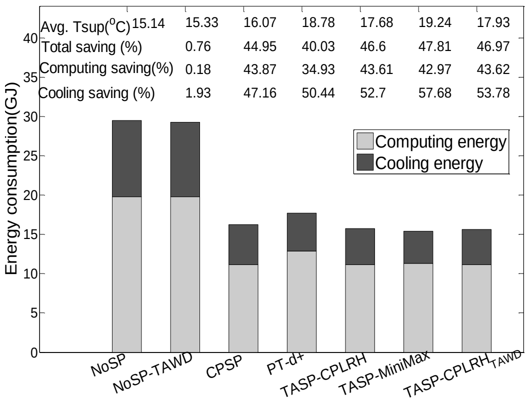


Figure 4.9: Energy Consumption of Schemes under a Heterogeneous Data Center with Server Type 3 and 4. Average $(|A'(t)| = 25)$ for TASP.

solution) which can be successfully manged under constant workloads, can not be adequately performed under realistic workloads. The reason is, under realistic workloads, the variation of cooling power can be because of variation in workload rather than adjusting the active server set size.

Performance of TASP and TAWD under Nonlinear Power Curves

To study the effect of the nonlinearity of the power curve on the power consumption, this section uses a state of the art server, server Type 5, whose power curve exhibits nonlinearity. The power curve of the server is fit to the polynomial function $f(u) = -4301u^6 + 15419u^5 - 21798u^4 + 15326u^3 - 5490u^2 + 1092u + 92.8$. Some experiments are then performed for a homogeneous data center case of fifty chassis of this server to calculate the energy consumption of TASP and TAWD schemes under

SPECweb and HotMail workload traces. The results shown in Fig. 4.11 interestingly show that under no server provisioning TAWD decreases the total energy by 12% which is much higher than the case where systems are assumed to have linear power curve (see Fig. 4.8). This saving is the outcome of consolidating the workload on thermal efficient servers and allowing them to operate at higher utilization where their computing energy efficiency is higher. Similarly, the energy saving benefit of TASP-LRH$_{\text{TAWD}}$ increases. The reason is, TAWD portion of TASP-LRH$_{\text{TAWD}}$ shifts the workload toward the more energy efficient servers and less to the other active servers compared load balancing scheme used in TASP-LRH (e.g., see Fig. 4.10 where active servers' average utilization is 35% by using TASP-LRH whereas average utilization for energy efficient servers is 80% by using TASP-LRH$_{\text{TAWD}}$). Since PT-d+ uses thermally oriented load balancing (i.e., MinHR based workload distribution), its energy savings is much lower than TASP-LRH$_{\text{TAWD}}$ which make thermally efficient servers to operate at their high energy efficiency level. Note that consolidation workload on the thermally efficient servers mitigates the hot spots. The average utilization of thermal efficient computing nodes in PT-d+ as shown in Fig. 4.10 are a little lower than LRH because the PT-d+' active server set size is larger than TASP-LRH. If active server set size of both PT-d+ and TASP-LRH would be the same, the average utilization of the most thermal efficient servers under PT-d+ would be higher than their average utilization under TASP-LRH.

Discussion on results: A significant observation is that non linear energy proportionality (observed in power-utilization curve of modern servers) can help in the savings of TASP and TAWD. Provisioning can consolidate the workload to fewer servers, thus increasing the per-server utilization with minimal increase in the energy consumption. Note that since consolidation of workload is applied on thermal efficient

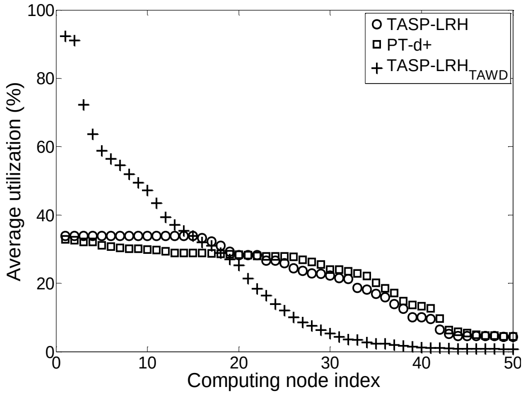


Figure 4.10: Average Utilization for Computing Nodes Indexed by LRH.

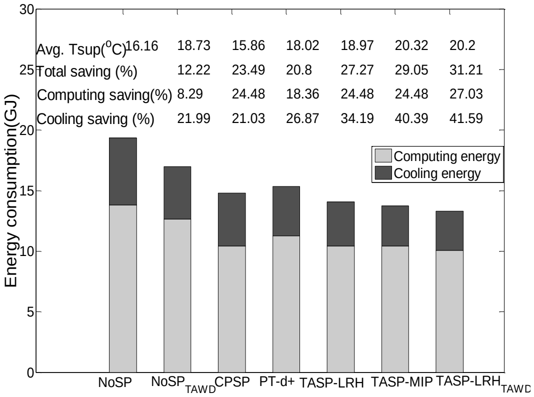


Figure 4.11: Energy Consumption of Schemes under a Homogeneous Data Center with Server Type 5.

Table 4.3: Performance Violations of TASP and SurgeGuard

Trace	Method	server-reserve*	$\lvert A' \rvert$	delay d(ms)	Perf. viol.(%)	violated d^{ref} (ms)	saving**
HotMail	TASP	0	31	5.5	6.18	6.7	24.96
	SurgeGuard	5	25	6.1	40	6.7	22.2
	SurgeGuard	10	30	5.7	10.23	7.4	12.57
SPECweb	TASP	0	15	4.8	2.6	10.1	60
	SurgeGuard	5	12	5.1	8.02	9.5	59.8
	SurgeGuard	10	17	4.8	3.3	10.9	47.6

* *server-reserve* is expressed as the number of servers in reserve.

** energy saving percentage of TASP-LRH under the corresponding "method" compared to NoSP.

servers, hot spots are unlikely to occur and the total energy is always minimized.

4.6.3 TACOMA QoS Violations

This section evaluates TASP for predicting the minimum active server set size and its impact on performance violations, compared to SurgeGuard [44] as described in Section 4.4.2 in this chapter. Since, the average workload prediction scheme is not available for SurgeGuard Kalman filtering is used to predict the workload arrival rate. The performance violation is measured as the percentage of slots where the average response time of servers goes above the reference response time. Results for TASP-LRH with HotMail and SPECweb workloads shown in Table 4.3 indicate that for almost the same active server set size, TASP saves more energy and yields much lower performance violations than SurgeGuard. The reason is TASP leverages the correlation between the peak and the average arrival rate during an epoch, whereas SurgeGuard uses the same static value of over provisioning (i.e., *server-reserve* for all epochs). Therefore, sometimes it underprovisions and sometimes it overprovisions. Even under TASP performance violation still happen due to unpredictable spikes in the workload. These violations are conjuncted to be predominantly eliminated by using a hybrid approach of proactive and reactive server provisioning (e.g., quickly activating servers whenever an unpredicted spike is observed). The investigations of such a hybrid scheme is left for future work.

4.7 Summary

This chapter proposes *thermal aware server provisioning* (TASP) and *workload distribution* (TAWD) schemes integrated into a two-tier Thermal-Aware Computing and cOoling Management Architecture (TACOMA). The chapter examines the performance of TACOMA over modern servers which tend to have *low idle power* and a *nonlinear power-utilization curve*. The analytical study and simulation results show that non thermal-aware server provisioning schemes are *insufficient* and even *inefficient* in improving the energy efficiency of data centers with modern servers, as their impact on cooling power increases (due to the creation of hot spots by highly utilizing the active servers) may outweigh their impact on computing power decrease. This phenomenon is due to the cooling-computing power tradeoff, which can manifest in air-cooled data centers. The chapter provides theoretical work on the conditions that cause the tradeoff to occur. TASP addresses inefficiencies such as this tradeoff through formulating the server provisioning problem as a mixed integer program and a minimax optimization for homogeneous and heterogeneous data center cases, respectively. In both cases, the objective is to minimize the *sum of cooling and computing power*. The results show that TASP solutions save up to 20% of the energy compared to previous non thermal aware and a newly published thermal aware server provisioning schemes. Further, the nonlinearity of the power-utilization curve of modern servers can be leveraged by the proposed TAWD to shift the workload dispatching toward the thermally and power efficient servers, and utilize them at their high energy efficiency level without creating hot spots. TASP and TAWD respect the performance constraints by considering the maximum affordable load of each server (according to a queuing model used) into the determination of the minimum number of required active servers servers, for TASP, and the determination of the workload distribution, for TAWD.

Chapter 5

ENERGY COST OPTIMIZATION AT GLOBAL DATA CENTER LEVEL

The solutions of this chapter and the two next chapters are on designing cost-aware workload management at the global data center level. This chapter primarily concerns algorithm design of the cost-aware workload management scheme in the presence of servers with non-zero idle power. The results show that the cost-aware workload management is generally an NP-hard problem, and that the proposed greedy solution has a performance near to that of the optimal solution. The results of the greedy solution is used in the next two chapters which account for more complex energy cost models (joint optimization of energy cost and carbon footprint, and an extension of the energy cost model to incorporate peak power cost). Further, the modeling and the solutions of this chapter accounts for stateful applications (applications that need to keep track of the state of the online users), in addition to the stateless applications (regular Internet workload with no state information).

In summary, this chapter presents Dynamic Application Hosting Management (DAHM), a workload management scheme for geographically distributed data centers, which decides on the number of active servers and on the workload share of each data center. DAHM achieves cost-efficient workload management by taking into account: (i) the spatio-temporal variation of energy cost, (ii) the data center computing and cooling energy efficiency, (iii) the live migration cost, and (iv) any SLA violations due to migration overhead or network delay. DAHM is modeled as *fixed-charge min-cost flow* and *mixed integer programming* for stateless and stateful applications, respectively, and it is shown to be NP-hard in both cases. The chapter also develops heuristic algorithms and proves, when applications are stateless, that the approximation ratio on the minimum total cost is bounded by the cost of running one idle server at each data center over the entire budgeting period. Further, the heuristics are evaluated in a simulation study using realistic parameter data; compared to a performance-oriented application assignment, i.e., hosting at the data center with the least delay, the potential cost savings of DAHM reaches 33%. The savings come from reducing the total number of active servers as well as leveraging the cost efficiency of data centers. Through the simulation study, we further explore how relaxing the delay requirement for a small fraction of users can increase the cost savings of DAHM.

In the rest of the chapter we first present the system model under which we study DAHM (Section 5.1). Next, we present the online algorithms for DAHM in Section 5.2 and provide a theoretical proof for performance bound guarantee of the greedy algorithm. We evaluate DAHM in Section 5.3 and conclude the chapter in Section 5.5.

5.1 Problem Formulation

In this section, we formally define DAHM problem. The DAHM system model are mainly consistent with Section 2.7, and Fig. 2.9. The specific assumptions, however, are that data centers get their required power from the grid, and that applications can be either stateless or stateful. Further, the notations are mainly given in Table 2.2 with this-chapter-specific-notations in Table 5.1.

We frame DAHM as an optimization problem. The optimization is a two level process: (i) deriving the number of required active servers at each data center i, ($y_{i,t} \in \mathbb{N}_0,\ 0 \leqslant y_{i,t} \leqslant Y_i$, where Y_i denotes the total number of servers at data center i), and (ii) deriving the traffic fractions $\lambda_{i,j}(t)$ from each area j to each data center i. The optimization is performed regularly with equal time intervals, namely slots.

DAHM is designed according to data center models introduced in Chapter 2

Table 5.1: Symbols for DAHM Problem Formulation

Symbol	Definition
u_i^{th}	threshold util. of servers at DC i
β	migration cost per migration
γ	performance violation cost per each user
si	% of new users over an slot
so	% of users to sign out over an slot

(Section 2.7) with some additional models to account for migration cost of stateful applications, as briefly described below:

Performance Modeling The results of this chapter work for any linear performance model (e.g., performance model based on $M/M/n$ and performance model based on CPU threshold utilization model described in Section 2.7.1). For the sake of notation brevity, the chapter develops the formulation and the analysis based on CPU threshold utilization performance model to account for service delay. Following to this model, we denoted by u^{th}, the threshold CPU utilization of a server. By bounding the maximum workload offered to a server, we bound the service delay by d'^{ref}. Therefore, for a data center i to receive workload from front-end j, we should have $d''_{i,j}(t) + d'^{ref} \leq d^{ref}$, where $d''_{i,j}(t)$ denote the network delay and d^{ref} denote the reference delay.

Energy Costs We use energy cost as cost per unit of energy consumed. Following computing power consumption model of Section 2.7.1, and considering PUE to model the cooling energy, the energy cost can be written as follows:

$$Cost_i^{energy}(t) = \left(\frac{\sum_j \lambda_{i,j}(t)}{\mu_i}(t) p_i^{util} + y_i(t) p_i^{idle} \right) PUE_i(t) \alpha_i(t), \tag{5.1}$$

where the '+' indicates that only when the difference is positive the cost is considered, otherwise there is no cost. Also recall that the phrase $\frac{\sum_j \lambda_{i,j}(t)}{\mu_i}$, gives the total data center utilization (i.e., $u_i(t)$) where μ_i denotes the service rate of the servers in data center i.

Migration Cost Dynamic workload distribution for stateful applications may require live migration (i.e., online users' state information should migrate from the source to the destination data center). Migration imposes a cost in terms of increase in network bandwidth consumption, and delaying the service for the affected online users.

Therefore, we consider a uniform, per-user migration cost, β, assuming equal-sized state information for all users. The calculation of the migration cost is based on the number of online users who have been migrated, as follows. Eq. 5.1 suggests that if a front-end assignment to a data center between two intervals changes, then migration is performed. Therefore, we can calculate the number of migrated users for each data center and front-end by calculating the difference in the number of assigned users between two consecutive slots. However, we choose not to directly take the difference between the previous slot's $(t-1)$ assignment and the next slot's (t) assignment, i.e., because we have to account for the users that are signing out in slot $t-1$ (and therefore their connections are not migrated) and the users that are signing in, in slot t (and therefore their connections did not exist at migration time). Let $si(t)$ denote the average fraction $(0 \leqslant si \leqslant 1)$ of new users out of the total users at each area during slot t, and $so(t-1)$ denote the average fraction $(0 \leqslant so \leqslant 1)$ of users at each area who sign out during slot $t-1$, then the migration cost for a data center i at time t can be formulated as

$$\text{cost}_i^{\text{migration}}(t) = \beta \sum_j \Big((1 - si(t))\lambda_{i,j}(t) - (1 - so(t-1))\lambda(t-1)\Big)^+, \tag{5.2}$$

where "+" indicates that only when the difference is positive the cost is considered, otherwise there is no cost. Each of the si and so parameters can be estimated from the other based on preservation of flow, expressed by this relation: $\Lambda_j(t)(1-si(t)) = \Lambda_j(t-1)(1-so(t-1))$ (i.e. the users that did not sign out in slot $t-1$ should be equal to the online users that did not just sign in, in the slot t), and that one of them is given as input for a given time. Note that si and so also imply the linear conversion from request arrival rate to the user arrival rate.

The decision to migrate workload is justified by the premise that it can complete its execution at a lower energy cost on another data center. The migration depends on two parameters: (i) the longevity of user connection; naturally, it is rarely beneficial to migrate a short running job as the benefit does not outweigh the migration costs; and (ii) the migration cost; if the migration cost is much higher than the difference between energy cost efficiency of two data centers for processing an online user workload, the migration never happens. If the migration cost is much lower than the difference between energy-cost efficiency of two data centers, it always happens. In our simulation study (Section 5.3), we assume long connections for the online users, and investigate the migration cost impact on the DAHM cost saving with respect to the average energy-cost benefit of migration. We refer DAHM problem as the DAHM for zero migration cost case, and the DAHM for non-zero migration cost case problem in the rest of chapter, where the former assumes $\beta = 0$ (i.e. stateless applications), and the latter assumes $\beta \neq 0$ (i.e., stateful applications).

5.1.1 Dynamic Application Hosting Management (DAHM) Optimization Framework

The problem can be summarized as follows:

> **DAHM Problem:** Given an application with a specific delay requirement d^{ref}, a cloud with N data centers in which the application can be hosted in a dynamic way, a spatio-temporal variation of the electricity price, $\alpha_i(t)$, a spatio-temporal variation of the number of the online users $\Lambda_j(t)$, find the hosting for each slot t that minimizes the sum of energy and migration cost, Eq. 5.1 and Eq. 5.2.

All aforementioned costs are assumed to be monetary. We can model the application hosting problem as an optimization problem where the objective is minimizing the total cost as shown in Fig. 5.1.

Cost minimization is subject to the following constraints:

- *Service constraint* (Eq. 5.4), which asserts that all users of every area should be assigned to a data center, and that there are no double assignments in either direction.

- *Idle power constraint* (Eq. 5.5), which ensures that the idle power consumption of all active servers is accounted.

- *Capacity constraint* (Eq. 5.6), i.e., the number of assigned active servers to the application in a data center should not exceed the available servers (denoted by Y_i) in that data center.

- *Performance constraint* (Eq. 5.7), i.e., the traffic of end users should be split among data centers whose network and service delay is less than the users' delay requirement.

Minimize

$$\begin{aligned}
&\text{Cost} = \text{Cost}^{\text{energy}} + \text{Cost}^{\text{migration}} \\
&= \textstyle\sum_{t=1}^{S} \Bigg(\sum_i \Big(u_i(t) p_i^{util} + y_i(t) p_i^{idle} \Big) \text{PUE}_i(t) \alpha_i(t) + \\
&+ \beta \textstyle\sum_i \sum_j \Big((1 - si(t)) \lambda_{i,j}(t) - (1 - so(t-1)) \lambda_{i,j}(t-1) \Big)^{+} \Bigg)
\end{aligned} \tag{5.3}$$

subject to

$$\text{(Service constraint) } \forall i,j,t: \quad \sum\nolimits_i \lambda_{i,j}(t) = \Lambda_j(t), \tag{5.4}$$

$$\text{(Idle power constraint) } \forall i,j,t: \quad y_i(t) \in \mathbb{N}_0 \text{ and, } y_i(t) \geq \frac{\sum_j \lambda_{i,j}(t)}{\mu_i u_i^{th}}, \tag{5.5}$$

$$\text{(Capacity constraint) } \forall i,t: \quad 0 \leqslant y_i(t) \leqslant Y_i, \tag{5.6}$$

$$\text{(Performance constraint) } \forall i,j,t: \quad d_i'^{\text{ref}} + d''_{i,j}(t) \lambda_{i,j}(t) \geq 0. \tag{5.7}$$

Figure 5.1: Mixed Integer Programming (MIP) Formulation of the Dynamic Application Hosting Management (DAHM) Problem.

A solution to this problem would specify, at each slot, how many servers in each data center should be assigned to the application (i.e., $y_i(t)$) and what portion of each area's traffic should be assigned to which data center (i.e., $\lambda_{i,j}(t)$). Observe that some of the variables are reals (i.e., $\lambda_{i,j}(t)$) and some are integers (i.e., $y_i(t)$). Therefore, due to linearity of all equations (both the objective function and the constraints), the problem is a Mixed Integer Programming, (MIP). MIP is a well-known and general NP-hard problem class for which generic solutions have high computational complexity. For the case of zero migration cost, DAHM can be formulated as a more specific problem that is a Fixed Charge Min Cost Flow (FCMCF). FCMCF is also NP-hard [66] but, compared to MIP, more efficient approximation methods have been developed [29] (FCMCF can be solved by MIP but not vice versa). We also show that DAHM is NP-hard by reducing an NP-hard sub-case of FCMCF to it.

Lemma 5.1.1. *The DAHM problem is NP-hard.*

Preliminary: We reduce FCMCF to DAHM. In FCMCF [66] a graph $G = (V, E)$ with nonnegative capacities capacity_i and nonnegative costs w_i for each edge i is given with the edge cost defined on each edge's flow f_i as follows: $w_i = \text{flow-cost}_i f_i +$ fixed-charge_i, when $f_i > 0$ and $w_i = 0$ when $f_i = 0$. The question is whether there is a subset $A \subseteq E$ of the edges of G such that the flow from the source to the sink in (V, A) is at least F and the cost is at most W. FCMCF is known to be NP-hard even on a graph with two nodes and a set of multiple edges between them [66] (this case solves Knapsack).

Proof. Given a two-node FCMCF instance with a set of multiple edges between them, we construct a DAHM instance as follows: let migration cost be zero: $\beta = 0$ (i.e., the problem becomes memoryless and the index t can be removed), PUE $= 1$ and electricity cost $\alpha_i = 1$. Also, the delay constraint is relaxed ($d^{\text{ref}} = \infty$). We group the edges such that the capacities capacity_i and costs flow-cost_i and fixed-charge_i are respectively equal within each edge group. Let N be equal to the the number of these groups, and let Y_i be equal to the number of edges at each group i. Set $\mu_i = 1$ (i.e., service rate) for all edges. We map the flow F to the workload arrival rate, i.e., let $M = 1$ and $\Lambda_1 = F$. Finally, let fixed-charge_i and flow-cost_i be p_i^{idle} and p_i^{util} respectively. It is easy to see that the instance of FCMCF has a solution if and only if there is a solution to DAHM by flow split and number of servers of cost at most W. Therefore, DAHM is NP-hard even with zero migration cost, and no network delay constraint. □

5.2 Solutions to DAHM

This section presents DAHM solutions for both zero and non-zero migration cost cases.

5.2.1 DAHM solution for Zero Migration Cost Case

In this case, a one-slot DAHM instance can be modeled as an FCMCF. To illustrate the modeling of DAHM as an FCMCF, at slot t, without loss of generality, we consider a simple case that $M = 3$, and $N = 4$. We can make a graph by adding a source and sink node as shown by Fig. 5.2 such that the source is connected to the front-ends by edges whose capacity is equal to the corresponding workload arrival rate of the front-ends (mapping the service constraint). Each data center is connected to the sink by multiple homogeneous edges such that the number of edges equals to the number of available servers (i.e., Y_i) (mapping the capacity constraint), where the capacity of each edge equals to the maximum affordable workload by each server at data center i (i.e., $\lambda_i^{th} = u_i^{th}\mu_i$) (mapping to the capacity constraint) and, the fixed cost and flow dependent cost are set according to the idle energy cost and the utilization energy cost as shown in Fig. 5.2 (in the example $Y_1 = 2$, $Y_2 = 1$, and $Y_3 = 1$, and $Y_4 = 2$) (mapping the objective function). Finally, the edges between front-ends and data centers are added under no capacity constraint and zero cost. An edge between a data center and a front-end is added if and only if the delay requirement of the front-end can be met by the data center (mapping the delay constraint). A solution flow $F = \sum_j \lambda_{i,j}$ to FCMCF can be converted to a DAHM solution by mapping the set of selected edges between data centers and the sink to $y_i(t)$, and the flow between front-ends and data centers to $x_{i,j}(t)$.

The LP relaxation of FCMCF (i.e. converting all integer variables to reals) does not provide a solution that is tight to the optimal [29]. However, there has been much work that suggests to use Benders Decomposition [35] or branch-and-bound methods to find the exact solution for FCMCF. Both of these solutions have exponential computation cost in the worst case. We used branch-and-bound to find the optimal in our simulation study and to evaluate the proposed solution.

There is a 2-approximation algorithm [29], which is based on adding an exponential number of constraints to the problem; consequently, it is very computationally expensive to solve and therefore prohibitive for finding a solution in a time-constrained manner. For that reason, we designed a greedy algorithm, "Greedy", and show that it has a small approximation ratio (see Lemma 5.2.1), Propositions 5.2.2, and 5.2.2).

Greedy Algorithm to Solve DAHM for Zero Migration Cost Case

The Greedy algorithm associates a Cost Efficiency Metric (CEM) to each data center i at time t as follows: $\mathrm{CEM}_{i,t} = \frac{(p^{idle}+p_i^{util}u_i^{th})\alpha_i(t)PUE_i(t)}{u_i^{th}}$, which equals to the average cost of a data center normalized to its per server capacity. The idea is to use a linear energy cost for data centers and solve the DAHM problem approximately using linear programing and more specifically min-cost flow method (using the energy cost below fixed charge at FCMCF becomes zero, fixed-charge = 0, consequently FCMCF becomes a min-cost flow problem) which have polynomial-time complexity. In this case, the energy cost in Eq. 5.3 becomes as follows:

$$Cost^{energy}(t) = \sum_i \sum_j \lambda_{i,j}(t) CEM_i(t). \tag{5.8}$$

Using the above energy cost, the variable $y_i(t)$ is removed from the objective function (i.e., Eq. 5.3) which will be derived after the solutions for $x_{i,j}(t)$s are found as follows: $y_i(t) = \left\lceil \frac{\sum_j \lambda_{i,j}(t)}{u_i^{th}} \right\rceil$. Similarly the capacity constraint (i.e., Eq. 5.6) for each data center and slot becomes $\frac{\sum_j \lambda_{i,j}(t)}{u_i^{th}(t)} \leqslant Y_i$.

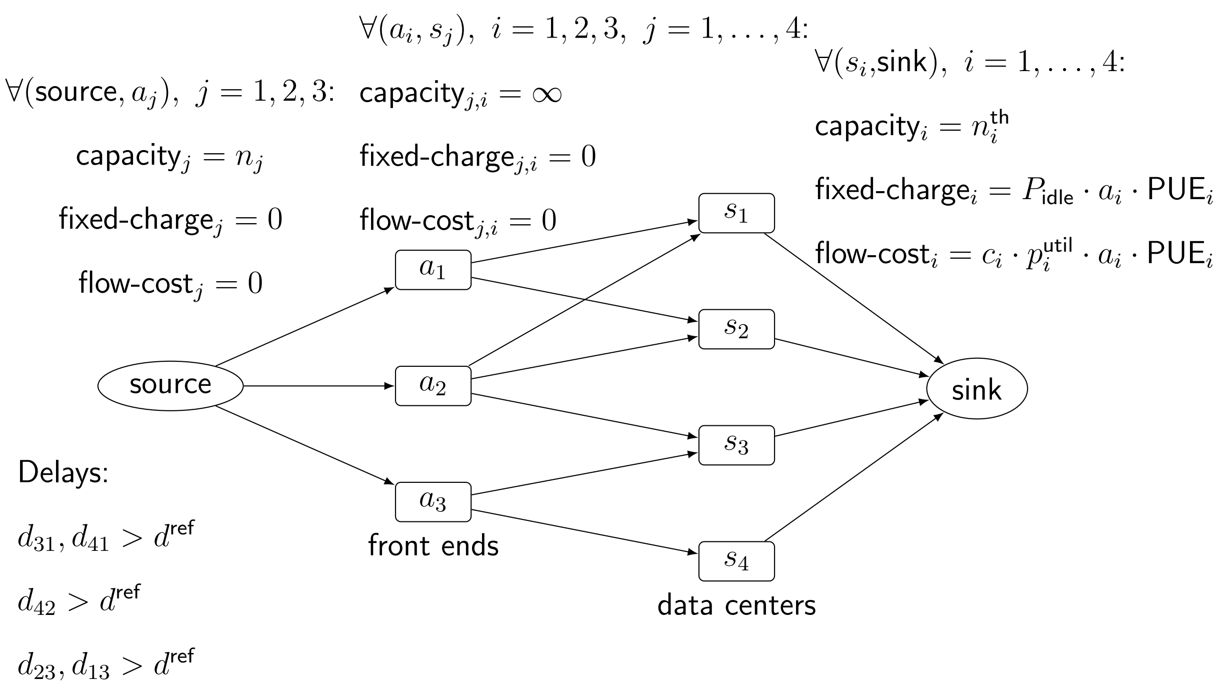

$\forall(a_i, s_j),\ i = 1,2,3,\ j = 1,\ldots,4$:
capacity$_{j,i} = \infty$
fixed-charge$_{j,i} = 0$
flow-cost$_{j,i} = 0$
$\forall$(source$, a_j),\ j = 1,2,3$:
capacity$_j = n_j$
fixed-charge$_j = 0$
flow-cost$_j = 0$
$\forall(s_i$,sink$),\ i = 1,\ldots,4$:
capacity$_i = n_i^{\mathsf{th}}$
fixed-charge$_i = P_{\mathsf{idle}} \cdot a_i \cdot$ PUE$_i$
flow-cost$_i = c_i \cdot p_i^{\mathsf{util}} \cdot a_i \cdot$ PUE$_i$
s_1
s_2
s_3
s_4
a_1
a_2
a_3
source
sink
front ends
data centers
Delays:
$d_{31}, d_{41} > d^{\mathsf{ref}}$
$d_{42} > d^{\mathsf{ref}}$
$d_{23}, d_{13} > d^{\mathsf{ref}}$

Figure 5.2: An Example of Modeling DAHM for Zero Migration as a FCMCF Problem.

Lemma 5.2.1. *The Greedy algorithm (5.2.1) is a N-approximation for DAHM when these three assumptions hold: (i) zero migration cost, and (ii) either (a) servers over the cloud have uniform IPR, or (b) for any two data centers i, k, $CEM_i \leqslant CEM_k \Rightarrow IPR_i \leqslant IPR_k$*[1]*.*

We use C_i to denote the numerator of CEM (see Section 5.2.1), and overload the notation Y_i to denote the data center i. Note that (i) Greedy prefers data centers with lower CEM because it optimizes the energy cost in Eq. 5.8, and (ii) we assume that Eq. 5.8 has a unique optimal solution which is true if no two CEMs are equal

[1] The 2$^{\text{nd}}$ condition is highly likely to happen in practice, since (i) CEM is dominantly affected by the servers' power model as PUE and electricity price do not vary as much, and (ii) low IPR values also reduce the PUE, and finally (iii) modern servers are exhibit low IPR values without any increase in their peak power.

(we can add a minute value to one of the equal CEMs to maintain the uniqueness without significantly changing the problem). The latter assumption ensures that as long as an efficient data center has available capacity, no other is used. The proof of the lemma is as follows:

Proof. Assume all servers within each data center are homogeneous (we argue on the case of heterogeneous data centers right below this proof). Due to the homogeneity of each data center and the linearity of the power model (Eq. 5.1), it follows that the total power of the assigned servers in a data center is *equivalent* to that of m fully utilized servers (i.e., $u = u^{th}$) and one under-utilized server. Let $\mathbb{N}$ denote the set of all data centers. We denote as $W \subseteq \mathbb{N}$ the set of data centers where, for each data center in W, Greedy assigns up to one server which is under-utilized, and as $K = \mathbb{N} - W$ the set of data centers where Greedy assigns servers of equivalent power of at least one fully utilized server. Assume that Greedy yields $y'_{i,t}$ fully utilized (i.e., $u = u_i^{th}$) and one under-utilized server in K. Let the total cost for the fully utilized servers in the set K be k_1, and the total cost for the under-utilized servers be k_2, and the total cost of *each* data center in the set W be w, we prove that each of k_1, k_2, and w provide a lower bound on the optimal cost, i.e., $Cost(Optimal) \geqslant \max(k_1, k_2, w)$.

- Hypothesis 1: k_1 is a lower bound on the optimal cost. By contradiction, assume Optimal pays less than k_1, then one of the two must be true:
  - *Splitting the workload of one or more fully utilized servers across other data centers achieves cost less than k_1.* Without loss of generality, assume Optimal splits the workload of a fully utilized server at data center 1 (as chosen by Greedy) onto data centers 2 and 3 . Since Greedy chose data center 1, it must be true that $\text{CEM}_1 \leqslant \text{CEM}_2$, and $\text{CEM}_1 \leqslant \text{CEM}_3$. Assume the q portion of the fully utilized server workload is assigned to

a server at data center 2 (and $1-q$ to data center 3). Also consider the most favorable scenario where all servers in all three data centers are truly energy proportional. Then the total cost of the workload becomes

$$\frac{q\lambda_1^{th}C_2}{\lambda_2^{th}} + \frac{(1-q)\lambda_1^{th}C_3}{\lambda_3^{th}} \geq \frac{q\lambda_1^{th}C_1}{\lambda_1^{th}} + \frac{(1-q)\lambda_1^{th}C_1}{\lambda_1^{th}} \geq C_1,$$

where λ_i^{th} is the upper workload arrival rate *per* server that can be hosted at data center i. Hence, the assumed case is contradicted.

- *Merging the workload of one or more fully utilized servers achieves cost less than* k_1. Without loss of generality, assume that Optimal merges the workload of two fully utilized servers in data centers 1 and 2 (as chosen by Greedy) onto data center 3. Also, without loss of generality, assume that $\text{CEM}_1 \leqslant \text{CEM}_2$ (one data center must be more efficient than the other due to the uniqueness assumption). It follows that $\lambda_3^{th} \geq \lambda_1^{th} + \lambda_2^{th}$. Since Greedy prefers data centers 1 and 2 over data center 3, it also follows that $\text{CEM}_1 \leqslant \text{CEM}_2 \leqslant \text{CEM}_3$. Also, assuming the most favorable scenario where all servers in all three data centers are truly energy proportional, then we have:

$$\frac{\lambda_1^{th}C_3}{\lambda_3^{th}} + \frac{\lambda_2^{th}C_3}{\lambda_3^{th}} \geq \frac{\lambda_1^{th}C_1}{\lambda_1^{th}} + \frac{\lambda_2^{th}C_2}{\lambda_2^{th}} \geq C_1 + C_2,$$

which contradicts the assumed case.

Hence, both the assumptions are contradicted, and Hypothesis 1 holds.

- Hypothesis 2: k_2 is a lower bound on the optimal cost. Since $k_2 \leqslant k_1$ (for each data center in K, there is up-to-one under-utilized server whereas there are one-or-more fully utilized servers), Hypothesis 1 proves this hypothesis as well.

- Hypothesis 3: For *any* data center in W, the cost w of its under-utilized server Y_w is a lower bound on the optimal cost. We consider the case that there exist other data centers that respect the delay requirement, otherwise Greedy would have no choice but to match the Optimal and yield the same cost. For the *total* cost of Optimal to be less than w, either (i) there is an available server with a lower CEM than the CEM of Y_{w}, or (ii) there is a data center on which the optimal solution can put the workload of Y_{w} to reduce the cost below w.

Case ii contradicts with the definition of Greedy: this would be possible if the server had a lower IPR than Y_{w}, which contradicts with Lemma 5.2.1's condition-*iii-a* or *iii-b*.

For case *i*, we observe that, for Greedy to select Y_{w} instead of any other eligible server with lower CEM, it must be so because all other eligible servers are fully utilized. Assume, by contradiction, that the optimal cost is less than w. By Hypothesis 1, we know that the optimal solution can not pay less than Greedy for the fully utilized servers (i.e., k_1). Therefore, for Optimal to achieve a lower total cost, it must merge the workload of some of fully utilized servers and the under-utilized one (i.e., server Y_{w}) onto any other server. This is possible only if the IPR of the target server is less than of that Y_{w}, which contradicts with Lemma 5.2.1's condition *iii-a* or *iii-b*.

Combining the above Hypotheses, $Cost(Optimal) > \max(k_1, k_2, w)$. The lower bound on total cost for the data centers in the set W is $|W|Cost(Optimal)$. We know that $Cost(Greedy) = k_1 + k_2 + |W|w$, then according to Hypotheses 1, 2 and 3, it follows that:

$$\begin{aligned}\tfrac{Cost(Greedy)}{Cost(Optimal)} &= \tfrac{k_1+k_2+|W|w}{Cost(Optimal)} \\ &\leqslant \tfrac{Cost(Optimal)+Cost(Optimal)+|W|Cost(Optimal)}{Cost(Optimal)} \leqslant \tfrac{(|S|+1)Cost(Optimal)}{Cost(Optimal)} = |S|+1.\end{aligned} \tag{5.9}$$

□

For the ***heterogeneous case*** of data centers, it is easy to see that the approximation ratio is the number of the classes of servers in the cloud. A tight example is as follows: assume M=3 and $|N$=3, where each data center has only one available server and that all have IPR=1 (no utilization-dependent cost). Further, assume Y_1 and Y_2 have the same CEM, where the total cost and capacity of a fully utilized servers for them is 1 and 10, whereas they are $1 + \varepsilon$, and 10 respectively for s_3. (i.e., $\text{CEM}_1 = \text{CEM}_2 = \frac{1}{10} < CEM_3 = \frac{1+\varepsilon}{10}$). Also assume Y_1 can only respect the delay requirement of front-end one,denoted by a_1 workload, similarly Y_2 can only respect the delay requirement of front-end number two, denoted a_2. But, the delay requirement of all areas can be respected by Y_3. Finally, assume each area has $\Lambda_j = 1$ one user. Greedy selects s_1 to provide service for a_1 workload, s_2 for a_2, and s_3 for a_3, hence it incurs $3 + \varepsilon$ cost in total. However, the optimal solution selects only s_3 to provide service for all areas, and incurs only $1 + \varepsilon$ cost. It is worth noting that the worst case situation happens only if utilization-dependent cost of servers is zero, and that number of required servers are low, neither of which are the case in practice. The following proposition, using the results of this lemma, gives another approximation ratio without making any assumption on the data centers' IPR. The ratio is specially very intuitive and show that the performance of Greedy solution is indeed very close to that of the optimal solution for practical cases.

Proposition 5.2.2. *The total cost of Greedy, satisfies the following , i.e., Cost(Greedy)* $\leq$ *Cost(Optimal)* $+ S \sum_i PUE_i \alpha_i(t))p_i^{idle}$

Proof. This directly follows from Lemma 5.2.1. According to Hypothesis 1 in Lemma 5.2.1, the cost of Greedy solution for servers that are fully utilized is lower than that of Optimal. Further, the cost of utilization power of the remaining servers is lower for Greedy, since Greedy optimizes the utilization power. □

Given that data centers usually require thousands of servers, the worst case cost difference of Greedy versus Optimal solution as given in the above proposition is negligible (at each slot the cost of Greedy compared to the optimal solution is at most increased by the cost of one idle server at each data center).

Proposition 5.2.3. *Greedy, under the conditions of Lemma 5.2.1, is a 2-approximation ratio when there is no network constraint delay or it is universally satisfied, i.e.,* $d_{ij} = d_i^{\prime ref} + d''_{i,j} \leqslant d^{ref},\ \forall\ i = 1 \ldots |S|$ *and* $j = 1 \ldots |A|$.

Proof. In this case, Greedy incurs at most one under-utilized server. By contradiction, assume that there are two or more under-utilized servers in S. If the data centers containing the underutilized servers are of equal CEM, then we can merge their workload and result into only one under-utilized server without altering the total cost, thusly contradicting the assumption. Conversely, if those data centers have unequal CEMs, it will contradict with the definition of Greedy which does not assign another data center before it fully uses the one with smaller CEM. Hence, Greedy yields only one under-utilized server; specifically, either $|W|$=1 and k_2=0, or $|W|$=0 and $k_2 \neq 0$, or $|W|$=0 and k_2=0. Therefore, we can safely conclude from Lemma 5.2.1 that the approximation ratio of Greedy in this case is 2. □

5.2.2 DAHM Solution for Non-zero Migration Cost Case

Similar to the zero migration cost case, the optimal solution for this case can be obtained using the branch-and-bound technique. However, the optimal solution to the problem can only be obtained offline where all information about the workload and electricity price is available in advance. In this case, DAHM generalizes FCMCF. Similar to DAHM for zero migration cost case, we can model DAHM at each time as a FCMCF for the non-zero migration cost case. However, the per-epoch solutions of

DAHM in the non-zero migration cost case depend on each other (i.e. each epoch's solution depends on the previous epoch's solution), and, to our knowledge, there is no way to connect those FCMCF instances in such a way that migration cost is incurred and flow conservation law is preserved, in order to find a combined optimal solution for the entire period.

We devise the following online algorithms that solve the problem at the beginning of each epoch, t, based on the current hosting state (i.e., $x_{i,j,t-1}$) of the application, the electricity price ($e_{i,t}$) and the next epoch's traffic behavior (i.e., distribution and population of online users).

OnlineMIP Algorithm

The online version of DAHM (Eq. 5.3), i.e., without summation over time in all the terms in Eq. 5.3, is solved using the branch-and-bound technique.
The online version of DAHM (Eq. 5.3), i.e., without summation over time in all the terms in Eq. 5.3, is solved using branch-and-bound.

OnlineGreedy Algorithm

The algorithm accounts for the online version of DAHM problem (Eq. 5.3), i.e., without summation over time in all the terms in Eq. 5.3, and uses Eq. 5.8 as the energy cost model in the objective function. It solves DAHM at each slot using linear programing.

OnlineCOB, Cost Oblivious Algorithm

We use conventional performance oriented load balancing assignment as a baseline algorithm to evaluate the cost efficiency of our approach. In this approach, (i) each area is assigned to a data center whose delay is the least among all other data centers,

(ii) load is balanced among data centers whose delay with respect to areas are the same. This is the approach that is currently used for mirror severs [41]. Also, the number of servers at each data center is dynamically adjusted at each epoch according to the size of incoming traffic. This algorithm is referred as Online Cost OBlivious, *OnlineCOB* in the rest of the chapter.

5.3 Evaluation

We simulate a cloud consisting of three data centers. Their characteristics are set according to realistic data. To this end, we assume data centers are located at the following three locations: Atlanta, GA; Houston, TX; and Mountain View, CA, namely DC1, DC2 and DC3, respectively. These locations correspond to the location of three major Google data centers. We used the historical electricity prices for the above locations [117] (see Fig. 5.3). Note that, in reality, each data center provider may have different electricity price contracts, i.e., lower electricity price than households. However, the electricity cost can be defined according to the actual electricity price or the type of energy source (green or brown). The electricity price of Fig. 5.3 is used as an *example* to show the cost saving benefit of DAHM by leveraging electricity cost.

To model the front-end coverage of the data centers we measure the network delay from the simulated data center locations to all US states using `traceroute`. We choose one IP address for each state (e.g., IP address of state universities) and run `traceroute` through three servers of the simulated data center locations (provided by "www.traceroute.org") to all 51 IP addresses. We ran `traceroute` hourly for 24 hours. As we did not find a server in Georgia, the location of DC3, to run `traceroute`, we chose Florida instead. The summary of results, shown in Fig. 5.5, indicates that the delay is highly correlated with the distance. Also the delay depends on the source

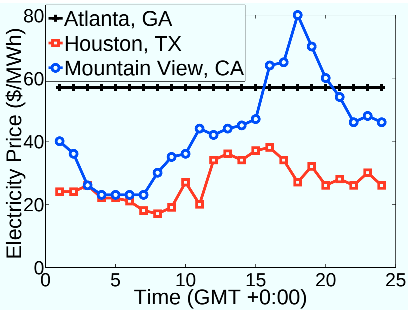


Figure 5.3: Hourly Electricity Price Data for Three Major Locations of Google IDCs on May 2nd, 2009 (data are taken from [97]).

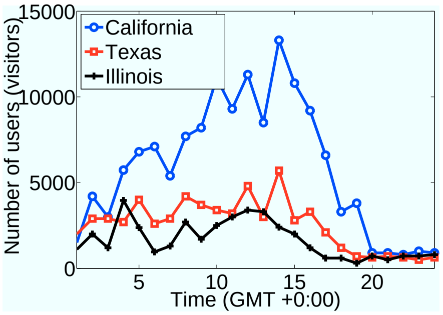


Figure 5.4: Hourly Number of Online Users from Three States for an Entertainment Web Site Hosted at GoDaddy.com.

network from which `traceroute` is run. Moreover, the daily variation of delays were negligible (within 1 ms).

Data center types Three homogeneous (identical) data centers are considered for the simulation with contemporary servers (e.g., IBM Systems x3650 M2: idle power 100 and peak power 320 watt) and very low PUE (we use 1.3, which is the PUE of the state of the art data centers [8]). To show the efficiency of DAHM solution under different energy proportionality of servers, the Idle to Peak power Ratio (IPR) [117] of servers is varied between zero (ideally energy-proportional server) and 0.6 (old servers). The maximum number of servers for each data center is set to 25 which matches the workload intensity range used in the simulations.

To model the utilization of servers, we assume that each online user imposes 0.00005 utilization to each server (i.e., $c = 0.0005$) and that each server can at most handle requests from 2000 online users. The server utilization thresholds, u_i^{th}, are set

Table 5.2: Data Centers' Characteristics

DC	Elec. price model	Servers' peak power	PUE	case*
DC1	Mountain View, CA.	320	1.3	homogen.
DC1	Mountain View, CA.	400	1.5	heterogen.
DC2	Houston, TX.	320	1.3	homogen. and heterogen.
DC3	Atlanta, GA.	320	1.3	homogen. and heterogen.

* The characteristics of DCs for the homogeneous and heterogeneous case study.

to 75%[2]. The d^{ref} is set to 66 ms, and data centers' reference delay, d'^{ref} is set to 6 ms [32].

Workload Model We used one day (March 17, 2011) of workload trace of an entertainment Web site hosted at *GoDaddy.com*. Using *Google Analytics*, we collected the hourly total number of visitors to the Web site from different USA states (see Fig. 5.6). The workload is scaled up to the data centers' capacity. Also we assume 50% of the users are new users (i.e., $si(t) = 0.5$).

Experiments Performed We performed different experiments to show the cost saving of DAHM with respect to the energy proportionality of servers (see Section 5.3.1), migration cost (see Section 5.3.2), heterogeneity of data centers (see Section 5.3.3), and workload variation (see Section 5.3.4).

We used *GNU Linear Programming Kit (GLPK)* solver under MATLAB 2009, to run the branch-and-bound algorithm on MIP. GLPK is also used to run our Greedy algorithms. **All of the cost savings are with respect to OnlineCOB**.

Under non-zero migration cost, the offline optimal, namely OfflineMIP, is im-

[2] This value was determined from anecdotal Web searching. It does not affect the validity of the results but only the amount of savings.

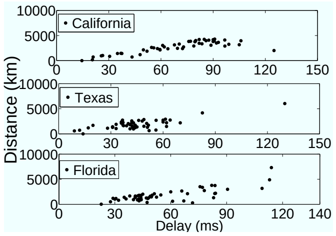


Figure 5.5: The Network Delay between Servers from Texas, California, and Florida to all other States in USA versus Distance between States.

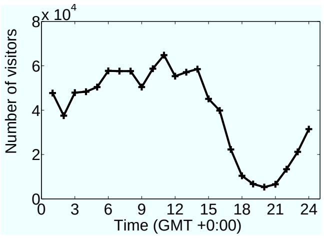


Figure 5.6: Hourly Number of U.S. Online Users for an Entertainment Web Site Hosted at GoDaddy.com on 17^{th} March, 2011.

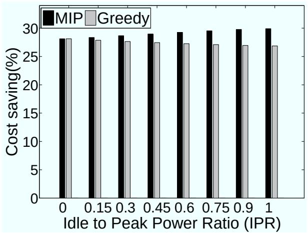


Figure 5.7: DAHM Cost Savings w.r.t. OnlineCOB over Different IPR of Servers and Zero Migration Cost (homogeneous DCs).

plemented using branch-and-bound and used to evaluate the proposed online solutions.

5.3.1 DAHM Cost Efficiency for Zero Migration Cost Case

The DAHM cost saving under different IPR of servers, shown in Fig. 5.7, interestingly indicates that the efficiency of Greedy is better than the theoretical bound (see Section 5.2.1). The same figure shows that at IPR= 0 (energy proportional case) Greedy and optimal solution incur the same cost. This is expected because in this case DAHM becomes a simple linear programing problem. The DAHM cost saving is due to both leveraging the variation of electricity price and minimizing the number of required servers across all data centers (see Fig. 5.13). The cost saving of the optimal solution increases for higher IPR, because consolidation of servers incurs more cost saving. The results in Fig. 5.7 show the benefit of the DAHM optimal solution over previous workload distribution schemes [70, 96]; the cost saving of those schemes was maximized under ideally energy-proportional servers and decreased significantly when the servers had an IPR greater than zero.

5.3.2 DAHM Cost Efficiency for Non-zero Migration Cost Case

To study the migration cost impact on DAHM cost saving, we choose a migration cost comparable to the *reference energy-cost benefit of a migration* denoted by β_0, which is defined as the difference between the energy-cost of the most and the least cost-efficient data centers for one online user. Fig. 5.8 shows that when the migration cost is less than $5\beta_0$, DAHM cost saving only drops from 27% down to 23% (with respect to OnlineCOB). The reason for the small drop is that when the workload share among the data center changes, new users do not have any migration, and that the benefit of migration is usually more than β_0, since it helps to consolidate servers. Therefore, if the migration cost is comparable to the cost efficiency difference of data centers, DAHM can still saves a significant cost due to reducing number of servers. The cost saving of DAHM diminishes down to 2.5% for very high migration cost cases 10-20 times β_0. The reason is that, since the migration is always applied to a portion of users (see Section 5.1), a very high migration cost prevents the workload share changes of data centers for total cost minimization. For the rest of the experiments we adjust $\beta = 5\beta_0$, and refer to it as the non-zero migration cost case.

For non-zero migration cost, neither OnlineMIP nor OnlineGreedy provide an optimal solution. As shown in Fig. 5.9, the cost saving of OnlineMIP is marginally greater than the cost saving of OnlineGreedy. The optimal solution under non-zero migration cost can only be achieved offline. Comparing DAHM offline optimal with respect to the online solutions[3], we find out that the offline optimal always achieves

[3] Due to high time complexity of the offline optimal algorithm, we just ran the algorithm for few

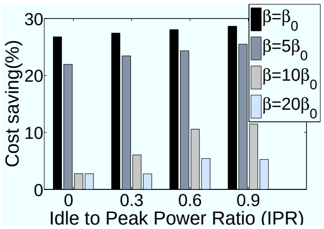


Figure 5.8: Cost Savings of MIP Solution w.r.t. OnlineCOB over Different IPR of Servers and Migration Cost β (Homogeneous DCs).

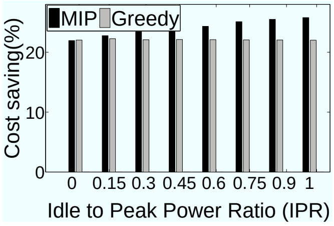


Figure 5.9: DAHM Cost Savings w.r.t. OnlineCOB under Different IPR of Servers and Non-zero Migration Cost (Homogeneous DCs).

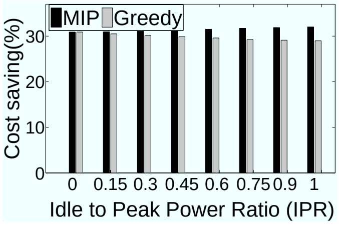


Figure 5.10: DAHM Cost Savings w.r.t. OnlineCOB under Different IPR and Zero Migration Cost (Heterogeneous DCs).

up to 1% better cost saving over the online solutions with respect OnlineCOB. Its cost saving accumulates with the increase in simulation time. Developing online algorithm with a competitive bound is left for future work.

5.3.3 DAHM Performance under Heterogeneous Data Centers

To investigate the potential saving of heterogeneous data centers and the heterogeneity's effect on the total cost efficiency of DAHM, we make DC1 to be less energy-efficient than DC2 and DC3. To this end, the PUE of DC2 is changed to 1.5 and the peak power of servers is changed to 400 W (see Table 5.2). The results in Figs. 5.10 and 5.11 show that DAHM cost saving increases from the range of 27-30% for the homogeneous data center case to 30-32% for the heterogeneous data center case. Also, in contrast to the case of homogeneous data center (§ 5.3.2) where

hours instead of entire 24 hours.

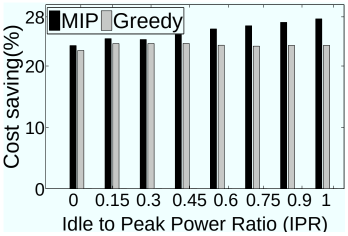


Figure 5.11: DAHM Cost Savings w.r.t. OnlineCOB under Different IPR, and Non-zero Migration Cost (Heterogeneous DCs).

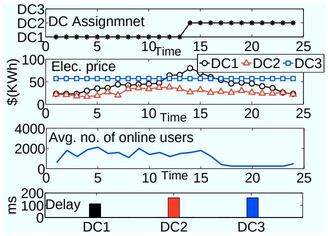


Figure 5.12: The Data Center Host and Workload Density of an Area over time (homogeneous DCs).

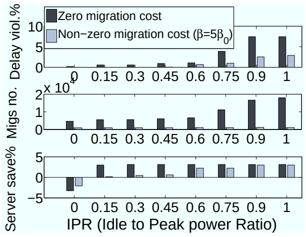


Figure 5.13: The performance of OnlineMIP with respect to OnlineCOB under Different IPR of Servers (homogeneous DCs).

Greedy's cost saving for non-zero migration cost decreases with respect to servers IPR, Greedy's saving in this case increases, yet marginally. The reason is that, in this case, minimizing the number of active servers over the cloud yields more cost saving.

5.3.4 DAHM Performance under Various QoS Requirement

To investigate how workload variation can be leveraged to save more cost, we used Lagrangian relaxation to move the performance constraint into the objective function (see Eq. 5.3 and 5.7) and adjust the Lagrangian multiplier (which is γ into the number of users whose delay is violated) to force the solution to perform tradeoff between energy cost and delay violation minimization. Since the current simulation setup would not yield a lot of delay violations (all DCs have low delays with most states), we change the simulation setup to restrict the data centers coverage area to at most half of the areas with low delay (this artificially makes areas out of the coverage

Table 5.3: Cost-saving and Delay Tradeoff of DAHM Compared to OnlineCOB

		$\gamma = \gamma_0$		$\gamma = 2\gamma_0$		$\gamma = 10\gamma_0$	
DC case	Algorithm	saving(%)**	viol.(%)**	saving(%)	viol.(%)	saving(%)	viol.(%)
homogen.	MIP	17-25	15-20	12-19	0-6	13-15	0
	OnlineMIP	13-25	0.1-7	11-18	0-2	9-14	0
heterogen.	MIP	22-28	10-18	21-24	1-8	20-22	0
	OnlineMIP	21-26	4-15	19-21	0.5-2.5	15-19	0

* The value of γ_0 is set to 0.000001.

** The saving values are given in a range from IPR=0 to IPR=1.

of a DC to have a delay above the constraint). The latter setup allows to investigate the potential cost-performance tradeoff under variability of network delays. With this setup, DAHM saves 9-15% cost in the case of no delay violation.

The results in Table 5.3 show that allowing delay violations for up to 1% of the users improves the cost saving of DAHM to 13-22% depending on the IPR value and migration cost. This saving can be explained using results in Figs. 5.12 and 5.13 as follows.

5.4 DHAM Implementation Issues in Practice

As summarized by Table 5.4, DAHM is NP-hard. We provide numerical results to evaluate the polynomial-time greedy algorithms and show they can have an approximation ratio very near to the optimal solution for the zero migration cost case. As a practical example, Greedy and GreedyOnline take a fraction of a second to compute the number of servers and workload share for a hypothetical cloud of 100 data centers, whereas MIPOnline takes around half an hour, all running on a 2.8 GHz Intel Pentium system, as performed in a side experiment.

Table 5.4: Summary of DAHM Problem and Solutions Characteristics

mig. cost	problem formulation	optimal sol.	complexity	approx. ratio
zero	FCMCF	Branch&Bound	NP-hard	2 (see [29]), and see Prop. 5.2.2,
non-zero	MIP	Branch&Bound	NP-hard	not known

For the homogeneous data center case, DAHM cost saving comes only from leveraging electricity cost and its magnitude depends on the number of available servers in the data centers, the delay constraints and the algorithm (MIP or Greedy). The maximum cost saving was 40% when there was no limit on the number of servers and delay. We provide numerical results to show how performance of DAHM is affected through the aforementioned parameters in the subsections above. The cost saving difference between Greedy and MIP algorithm diminishes as the workload and number of servers are scaled up, and the Appendix 5.2.1 shows the Greedy algorithm approximation ratio for the general case of workload volume.

In practice, different classes of workload may have different SLA and delay requirements. Incorporating the class of workload into the cost model does not change the nature of the problem, yet it needs more parameters to express the problem. Also, it adds the flexibility of DAHM to move workload of a lower class to the most cost efficient data centers to yield more cost savings. An exhaustive study of this modeling is left for future work.

In our simulation study, we assume that at the beginning of each slot, the input about workload and electricity price is available; but, in practice, this information should be predicted. Both workload and electricity are predictable, however the prediction error may marginally decrease the overall cost saving. DAHM can be considered as a central controller and should be frequently updated with information on network delay, electricity price and history of workload from data centers. Since these data should be sent at each epoch, and each epoch is nominally around half an hour to several hours, its overhead is negligible.

5.5 Summary

This chapter presents problem formulation and algorithms for DAHM, which allow cloud providers to host Web allocation cost efficiently in a dynamic fashion. The problem is formulated according to a cost model that accounts for energy cost of data centers, delay requirement, and traffic behavior of applications as well as live migrations. We show that the problem is generally NP-hard and that in the zero migration cost case, the problem can be modeled as FCMCF. We also show that the polynomial-time Greedy algorithm can provide a performance near to that of Optimal solution. Further, a simulation study is performed using realistic data and we make the following conclusions: (i) dynamic workload and server management minimizes the total number of servers over cloud and yields significant cost savings by removing idle power cost; (ii) dynamic server and workload management can leverage the temporal and spatial variation of electricity price, workload and data centers' energy efficiency to minimize total cost, and (iii) relaxing the delay requirement of a few users by incorporating the SLA revenue lost in the cost model can increase the total cost efficiency; this is due to (a) periods of low and high online user population over different areas are not simultaneously happened, and (b) assigning users of areas at periods of low online user population to data centers which are in service for other areas reduces total number of active servers while it may incur delay violation for a few fraction of population.

Chapter 6

JOINT OPTIMIZATION OF ELECTRICITY COST AND CARBON FOOTPRINT AT GLOBAL DATA CENTER LEVEL

The problems and the solutions of this chapter further extend the cost management of the global data center level as given in the previous chapter to incorporate carbon footprint capping across data centers. In particular, the chapter presents online workload and server management solutions towards achieving carbon neutrality for a cloud consisting of several geo-distributed data centers while minimizing the electricity bill. Carbon neutrality necessitates imposing a cap on the carbon footprint of data centers to reduce brown energy consumption (e.g., coal and Natural Gas). However, a joint electricity cost minimization and carbon footprint capping favors an offline solution with entire future information. We make use of Lyapunov optimization to design online solutions, namely OnlineCC and OnlineCC-T, to jointly optimize electricity cost and carbon footprint of a set of geo-distributed data centers. OnlineCC and OnlineCC-T leverage one slot and T-slot Lyapunov optimization, respectively, to provide solutions for both stationary and non-stationary data center parameters (e.g., workload). The performance of these solutions is analytically studied against the offline solution with the entire future information and an offline solution with limited lookahead information. We prove that both the online solutions achieve a near-optimal operational cost (electricity cost) compared to the offline algorithms, while deterministically bounding the potential violation of carbon footprint target, depending on the Lyapunov control parameter. A salient feature of the proven bound on OnlineCC's carbon cap violation, is that it can be estimated without the need to solve the offline solution, which is usually hard to solve due to the curse of dimensionality. The bound is also used to design a heuristic for adjusting the value of the Lyapunov control parameter, by significantly reducing its search space. The study also accounts for designing OnlineCC based on an actual data center nonlinear energy consumption model. We show that in this case OnlineCC can be modeled as a convex optimization problem. Finally, we perform a trace-based simulation and a small scale experiment to complement the analysis. The results show that OnlineCC reduces cost by more than 18% compared to a prediction-based online solution while resulting in equal or smaller carbon footprint.

Similar to the previous Chapter, we focus on the electricity operational costs optimization to make the operational cost more efficient. Adapting the results of the previous chapter (Chapter 5), we formulate OnlineCC as a linear programming. In particular, number of active servers is approximated as a real-type decision variable. This simplification significantly improves the computation efficiency of the solution, yet it has a minimal effect on the performance of the solution.

This chapter also links the solutions of the local and the global data center level. TACOMA is a dynamic server and workload management solution which can be locally used by each data center. Instead of integrating TACOMA, we choose to use TACOMA's results in the cost management at the global data center level. In Chapter 4 it is shown that the cooling energy varies depending on the active server set selection algorithm (e.g., TASP and CPSP in TACOMA), and so does the PUE. We consider that the PUE can be characterized using profiling which is a function of data center physical layout, workload, and active server set selection algorithm (see Chapter 4). Using such a PUE model we device Thermal-aware OnlineCC which takes into consideration the impact of global workload and server management on the thermal conditions and on the cooling energy of data centers.

In the rest of the chapter, we first briefly describe the system model (Section 6.1). This section also accounts for formulating an offline optimization problem, namely **P1**, to minimize the energy cost and cap the carbon footprint of a cloud (Section 6.1.3). Next, we present OnlineCC, our online solution to **P1** in Section 6.2, followed by the performance analysis of OnlineCC with respect to several variants of the offline optimal solution. Next, we present Thermal-aware OnlineCC, which further extends OnlineCC to optimize the energy consumption of data centers (Section 6.3. We evaluate the online solutions by (i) using a real-world trace based simulation study in Section 6.4, (ii) using an actual data center energy consumption model in Section 6.4.5, and (iii) using a small scale experimental study in Section refsec:onlinecc:experiment. We conclude in Section 6.6. A summary of optimization problems and the proposed algorithm is given in Table. 6.1.

6.1 Problem Formulation

The problem setting of this chapter is based on the aforementioned system model in Chapter 2. The modeling and the optimization framework are particularly designed to make use of the results from Chapter 4 in order to account for realistic data center cooling power consumption models, and Chapter 5 in order to approximate servers power consumption without significantly affecting the performance of the solutions.

In summary, the optimization framework is designed for a set of N geo-distributed data centers which receive their workload form a total number of M front-ends. Optimizations is performed regularly per each discrete equal time intervals, namely slots, and within a budgeting period (the period for the carbon budget or cap) consisting a total number of S slots. Similar to DAHM solution (see Chapter 5), the optimization is a two level process: (i) deriving the number of required active

Table 6.1: Summary of Variables, Optimization Problems and Algorithms for Joint Optimization of Cost and Carbon Footprints

Symbol	**Definition**
Ψ	carbon cap
ψ	time-average carbon cap
X	virtual queue
V	Lyapunov control param.
X_{lim}	see Lemma 6.2.2
$b_{max}(t)$	maximum carbon footprint over slot t
$b_{min}(t)$	maximum carbon footprint over slot t
b_{max}	maximum per-slot carbon footprint over S
b_{min}	minimum per-slot carbon footprint over S
Ψ_{max}	maximum total carbon footprint over S
Ψ_{min}	minimum total carbon footprint over S
θ	see periodicity assumption in Section 6.1.3
Problem	**Definition**
P1	Offline cost and carbon footprint optimization problem
P2	Variant of **P1** with T slot future data
P3	Thermal-aware variant of **P1**
P4	Variant of **P1** with nonlinear power consumption model
Solution	**Definition**
OnlineCC	Online solution to **P1** and **P2** designed based on one-slot Lyap. optimization
OnlineCC-T	Online solution to **P1** designed based on T-slot Lyap. optimization
Thermal-aware OnlineCC	Online solution to **P3** and **P4** designed based on convex programming and one-slot Lyap. optimization

servers at each data center i, ($y_{i,t} \in \mathbb{N}_0,\ 0 \leqslant y_{i,t} \leqslant Y_i$, where Y_i denotes the total number of servers at data center i), and (ii) deriving the traffic fractions $\lambda_{i,j}(t)$ from each area j to each data center i.

Performance Model We base the performance model on the $M/M/n$ queuing theory as described in Chapter 2. Following the $M/M/n$ queuing theory, the performance requirements can be expressed as the following set of equations:

$$\begin{aligned} &\textstyle\sum_i \lambda_{i,j}(t) = \lambda_j(t),\ \forall j,t\,\text{[service]},\\ &n_i(t)\mu_i > \textstyle\sum_j \lambda_{i,j}(t),\ \forall i,t\,\text{[queuing stability]},\\ &\tfrac{1}{n_i(t)\mu - \sum_j \lambda_{i,j}(t)} \leq d'^{ref},\ \forall i,t\ \text{ [service delay] },\\ &(d^{ref} - (d'^{ref} + d''_{i,j}(t))\lambda_{i,j}(t) \geq 0,\ \forall i,j,t\ \text{ [total delay]},\\ &y_i(t) = (1 + y_i^{slack})n_i(t) \leq Y_i,\ \forall i,t\,\text{[data center capacity]}. \end{aligned} \tag{6.1}$$

In the above, "service" constraint guarantees providing service for all requests, "queuing stability" constraint ensures the queuing model convergence, "service delay" constraint estimates the service delay using the $M/M/n$ queuing model and guarantees maintaining reference average service delay, and "total delay" asserts that the sum of service and network delay is below the reference, d^{ref}. Further, "data center capacity" ensures that number of required servers to maintain the average reference service delay and the additional number of servers (i.e., y^{slack}) to maintain the reference delay during workload spikes are within the data center capacity.

6.1.1 Power Supply and Demand Modeling

Following the results from Chapter 5, we model the average one-slot energy consumption of an active server, denoted by p_i. Then $y_i(t)p_i$ estimates the total one-slot energy consumed by active servers in data center i. The total one-slot energy consumption of the data center can be estimated as follows: $p_i^{tot}(t) = PUE_i(t)p_iy_i(t)$.

As mentioned earlier, PUE of a data center usually varies over time depending on several factors. Of particular interest is air-cooled data centers for which PUE varies with respect to workload and active server set (see Chapter 4). When considering variation of PUE with respect to workload, the total energy consumption of data centers becomes a nonlinear function of number of active servers (see Chapter 4). In this chapter, we first analytically study non-thermal aware carbon capping problem where data center total energy consumption can be formulated as a linear function. Next in Section 6.3, we extend the problem formulation and the solutions for thermal aware carbon capping problem, where workload management is performed with respect to the temperature distribution in data center room and its impact on data center total energy consumption and the PUE.

Linear Power Consumption Model

In this model, similar to the most of existing work [12, 46, 77, 97], we assume PUE is independent of the workload and workload placement, where total energy consumption can be calculated as a linear function of number of active servers as follows:

$$p_i^{tot}(t) = PUE_i(t) p_i y_i(t) \tag{6.2}$$

Note, PUE may vary over time (e.g., depending on temperature [46]).

We proceed with the rest of the study considering linear power consumption model (6.2). Particularly, all of the theoretical results are obtained under the linear power consumption model assumption. We discuss how to extend the results for thermal aware cost and carbon footprint optimization problem in Section 6.3.

Given $p_i^{tot}(t)$, and the available renewable energy, the power draw from the grid should be decided as follows:

$$\begin{aligned} &p_i^{tot}(t) = g_i(t) + r_i(t), \forall i \text{ and } t, \\ &g_i(t) \geq 0, \forall i \text{ and } t. \end{aligned} \tag{6.3}$$

As shown in the above equation, we assume power draw from the grid is always positive.

6.1.2 Carbon Footprint Capping

Denote by $b(t)$ the total carbon footprint of the cloud over slot t: $b(t) = \sum_i b_i(t)$, where $b_i(t)$ can be calculated using the model (2.3). The cloud desires to follow the long-term carbon capping target, denoted by Ψ, which is typically expressed for a year of operation of a data center. Mathematically, for $\psi = \frac{\Psi}{S}$, the long term carbon capping constraint can be written as follows:

$$\frac{1}{S}\sum_{t=0}^{S-1} b(t) \leq \psi. \tag{6.4}$$

6.1.3 Optimization Framework

We also set the renewable energy operational cost to zero, since the primary cost for solar panels and wind turbines is the construction cost. Further, data centers would like to maximize the utilization of their on-site renewable energy. At each slot t, the operation cost is power procurement cost across all the data centers i.e., cost(t)=$\sum_i g_i(t)\alpha_i(t)$, where α denotes the electricity cost. Finally, we formulate the offline workload distribution strategy over the cloud to minimize the long-term average electricity cost of the cloud, which is demonstrated in the following optimization problem, namely **P1**:

$$\begin{array}{ll} \text{minimize}_{g,r,y,\lambda} & \bar{\text{cost}} = \frac{1}{S}\sum_{t=0}^{S-1}\sum_{i} g_i(t)\alpha_i(t), \\ \text{subject to:} & \text{(6.1), (6.3), and (6.4).} \end{array} \tag{6.5}$$

Observe that some of the variables are real (i.e., g_i and $\lambda_{i,j}$) and some are integer (i.e., y_i). Following Lemma 5.1.1 in can be proven that the well-known NP-hard Fixed-Charge Min-Cost Flow problem can be reduced to **P1**. However relaxing y to a real variable has a very negligible impact on the cost given the thousands of servers in data centers, as proven in Proposition 5.2.2 (Chapter 5). Therefore, we consider solving **P1** where all its decision variables are real. In this way, **P1** can be optimally solved using linear programming. Carbon capping constraint (6.4) couples the solution of **P1** over slots. Therefore, it is natural that optimally solving **P1** requires complete offline information (e.g., workload arrival rate, electricity price) which is impractical. To ensure there exist at least one feasible solution to **P1** and to design online solution we make the following assumptions which are practically not too constraining:

- **Boundedness assumption**: The cloud carbon footprint on every slots is upper-bounded by b_{max} which implies that the workload arrival rate and the carbon intensity associated with the cloud are finite for t=0, . . . , S−1, that is true due to the finite number of servers.

- **Feasibility assumption**: There exists at least one sequence of workload distribution policy over slots $t = 0, \ldots, S-1$ that satisfies **P1**'s constraints.

- **Periodicity assumption**: There exits θ number of continuous slots, $\theta << S$, during which if carbon footprint is minimized (i.e., $\forall k \in \theta : b(k){=}b_{min}(k)$, where $b_{min}(k)$ is the minimum possible carbon footprint for slot k which can be achieved by any workload distribution policy) then the average carbon footprint

Algorithm 6.1: OnlineCC Algorithm

```
function OnlineCC
    Initialize the virtual queue $X$.
    for every slot $t = 1 \ldots S$ (beginning of the slot) do
        Predict the system parameters over slot $t$.
        Solve the following problem:
        Minimize:
            $cost_{OnlineCC} = V \sum_i g_i(t)\alpha_i(t) + X(t) \sum_i b_i(t).$    (6.6)
        Subject to: (6.1), and (6.3).
        Update the virtual queue $X$ using (6.7).
    end for
end function
```

over θ becomes lower than that of average carbon cap (ψ). The parameter θ depends on the cycle variation of the cloud carbon footprint as well as the tightness of ψ, i.e., the proximity of the ψ to the minimum feasible average carbon footprint. Consider an extreme case where $\psi \geq b_{max}$, then θ equals to one. If ψ is very tight, then θ becomes close to the cloud cycle variation. Note, given the weekly and daily variation of data center system parameters (e.g., workload, electricity price, carbon emission), we have that $\theta << S$ even for the case where ψ is very tight.

6.2 OnlineCC: Online Cost and Carbon Footprint Optimization Solution

Carbon capping constraint (6.4) in **P1** couples the data center decisions across different time slots. Eliminating (6.4) from **P1** leads to an online problem, however we need a technique for managing the carbon capping requirement. There are some well-studied online algorithms such as Metrical Task System (MTS), and K-servers. These algorithms typically solve online problems to minimize cost, where the cost is given per system state and per state change. The Problem P1 is different in the sense that the coupling property (carbon capping constraint) is a requirement which needs to be met when operating over the budgeting period. Lyapunov optimization refers to the use of a Lyapunov function to optimally control a dynamical system. A Lyapunov function is a nonnegative scalar measure of the system state (e.g., in Problem P1, the deviation of carbon footprint from the cap at a given time). Typically, the function is defined to grow large when the system moves towards undesirable states (e.g., in Problem P1, violation of the cap). System stability is achieved by taking control actions that make the Lyapunov function drift in the negative direction towards zero. We leverage Lyapunov optimization to enables online control of carbon footprint cap. In accordance with Lyapunov optimization, we construct a (virtual) queue with occupancy $X(t)$ to include the total excess carbon footprint beyond the average carbon footprint until the time slot t. Using $X(0) = 0$, we propagate $X(t)$ values over slots as follows:

$$X(t+1) = max[X(t) - \psi, 0] + \sum_{i} b_i(t). \tag{6.7}$$

We design OnlineCC as given in Alg. 6.1 to solve the cost minimization in an online way. OnlineCC, solving the optimization problem in Alg. 6.1, requires only one slot ahead information as the inputs (i.e., $\lambda_j(t)$, $r_i(t)$, $\alpha_i(t)$, $\varepsilon_i^g(t)$, and $\varepsilon_i^r(t)$), since the problem in Alg. 6.1 removes the coupling property of **P1** (i.e., removing the constraint (6.4)). OnlineCC uses the control parameter V (see Alg. 6.1) to adjust the cost minimization and carbon capping tradeoff, for which we provide an analytically supported guideline for its adjustment.

6.2.1 OnlineCC Performance Analysis

We prove Lemma 6.2.1 which helps to prove the worst-case carbon capping violation of OnlineCC. It can be seen that OnlineCC minimizes the weighted sum of the electricity cost and the carbon footprint, weighted by V and $X(t)$, respectively. Lemma 6.2.1 presents a condition under which the second term of (6.6) outweighs its first term such that minimizing carbon footprint yields lesser value for OnlineCC objective function. The condition in Lemma 1 is related to the parameter X_{lim} which is a bound of the electricity cost difference over the carbon footprint difference across data centers for the entire time period. Mathematically, it is represented as $X_{lim}{=}\frac{c_{max}-c_{min}}{b'}$, where, $c_{max}{=}\max_{i,t}(\frac{p_i}{\mu_i}\alpha_i(t))$, $c_{min}{=}\min_{i,t}(\frac{p_i}{\mu_i}\alpha_i(t))$, $b{=}min_{i,k,t,i\neq k}(\frac{p_i}{\mu_i}\varepsilon_i^g(t){-}\frac{p_k}{\mu_k}\varepsilon_k^g(t)|\frac{p_i}{\mu_i}\varepsilon_i^g(t) \neq \frac{p_k}{\mu_k}\varepsilon_k^g(t))$. Next in Lemma 6.2.2 we prove how much farther X can grow beyond VX_{lim}. The upper bound of X specifies OnlineCC carbon violation (see (6.7)).

Lemma 6.2.1. *Suppose data centers require non-zero active servers, and for a given slot t, $X(t){\geq}VX_{lim}$, then OnlineCC minimizes carbon footprint for slot t.*

Proof. We prove the lemma for $y^{slack}{=}0$ for the sake of notation brevity. Let $p_i'^{tot}$, g_i', y_i', λ_i' and r_i' denote the value of parameters when minimizing carbon footprint. For OnlineCC to minimize the carbon footprint for any values of p_i^{tot}, g_i, y_i, λ_i, and r_i we should have the following:

$$\begin{aligned}
&V\textstyle\sum_i(p_i^{tot}(t)-r_i(t))\alpha_i(t)\\
&+X(t)\textstyle\sum_i(p_i^{tot}(t)-r_i(t))\varepsilon_i^g(t)+r_i(t)\varepsilon_i^r(t)\\
&\geq V\textstyle\sum_i(p_i'^{tot}(t)-r_i'(t))\alpha_i(t)\\
&+X(t)\textstyle\sum_i(p_i'^{tot}(t)-r_i'(t))\varepsilon_i^g(\tau)+r_i'(t)\varepsilon_i^r(t).
\end{aligned}$$

The rest of the proof is about obtaining a bound for which the above inequality always holds. Since the carbon intensity of renewable power is much lower than that of grid where brown energy forms its significant energy source, $\varepsilon_i^r(t) << \varepsilon_i^g(t)$, and that we consider zero cost for on-site renewable power, increasing the renewable energy favors both reducing the total electricity cost and carbon footprint i.e., $r_i(t) = r'_i(t)$. Therefore, given that $p_i^{tot} = y_i p_i$, to prove the above inequality it is sufficient to show:

$$X(t) \geq V \frac{\sum_i y'_i(t) p_i \alpha_i(t) - \sum_i y_i(t) p_i \alpha_i(t)}{\sum_i y_i(t) p_i \varepsilon_i^g(t) - \sum_i y'_i(t) p_i \varepsilon_i^g(t)}. \tag{6.8}$$

Using "service delay" constraint in (6.1) and the assumptions of $y^{slack} = 0$ and $y_i > 0$ we have that $y_i(t) = \frac{\sum_j \lambda_{i,j}(t)}{\mu_i} + \frac{1}{d\mu_i}$. Plugging this into (6.8), rearranging the terms, cancelling the term $\frac{p_i}{d\mu_i}$ from both numerator and denominator, and defining the parameters cost per flow, c as $c_i(t) = \frac{p_i \alpha_i(t)}{\mu_i}$, and carbon per flow, b' as $b'_i(t) = \frac{p_i \alpha_i(t)}{\mu_i}$, it is left to prove the following:

$$\begin{aligned} X(t) &\geq V \frac{\sum_i \sum_j \lambda'_{i,j}(t) c_i(t) - \sum_i \sum_j \lambda_{i,j}(t) c_i(t)}{\sum_i \sum_j \lambda_{i,j}(t) b_i(t) - \sum_i \sum_j \lambda'_{i,j}(t) b_i(t)} \\ &\leq V \frac{\sum_i \sum_j \lambda'_{i,j}(t) c_{max} - \sum_i \sum_j \lambda_{i,j}(t) c_{min}}{\sum_i \sum_j \lambda_{i,j}(t) b_1 - \sum_i \sum_j \lambda'_{i,j}(t) b_2}. \end{aligned} \tag{6.9}$$

where $c_{max} = \max_{i,t}(c_i(t))$, $c_{min} = \min_{i,t}(c_i(t))$, and b_1, and b_2 are such that

$$b_1 - b_2 = \min_{i,k,t,i \neq k} (b'_i(t) - b'_k(t) | b'_i(t) \neq b'_k(t)).$$

Note that for every $b'_i(t) = b'_k(t)$, we have $g_i(t) = g'_i(t)$, and $g_k(t) = g'_k(t)$ (this is because online algorithm first minimizes the carbon footprint and then the electricity cost) which means that such carbon factors do not affect the above inequality. Given $\sum_i \sum_j \lambda_{i,j}(t) = \sum_i \sum_j \lambda'_{i,j}(t)$, the lemma follows. □

Lemma 6.2.2. *Suppose $X(0) = 0$, then using OnlineCC, the virtual queue X is deterministically bounded as: $\forall t = 0 \ldots S-1, X(t) \leq V X_{lim} + \max(\theta - 2, 0)(b_{max} - \psi) + b_{max}$.*

We consider that $b_{max} > \psi$, otherwise the proof is trivial. We prove by induction. Clearly the lemma holds for $t = 0$. Now suppose it holds for t, we will prove it for

$t{+}1$. First, suppose $X(t){\leq}VX_{lim}$ then according to (6.7) the most that $X(t{+}1)$ can increase in one slot is $b_{max}{-}\psi$. Next, suppose $VX_{lim}{<}X(t){\leq}VX_{lim}{+}k(b_{max}{-}\psi){+}b_{max}$, for every integer $k{<}\theta{-}2$. If $b(t{+}1){\leq}\psi$ then we have:

$$X(t+1){=}X(t){-}\psi{+}b(t{+}1){\leq}X(t){-}\psi{+}\psi{\leq}X(t).$$

Otherwise $X(t{+}1)$ at most becomes $VX_{lim}{+}(k{+}1)(b_{max}{-}\psi){+}b_{max}$. Finally, suppose $X(t){=}VX_{lim}{+}\max(\theta{-}2,0)(b_{max}{-}\psi){+}b_{max}$. According to Lemma 6.2.1, once $X(t)$ exceeds VX_{lim}, OnlineCC minimizes carbon footprint. Thus the value of $X(t)$ implies that there has been $\theta{-}1$ number of slots that OnlineCC has minimized carbon footprint, yet it has incurred carbon footprint of b_{max} on each of those $\theta{-}1$ slots (worst case scenario). According to definition of θ, the minimum carbon footprint in $t{+}1$, i.e., $b_{min}(t{+}1)$ must satisfy $b_{min}(t{+}1){<}\psi$ which follows that $X(t{+}1){=}X(t){-}\psi{+}b_{min}(t{+}1) \leq X(t){-}\psi{+}\psi{\leq}X(t)$.

Now we present Theorem 6.2.3 which is built upon Lyapunov optimization [90], and makes use of Lemma 6.2.1 and Lemma 6.2.2 to provide the performance analysis of OnlineCC.

Theorem 6.2.3. *(Performance Bound Analysis): Suppose $X(0){=}0$, and that power demands and input workloads of data centers are bounded. Then, given any fixed control parameter $V{>}0$, OnlineCC achieves the following:*

1. *Assume data center parameters are i.i.d. over every slot, the time averaged cost under the online algorithm is within $\frac{B}{V}$ of the offline optimal time averaged cost value, cost*:*

$$\limsup_{t\to\infty}\tfrac{1}{t}\textstyle\sum_{\tau=0}^{t-1}\mathbb{E}\{cost_{OnlineCC}(\tau)\} \leq cost^{*} + \tfrac{B}{V}, \tag{6.10}$$

where $B = \frac{1}{2}({b_{max}}^{2}{+}\psi^{2})$, $cost_{OnlineCC}$ refers to the value of (6.6) as shown in Alg. 6.1, and $cost^{*}$ is the optimal solution to ***P1****.*

2. *The carbon footprint capping constraint is approximately satisfied with a bounded deviation as follows:*

$$\textstyle\sum_{t=0}^{S-1}\sum_i b_i(t) \leq \Psi + \min(R1, R2), \tag{6.11}$$

where,

$$R1{=}VX_{lim}{+}\max(\theta{-}2,0)(b_{max}{-}\psi){+}b_{max}$$

, and

$$R2{=}\sqrt{2}\sqrt{SB{+}V(Scost^{*}{-}\sum_{t=0}^{S-1}cost_{OnlineCC}(t))}.$$

Proof. First, the proof of (6.10) builds upon the recently-developed Lyapunov optimization technique [90]. We only show the key steps. Let's define a quadratic Lyapunov function $L(t)$ that measures the aggregate carbon deficit in the system: $L(t) = \frac{1}{2}X(t)^2$. Next, let's define the one-slot Lyapunov drift, $\Delta(t)$ as the expected change in the Lyapunov function over every slot as follows: $\Delta(t) = \frac{1}{2}(\mathbb{E}\{L(t+1) - L(t)\}|X(t))$. Now we derive the upper bound on $\Delta(t)$ as follows. By (6.7) we have that:

$$(X(t+1))^2 \leq (X(t) - \psi + \sum_i b_i(t))^2.$$

Squaring both side of (6.7), and given that $\sum_i b_i(t) = b(t) \leq b_{max}$ we have:

$$\begin{aligned}&[X(t+1)]^2 \leq [X(t)]^2 + [\psi]^2 + [\textstyle\sum_i b_i(t)]^2\\ &-2X(t)\psi + 2X(t)\textstyle\sum_i b_i(t) - 2\psi\sum_i b_i(t)\\ &\Rightarrow [X(t+1)]^2 - [X(t)]^2 \leq {b_{max}}^2 + \psi^2 - 2X(t)[\psi - \textstyle\sum_i b_i(t)].\end{aligned} \tag{6.12}$$

Now multiplying the above inequality by $\frac{1}{2}$, taking expectations over $X(t)$, and $b_i(t)$, conditioning on $X(t)$, we get the one-slot conditional Lyapunov drift $\Delta(t)$:

$$\Delta(t) \leq B - X(t)\mathbb{E}\{\psi - \textstyle\sum_i b_i(t)|X(t)\},$$

where $B{=}\frac{1}{2}({b_{max}}^2{+}\psi^2)$. Adding the cost as penalty term to the both side of the above inequality, i.e., $V\text{cost}(t)$, we get:

$$\Delta(t) + V\text{cost}(t) \leq B + V\text{cost}(t) - X(t)\mathbb{E}\{\psi - \textstyle\sum_i b_i(t)|X(t)\}. \tag{6.13}$$

Observe that OnlineCC as shown in Alg.(6.1) minimizes the right hand side of (6.13). The i.i.d. assumptions of input parameters, ensures the existence of an optimal stationary randomized policy π which can achieve as follows for solving problem **P2** at all $t = 0 \ldots S{-}1$: $\mathbb{E}\{b_i(t)\} \leq \psi$ and $\mathbb{E}\{\text{cost}^{\pi}\}(t){=}\text{cost}^*$, where cost^* is the offline optimal average cost (this can be proven using Caratheodory's theorem similar to the proof in [90]).

Using the fact that OnlineCC is constructed to minimize the R.H.S. of (6.13), we have:

$$\begin{aligned} &\Delta(t) + V\text{cost}_{OnlineCC}(t) \leq \\ &B + V\text{cost}^{alt}(t) - X(t)\mathbb{E}\{\psi - \textstyle\sum_i b_i(t)|X(t)\}, \end{aligned} \tag{6.14}$$

where *alt* represents any alternate policy (including stationary randomized policy π) that can be implemented over slot t. Then plugging the control decisions corresponding to the stationary randomized policy π, we get:

$$\Delta(t) + V\text{cost}_{OnlineCC}(t) \leq B + V\text{cost}^*. \tag{6.15}$$

Taking the expectations from both sides, summing the above over $t{=}0 \ldots S{-}1$, using the fact that $\Delta(t){\geq}0$, and dividing both sides by SV, we have:

$$\frac{1}{S}\mathbb{E}\{\sum_{t=0}^{S-1} \text{cost}_{OnlineCC}(t)\} \leq V\text{cost}^* + \frac{B}{V}.$$

Taking a lim sup as $S{\to}\infty$, we complete the proof.

Next, to prove (6.11) we first prove OnlineCC carbon violation never exceeds $R1$ as defined in theorem. This immediately follows from Lemma 6.2.2, and (6.7).

The reason is that, by definition of the virtual queue X, i.e., (6.7), the total carbon footprint violation of OnlineCC from cap (i.e., b) up to end of slot t is equal to $\max(X(t){-}\psi, 0)$. and that by Lemma 6.2.2 we have that

$$X(S) \leq VX_{lim}{+}\max(\theta{-}2, 0)(b_{max}{-}\psi){+}b_{max}.$$

To prove (6.11), it is left to show that OnlineCC carbon violation never exceeds $R2$, where $R2$ is defined in the theorem. From the carbon deficit queue dynamic (6.7), we have that:

$$X(t+1) - X(t) \geq \sum_i b_i(t) - \psi.$$

Summing the above over $t = 01, \ldots S$, and using the fact that $X(0){=}0$ we obtain the following:

$$\sum_{t=0}^{S-1} \sum_i b_i(t) \leq S\psi + X(S). \tag{6.16}$$

Similarly, summing (6.15) over the entire budgeting period, i.e., $t{=}0 \ldots S$, and using the fact that $L(0){=}X(0){=}0$ yields:

$$L(S) = 1/2X^2(S) \leq SB + SV cost^* - V \sum_{t=0}^{S-1} \text{cost}_{OnlineCC}(t). \tag{6.17}$$

Plugging (6.17) into (6.16), and using the fact that $\Psi{=}S\psi$ we prove the theorem as follows:

$$\sum_{t=0}^{S-1} b_i(t) \leq \Psi + \sqrt{2}\sqrt{SB + V(Scost^* - \sum_{t=0}^{S-1} cost_{OnlineCC}(t))}. \tag{6.18}$$

□

The results of Lemma 6.2.1 and (6.11) are important, since they provide a deterministic bound on the maximum carbon capping violation of OnlineCC. The intuition behind $R1$ in (6.11) is that the carbon cap violation is bounded by the maximum value that X can get which is equal to the sum of VX_{lim} (the upper bound value of X to minimize carbon footprint) and the total carbon footprint backlog accumulated when minimizing carbon footprint in the worst case (i.e., over θ). The salient feature of this bound (i.e., $R1$) is that the worst case carbon capping violation can be calculated without the need to solve the optimal offline solution (as opposed to the bound of $R2$ which is derived using standard steps of Lyapunov optimization). This is important, since the optimal offline solution is hard to solve due to the curse of dimensionality problem. Further, per slot carbon capping violation from bound $R1$ equals to $\frac{R1}{S}$. Since R1 is independent of the budgeting period length (S), the per slot carbon capping violation becomes tight for large S as long as X_{lim} and θ do not vary significantly with increasing S. A tight bound on the carbon capping violation of OnlineCC is of utmost importance as it can be used to adjust the additional carbon credit required for the cloud to achieve carbon neutrality. Finally, $R1$ can be used to adjust the value of V as described in the section below.

How to Choose V Value?:

OnlineCC uses a control parameter $V{>}0$ that affects the distance from optimality. Particularly, according to Theorem 6.2.3, the algorithm achieves an average cost no more than $O(1/V)$ distance above the optimal average cost, while the large value of V comes at the expense of an $O(V)$ tradeoff in achieving the carbon cap. In this section we present two heuristic solutions which intuitively guide on how to choose V value.

First solution: According to (6.7) the aggregated carbon violation until time t equals $\max(X(t){-}\psi)$. Suppose we choose $V{=}V_{min}$, where $\psi{=}V_{min}X_{lim}$, then according to Lemma 6.2.1, OnlineCC minimizes carbon footprint whenever either the carbon footprint over a slot exceeds ψ (due to peak workload) or the sum of backlog and the slot carbon footprint exceed ψ. This means that choosing $V{=}V_{min}$, OnlineCC most of the time yields a value lower than that of ψ, since the offline solution does not always minimizes the carbon footprint when the workload is at its peak (e.g., for loose ψ). This is particularly true because VX_{lim} is an upper-bound value for OnlineCC to minimize the carbon footprint. Choosing the right value for V such that OnlineCC achieves near or very close to the carbon cap (Ψ) depends on the variability of the electricity cost ($\alpha_i(t)$) and the carbon emission ($\varepsilon_i^g(t)$) of data centers over time. The right V value also depends on θ. The input workload, the carbon intensities and the electricity prices periodically have ups and downs (e.g., daily variation). Depending on the variation periodicity of these parameters, the optimal offline solution may violate the average carbon cap in some slots (e.g., where workload is at its peak), which can be compensated in future slots (e.g., where workload is low). The parameters variation period and the carbon cap value also determines θ. One heuristic solution to imitate this behavior, is to choose V such that on average around $\frac{\theta}{2}\psi$ violations above the average carbon cap is allowed, i.e., V such that $\frac{\theta}{2}\psi = VX_{lim}$ or $V = \frac{\theta}{2}V_{min}$. Further, for choosing V we should consider how tight the bound X_{lim} is, which can be approximately calculated using the data center input parameters (i.e., peak to mean ratio of the electricity cost and the carbon intensities).

Second solution: Another solution is to find the sweet V value to minimize the sum of cost and carbon capping violation from Optimal. According to (6.10) and (6.11), such minimization can be written as follows:

$$\text{Minimize } \frac{B}{V} + VX_{lim} + \max(\theta - 2, 0)(b_{max} - \psi) + b_{max}$$

This can be solved analytically, which yields $V = \sqrt{\frac{B}{X_{lim}}}$. We evaluate the performance of the two solutions in the experimental study.

In the following section, first we give a simple example to illustrate how OnlineCC works. Then, we study OnlineCC-T which is designed based on T-slot Lyapunov optimization. Next, we study the performance of OnlineCC with respect to an

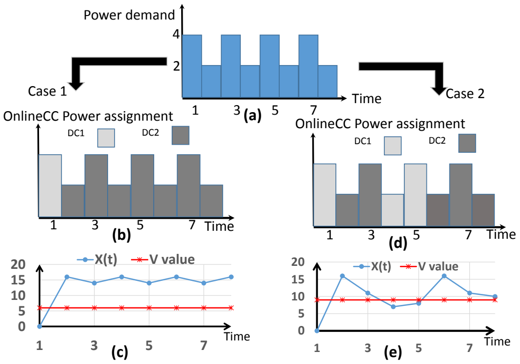


Figure 6.1: Illustrating OnlineCC and V Adjustment Solutions.

offline solution with T slots future information.

Numerical Examples

Consider a cloud consisting of only one front-end and two data centers (DC1 and DC2), each having identical power consumption and service rate per server (p=p_i, and μ=μ_i). Suppose, both the electricity cost and the carbon footprint are constant over time, their magnitude are comparable (in the same range) and that the data center with the lower cost has the higher carbon footprint and vice versa: $\forall t, \alpha_1(t)$=2 \$/J, $\epsilon_1^g(t)$=4 CO$_2$ g/J, $\alpha_1(t)$=4 \$/J, $\epsilon_1^g(t)$=2 CO$_2$ g/J. Observe that DC1 and DC2 are optimal destination for power demand to minimize electricity cost and carbon footprint, respectively. Finally, consider T=8, and a cyclic power demand as given in Fig. 6.1(a). Observe that in this setting the minimum feasible average carbon footprint equals to eight and X_{lim}=1. To illustrate the proposed solutions for OnlineCC and V adjustment, consider two cases. First, suppose ψ is equal to the minimum feasible value, i.e., ψ=8. According to the solution V_{min}=$\frac{\psi}{X_{lim}}$=8. Note, the offline solution chooses to minimize the carbon cap over all slots to satisfy the carbon cap target (by assigning the entire power demand to DC2). Assume we choose V=V_{min}. For this setting, as shown in Fig. 6.1(b) except the first slot, OnlineCC assigns the

Algorithm 6.2: OnlineCC-T Algorithm

```
Initialize the virtual queue $X$
for every $T$ frame $k = 0 \ldots K-1$ (beginning of the frame k) do
    Predict the system parameters over slot $(kT, kT+T-1)$
    Minimize:
        $\text{cost}_{\text{OnlineCC-T}}(k) = V\sum_{\tau=kT}^{KT+T-1}\sum_i g_i(\tau)\alpha_i(\tau) + X_i(kT)\sum_{\tau=kT}^{KT+T-1}\sum_i b_i(t)$    (6.19)
    Subject to: (6.1), and (6.3)
    Update the virtual queue $X$ as $X(kT+T) = max(X(kT) - T\psi, 0) + \sum_{\tau=kT}^{kT+T-1} b_i(\tau)$
end for
```

power demand to DC2, since assigning power to DC2 causes to minimize the carbon footprint. This is because X value (see (6.7)) quickly exceeds V value (see Fig. 6.1(c)) resulting in the second term of OnlineCC objective function (see (6.6)) outweighing the first term such that minimizing the carbon footprint yields smaller value for OnlineCC objective function compared to minimizing the electricity cost.

Next, suppose ψ gets a larger feasible value i.e., ψ=9. By choosing V=V_{min}=ψ=9, it can be seen in Figs. 6.1(d) and (e) that as soon as either the slot carbon footprint exceeds ψ (slots 3 and 7), or the sum of the backlog and the slot carbon emission exceeds ψ (slots 2 and 6), X value exceeds V and that OnlineCC minimizes the carbon footprint by assigning the power demand to DC2. This results an average carbon footprint of 8.2, a value less than ψ (in agreement with Section 6.2.1).

6.2.2 OnlineCC-T: Leveraging Predictability of Parameters

The competitive cost ratio of OnlineCC, (Theorem 6.2.3) is based on i.i.d. assumption of data center system parameters. Therefore, there is this concern that the results may not hold depending on the nature of parameters (e.g., non-stationary distribution for renewable energy). Further, data center system parameters, usually have seasonality pattern (e.g., daily and weekly), where the information can be predicted with reasonable accuracy. Therefore, there is this question that whether the performance of OnlineCC can be improved by leveraging the predictability of information in T slots ahead ($T \geq 1$) rather than one slot ahead. To address the above, we also develop and evaluate an online solution using T-slot Lyapunov optimization, namely OnlineCC-T. The parameter T can be viewed as the time required for the system to reach near steady state. This is due to the fact that systems with non-i.i.d. parameters such as workload arrival rate may have the systems states that yield low workload arrival rates or large arrival bursts for many time slots within every T. Similarly, the system within T may have many slots with very low renewable energy and low electricity price, and many slots of large renewable energy burst or high electricity price. However, the variation ofsystem parameters for every T can be assumed stationary (an i.i.d. system case can be viewed as a special case where $T{=}1$). Such definition is consistent with the general nature of workload, solar energy, and electricity price as they usually have a daily (or weekly) basis variation pattern. OnlineCC-T, given in Alg. 6.2 can be viewed as a general case of OnlineCC where the optimization problem in Alg. 6.1is solved over every T slots. Further, the carbon deficit queue (X) is updated over every T slot. The following theorem shows the performance of OnlineCC-T with respect to the offline optimal solution.

Theorem 6.2.4. *Suppose* $X(0){=}0$*, and that power demand and input workload of* *data centers are bounded. Further, let $cost_T^*$ denotes the optimal solution to the problem **P1** over every T slot, $cost_{OnlineCC-T}$ denotes the OnlineCC-T per-T-slot cost as defined in Alg. 6.2, and B denotes a finite constant parameter as follows: $B{=}\frac{1}{2}(b_{max}^2{+}\psi^2)$. Then given integer numbers $k{=}0,1\ldots K$, where $t = kT$ and $S{=}KT$, and given any fixed control parameter $V{>}0$, OnlineCC-T achieves the following:*

1. *If data center parameters are i.i.d. over every T-slots, then the time averaged cost under the online algorithm is within $\frac{BT}{V}$ of the offline optimal time averaged cost, $cost_T^*$:*

$$\begin{aligned} &\bar{cost}_{OnlineCC-T} = \\ &\limsup_{t\to\infty}\tfrac{1}{t}\textstyle\sum_{k=0}^{K-1}\{\sum_{\tau=kT}^{kT+T-1} cost_{OnlineCC-T}(\tau)\} \\ &\le cost_T^* + \tfrac{BT}{V}. \end{aligned} \tag{6.20}$$

2. *The carbon footprint capping constraint is approximately satisfied with a bounded deviation as follows:*

$$\textstyle\sum_{t=0}^{S-1}\sum_i b_i(t) \le \Psi + \min(R1, R2), \tag{6.21}$$

where,

$$R1{=}VX_{lim}{+}\max(\theta_T{-}2,0)(Tb_{max}{-}T\psi){+}Tb_{max}$$

, and

$$R2{=}\sqrt{2}\sqrt{SB{+}V(Kcost_T^*{-}\sum_{k=0}^{K-1}cost_{OnlineCC-T}(k))}$$

Proof. The similar steps to the proof of Theorem 6.2.3 can be taken to prove this theorem. The only difference is that we need to define the T-slot Lyapunov optimization drift. □

The above theorem, shows that OnlineCC and OnlineCC-T have similar performance with respect to offline optimal solution. In particular, the competitive ratio of OnlineCC-T cost over cost_T^* (average optimal cost over T slots) is similar to the competitive ratio of OnlineCC cost over cost^* (average optimal cost over every slot). Further, the similar steps as Section 6.2.1 can be taken to choose V for OnlineCC-T. The first solution in Section 6.2.1 suggests to use a V_{min} for OnlineCC-T such that $T\psi{=}V_{min}$. The second solution of Section 6.2.1 suggests a V value equal to $\sqrt{\frac{BT}{X_{lim}}}$.

6.2.3 Offline Solution with T Slots Lookahead Information

Suppose we divide the entire budgeting period into K equal time frames each consisting of $T{\geq}1$ time slots such that $S{=}KT$. Consider an instance of **P1** where the optimal solution for the problem equals to the sum of optimal solutions for each of independent K frames (note we overload the parameter T to define OnlineCC-T and the offline solution with limited lookahead information). The question is how is the performance of OnlineCC for such an instance of the problem. This is important, since the performance of OnlineCC is often compared to a prediction based solution e.g. the solution in [69] and our reference online solution, OnlineH (see Section 6.4). The optimal solution of **P3** is a representative of a prediction based solution which operates near optimal solution. Hence, we make use of offline algorithm with T-slot lookahead information as a benchmark. To this end, we divide the carbon capping target into chunks per each frame, denoted by Ψ_T as follows $\Psi_T{=}\frac{\Psi}{K}$. Then, at the beginning of every k-th frame, for $k = 0, 1, \ldots K-1$, the offline decisions are chosen to solve the following problem, namely **P2**:

$$\boxed{\begin{array}{ll} \text{minimize}_{g,r,y,\lambda} & \bar{\text{cost}} = \frac{1}{T}\sum_{t=kT}^{kT+T-1}\sum_i g_i(t)\alpha_i(t) \\ \text{subject to:} & (6.1), \text{ and } (6.3) \\ & \sum_{t=kT}^{kT+T-1}\sum_i b_i(t) \leq \Psi_T. \end{array}} \quad (6.22)$$

We also consider that **P2**, similar to **P1**, fulfills the **Boundedness**, **Feasibility**, and

Periodicity assumptions.

Next, building upon Lyapunov optimization technique, we formalize the performance analysis of OnlineCC in Theorem 6.2.5 where $cost_k^*$ denotes the optimal cost of **P2** for each frame k.

Theorem 6.2.5. *(Performance Bound Analysis w.r.t. offline with T slots lookahead information): Consider the optimal solution to **P2** under the aforementioned boundedness and feasibility assumptions and define B' as follows:*

$$B' = BT{+}1/2T(T{-}1)(b_{max}^2{-}b_{max}\psi{+}\psi^2).$$

Then, for any $T{\in}\mathbb{N}$, and $K{\in}\mathbb{N}$ such that $S{=}KT$, the following statements hold.

1. *If data center parameters are i.i.d. over every slots, then the time averaged cost under the online algorithm is within $\frac{B'}{V}$ of the offline solution with T slot lookahead information:*

   $$\tfrac{1}{K}\textstyle\sum_{\tau=0}^{S-1} cost_{OnlineCC}(t) \leq \frac{B'}{V} + \frac{1}{K}\sum_{k=0}^{K} cost_k^* \qquad (6.23)$$

2. *The carbon footprint capping constraint is approximately satisfied with a bounded deviation as follows:*

   $$\textstyle\sum_{t=0}^{S-1}\sum_i b_i(t) \leq K\Psi_T + \min(R1, R2), \qquad (6.24)$$

   where,

   $$R1{=}VX_{lim}{+}\max(\theta{-}2,0)(b_{max}{-}\psi){+}b_{max},$$

   and

   $$R2{=}\sqrt{KB'{+}V(\sum_{k=0}^{K} cost_{T,k}^*{-}\sum_{k=0}^{K}\sum_{t=kT}^{kT+T-1} cost_{OnlineCC}(t))}$$

Proof. To prove (6.23), let's define the T-slot Lyapunov drift, $\Delta_T(t) = L(t+T) - L(t)$ as the expected change in the Lyapunov function over every T-slot, where the Lyapunov function is defined in Theorem 6.2.3. We define $k \in \mathbb{N}$ where $k = 0 \dots K-1$. Consider $\Delta(kT)$, summing (6.13) over a T-slots, i.e., $\tau = kT \dots kT+T-1$, yields:

$$\begin{aligned}&\Delta_T(kT) + V \textstyle\sum_{\tau=kT}^{kT+T-1} cost(\tau) \leq TB + V \textstyle\sum_{\tau=kT}^{kT+T-1} \mathrm{cost}(\tau)\\ &- \textstyle\sum_{\tau=kT}^{kT+T-1} X(\tau)\mathbb{E}\{\psi - \sum_i b_i(\tau)\}.\end{aligned} \tag{6.25}$$

According to (6.7) we have that:

$$\begin{aligned}&\forall \tau \in [kT, kT+T-1]:\\ &X(kT) - (\tau - kT)\psi \leq X(\tau) \leq X(kT) + (\tau - kT)(b_{max} - \psi).\end{aligned}$$

Consequently we have:

$$\begin{aligned}&\textstyle\sum_{\tau=kT}^{kT+T-1} X(\tau) \sum_i b_i(\tau) - X(\tau)\psi\\ &\leq X(kT) \textstyle\sum_{\tau=kT}^{kT+T-1} (\sum_i b_i(\tau) - \psi) + \frac{1}{2}T(T-1)(b_{max}^2 - b_{max}\psi + \psi^2).\end{aligned}$$

Plugging the above into (6.25) we get:

$$\begin{aligned}&\Delta_T(kT) + V \textstyle\sum_{\tau=kT}^{kT+T-1} cost(\tau) \leq B' + V \textstyle\sum_{\tau=kT}^{kT+T-1} \mathrm{cost}(\tau)\\ &-X(kT) \textstyle\sum_{\tau=kT}^{kT+T-1} \mathbb{E}\{\psi - \sum_i b_i(\tau)\},\end{aligned} \tag{6.26}$$

where B' is as follows:

$$B' = BT + \frac{1}{2}T(T-1)(b_{max}^2 - b_{max}\psi + \psi^2).$$

Note that, OnlineCC explicitly minimizes the right-hand side of the above inequality. Thus, by applying OnlineCC on the left-hand side and considering the optimal T-slot lookahead policy on the right-hand side of (6.26), denoted by cost_k^*, and summing the inequality over $k \in \{0, 1, \dots K\}$ we obtain the following inequality:

$$L(KT) - L(0) + V \sum_{t=0}^{KT-1} \mathrm{cost}(t) \leq KB' + V \sum_{k=0}^{K} \mathrm{cost}_k^*. \tag{6.27}$$

Dividing by VK, using the fact $L(KT) > 0$ and $L(0) = 0$ and rearranging terms yields:

$$\frac{1}{K}\sum_{\tau=0}^{S-1}\text{cost}(t) \leq \frac{1}{K}\sum_{k=0}^{K}\text{cost}_k^* + \frac{B'}{V}.$$

To prove (6.24), following the carbon deficit queue dynamic specified by (6.7), we have for any $t \in [kT, kT+T-1]$, and any $k = 0, 1 \ldots K$:

$$X(t+1) - X(t) \geq \sum_i b_i(t) - \psi.$$

By summing the above inequality over a T-slot frame: $t{=}kT \ldots kT{+}T{-}1$, we obtain:

$$X(kT+T) - X(kT) \geq \textstyle\sum_{t=kT}^{kT+T-1}\sum_i b_i(t) - T\psi.$$

Thus:

$$\sum_{t=kT}^{kT+T-1}\sum_i b_i(t) \leq T\psi + X(kT+T) - X(kT). \tag{6.28}$$

Summing (6.28) over k=0, 1 ... $K-1$, and using the fact that $X(0)$=0 we obtain the following:

$$\sum_{t=0}^{KT-1}\sum_i b_i(t) \leq KT\psi + X(KT). \tag{6.29}$$

Further, from (6.27) we have:

$$\frac{1}{2}X^2(KT) \leq KB' + V\sum_{k=0}^{K}\text{cost}_k^* - V\sum_{t=0}^{S-1}\text{cost}_{OnlineCC}(t).$$

Plugging (6.29) in the above inequality we obtain:

$$\sum_{t=0}^{KT-1}\sum_i b_i(t) \leq \Psi + \sqrt{2}\sqrt{KB' + V(\sum_{k=0}^{K}\text{cost}_k^* - \sum_{t=0}^{S-1}\text{cost}_{OnlineCC}(t))}.$$

□

The above results are very similar to the results of Theorem 6.2.3. However, in practice it is easer to design OnlineCC which is competitive to Offline solution with T lookahead. The reason is that the parameter V can be found in an easier way depending on the parameters within every frame T.

6.3 Thermal-aware Cost Minimization and Carbon Footprint Capping

This section extends the problem formulation and the solutions for data centers with non-uniform heat distribution, where the servers' power consumption affect the temperature distribution and the cooling energy of the data centers. From Chapter 4 recall that the heat recirculation coefficients for all pairs of servers in a data center can be modeled considering the data center layout and thermodynamic conditions: $H_i = \{h_{i,m,n}\}_{Y_i \times Y_i}$ where each $h_{i,m,n}$ denotes the fraction of heat that flows from server n to server m among all Y_i servers of data center i [110]. Denote by $T_{sup,i}$, the supplied temperature of the cooling system at data center i, the maximum allowed cooling system's supplied temperature is limited by the following constraint: $T_{sup,i}(t) \leq T_{red,i} - \max(H_i \mathbf{p}_i(t))$, where $\mathbf{p}_i(t)$ denotes the power consumption vectors of servers at data center i, i.e., $\mathbf{p}_i(t) = \{p_{i,1}(t) \ldots p_{i,m}(t), \ldots p_{i,Y_i}(t)\}$. Cooling power can be modeled as the ratio of computing power over *coefficient of performance* (CoP) of the cooling system. CoP is the ratio of the heat removed (i.e., computing power) over the work required to remove that heat (i.e., cooling power) and is typically a supper-linear function of the supplied temperature ($T_{sup,i}$). In order to account for the active server set, we introduce the binary variable $x_{i,m}$ which captures the power state of server m at data center i (i.e., "1" for active state and "0" for inactive state) and denote $\mathbf{x}_i$ to represent its vector. Also assume $p_{i,m}$ denotes the average power consumption of server m at data center i in the active state. Then, Hadamard product of $\mathbf{p}_i$ and $\mathbf{x}_i$ denoted by $\mathbf{p}_i \circ \mathbf{x}_i(t)$ gives the power consumption vector of the active servers. Now, the total data center power consumption (power consumption of active servers and the cooling system) can be formulated as follows:

$$\begin{aligned}
&p_i^{tot}(t) = \textstyle\sum_m p_{i,m} x_{i,m}(t)(1 + \frac{1}{CoP(T_{sup,i}(t)} \\
&T_{sup,i}(t) \leq T_{red,i} - \max H_i(\mathbf{p}_i \circ x_i(t))), \\
&\textstyle\sum_m x_{i,m}(t) = y_i(t).
\end{aligned} \tag{6.30}$$

Observe that the above data center power consumption model captures the impact of active server set on the temperature distribution in the data center room and consequently on the cooling power consumption. Using this model, the offline thermal aware cost minimization and carbon footprint problem, namely **P3**, can be formalized as follows:

$$\boxed{\begin{array}{ll}
\text{minimize}_{g,r,y,\lambda,x} & \bar{\text{cost}} = \frac{1}{S}\sum_{t=0}^{S-1}\sum_i g_i(t)\alpha_i(t), \\
\text{subject to:} & \text{(6.1), (6.30), and (6.4).}
\end{array} \qquad (6.31)}$$

A solution to this problem would specify, the active server set of each data center (i.e., $x_{i,m}(t)$ which also implies $y_i(t)$), the workload distribution policy of front-ends and data centers (i.e., $\lambda_{i,j}(t)$), and the power supply model of each data center ($g_i(t)$ and $r_i(t)$). Observe that some of the variables are reals (i.e., $\lambda_{i,j}(t)$, $g_i(t)$ and $r_i(t)$) and some are binary (i.e., $x_{i,m}(t)$). Therefore, **P3** is a nonlinear (due to non-linearity of Eq. 6.30) binary programming. Following the discussion in Section 6.1.3, and similar to problem **P1**, the problem **P3** is a NP-hard problem. However, relaxing the binary variable x to a real variable does not significantly help to derive a computation efficient solution to **P3**: (i) **P3** is a large-scale optimization problem; in particular, its number of variables compared to that of **P1** is increased by the order of number of servers at data centers, which is typically in the order of thousands, (ii) **P3** is a non-convex problem, since CoP is typically a super-linear function of the supplied temperature, $T_{sup,i}(t)$ (e.g., a quadratic function [110]). The later would suggest that if **P3** is solved using distributed optimization techniques (solutions to

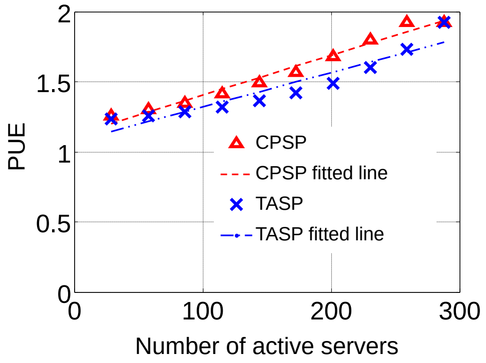


Figure 6.2: BlueCenter PUE when using TASP (i.e., TASP-LRH in Section 4.4) and CPSP.

solve large-scale optimization problems [77]), then there would be no guarantee on the convergence of the solution to a local optimal solution. The following section gives a heuristic solution to solve **P3** in a computation-efficient and online way.

6.3.1 Thermal-aware OnlineCC

In order to find a computation efficient solution to **P3**, we decompose the problem into two distinct optimization problems; thermal-aware active server set selection problem, and cost and carbon footprint optimization problem. The idea is to model the energy consumption of the data centers for a given number of required servers and a given active server set selection algorithm, and use that model in the cost and carbon footprint optimization problem. In this way, we remove the complexity of the active server set selection from **P3** (similar to **P1** which does not account for the active server set selection). We already devised a computation-efficient thermal-aware active server selection (i.e., TASP-LRH in Section 4.4) where for a given number of required active servers it finds the active server set to competitively minimize the sum of cooling and computing energy of the data center with respect to optimal solution. In particular, TASP-LRH uses a model similar to (6.33) to calculate the

Algorithm 6.3: Thermal-aware OnlineCC Algorithm

```
procedure INITIALIZATION( )
    Initialize the virtual queue $X$
    for each data center $i$ do
        find $\gamma_i$ and $\gamma_{0,1}$ (6.33) using TASP-LRH solution (Section 4.4)
    end for
end procedure
procedure ONDATACENTERLAYOUTCHANGE(()i)
    Update the heat recirculation model
    Update $\gamma_i$ and $\gamma_{0,1}$ (6.33) using TASP solution (Section 4.4)
end procedure
procedure ONSLOTTIMEOUT( )
    Predict the system parameters over slot $t$.
    Solve the following problem:

        Minimize: $cost_{OnlineCC} = V \sum_i g_i(t)\alpha_i(t) + X(t) \sum_i b_i(t).$        (6.32)

    Subject to: (6.1), and (6.33).
    Update the virtual queue $X$ using (6.7).
end procedure
```

data center total energy consumption. The computation complexity of TASP-LRH is $O(Y^3)$ (where Y denotes the number of servers in a data center), and its performance is analytically and experimentally evaluated against the optimal solution. We run this algorithm to model the energy consumption of BlueCenter (and its PUE) for a given number of required active servers. BlueCenter is an NSF funded small scale data center facility located in the ASU campus [57] which offers experimentation environment with innovative data center management schemes. It is an air-cooled data center with a layout similar to contemporary data centers. It has physical dimensions of 27.6” × 28” × 11.8”. There are total of 288 servers each of which consumes 300 W at peak utilization. The chiller supplies cold air with a flow rate of $5\,m^3/s$ from a single CRAC and follows a regular hot-cold aisle structure. Similar to the contemporary data centers, BlueCenter has a nonuniform heat distribution across servers such that the hot spot temperature varies depending on how many servers and which servers are active. The heat recirculation model of BlueCenter is derived and validated in [56]. We run some experiments to calculate PUE of BlueCenter for various input workload and number of required active servers using two algorithms: TASP-LRH and a reference non-thermal aware server selection algorithm namely Computing Power aware Server Provisioning (CPSP) (see Chapter 4). CPSP is oblivious to the cooling energy, it resizes the active servers to the incoming workload without considering the impact of the active server set on the cooling energy. Results as shown in Fig. 6.2 depicts that PUE varies depending on the active server set selection algorithm and the number of active servers. The results expectedly show that TASP yields lower PUE than CPSP, since it minimizes the sum of the cooling and the computing energy.

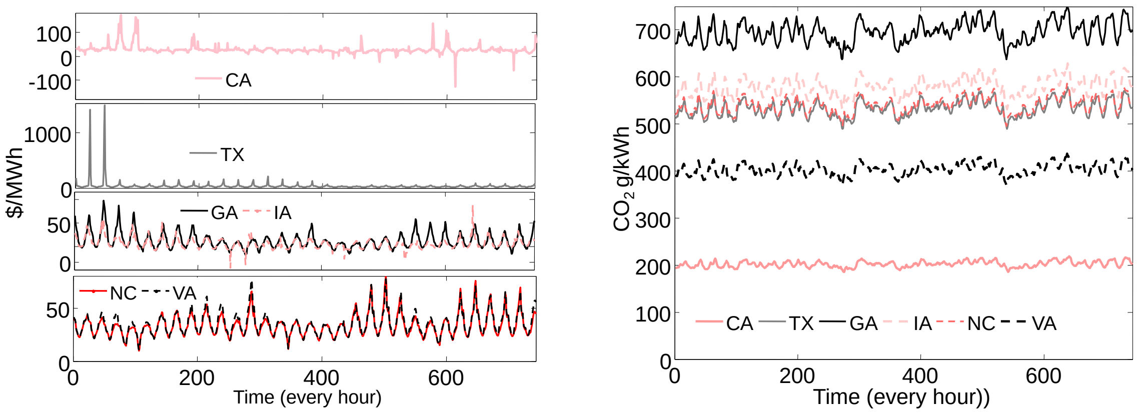
We use regression to model BlueCenter’s PUE as a linear function of number of active servers when using TASP (i.e., TASP-LRH) and CPSP. Both models incur around 7% mean absolute error and a R^2 of around 0.9. The models can be represented as $PUE(y){=}\gamma y{+}\gamma_0$ where TASP yields $\gamma{=}0.0025$, and $\gamma_0{=}1.07$, and CPSP yields $\gamma{=}0.0028$, and $\gamma_0{=}1.12$. Using this PUE model, p^{tot}, the total energy consumption of a data center becomes a nonlinear function of number of active servers as follows:

$$p_i^{tot}(t){\geq}p_i\gamma_i y_i^2(t){+}p_i\gamma_{0,i}y_i(t). \qquad (6.33)$$

We intensionally, use inequality "$\geq$" instead of equality in the above model. Incorprating the above model in **P3** instead of (6.30), we make a new convex optimization problem, namely **P4** where its scale (in terms of number of decision variables) is equal to that of **P1**. In this way, **P4**, similar to **P1** can be solved using exiting convex optimization solvers. Since both the electricity cost minimization (for positive electricity cost models) and the carbon footprint minimization favor low energy consumption, the solution of **P4** are still true when using the inequality for (6.33) instead of an equality. Further, for rare cases where electricity pricers are negative (negative prices are a price signal on the power wholesale market that occurs when a high power generation plant meets low demand. This is because some of power generation plants cannot be shut down and restarted in a quick and cost-efficient manner.), the outcomes as a result of using either inequality or equality in (6.33) are equivalent (as long as the upper bound of p^{tot} is appropriately set). This is because, the performance model (6.1) favors large number of servers.

Note that the problem **P4** which uses (6.33) as the data center power consumption model implies that the corresponding active server set selection algorithm to the parameters γ_i, and $\gamma_{0,i}$ runs locally at each data center as shown in Fig.2.9.

Therefore it is left to design an online solution similar to that of OnlineCC in order to solve **P4** in an online way. Lyapunov optimization, as used to design OnlineCC, does not make any assumption on the cost model of OnlineCC. This means that we can update the power consumption model of OnlineCC to (6.33) and use it to solve **P4** in an online way. Note, in this way, except, Lemma 1 and Lemma 2, and the parts of the Theorems that use these Lemmata's results, all other results hold true. Particularly, it is guaranteed that OnlineCC for both power consumption models satisfies the carbon cap and achieves a cost with optimality distance of $\frac{B}{V}$, when it runs for sufficiently large S.



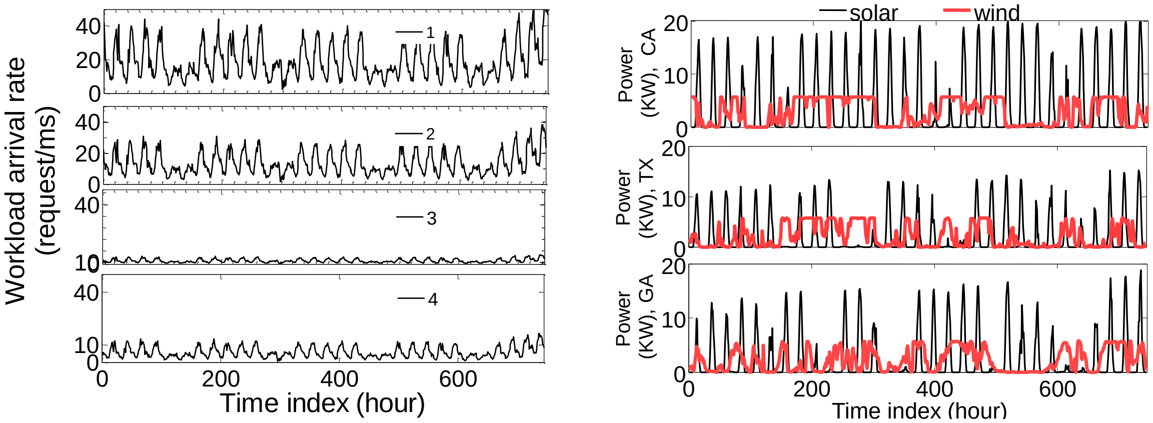
(a) (b)

Figure 6.3: Hourly Traces for August: (a) Electricity Price, and (b) Carbon Emission.

The aforementioned online thermal aware cost and carbon footprint capping is summarized in three procedures as given in Alg. 6.3, namely Thermal-aware OnlineCC. The "Initialization" procedure similar to OnlineCC solution initializes the virtual queue X. It also runs the data centers active server selectin algorithm (e.g., TASP-LRH) to find each data centers' energy consumption model when deploying the corresponding active server seltion algorithm (i.e., finding γ_i and $\gamma_{0,i}$ of (6.33)). The parameters γ_i and $\gamma_{0,i}$ are updated whenever there is any change in the data center physical layout and consequently in its heat recirulation model (see the procedure "OnDataCenterLayoutChange"), whcih is an infrequent event. Finaly, the procedure "OnSlotTimeOut" at each time slots runs an algorithm similar to that of OnlineCC. The only difference is that Thermal-aware OnlinECC usee (6.33) in the procedure instead of (6.3).

6.4 Evaluation

We simulate a cloud consisting of six data centers located at CA, TX, GA, IA, NC and VA, most of which correspond to Google's data centers' locations. The



(a) (b)

Figure 6.4: Hourly Traces for August: (a) Front-ends' Workload, and (b) Solar and Wind Power.

data centers are assumed to be homogeneous in terms of power consumption and computing characteristics, such that all the electricity cost savings and the carbon footprint reduction only comes from spatio-temporal variation of the electricity cost and the carbon footprint. Servers in each of the data centers are assumed to consume 300 W at peak utilization and data centers are considered to have an ideal PUE of 1 (Section 6.4.5 evaluates the solutions according to PUE model of a real data center). We set the slot length to one hour, S to one month, and use realistic hourly traces of the electricity price, carbon intensity, renewable power, from data centers' locations. To ensure data consistency, all traces are chosen from the month of July and August, since the workload traces were available for only these two months. Particularly, we obtain the hourly Locational Marginal Prices (LMP) of the aforementioned locations in August 2012, from their corresponding RTO/ISO website (see Figure 6.3(a)). Further, we estimate the hourly carbon emission intensity of our six data centers by calculating the weighted average of carbon intensities of fuels in Table 2.1 where the weights are taken from the available hourly electricity fuel mix of data center locations in August 2012 (see Figure 6.3(b)).

We consider four front-ends, corresponding to four time-zones in the U.S., and use two months (July and August) of NASA workload Internet trace [1]. The workload of each front-end is scaled proportionally to the number of Internet users and shifted according to the time zone for each front-end in the corresponding area, as shown in Fig. 6.4(a). Each data center has 280 servers, and the intensity of the workload is such that at peak, 70% of servers in the entire cloud are required to be activated. We assume that a data center can receive workload from any of the front-ends. Note, in practice there might be a large number of front-ends. However, under the assumption that front-ends can send their requests to all data centers, high number of front-ends does not affect the results as long as the study accounts for at least one front-end in each time zone. This is because the aggregated workload of all front-ends is affected by the time-variation of workload at different time-zones.

To capture the availability of wind and solar energy, we use the traces [2] for three sites located in the data center locations of CA, TX and GA. We use the wind speed and the rated power to calculate the wind power, and Global Horizontal Irradiance (GHI) and the ambient temperature to calculate the solar power using models described in [125]. The renewable infrastructure capacity (i.e., PV cells and wind turbines) are considered to be equal for all three data centers (see Fig. 6.4(b)).

Prediction results: We use one month of training data (July traces) and build weekly and daily Seasonal Auto Regressive Integrated and Moving Average (SARIMA) prediction model to predict workload and solar energy, respectively. Further, we use ARMA prediction model for wind energy. The lag one (one hour-ahead) prediction error is 14%, 12% and 18% for workload, solar and wind energy respectively. The error goes up to 20%, 18% and 52% for 24 lag (24 hour ahead) prediction of workload, solar and wind energy respectively. Since wind and solar traces contain some values of zero or nearly zero, we report 95 percentile mean absolute percentage error of these two traces (e.g., lag one mean absolute error of the solar energy is 25%).

Experiments Performed The following algorithms are used to evaluate OnlineCC:

- **MinCost** (reference algorithm): performs workload distribution in the cloud to first minimize the electricity cost and then the carbon footprint.
- **MinCarbon** (reference algorithm): performs workload management to first minimize the carbon footprint and then the electricity cost in the cloud.
- **Optimal**: offline optimal solution to **P1**.
- **OnlineCC, PP**, and **OnlineCC, P**: Alg. 6.1 with Perfect Prediction (PP), and Predicted data (P).
- **OnlineCC-T, PP**, and **OnlineCC-T, P**: Alg. 6.2 with Perfect Prediction (PP), and Predicted data (P).
- **OnlineH, PP**, and **OnlineH, P** (similar to heuristic solution of [69]): OnlineH divides the given carbon cap for a month (i.e., Ψ) into chunks per day (i.e., T=24 hours/slots), where data center parameters can be predicted, and solves the problem **P1** over T. OnlineH satisfies the carbon cap in a best-effort manner, since the feasible carbon cap for a T-slot depends on the workload intensity, the availability of renewable energy and the carbon intensity on that T-slot. Similar to OnlineCC, we implemented OnlineH PP, and OnlineH P. Finally, we use the Receding Horizon Control (RHC) technique to minimize the impact of the prediction error on the performance of OnlineH. Accordingly, the solution at time slot t is calculated by solving optimization problem of **P1** over the time frame T=24, given the solution at time $t-1$, and the predicted information

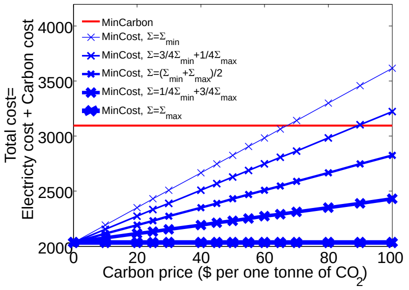


Figure 6.5: The Monetary Comparison of MinCost and MinCarbon for a Given Carbon Cap (Ψ) and a Given Carbon Price.

over T. Using RHC the impact of prediction error on OnlineH performance degradation becomes minimal.

Note, when we use online algorithms without any postfix i.e. P or PP, it implies that the algorithms are run with perfect prediction (PP). MinCost and MinCarbon can be viewed as representative of the previous schemes which solely focus on either cost minimization (e.g., [12, 96, 97]) or carbon footprint minimization (e.g., an algorithm in [46]). The carbon footprint target of the cloud, i.e., Ψ, is clearly a value between the carbon emissions achieved by MinCost and MinCarbon solutions for feasibility assumption. We use MATLAB and GNU Linear Programming Kit (GLPK) to solve all of the algorithms. In the experiments we justify Lemma 1, Theorem 1, 2 and 3, and the solution of Section 6.2.1. Further, we study the performance of OnlineCC versus OnlineH under various parameters i.e., V value (§ 6.4.2), and carbon cap (Ψ) (§ 6.4.3). We also study the performance of OnlineCC-T for 1≤T≤24 for various availability of the renewable energy and prediction error (§ 6.4.4). Finally, we evaluate OnlineCC when using a real data center (BlueCenter) PUE model (Section 6.4.5).

6.4.1 Electricity Cost and Carbon Reduction Tradeoff for Data Centers

Given existing carbon capping policies and carbon pricing, this experiment compares monetary values of electricity cost reduction versus carbon reduction for data centers. The purpose is to study whether a MinCost or a MinCarbon strategy is sufficient for the cloud to manage its cost and carbon footprints. The existing carbon capping policies typically define the carbon price as the amount that must be paid for the right to emit one tonne of CO_2 into the atmosphere which is given in the form of a carbon tax or a requirement to purchase permits to emit (e.g., cap and trade). Of particular interest is carbon pricing in USA where our simulated cloud is located. However, there is no nationwide carbon tax leveled in USA. For this reason, we evaluate monetary values of carbon reduction for a range of carbon prices. We consider that the cloud has to purchase carbon credits per excess carbon footprints from the cap. We compare the cloud total cost as a result of applying MinCost and MinCarbon, where the total cost is the sum of the electricity cost and the carbon cost. Fig. 6.5 shows the total cost for a given carbon cap versus carbon price for a tonne of carbon ranging from \$0 to \$100 which is larger than the existing carbon prices. The carbon price in Australia, for instance, is set to \$24.15 for the 2013-2014 financial year [9]. Also the carbon price in USA is estimated around \$37 [7]. The carbon price is typically estimated from the monetary value of the future damage from climate change associated with an increase in CO_2 emissions in a given year, referred to as the social cost of carbon (SCC). Estimates of the SCC are highly uncertain, and researchers have produced a wide range of values [10]. SCC for USA is estimated \$21 for 2010 [10] and \$37 for 2013-2014 [7]per one tonne of carbon.

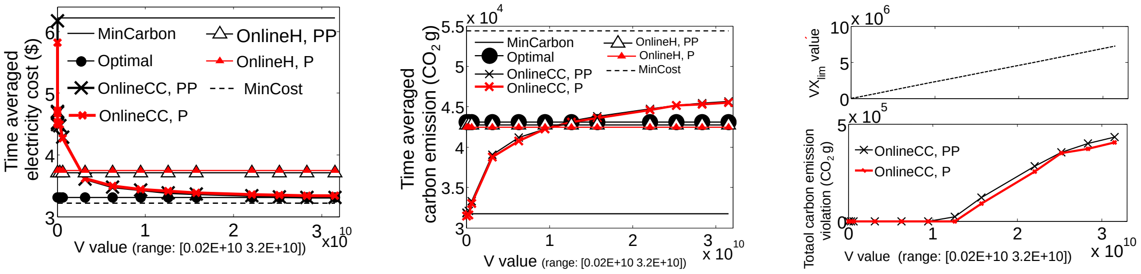
Therefore, the carbon price of \$50-\$100 is very large which is used to evaluate the performance of MinCost in the worst-case. As shown in Fig. 6.5, we use the total cost of MinCarbon as the reference which yields the minimum feasible cap (Ψ_{min}). The total cost of MinCarbon, therefore, is equal to its electricity cost, since it yields zero excess carbon from the cap. MinCost, however, incurs carbon cost depending on the cap, since it yields larger carbon footprint than that of MinCarbon (i.e., Ψ_{max}). The results indicate that for the carbon price of \$0-\$60 the total cost of MinCost is always (i.e., independent of the cap) significantly smaller than that of MinCarbon, which means that the electricity cost saving of MinCost significantly outweighs the monetary value of the carbon reduction. Also when the cap increases form the minimum cap (Ψ_{min}), the total cost of MinCost is smaller than that of MinCarbon even for very small carbon cap value ($\Psi = \Psi_{min}$). The results suggest that data centers can significantly save cost by leveraging the spatio-temporal variation of the electricity even when it comes with the expense of violating the carbon cap. The reason is that the existing carbon prices and the spatio- and temporal variation of electricity prices is such that the monetary value of carbon reduction is significantly smaller than the monetary value of electricity cost reduction. However, cloud operators many times prefer to operate under the cap for their credential and for their responsibility to the environment which necessitates a workload management strategy to minimize the cost while maintaining the carbon cap, i.e., a solution such as OnlineCC.

6.4.2 Performance of the Solutions versus Lyapunov Control Parameter

First, we run an experiment without on-site renewable energy for data centers. In order to run Optimal solution we run this experiment for S=168 (i.e., one week of data) since running the Optimal solution for larger number of slots takes huge time (due to the curse of dimensionality problem). We set Ψ to 7.2 CO_2 Mg (ψ=43 CO_2 Kg), the mean total carbon footprint of MinCost and MinCarbon, and vary V starting from V=V_{min}=0.02×10^{10}, where X_{lim} equals to 0.00023 in the data set (see the first solution in Section 6.2.1). The second solution of Section 6.2.1 suggests a V value equal to $\sqrt{\frac{B}{X_{lim}}}$, where $B = \frac{1}{2}(b_{max}^2 + \psi^2)$. Similar to the first solution, this solution also gives an estimation of V and depends on the tightness of B and X_{lim}. Considering that in our data set, b_{max}=0.13 CO_2 Mg, this solution suggests a V value around 0.0006×10^{10} that is lower than V_{min}. Therefore, we proceed investigating V value using the first solution. In practice, either of the solutions of Section 6.2.1 can be used as a start point to investigate the right V value.

The results, shown in Figs. 6.6(a) and (b), being consistent with Theorem 6.2.3, clearly demonstrate the electricity cost and carbon reduction tradeoff which is managed by OnlineCC V parameter. Further, interestingly the prediction error has very negligible impact on the performance of the both online solutions i.e., OnlineCC and OnlineH. This is due to the relatively low prediction error of workload for both lag one and 24 hours. Comparing OnlineCC with OnlineH from Fig. 6.6(a) and (b), it can be seen that for $V{>}1{\times}10^{10}$, OnlineCC achieves a lower cost than OnlineH while satisfying the carbon footprint cap for $V{<}2{\times}10^{10}$. In particular, under the condition that both OnlineH and OnlineCC satisfy the carbon cap, OnlineCC achieves cost saving up to 18% more than OnlineH depending on the value of V. This indicates that OnlineCC surpasses OnlineH when V is appropriately adjusted. OnlineCC violates carbon cap for $V{>}2{\times}10^{10}$.

However, as shown in Fig. 6.6(c) and being in agreement with Theorem 6.2.3, the carbon violation is much lower than the proven upper-bound. Particularly, as shown in the top graph of Fig. 6.6(c), for large V where OnlineCC violates carbon cap, VX_{lim} itself is larger than the total carbon cap violation. According to Fig. 6.6(c), the per slot (i.e., time averaged) carbon capping violation for $V{=}2{\times}10^{10}$ is around 50% of ψ, while the actual time averaged carbon capping violation for this V value is around 4.6% of ψ (see Fig. 6.6(b)). Note, the bound of Theorem 6.2.3 becomes tighter



(a)

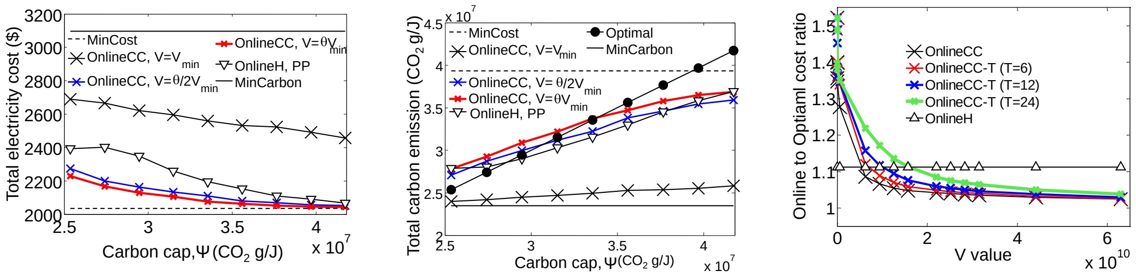
(b)

(c)

Figure 6.6: OnlineCC and OnlineH Performance versus Optimal with and without Prediction Error: (a) Average Cost, (b) Average Carbon, (c) Total Carbon Violation form the Cap, i.e., Ψ=7.25 CO_2 Mg. It can be seen that the violation in the given range of V value is up to 6% of the carbon cap target.

with increasing the budgeting period length. This can be clearly seen from Fig. 6.8(a) which demonstrates the carbon footprint violation of OnlineCC over the budgeting period of one month (S=744) as opposed to the results of Fig. 6.6(c) which are given for the budgeting period of one week $S = 168$. The figure shows that the OnlineCC average carbon capping violation for V=2×10^{10} is still around 5%, while the value of VX_{lim} divided by the one month budgeting period i.e., S=744 is around 11%. The OnlineCC worst case carbon violation also depends on θ in addition to the value of VX_{lim} (see $R1$ in (6.11)), however this result shows how the bound becomes tighter with increasing S.

Figs. 6.6(a) and (b) show that there exists V for which OnlineCC achieves near one cost competitive ratio with respect to Optimal, while maintaining the carbon cap. In agreement with discussion in Section 6.2.1, the same figures show that choosing V=V_{min}=0.02×10^{10}, OnlineCC yields an output almost equal to that of MinCarbon. Considering the daily variation of the workload, the daily variation of the electricity price, and the carbon intensity as demonstrated in Fig. 6.3(b), the parameter θ can be



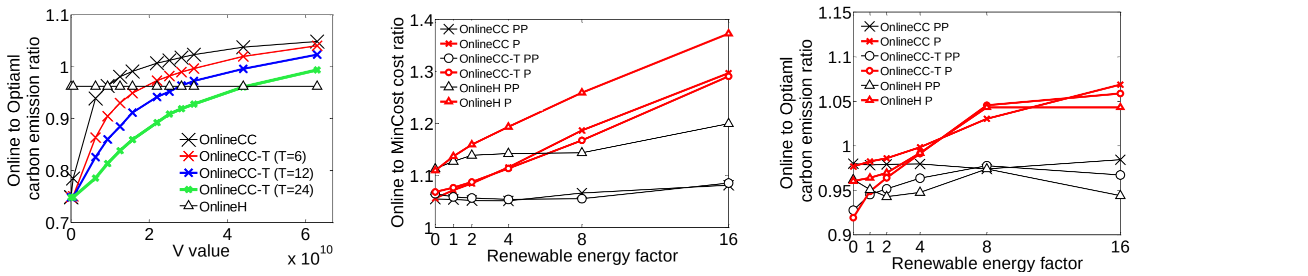
(a) (b) (c)

Figure 6.7: OnlineCC and OnlineH Performance versus the Magnitude of Carbon Cap (Ψ): (a) Total Electricity Cost, (b) Total Carbon Emission, and (c) OnlineCC-T to Optimal Carbon Footprint Ratio for Different Values of V and T.

overestimated as the number of slots for a day, i.e., 24. According to Section 6.2.1, for a sufficiently tight X_{lim} value, OnlineCC for $V{=}\frac{\theta}{2}V_{min}$ (i.e., $V{=}0.2 \times 10^{10}$) achieves a performance near to Optimal. However, in our dataset X_{lim} is not very tight, e.g., the value of $(\alpha_{max}{-}\alpha_{min})$ is around 3 times greater than the average electricity price differences in data centers (see Fig. 6.3 (a) and (b)). Therefore, OnlineCC for a V value around $0.6{\times}10^{10}$ achieves near Optimal solution performance. In general, due to the variability of input data parameters, the task of deciding the tightness of X_{lim} becomes tedious. This means that we need to run number of trials to find a right value for V. However, the heuristic solution of Section 6.2.1 significantly reduces its search space.

6.4.3 Performance of the Solutions versus the Carbon Cap

In the rest of experiments including this experiment we run the experiments for the entire one month of traces. We compare the carbon footprint of the online solutions against Optimal. Also motivated by the result of the previous section, which suggests Optimal solution achieves a cost very close to MinCost, we compare



(a) (b) (c)

Figure 6.8: OnlineCC-T to Optimal Carbon Footprint Ratio for Different Values of V and T, (b) OnlineCC and OnlineCC-T (T=24) to MinCost Cost Ratio versus Prediction Error and Different Availability of Renewable Energy, and (c) OnlineCC and OnlineCC-T (T=24) to Optimal Carbon Footprint Ratio versus Prediction Error and Different Availability of Renewable Energy.

the electricity cost of online solutions against MinCost.

We vary the carbon cap Ψ from $\Psi{=}\Psi_{min}$, up to Ψ_{max}, where Ψ_{min} and Ψ_{max} denote the minimum and maximum total carbon footprint of the cloud achieved by MinCarbon and MinCost, respectively. Further, for a given Ψ, we run OnlineCC for three values of V: $V{=}V_{min}$, $V{=}\frac{\theta}{2}V_{min}$, and $V{=}\theta V_{min}$, where $\theta{\simeq}24$ and V_{min} is calculated for the given carbon cap. The results, shown in Fig. 6.7(a) and (b) indicate that when Ψ is tight, i.e., Ψ is close to Ψ_{min}, OnlineH and OnlineCC for large V values slightly violate the carbon cap (see Fig. 6.7(b)). However, in the same situation, OnlineCC for $V{=}V_{min}$ satisfies the cap (see Fig. 6.7(b)). Interestingly, when Ψ is tight, OnlineCC yields a carbon footprint lower (for $V{=}\frac{\theta}{2}V_{min}$) or very close to that of OnlineH, yet achieves 10% lesser cost than that of OnlineH (see Fig. 6.7(a)). Note OnlineCC violates cap for high V value, however, the violation never exceeds the upper bound defined in Theorem 6.2.3 (see Fig. 6.7(b)).

When the carbon cap is loose, i.e., it is close to Ψ_{max}, the offline solution is comparable with offline solution with T ahead information (Section 6.2.3). It can be seen in Fig. 6.7(a) and (b) that for Ψ close to Ψ_{max}, OnlineH achieves a performance very close to that of Optimal, suggesting that an Offline solution with a T value slightly greater than 24 slots achieves a similar performance to that of offline solution with entire future information. It can be seen that for loose carbon cap still OnlineCC with appropriate V value surpasses that of OnlineH in terms of cost saving, suggesting the competitiveness of OnlineCC against Optimal in agreement with Theorem 6.2.5.

6.4.4 Performance of the Solutions versus Parameters' Prediction Error

First, we run OnlineCC-T for different values of T and calculate the cost ratio and the carbon footprint ratio of the online solutions over MinCost and Optimal, respectively (note, OnlineCC corresponds to OnlineCC-T for T=1). The results as shown in Fig. 6.7(c) and Fig. 6.8(b) indicate that OnlineCC for every T value achieves near Optimal performance albeit for different values of V (as explained in Section 6.2.4). Results also show that OnlineCC-T (T=24) has higher performance over OnlineCC-T ($T < 24$) since it achieves near one cost ratio, while reducing carbon footprint more than the other variants of OnlineCC-T solutions.

Next, we run some experiments where we use on-site renewable energy for three data centers at locations CA, TX and GA. We run the experiments for various renewable energy availability by scaling the renewable traces with factors of [0 1 2 4 8 16]. As a result of this scaling, the renewable energy forms 0% up to 70% of the total energy consumption by the cloud when using Optimal algorithm. We run OnlineCC-T PP, OnlineCC-T P for T=1 and T=24. Further we show the results for V values that OnlineCC-T, PP maintains the carbon cap. As shown in Figs. 6.8(b) and (c), prediction error is a downgrading factor to the performance of all online solutions. In particular, for the situation where all online solutions with perfect prediction (PP) maintain the cap and achieve a cost ratio near to 1, the solutions with predicted data (P) violates the cap and achieve a large cost ratio with respect to Optimal. The impact of prediction error worsens with increasing the available renewable energy. The results suggest to use a more efficient prediction technique to predict the availability of the renewables with low prediction error.

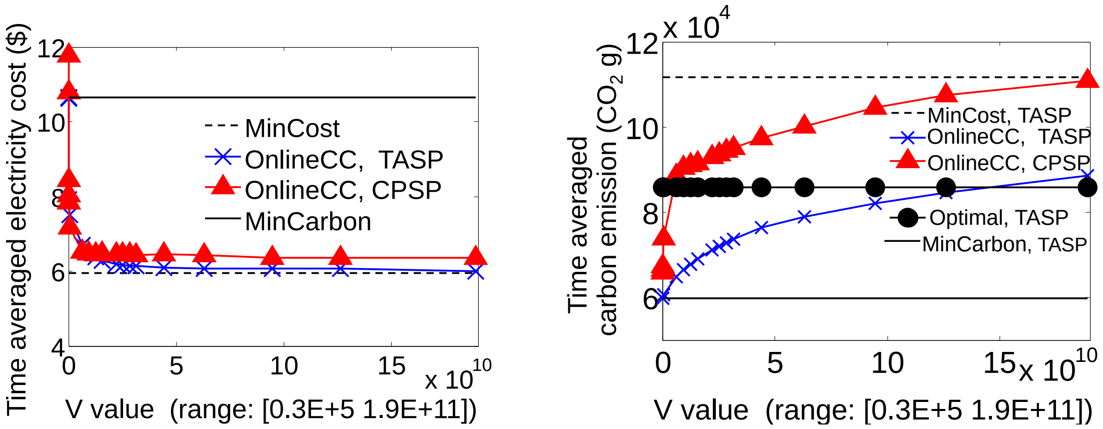
From Figs. 6.8(b) and (c), it can also be seen that both OnlineCC, PP and OnlineCC-T, PP achieves 10-20% more cost saving than OnlineH, PP while all the solutions satisfy the carbon cap. Interestingly, OnlineCC-T, P for both T=1 and T=24 achieves lower cost (around 5-10%) and lower (or very close) carbon footprint than OnlineH, P.

Finally, the results of Figs. 6.8(b) and (c) show that OnlineCC-T, P and OnlineCC-T, PP for T=24 achieves slightly lower carbon footprint than OnlineCC, while achieving the same cost saving. Interestingly, the impact of prediction error on the performance of OnlineCC-T T=24 and OnlineCC is very similar even when the availability of the renewable energy is high. This is because of the high renewable energy prediction error even for lag one.

6.4.5 Performance of Thermal-aware OnlineCC

This section evaluates Thermal-aware OnlineCC when using BlueCenter's energy consumption model. We use IBM ILOG CPLEX Studio version 12.5, the MATLAB connector of CPLEX and the function "`cplexqcp`" to run MinCost, MinCarbon and OnlineCC. All algorithms are run when using (6.33) as the energy consumption of all six data centers, where γ_i and $\gamma_{0,i}$ are set according to BlueCenter's PUE models (one round all data centers use TASP PUE model and another round all data centers use CPSP PUE model).

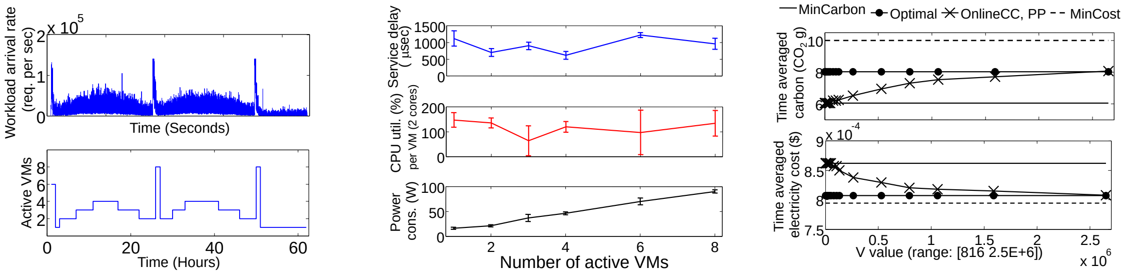
Results, as shown in Fig 6.9, in agreement with Theorem 6.2.3, indicates that



(a)

(b)

Figure 6.9: OnlineCC Performance when using BlueCenter TASP and CPSP PUE Models: (a) Time Averages Electricity Cost versus V, and (b) Time Average Carbon Footprint versus V.



(a)

(b)

(c)

Figure 6.10: Experiment Traces and Results: (a) Microsoft Hotmail Traces and Number of Active VMs in the Experiment, (b) Power and Performance Measurements, and (c) OnlineCC Performance over Experimental Data.

Thermal-aware OnlineCC has a similar performance trend to Section 6.4.2 where a linear energy consumption model is used. The results also show that for a V value that both non-thermal-aware OnlineCC (i.e., OnlineCC CPSP) and Thermal-aware OnlineCC (i.e.e, Online TASP) satisfy the carbon cap, OnlineCC TASP yields 9% lower average cost than OnlineCC CPSP. This is because TASP solution decrease the total energy consumption of data centers by choosing servers that have less contribution in the data center heat recirculation (see Chapter 4).

6.5 Experimental Evaluation of OnlineCC

We implemented a small-scale experiment using real systems in order to validate the performance model (Eq. 6.1), as well as to validate the simulation results. We use two Intel(R) W2600 Pedestal server, 2 × Intel Quad 1.8 GHZ CPU (32 cores), and 32 GB RAM as the test servers. KVM hypervisor is used to create a virtualized environment with four virtual machines (VMs) in each system, and each VM is assigned two V-CPU and 1G of RAM. Each server emulates a data center and each VM emulates a physical server in our model. In other words, the control parameter in our experiment is the number of active VMs. Ubuntu Linux server 12.04 LTS 64-bit is installed as the VM operating system. One line of the future work is to extend the experimental study using BlueCenter Infrastructure[57], which offers small scale data center for experiments.

We developed a server-client program in C generating TCP-based requests on image files with size distribution following Pareto distribution and ranging from 0.3KB to 90KB, in accordance to a study on the file size distribution of web image content [101]. The server-side program performs image transcoding for each file, yielding in CPU-intensive operations. Each VM hosts the server-side program. The client-side programs run from a separate machine, where the workload arrival rate is taken from the two and half days of Microsoft Hotmail traces [113] (see Fig. 6.10 (a)) (the original Microsoft trace is a little bit modified by removing large spikes for the sake of do-ability in our testbed). The original trace gives the normalized average workload arrival (between 0 and 1). We first perform several trial runs with different workload arrival rates to model the servers' service rate (μ). Accordingly, we adjust the service rate μ to 6700 request per second and the reference average service delay d'^{ref} (including service time) to 0.0012 ms. Then we scale the traces, so that using (6.1) the total number of servers needed is eight with zero slack (see Fig. 6.10 (a)). With zero slack (y^{slack}=0) we expect our model to meet the reference average service delay d'^{ref}. Next, we choose some slots (in hour) of the trace, which represent all of the workload variation range (number of needed VMs vary from 1 to 8). We run 10 minutes of the actual Hotmail traces in all those slots (in second resolution) with appropriate number of VMs and log the servers power consumption, VMs' CPU utilization, and the turn around time of the requests. Considering that the servers and the workload generators are connected using an internal network, the turn around time of the requests are assumed to emulate data center service delay (including queuing delay and service time). Results in Fig. 6.10(b) indicates that the average turn around time satisfy the reference average service delay. Note that we subtract a certain amount of power from the measurements such that the total power consumption is roughly in proportion to the number of active VMs (i.e., enabling a new VM has a similar effect on power consumption as turning on a new physical server) as given in Fig. 6.10(b). For parameters that cannot be captured by our system (e.g., electricity price (α_i), carbon intensity (ε_i)), we use first two and half days of real-world trace data of DC1 and DC2 as presented in the simulation section.

Fig. 6.10(d) shows the results of running OnlineCC on the entire Hormail traces using the measurement data. We adjust Ψ=480 CO_2 g (i.e., ψ=8 for 60 slots), the mean carbon footprint achieved by MinCost and MinCarbon algorithms. We find that X_{lim}=0.01, and consequently V_{min}=816. The results are consistent with Section 6.2.1 and the simulation study where OnlineCC performs close to MinCarbon algorithm for V=V_{min}. Having daily variation in the system parameters we have θ=24 hours. Since, X_{lim} is not as tight (see Section 6.2.1) OnlineCC achieves near Optimal performance for V=1×10^6 (see Fig. 6.10(d)).

6.6 Summary

We developed online solutions, OnlineCC, OnlineCC-T, and Thermal-aware OnlineCC, which help a cloud, consisting a set of geo-distributed data centers, to move toward carbon neutrality in a cost efficient way. OnlineCC, balances workload across data centers of a cloud to minimize the electricity bill while satisfying the cloud's carbon footprint cap. Carbon neutrality can be achieved by purchasing additional carbon credit for the remaining offset from carbon cap. OnlineCC-T, further extends OnlineCC to account for the case where data center parameters exhibit non-stationary characteristics. Thermal-aware OnlineCC extends OnlineCC to account for the impact of global server and workload management on the temperature distribution and on the cooling energy of data centers. All the online solutions, building upon Lyapunov optimization, may violate the carbon cap up to a proven bound depending on the control parameter, namely V. We extended the Lyapunov optimization results such that the worst case carbon cap violation of OnlineCC and OnlineCC-T can be calculated without the need to solve the optimal offline solution. The performance of the online solutions heavily depends on the control parameter, V, where we devised some heuristics to adjust its value. We evaluated the performance of the online solutions, analytically and experimentally in a simulation study, against the optimal solution with entire information, an offline solution with limited lookahead information, and a prediction-based scheme. Further, Thermal-aware OnlineCC is evaluated using an actual data center energy consumption traces. We also evaluated OnlineCC in a small scale experiment. The studies show positive results.

Chapter 7

PEAK POWER DEMAND SHAVING AND CARBON FOOTPRINT CAPPING AT GLOBAL DATA CENTER LEVEL

Global workload management, exploring temporal and spatial diversities in data centers, has been shown to be promising in optimizing data centers' operational cost and carbon footprint capping. Nevertheless, in this chapter we consider some key aspects that have not been explored in this approach. First, the data centers cost model are often simplified to only account for energy cost. Yet, data centers spend significant cost in provisioning their power infrastructure and the peak power draw from utilities, regardless of the energy actually consumed. In this chapter, we extend OnlineCC model to account for peak power shaving and energy buffering. Second, global workload management favors an offline solution, due to the time coupling to manage energy storage devices, carbon capping and peak power cost minimization. Yet the online algorithms are often designed to address each of the aforementioned coupling factors separately, disregarding their management implications on each other and more importantly the practical considerations (e.g., sensitivity to the prediction error). In this chapter we design OnCMCCLyp which extends OnlinCC (Chapter 6) to utilize the predicted information within T future slots. OnCMCCLyp uses T future slots information to smoothen the power draw and consequently decrease the peak power cost, and T-slot Lyapunov technique to dynamically mange the carbon capping requirements over the entire budgeting period in an online way. We implement OnCMCCLyp using stochastic programming approach to remove/alleviate the harmful impact of the prediction error (particularly on the peak power shaving) without significantly affecting the size of the problem. We, further, adapt the Alternating Direction Method of Multipliers to design the distributed algorithms with linearly convergence for OnCMCCLyp in accordance to the data center confidentiality and the solution scalability requirements in practice. We perform a comprehensive trace-based study using realistic traces to complement the analysis. In particular, the results, in agreement with the analysis, show that OnCMCCLyp achieves near optimal solution performance, when T is sufficiently large and that the information over T is accurately available. Prediction error, however, downgrade the performance of OnCMCCLyp by increasing the cost (electricity cost \$/J, and peak power cost \$/W) up to 45% compared to the offline optimal solution. Our stochastic programming based solution is shown to remove up to 66% of such a harmful impact of the prediction error. We also show that the distributed implementation of OnCMCCLyp converges quickly (around tens of steps).

In the rest of the chapter we first frame the global workload management problem as a linear programming (Section 7.1) which extends the models of OnlineCC (Chapter 6)to include peak power shaving and energy buffering. Next, We develop Online Cost minimization and Carbon Capping (OnCMCC) solution, which leverages predictability of data center parameters (e.g., workload) over T to solve the problem in an online way. Next, we design OnCMCCLyp which extends OnCMCC for using T-slot Lyapunov optimization technique to jointly minimize the cost (electricity cost \$/J, and peak power cost \$/W) and the carbon footprint across data centers (Section 7.2.2). T slots future information is used to smooth the peak power draw, while data center carbon footprint dynamics is stabilized through Lyapunov optimization (Theorem 7.2.1). We also introduce and design a stochastic programming approach to model and solve the global workload management solutions (OnCMCC and OnCMCCLyp) in the presence of the parameters' prediction error (Section 7.3). Then, we design a 2-block ADMM based algorithm to solve OnCMCC and OnCMCCLyp in

Table 7.1: Summary of Variables, Problems and Solutions of Peak Power Minimization

Sym.	**Definition**	**Sym.**	**Definition**
p_0	stipulated power	S'	no. of slots for peak power billing period
dschrg	ESD discharge rate	D	ESD max discharge rate
chrg	ESD charge rate	C	ESD max charging rate
E	ESD capacity	η	ESD energy inefficiency
ϕ	ESD cost per charge/discharge		

Symbol and formula	**Definition**
$e_i(t+1) = e_i(t) + \eta \text{chrg}_i(t) + \frac{1}{\eta}\text{dschrg}_i(t)$	ESD energy level
P5	global workload management optimization problem
P6	stochastic counterpart of **P5**
OnCMCC	predictive based online solution
OnCMCCLyp	$T-$slot Lyapunov based solution

Sym.	**Definition**	**Sym.**	**Definition**
W	set of stochastic scenarios	ρ	penalty term in Alg. 7.3
w	a scenario in W	γ	Lagrangian multiplier in Alg.,7.3
ζ	prediction error rand. var.	a	auxiliary variable in Alg.7.3

a distributed way both of which are linearly convergent (Section 7.4). The distributed algorithm of OnCMCCLyp is shown to be more efficient than that of OnCMCC in terms of scalability and confidentiality. Finally, we perform a real-world trace based study to complement our analysis (Section 7.5).

7.1 Problem Formulation

We introduce our model for data center energy cost and peak power cost in this section. It builds on the models of OnlineCC (see Chapter 6), which is in turn related to the system model described in Chapter 2. The key change we make to OnlineCC is to incorporate charges for the peak power draw form utility, and energy buffering into the optimization framework. This is a simple modeling change, but one that creates significant algorithmic challenges (see Sections 7.2, 7.3 and 7.4 for more details). In the following we describe cloud power supply and demand model when considering energy storage into consideration, with the notations specific to this chapter in Table 7.1 (see Tables 2.2 and 6.1 for other variables).

7.1.1 Power Supply and Demand Modeling

In order to smoothen the grid power draw and remove/reduce the peak power cost, the related work propose to utilize the existing batteries in data centers. The batteries are primarily deployed to power data centers for a duration which it takes the diesel generators to get activated during utility outages. The diesel generators' start-up time is about a few seconds, while existing UPS typically can power the data center for about 10-15 minutes. To model energy storage, we denote the energy storage level at time t by $e_i(t)$, and the charge/discharge energy during time slot t by $\mathrm{chrg}_i(t)$ and $\mathrm{dschrg}_i(t)$, respectively. There is limit on the maximum charging and discharging rate denoted by C and D, respectively. An ESD has limited capacity, further it is associated with a cycle-life i.e., the number of charging/discharging cycles that can be accomplished during the lifetime of the device for a given depth of discharge. Furthermore, depending on the ESD functionality in a data center some of its capacity is reserved for using during the power outages. Therefore, we denote E the capacity of the ESD which can be used to manage energy cost and renewable energy utilization, without affecting the data center availability and without violating the given depth of discharge. We assume that the efficiencies of ESD charging and discharging are the same, denoted by $\eta \in [0, 1]$, e.g., $\eta = 0.8$ means that only 80% of the charged or discharged energy is useful when charging or discharging. Energy level of an ESD over time satisfies the following:

$$\begin{aligned}
&\forall i, t: \quad e_i(t+1) = e_i(t) + \eta \mathrm{chrg}_i(t) + \tfrac{1}{\eta} \mathrm{dschrg}_i(t) [\text{ESD energy level}], \\
&\forall i, t: \quad 0 \leq e_i(t+1) \leq E, 0 \leq e_i(0) \leq E_i, \\
&\forall i, t: \quad 0 \leq \mathrm{chrg}_i(t) \leq C, 0 \leq \mathrm{dschrg}_i(t) \leq D.
\end{aligned} \tag{7.1}$$

Energy storage devices have some other physical limitations such as self discharge rate, which are ignored for notation brevity. Finally, in any slot, one can either recharge or discharge the battery or do neither, but not both. This means that for all t and i we have:

$$\forall i, t: \quad \mathrm{chrg}_i(t) \mathrm{dschrg}_i(t) = 0. \tag{7.2}$$

Today's data centers are increasingly have some form of local renewable energy available, with solar and wind being the most popular ones. The energy availability of thees sources is closely tied to the external conditions (e.g., wind speed and temperature). Their fluctuation and variability introduce a significant challenge for data centers management. We assume data centers also gets their power partially from the available on-site energy denoted by $r_i(t) \leq R_i$. We study the impact of their prediction on data center management. For every data center i and all time t the energy demand and supply should be balanced as follows:

$$\begin{aligned}
&\forall i, t: \qquad g_i(t) + r_i(t) + \mathrm{dschrg}_i(t) = p_i^{tot}(t) + \mathrm{chrg}_i(t), \\
&g_i(t) \geq 0.
\end{aligned} \tag{7.3}$$

7.1.2 *Offline Optimization Framework*

Cost model in this chapter extends that of OnlineCC to account for peak power cost (see § 2.7.2) and the operation cost of energy storage devices. Reducing peak power, allowing under provisioning of power infrastructure, indirectly helps to reduce power capital expenditure. The peak power cost is usually calculated per the peak excess power draw from stipulated power seen at any point in a given billing period, denoted by S'. In addition to the energy cost (cost per KWh) and the peak power cost, we consider that the data center operational cost accounts for the cost per maximum charging and discharging denoted by $\phi_{i,C}$, and $\phi_{i,D}$, respectively which depends on the ESD characteristics (e.g., number of cycle life for a given depth of discharge). We keep the program linear by approximating battery cost model as a linear function where the cost is inured proportionally to the charging/discharging rate with respect to maximum charging and discharging. Therefore, the time-averaged operational cost of data centers over S slots, can be written as the following optimization problem, namely **P5**:

$$\boxed{\begin{aligned} \text{minimize} \quad & \tfrac{1}{S}\textstyle\sum_{t=1}^{S}\sum_i g_i(t)\alpha_i(t) + \tfrac{\text{chrg}_i(t)}{C_i}\gamma_{i,C} + \tfrac{\text{dschrg}_i(t)}{D_i}\gamma_{i,D}, \\ & + \textstyle\sum_{t'=1}^{T'} \Big(\max_{1\le k\le T}(g_i((t'-1)T+k) - p_0 \Big)^{+} \beta_i, \qquad (7.4) \\ \text{subject to} \quad & (6.1),\ (6.4),\ (7.1),\ \text{and}\ (7.3). \end{aligned}}$$

Similar, the solutions of Chapter 6, we can relax the integer constraint in (6.1) (i.e., y_i number of active servers) and round the resulting solution with minimal increase in cost, as proven in Prop. 5.2.2. Also observe that **P5** disregards the non-convex and non-linear constraint (7.2), however the following lemma asserts that the optimal solution to **P5** never chooses to simultaneously charge and discharge from ESDs. This is intuitively clear, because charging and discharging the ESD in the same slot incur additional battery cost and energy due to the battery inefficiency. It is, thereby, beneficial to instead satisfy the demand form the grid.

Lemma 7.1.1. *The optimal solution to* ***P5*** *for every data center* i *and time* t *always chooses* $chrg_i(t)\,dschrg_i(t) = 0$.

Proof. We prove by contradiction. First, assume $\mathrm{chrg}_i(t) > 0$ and $\mathrm{dschrg}_i(t) > 0$ and $\eta\mathrm{chrg}_i(t) - \frac{1}{\rho}\mathrm{dschrg}_i(t) = \eta c' \geq 0$, which means that the energy level of ESD increases. Also given that $0 \leq \eta \leq 1$, it follows that $c' = \mathrm{chrg} - \frac{1}{\eta^2}\mathrm{dschrg}_i(t) \leq \mathrm{chrg}_i(t) - \mathrm{dschrg}_i(t)$, and obviously $c' \leq \mathrm{chrg}_i(t)$. Since charging rate is higher than discharging rate, energy (and power) cost increases (see (7.3)). Due to linearity of data center energy and peak power cost, we define the parameter a to denote the energy and cost increase rate. We have that:

$$a(\mathrm{chrg}_i(t) - \mathrm{dschrg}_i(t)) + \frac{\mathrm{chrg}_i(t)}{C} + \frac{\mathrm{dschrg}_i(t)}{D} \geq ac' + \frac{c'}{C}.$$

In the above, the left side denotes the value of the objective function when the solution decides only on charging the ESD with the rate c', i.e., $\mathrm{chrg}_i(t) = c'$, and $\mathrm{dschrg}_i(t) = 0$. This means that there is a feasible solution which results in the same amount of increased energy level in ESD as the optimal solution, and yet it results a lower increase in the value of the objective function than the optima solution, which contradicts the optimality of the solution.

Next, assume $\mathrm{chrg}_i(t) > 0$ and $\mathrm{dschrg}_i(t) > 0$ and $\eta\mathrm{chrg}_i(t) - \frac{1}{\eta}\mathrm{dschrg}_i(t) = -\frac{1}{\eta}d' \leq 0$, which means that the energy level of ESD decreases. Also given that $0 \leq \eta \leq 1$ it follows that $d' = \mathrm{dschrg}_i(t) - \eta^2\mathrm{chrg}_i(t) \geq \mathrm{dschrg}_i(t) - \mathrm{chrg}_i(t)$, and obviously we have $d' \leq \mathrm{dschrg}_i(t)$. Since discharging rate is higher than charging rate, energy (and power) cost decreases. Similar to the previous case we define the parameter a to denote the energy and cost decrease rate. We have that:

$$a(\mathrm{chrg}_i(t) - \mathrm{dschrg}_i(t)) + \frac{\mathrm{chrg}_i(t)}{C} + \frac{\mathrm{dschrg}_i(t)}{D} \geq -ad' + \frac{d'}{D}.$$

In the above, the left side denotes the value of the objective function when the solution decides only on discharging the ESD with the rate c', i.e., $\mathrm{chrg}_i(t) = 0'$, and $\mathrm{dschrg}_i(t) = d'$. This means that there is a feasible solution which results in the same amount of decreased energy level in ESD as the optimal solution and yet higher reduction in the value of the objective function, which contradicts the optimality of the solution. □

The problem **P5** as described above (given relation of number of servers to a continuous variable) is a linear programing which can be optimally solved using the existing linear programming solvers. However, the solutions of **P5** over time are *dependent* due to the several sources of coupling factors: (i) the peak power cost is calculated over every $S' \geq 1$ slots (7.4), as a result it couples the solutions over S', (ii) the ESDs' dynamics (7.1) and the carbon capping constraint (6.4) couples the solutions over time. In practice, the billing period (S') is typically a month, and the carbon cap is typically given over a year of operation of the data centers. This means that S is typically equals to the number of slots for a year. Therefore, in practice, it becomes impractical to solve **P5**: (i) it is infeasible to predict the parameters over a large number of slots of S, (ii) even if we use historical data (e.g., the data from the previous year), traditional approaches to construct optimal policies of **P5** involve the use of Markov Decision Theory and Dynamic Programming [26, 116]. It is well known that these techniques suffer from the "curse of dimensionality" where the optimal strategy computing complexity exponentially grows with the system size [105].

7.2 Online Solutions

In this chapter, we study and propose online solutions to solve **P5**. The performance of the online solutions are based on (i) the feasibility assumption which ensures that **P5** has non-zero feasible solutions, (ii) the bounded assumption which ensures that the total one-slot cloud's carbon footprint is bounded by b_{max}, i.e., $b(t) \leq b_{max} \, \forall t$, and (iii) the predictability assumption which ensures that the data center parameters are predictable over T slots with reasonable accuracy, and their most variabilities fall within T slots. Observe that, the assumptions are not constraining in practice, and that the last assumption is consistent with the daily variability of the data center parameters for a T value that is daily basis.

7.2.1 OnCMCC: Predictive Online Solution

We design the online solution , namely OnCMCC, to solve the problem **P5** over $T \leq S'$, where T consists of slots for one or more days (e.g., T=24 or T=48 for hourly basis slots). In this solution we also use β' for the peak power cost where $\beta' = \frac{T}{S'} \beta$. The energy cost efficiency of OnCMCC is lower than that of the offline optimal solution, since it ignores the energy migration of ESDs across T slots, and it cannot leverage the variation of the data center parameters across T slots to further smoothen the power and consequently optimally decrease the peak power cost. However, OnCMCC is inspired by the observation that the variation of the data center parameters across days is usually lower than the variation across slots within days. Given the limited ESDs' sizes, therefore, the ESDs most likely to be best utilized to leverage the daily variation of the data center parameters. As a result, the energy migration of ESDs over every T slots become very negligible. Similarly, most of the power smoothing can be obtained by leveraging the daily variation of the parameters. The availability of renewable energy, however, not only significantly varies within days (solar energy is only available in the days), but also significantly varies across days and even months in a year depending on the weather conditions and geographical locations. However, due to the limited size of ESDs and their physical limitations (e.g., self-discharge), it is impractical to migrate renewable energy across such long periods, making the cost optimality distance of OnCMCC negligible when carbon capping requirement is relaxed. Note, OnCMCC can only satisfy the carbon cap in a best-effort manner, since the feasible carbon cap for a T-slot depends on the workload intensity, the availability of renewable energy and the carbon intensity on that T-slot. Due to the intermittent nature of the renewable power, therefore, OnCMCC may significantly violates the carbon cap, making it inefficient particularly when cloud needs to perform under (relatively tight) carbon capping requirement. To avoid this problem, we extend OnCMCC to leverage the T-slot Lyapunov optimization in order to account for the dynamics of carbon footprint over S.

7.2.2 OnCMCCLyp: T-slot Lyapunov Optimization Solution

In this section we extend OnCMCC to make use of T-slot Lyapunov optimization, namely OnCMCCLyp, for carbon capping requirement. In accordance of Lyapunov optimization, we define a virtual queue [90] with occupancy $X(t)$ equal to the maximum excess carbon footprint beyond the average carbon footprint over every T-slot. Using $X(0)$=0, we propagate the $X(t)$ values over every T-slot as follows:

$$X(t_0 + T) = max[X(t_0) - T\psi, 0] + \sum_{\tau=t_0}^{t_0+T-1} \sum_i b_i(\tau). \tag{7.5}$$

Building upon Lyapunov optimization technique we design OnCMCCLyp as given in Algorithm 7.1. The parameter V Algorithm 7.1 is the Lyapunov control parameter which manages the cos carbon footprint reduction tradeoff. It can be seen that OnCMCCLyp, requires only the T ahead information as the inputs. The algorithm removes the coupling property of **P5** by (i) removing the constraint (6.4)), and (ii) managing the energy storage dynamics over window $(t{+}T{-}1)$ rather than S and managing peak power reduction over $(t, t{+}T{-}1)$, rather than S'. It can be seen that OnCMCCLyp leverages both the predictability of data center input parameters over window of T and Lyapunov optimization to design the online algorithm. The predicted input parameters of the time frames T helps to optimally manage the operational cost according to the variation of the parameters within the frame T as described in Section 7.2.2. The Lyapunov technique is used to stabilize the carbon footprint dynamics across T-slots. In order to evaluate OnCMCCLyp, we theoretically compare its performance against the offline optimal solution of problem **P5** for the case of (i) $S'{=}T$, and (ii) the energy storage dynamics only depends on the window of T. In other words, we consider that the operational cost and energy storage can be optimally managed using T slots future information, and evaluate how OnCMCCLyp can manage the carbon cap (i.e., Ψ) without excessively increasing the operational cost. The theoretical results, in particular, extend Theorem 6.2.4, to evaluate OnCMCCLyp as follows

Theorem 7.2.1. *(Performance Bound Analysis of OnCMCCLyp): Suppose* $X(0){=}0$*, and that the maximum carbon footprint of the cloud over every* T *slot is upper bounded by* Tb_{max}*. Also define* $cost^*_T$ *as the optimal solution to the special case of problem* ***P5****, where* $S'{=}T$*, and for every* t_0 *the beginning slot in every frame* T*, we have* $e_i(t_0 + T) = e_i(t_0)$*. Further, suppose data center parameters are i.i.d. over every* T*-slots, and let cost(*τ*) and* $b(\tau)$ *to denote the OnCMCCLyp cost and carbon footprint, respectively for slot* τ*. Then for* $V > 0$*, and the integer variable* $k = 0, 1, \ldots K$ *where*

$S = KT$ we have the following:

$$\begin{aligned} cost_T = &\ \limsup_{K\to\infty} \tfrac{1}{K} \textstyle\sum_{k=0}^{K-1} \mathbb{E}\{\sum_{\tau=kT}^{kT+T-1} cost(\tau)\} \\ &\leq cost_T^* + \tfrac{B}{V}, \end{aligned} \tag{7.6}$$

$$\sum_{t=0}^{S-1} \sum_i b_i(t) \leq \Psi + \sqrt{2}\sqrt{KB + V(Kcost_T^* - \sum_{k=0}^{K-1} \sum_{\tau=kT}^{kT+T-1} cost(\tau))}, \tag{7.7}$$

where $B = \frac{1}{2}(T^2 b_{max}^2 + T^2\psi^2)$.

Proof. To prove (7.6), let's define a quadratic Lyapunov function $L(t)$ that measures the aggregate carbon deficit in the system: $L(t) = \frac{1}{2}X(t)^2$. We define $k \in \mathbb{N}$ where $k = 0 \ldots K-1$. Next, let's define the T-slot Lyapunov drift, $\Delta_T(kT)$ as the expected change in the Lyapunov function over every T slots as follows: $\Delta_T(kT) = \frac{1}{2}(\mathbb{E}\{L(kT+T) - L(kT)\}|X(kT))$. Now we derive the upper bound on $\Delta_T(kT)$ as follows. By (7.5) we have that $(X(kT+T))^2 \leq (X(kT) - T\psi + \sum_i \sum_{\tau=kT}^{KT+T-1} b_i(t))^2$. Squaring both side of (7.5), and given that $(\sum_{\tau=kT}^{KT+T-1} b_i(\tau))^2 \leq T^2 b_{max}^2$ we have:

$$\begin{aligned} &[X(kT+T)]^2 \leq [X(kT)]^2 + T^2[\psi]^2 + [\textstyle\sum_{\tau=kT}^{KT+T-1} b_i(\tau))]^2 \\ &-2TX(kT)\psi + 2X(kT)\textstyle\sum_{\tau=kT}^{KT+T-1}\sum_i b_i(\tau) - 2T\psi \sum_{\tau=kT}^{KT+T-1}\sum_i b_i(\tau) \\ &\Rightarrow [X(kT+T)]^2 - [X(kT)]^2 \leq \\ &T^2 {b_{max}}^2 + T^2{\psi}^2 - 2X(kT)[T\psi - \textstyle\sum_{\tau=kT}^{KT+T-1}\sum_i b_i(\tau)]. \end{aligned} \tag{7.8}$$

Now multiplying the above inequality by $\frac{1}{2}$, taking expectations over $X(kT)$, and $b_i(\tau)$, conditioning on $X(kT)$, we get the T-slot conditional Lyapunov drift $\Delta_T(kT)$:

$$\Delta_T(kT) \leq B - X(kT)\mathbb{E}\{T\psi - \textstyle\sum_{tau=kT}^{kT+T-1}\sum_i b_i(\tau)|X(kT)\},$$

where $B = \frac{1}{2}(T^2 {b_{max}}^2 + T^2\psi^2)$. Adding the cost as penalty term to the both side of the above inequality, i.e., $V\sum_{\tau=kT}^{KT+T-1} \text{cost}(\tau)$, we get:

$$\begin{aligned} &\Delta_T(kT) + V\textstyle\sum_{\tau=kT}^{KT+T-1} \text{cost}(\tau) \leq \\ &B + V\textstyle\sum_{\tau=kT}^{KT+T-1} \text{cost}(\tau) - X(kT)\mathbb{E}\{T\psi - \sum_{\tau=kT}^{KT+T-1}\sum_i b_i(\tau)|X(kT)\}. \end{aligned} \tag{7.9}$$

Observe that OnCMCCLyp as shown in Alg.(7.1) minimizes the right hand side of (7.9). The i.i.d. assumptions of input parameters, ensures the existence of an optimal stationary randomized policy π which can achieve as follows for all $k = 0 \dots K-1$: $\mathbb{E}\{\sum_{\tau=kT}^{kT+T-1} b^{\pi}(\tau)\} \leq T\psi$ and $\mathbb{E}\{\sum_{\tau=kT}^{kT+T-1} \text{cost}^{\pi}(\tau)\} = \text{cost}_T^*$, where cost_T^* is the offline optimal T-slot averaged cost of problem P1 under Theorem's conditions (this can be proven using Caratheodory's theorem similar to the proof in [90]).

Using the fact that OnCMCCLyp is constructed to minimize the R.H.S. of (7.9), we have that:

$$\begin{aligned}&\Delta_T(kT) + V\textstyle\sum_{\tau=kT}^{kT+T-1}\text{cost}(\tau) \leq \\ &B + V\text{cost}_T^{alt} - X(kT)\mathbb{E}\{(T\psi - \textstyle\sum_{\tau=kT}^{kT+T-1}\sum_i b_i^{alt}(\tau))|X(kT)\},\end{aligned} \tag{7.10}$$

where *alt* represents any alternate policy (including stationary randomized policy π) that can be implemented over slot t. Then, plugging the control decisions corresponding to the stationary randomized policy π, we get:

$$\Delta_T(kT) + V\sum_{\tau=kT}^{kT+T-1}\text{cost}(\tau) \leq B + V\text{cost}_T^*. \tag{7.11}$$

Taking the expectations from both sides, summing the above over k=0 ... K−1, using the fact that $\Delta_T(kT)$≥0, and dividing both sides by KV, we have:

$$\frac{1}{K}\mathbb{E}\{\sum_{k=0}^{K-1}\sum_{\tau=kT}^{kT+T-1}\text{cost}(\tau)\} \leq \text{cost}_T^* + \frac{B}{V}.$$

Taking a lim sup as $K \to \infty$, we complete the proof.

To prove (7.7), from the carbon deficit queue dynamic (7.5), we have that:

$$X(kT+T) - X(kT) \geq \sum_{\tau=kT}^{kT+T-1}\sum_i b_i(t) - T\psi.$$

Summing the above over $k = 01, \dots K-1$, and using the fact that $X(0)$=0 we obtain the following:

$$\sum_{k=0}^{K-1}\sum_{\tau=kT}^{kT+T-1}\sum_i b_i(t) \leq KT\psi + X(KT). \tag{7.12}$$

Algorithm 7.1: OnCMCCLyp Algorithm

```
Initialize the virtual queue $X$
for every slot $t = 1 \dots S$ (beginning of the slot) do
    Predict the system parameters over the window $t + T - 1$
    Solve the following problem:
    Minimize:
        $V\Big(\frac{1}{T}\sum_{\tau=t}^{t+T-1}\sum_i g_i(t)\alpha_i(t) + \frac{c_i(t)}{C}\phi_{i,C} + \frac{d_i(t)}{D}\phi_{i,D}$
        $+ \sum_i \max_{t\leq\tau\leq t+T-1}(g_i(\tau) - p_{i,0})^+\beta_i\Big) - X(t)\sum_{\tau=t}^{t+T-1}\sum_i b_i(\tau)$    (7.15)
    Subject to: ((6.1), (7.1), and (7.3).
    Update the virtual queue $X$ using (7.5).
end for
```

Similarly, summing (7.11) over the entire budgeting period, i.e., k=0 . . . K−1, and using the fact that $L(0)$=$X(0)$=0 yields:

$$L(KT) = 1/2X^2(KT) \leq KB + KVcost_T^* - V\sum_{k=0}^{K-1}\sum_{\tau=kT}^{kT+T-1}\text{cost}(\tau). \tag{7.13}$$

Plugging (7.13) into (7.12), and using the fact that Ψ=$KT\psi$ we prove the theorem as follows:

$$\begin{aligned}&\textstyle\sum_{k=0}^{K-1}\sum_{\tau=kT}^{kT+T-1}\sum_i b_i(t) \leq \\ &\Psi + \sqrt{2}\sqrt{KB + V(Kcost_T^* - \textstyle\sum_{k=0}^{K-1}\sum_{\tau=kT}^{kT+T-1}\text{cost}(\tau))}.\end{aligned} \tag{7.14}$$

□

According to Theorem 7.2.1, the OnCMCCLyp achieves average cost no more than $O(1/V)$ distance above the optimal average cost of P4 under the Theorem's conditions. The large value of V comes at the expense of an $O(V)$ tradeoff in achieving carbon cap.

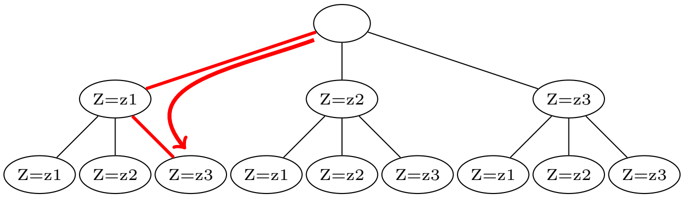


Figure 7.1: A Sample Scenario Tree for the Example Random Variable Z with Three Possible Values over Two Stages.

7.3 Stochastic Programming Approach

The performance of the online solutions depends on the predictability of the parameters over T. To realize the impact of the prediction error consider an example where we have a cloud consisting of two data centers DC1 and DC2 with no carbon capping requirement. Suppose an hourly basis slots, $T = S' = 1$, and that for a given slot t, $\alpha_1(t)$=1 \$/KWh, and $\alpha_2(t)$=2 \$/KWh. Also assume the stipulated power for both of DC1 and DC2 is equal to 8 KW, p_0=8, and β=10 \$/KW). Further, assume the predicted power demand (p^{tot}) is 8 KW, wheres as the actual power demand is 9 KW. The online solution OnCMCC to minimize the electricity cost uses the predicted power demand as the input and assigns the entire demand to DC1 to achieve the optimal cost of 8. The actual cost, however, becomes 19, since the actual demand assigned to DC1 is 9 KW. Observe that the prediction error has a very harmful impact on the peak power shaving. This is because the optimal approach, as seen in the example, is to utilize the data centers with low electricity cost as much as possible without violating their stipulated power (DC1 in the example). Therefore, any under prediction error (e.g., under prediction of the workload or over prediction of the available renewable power) is likely to violate the stipulated power of those data centers, resulting in unexpected increase in the peak power cost. As a result, the cost efficiency of the solutions decrease since the peak power cost contributes a significant portion of the data center operational energy cost (we typically have $\beta >> \alpha(t)$). Therefore, we choose to solve the problem given non-zero prediction error. Accordingly, we use stochastic programming to take into consideration the randomness of the predicted input parameters. Stochastic programming is an approach for modeling optimization problems that involve uncertainty and takes advantage of the fact that probability distributions governing the data are known or can be estimated (e.g., from historical data). The major issue in developing the stochastic problem formulation is the modeling of the uncertainties. We characterize and model uncertainties in the form of scenarios (possible outcomes of the data), a typical scheme in stochastic programming approach [102]. The outcomes (scenarios) are generally described in terms of elements w of a set W. For instance, in our problem W can be the set of possible input workload over the budgeting period T. The goal is to find a policy that is feasible for all the possible parameter realizations (scenarios) and optimize the expectation of the objective functions given the input random variables and their probabilities. Stochastic programming has many variants including stochastic dynamic programming. Although our problem can be modeled as a stochastic dynamic programming, it's not an appropriate solution because of the large number of states at each time stage. This is partially because of the continuous dynamic states of ESDs. Discretization of the ESD states for multiple data centers causes the number of states at each stage of the stochastic dynamic programming to dramatically increase. We incorporate the stochastic scenarios in the original optimization problem and designed the "deterministic equivalent" of the stochastic problem which is a typical stochastic programming approach [102]. Consider a deterministic optimization problem of the objective function f, the constraint function of h, and the decision variable of x, i.e., minimize $f(x)$, subject to $h(x)$. Its stochastic programming counterpart can be written as follows:

$$\begin{aligned} &\text{Minimize} && \textstyle\sum_{w \in W} Pr(w) f(x,w), \\ &\text{Subject to:} && h(x,w)\ \forall w \in W. \end{aligned} \tag{7.16}$$

where Pr denotes the probability function.

To capture the randomness of the parameters and characterize the scenarios, we model the prediction error of the input workload of each front-end, Λ_j, the available renewable power at each data center i, $r_i(t)$, the electricity price at data center location i, α_i, and the carbon intensity of the grid power at each data center i, ε_i^g. We denote $r_i(t)$ the actual renewable energy available to data center i at time t and use $\hat{r}_i(t)$ for the predicted generation. We denote $r_i(t)$=$(1+\zeta_{i,r})\hat{r}_i(t)$, where $\zeta_{i,r}$ is the prediction error. We assume unbiased prediction $\mathbb{E}(\zeta_r)$=0, and denote the variance by Ψ_r^2 which can be obtained from historic data. These are standard assumptions in statistics. We use the similar assumptions for the prediction error of the input workload, the electricity price, and the carbon intensity: i.e., $\Lambda_j(t)$=$(1+\zeta_{j,\lambda})\hat{\Lambda}_j(t)$, $\alpha_i(t)$=$(1+\zeta_{i,\alpha})\hat{\alpha}_i(t)$, and $\varepsilon_i^g(t)$=$(1+\zeta_{i,\varepsilon})\hat{\varepsilon}_i(t)$, where we use hat superscript to distinguish between the actual and the predicted data, and denote ζ as the prediction error random variable. We also consider that the random variables (e.g., the prediction error of workload of each front-end, $\zeta_{j,\lambda}$) as independent random processes. As a result, the evolution of these stochastic processes is modeled as a multivariate random process. The marginal distribution for each of these random processes at any time step is assumed to be a normal distribution in consistent with the nature of unbiased prediction error. The sampling of these random processes results in scenarios representing the future realizations of the uncertainties. To define the scenarios, we approximate the marginal distribution of the random parameters (i.e., ζ) into discrete samples with equal probabilities. The multivariate random process has therefore $L_\lambda^M L_r^N L_\alpha^N L_\varepsilon^N$ samples at each time step, where L_λ, L_r, L_α, and L_ε (e.g., $L = 5$) denote the number of discretization levels used for workload demand, renewable power, energy price, and grid carbon intensity, respectively. The evolution of the random process for the entire T slots is a huge set of scenarios. In other words, this type of uncertainty modeling results in a multistage "scenario tree" with T branching stages and $L_\lambda^M L_r^N L_\alpha^N$ samples at each node of the tree (see Fig. 7.1 as an example). Each of the scenarios (i.e., a path from root to a leaf of the scenario tree) represents a possible future realization of the multivariate random process. Observe that the scenario tree for our problem is huge. For example consider five discretization levels for each random processes, hourly-basis slots and $T = 24$ to capture the daily variation of the input processes. Further assume a small cloud consisting of five data centers and ten front-ends. The scenario tree in this case has 24 branching stages where each node in the tree has 5^{25} children. Hence, the total number of scenarios (total number of paths in the scenario tree) is equal to 5^{600}. To solve the stochastic model, the multivariate random process with huge set of scenarios has to be approximated to a simple random process with finite set of scenarios and should be as close as possible to the original scenario tree.

7.3.1 Stochastic Scenario Generation

The computational effort for solving scenario-based optimization models depends on the number of scenarios, as the size of input variables and the constraints typically dramatically grows with increasing the number of scenarios. Hence, it is natural to look for scenario-based approximations of the random data processes that have a small number of scenarios, but still represent reasonably good approximations. The currently available scenario reduction techniques make use of probability metrics to choose a subset of scenarios [54, 59]. The scenario to be deleted is selected by comparing each scenario with the rest of the scenarios. For deleting scenarios from an initial set of scenarios, the process of one-to-one comparison has to be repeated times.

This is computationally expensive and is not suited for the huge set of initial scenarios (in particular the scenario reduction algorithms in [54] make use of algorithms very similar to "k-means" and "k-medoids" where the probabilistic measures are used to evaluate the distance of the scenarios). Similar to k-means, these solutions can be implemented efficiently using parallel programming to run on huge set of initial scenarios. Further, it should be noted that scenario reduction procedure does not need to run frequently. As long as the stochastic parameters (i.e., standard deviation) of the prediction error are remained constant, the scenarios do not change. Therefore, in practice, it is feasible to find a subset of scenarios from the entire set of initial scenarios. As a general case, where running algorithms on the complete scenario tree may not be feasible, one can use the following two steps heuristic approach:

- *Step one:* generate a scenario tree with reasonable size such that the scenario reduction algorithms can run with reasonable speed.
- *Step two:* apply scenario reduction algorithm (e.g., [54, Algorithms 2, and 3]) over the scenario tree to achieve the desired number of scenarios.

Depending on the number of initial scenarios and the resources available to run the algorithm at step two, the step one can be configured to generate a complete scenario tree or a reduced one. In the following we explain some of the possible strategies which can be used in the step one in order to build a scenario tree with a reasonable size.

- Use stochastic aggregation rules to reduce the number of initial input random processes (e.g., workload). Consider the random processes X and Y with normal distribution, $X \sim \mathbb{N}(\mu_1, \Psi_1^2)$, $Y \sim \mathbb{N}(\mu_2, \Psi_2^2)$, then $aX{+}bY$, where a and b are constant numbers, also has a normal distribution as follows, $aX{+}bY \sim$

$\mathbb{N}(a\mu_1+b\mu_2, a^2\Psi_1^2+b^2\Psi_2^2)$. We can use such stochastic aggregation rules to reduce the number of random processes as follows. First, data centers may use a combination of wind and solar energy where an aggregate random process of the two can capture their randomness. Second, in practice, number of front-ends (i.e., M) is very large. Suppose there is no restriction on the destination data center of front-ends, such that every front-end can get service from all the available data centers in the cloud. Then, the entire input workload of all front-ends can be aggregated into one single random process. In practice, however, there are always some restrictions such as network latency (proximity of front-ends to the data centers) and data availability, where every front-ends can get service from a subset of data centers. In this case we can group front-ends depending on their feasible destination data centers and aggregate the workload of each group.

- Ignore random processes which have relatively small prediction error to the other random data processes. Ignoring these processes, significantly reduces the initial scenario tree size with negligible impact in the solution.

- Trade accuracy with speed. We can also reduce the size of the initial scenario tree at the expense of lower accuracy of the solution. This can be achieved by (i) compacting the number of stages, (ii) using small number of discretization levels for the random data processes, and (iii) prioritizing the random data processes according to their importance in the problem and ignoring the randomness of the low priority ones. To compact the number of stages, we build the initial scenario tree with T', $T'{\leq}T$ stages and then for every $t{\in}T$ we use the scenarios of t' where $t'{\in}T'$ and $(t'-1)T/T'{\leq}t{\leq}t'T/T'$. Most of the scenario reduction algorithms reduce the number of scenarios without considering the impact of the scenarios on the solution. However, problem-specific strategies can be designed to remove the randomness of the less important random processes in order to increase the contribution of the more important data process in the final scenario set. In our problem, for instance, the available renewable power of data centers, i.e. solar and wind power, can be predicted with relatively high prediction error compared to the input workload and the electricity price. In practice, however, not all of data centers in a cloud utilize on-site renewable power, making the randomness impact of the available renewable power less important than the randomness impact of the input workload on the solution.

In practice, any other problem-specific strategies can be applied. Also an alternative to the two-steps scenario reduction solution is to use a solution such as the meta heuristic solution described [93] to generate the number of desired scenarios directly from the complete scenario tree without the need to perform scenario comparisons over the entire scenario tree. Generally, because the size of the final scenario set is significantly smaller than the initial scenario set (i.e., scenario tree), applying problem-specifics strategies helps to choose the right scenario subset. We study some of the above strategies and their impact in the experimental study.

7.3.2 Stochastic Programming Problem Formulation

The deterministic equivalent of the stochastic problem can be written taking into account the stochastic scenarios. Accordingly, following the model of (7.16), and given the set of scenarios W we formulate the stochastic counterpart of the problem **P5**, namely **P6** as follows.

To formulate **P6**, we need to rewrite the objective function and the constraints of **P5** that are affected by each element w of the scenario set W. We introduce a new decision variable of $x_{i,j}(t)$, $0 \leq x_{i,j}(t) \leq 1$ which denotes the fraction of workload from front-end j to be routed to data center i, i.e., $\lambda_{i,j}(t) = x_{i,j}(t)\Lambda_j(t)$. We use x instead of $\lambda_{i,j}(t)$, as it simplifies the modeling of **P6**. Note that the problem **P6** should solve for workload distribution, $x_{i,j}(t)$, number of active servers, $y_i(t)$, and charging, $c_i(t)$, and discharging, $d_i(t)$ of ESDs at each data center i. This is performed using the stochastic input parameters as elements w from the scenario set W, each of which is associated with the probability $Pr(w)$: the workload from each front-end j, $\Lambda_{j,w}(t)$, the parameters of data center i including the electricity price $\alpha_{i,w}(t)$, the available renewable power, $r_{i,w}(t)$, and the grid carbon intensity $\varepsilon^g_{i,w}(t)$. In the following we give an overview of how **P6** can be written according to the models of **P5** and the scenario set W.

Following the model (6.1), the service constraint can be written as follows:

$$\forall j, t: \quad \textstyle\sum_i x_{i,j}(t) = 1, \text{ [service]}. \tag{7.17}$$

Similar to (6.1) the performance constraints of **P6** can be written as follows:

$$\begin{array}{ll}
\forall i, t, w: & n_{i,w}(t)\mu_i > \sum_j x_{i,j}(t)\Lambda_{j,w}(t), \text{ [queuing stability]},\\
\forall i, t, w: & \frac{1}{n_{i,w}(t)\mu - \sum_j x_{i,j}(t)\Lambda_{j,w}(t)} \leq \text{dly}'^{ref}, \text{ [service delay] },\\
\forall i, j, t: & (\text{dly}^{ref} - (dly'^{ref} + dly''_{i,j}(t))\lambda_{i,j}(t) \geq 0, \text{ [total delay]},\\
\forall i, t, w: & y_{i,w}(t) \geq (1 + y^{slack})n_{i,w}(t) \leq Y_i, \text{ [capacity]},\\
\forall i, t: & y_i(t) = \sum_{w \in W} Pr(w) y_{i,w}(t) \text{[expected no of servers]}.
\end{array} \tag{7.18}$$

Observe that the above equations are counterpart of equations of (6.1) plus an additional equation for $y_i(t)$, i.e. “expected number of servers”. Similarly, the power supply demad balance and the carbon capping constraint can be written as follows:

$$\forall i, t, w: \quad g_{i,w}(t) + r_{i,w}(t) + \text{dschrg}_i(t) = p^{tot}_{i,w}(t) + \text{chrg}_i(t). \tag{7.19}$$

$$\begin{aligned} &b_i(t) = \textstyle\sum_{w \in W} Pr(w)(g_{i,w}\varepsilon^g_{i,w}(t) + r_{i,w}(t)\varepsilon^r_{i,w}(t)), \\ &\tfrac{1}{S}\textstyle\sum_{t=0}^{S-1}\sum_i b_i(t) \leq \tfrac{\Psi}{S} = \psi. \end{aligned} \tag{7.20}$$

Finally, given the scenario set W and the associated probabilities **P6** can be written to minimize the expected cost over scenarios as follows:

$$\boxed{\begin{aligned} \text{minimize} \quad & \tfrac{1}{S}\Big(\textstyle\sum_{w \in W} Pr(w) \sum_{t=0}^{S-1}(\sum_i g_{i,w}(t)\alpha_{i,w}(t) \\ & + \tfrac{\text{chrg}_i(t)}{C}\phi_{i,C} + \tfrac{\text{dschrg}_i(t)}{D}\phi_{i,D}) \\ & + \textstyle\sum_{t'=0}^{S/S'-1}(\max_{(t'-1)S' \leq \tau \leq t'S'} \\ & \big(g_{i,w}(\tau) - p_{0,i}\big)^+ \beta_i\Big). \\ \text{subject to} \quad & \text{(7.17), (7.18), (7.1), (7.19), and (7.20).} \end{aligned}} \tag{7.21}$$

The stochastic counterpart of the online solutions, OnCMCC, and OnCMCCLyp can be designed based on **P6** (see Alg. 7.2 for the stochastic counterpart of OnCMCCLyp).

7.4 Distributed Implementation of the Online Solutions

Since OnCMCC and OnCMCCLyp are linear programming, they can be efficiently solved centrally if all the necessary information can be collected at a single point, and that if the scale of the system in terms of number of data centers and front-ends, and number of stochastic scenarios are sufficiently small. However, in practice a distributed algorithm is preferable due to the following. First there is a strong case where the different parts of the system (i.e., data centers, front-ends, and the network) have different owners (e.g., Internet-scale systems usually outsource route services (e.g., Akamai). In such a setting a distributed algorithm is preferable to partially exchange information, and maintain the confidentiality of different parts of the system as much as possible. Second, the online problems, OnCMCC and OnCM-

Algorithm 7.2: Stochastic counterpart of OnCMCCLyp Algorithm

```
Initialize the virtual queue $X$
Generate the stochastic scenario set of $W$
for every slot $t = 1 \ldots S$ (beginning of the slot) do
    Predict the system parameters over the window $t + T - 1$
    Solve the following problem:
    Minimize:
    $\sum_w \mathsf{Pr}(w)\Bigg(V\Big(\frac{1}{T}\sum_{\tau=t}^{t+T-1}\sum_i\sum_i g_{i,w}(t)\alpha_{i,w}(t) + \frac{c_i(t)}{C}\phi_{i,C} + \frac{d_i(t)}{D}\phi_{i,D}$
    $+\sum_i \max_{t\leq\tau\leq t+T-1}(g_{i,w}(\tau) - p_{i,0})^{+}\beta_i\Big) - X(t)\sum_{\tau=t}^{t+T-1}\sum_i(g_{i,w}\varepsilon^{g}_{i,w}(t) + r_{i,w}(t)\varepsilon^{r}_{i,w}(t))\Bigg)$    (7.22)
    Subject to: ((7.1), (7.18), and (7.19).
    Update the virtual queue $X$ using (7.5).
end for
```

CCLyp become a large-scale linear optimization problems in practice. The number of decision variables are in the order of $(T \times N \times M))$. Particularly, the magnitude of number of front-ends, M can be very large. More importantly when designed as a stochastic problem as described in Section 7.3, the size of OnCMCC and OnCMCCLyp grows with increasing the number of scenarios. A distributed algorithm which can be efficiently parallelized is preferable to solve the problems in a computation efficient way.

We make use of the standard 2-block Alternating Direction Method of Multipliers (ADMM) which is a simple yet powerful algorithm to solve large-scale distributed convex and linear programming problems [27]. Particularly, ADMM is shown to linearly converge for linear programming problems, such as OnCMCC and OnCMCCLyp. The standard 2-block ADMM solves the problems where the objective function is separable over two sets of variables, which are coupled through an equality constraint as shown in the following example:

$$\begin{aligned} \text{minimize:} \quad & f_1(x) + f_2(z) \\ \text{subject to:} \quad & x \in C_1, \text{ and } z \in C_2, \\ & A_1 x + A_2 z = l. \end{aligned} \tag{7.23}$$

where x and z are the variable vectors forming two set of variables, C_1 and C_2 denote the non-empty polyhedral sets, f_1 and f_2 are convex functions, and A_1, A_2, and l are constant matrices and vector, respectively forming a coupling equality constraint for x and z variables. In accordance with ADMM we can form the augmented Lagrangian of (7.23) as follows [27]:

$$L_\rho = f_1(x) + f_2(z) + \gamma^T (A_1 x + A_2 z - l) + \rho/2 \| A_1 x + A_2 z - l \|_2^2,$$

where ρ denotes the penalty term, and γ denotes the Lagrangian parameter. ADMM solves the above dual problem with iterations from k=$0, 1 \dots$ until the convergence condition is stratified, where each iteration has three steps as follows:

1. $x^{(k+1)} := argmin_{x \in C_1} L_\rho(x, z^{(k)}, \gamma^{(k)})$,

2. $z^{(k)} := argmin_{z \in C_2} L_\rho(x^{(k+1)}, z, \gamma^{(k)})$,

3. $\gamma^{(k+1)} := \gamma^{(k)} + \rho(A_1 x^{(k+1)} + A_2 z^{(k+1)} - l)$.

It can be seen that x and z are updated in an alternating fashion. For large scale problems where functions f_1 and/ or f_2 are separable, and consist of sum of several functions, each of steps (1) and (2) can be decomposed and solved separately. This is precisely what allows ADMM algorithm to be useful in solving OnCMCCLyp in a distributed way as described below.

Algorithm 7.3: Distributed Algorithm to Solve OnCMCCLyp

```
initialize $\gamma^{0}_{i,j}(t) = 0 \, \forall i,\, j$, and $t$
initialize $\rho$
initialize $a^{(}_{i,j}0) = 0 \, \forall i,\, j$, and $t$
k := 0
repeat
    Step 1 (data center minimization): given $a^{(k)}_{i,j}(t)$ and $\gamma^{(k)}_{i,j}(t)$, update $\lambda^{(k+1)}_{i,j}(t)$, as follows:
        $\lambda^{(k+1)}_{i,j}(t) := \text{minimize}_{\lambda,g,y,r} \quad (7.15) + \sum_{i,j} \lambda_{i,j}(\gamma_{i,j}(t) - \rho a^{(k)}_{i,j}(t)) + \rho/2\lambda_{i,j}(t)^2$    (7.24)
        subject to: (6.1), (7.1), and (7.3)
    Step 2 (workload distribution minimization): given $\lambda^{(k+1)}_{i,j}(t)$, and $\gamma^{(k)}_{i,j}$ update $a^{(k+1)}_{i,j}(t)$ as follows
        $a^{(k+1)}_{i,j}(t) := \text{minimize:} \quad \sum_{i,j} \frac{\rho}{2} a_{i,j}(t)^2 - a_{i,j}(t)(\gamma^{k}_{i,j}(t) + \rho\lambda^{k}_{i,j}(t))$    (7.25)
        subject to: $\forall i,\, j,\, t: \; a_{i,j}(t) = \Lambda_j(t)$
    Step 3 (dual update): given $\lambda^{(k+1)}_{i,j}(t)$ and $a^{(k+1)}_{i,j}(t)$ update $\gamma^{(k+1)}$ as follows:
        $\gamma^{(k+1)}_{i,j} := \gamma^{(k)}_{i,j} + \rho(\lambda^{(k+1)}_{i,j}(t) - a^{(k+1)}_{i,j}(t)$
    k := k+1
until convergence condition satisfied
```

7.4.1 Distributed Algorithm to Implement OnCMCCLyp

For the sake of notation brevity we design the distributed algorithm for the case of one scenario i.e., $|W| = 1$, in this case $\mathbf{P6}$ is identical to $\mathbf{P5}$. Consider the problem OnCMCCLyp, we can divide the decision variables into two sets, *the data center local variables* which are the variables solely associated with each individual data center i.e., y_i, g_i, r_i, dschrg_i, chrg_i, and e_i, and *the complicating workload distribution variables*, the variables associated with each front-end and data centers i.e., $\lambda_{i,j}$. Observe that the objective function (7.15) only depends on the data center local variables. However, two inequality constraints couples the complicating variables $\lambda_{i,j}$ to the data center local variables i.e., y_i (6.1). In order to fit the problem in to two-block ADMM, we introduce a new set of auxiliary variables $a_{i,j} = \lambda_{i,j}$, and re-formulate the problem (7.15) as follows:

$$\boxed{\begin{array}{ll} \text{minimize:} & (7.15) \\ \text{subject to:} & (6.1),\ (7.1),\ \text{and}\ (7.3) \\ \forall i,\ j,\ t: & a_{i,j}(t) = \lambda_{i,j}(t), \\ \forall j,\ t: & \sum_i a_{i,j}(t) = \Lambda_j(t). \end{array} \qquad (7.26)}$$

This is a 2-block ADMM problem, where $a_{i,j}$s form the variables of one block, and all other variables form the second block (i.e., comparing this problem with the example (7.23), observe that $a_{i,j}(t)$s form the z vector, and other variables form the x vector where (7.15) forms f_1, and $f_2 = 0$). The augmented Lagrangian can then be readily obtained from (7.26). By omitting the irrelevant terms, we can form the optimization required for ADMM iteration steps (similar to the example given in §7.4) as presented in Alg. 7.3.

Remarks: The key points about the Alg. 7.3 is that first, the optimization problem in Step 1, i.e., (7.24), is a separable quadratic optimization problem over data centers where for the given $\gamma_{i,j}^{(k)}(t)$ and $a_{i,j}^{(k)}(t)$, each data center can separately solve for its local variables as well as $\lambda_{i,j}(t)$. In other words, (7.24) is a distributed algorithm where each y_i, g_i, r_i, c_i, d_i, e_i, and λ_i is optimized in parallel. Similarly, the optimization problem in Step 2 i.e., (7.25), is a separable quadratic optimization problem over front-ends, where for the given $\gamma_{i,j}^{(k)}(t)$ and $\lambda_{i,j}^{(k+1)}(t)$, each front-end j can solve (7.25) separately for its local variables, i.e., $a_{i,j}$s. Next, using Alg. 7.3 data centers only required to exchange $\lambda_{i,j}(t)$s and not their other local parameters (renewable energy, servers power characteristics, etc). Third, all the optimization

Table 7.2: ESD Characteristics

Parameters	FLA
Capacity (KW)	115
Cost per discharge ($)	0.65
Cycle life of one cell (cycles)	1200
Discharge rate, D (W)	5387.5
Discharge-to-charge ratio (D/C)	10
Efficiency, η (%)	80
Number of cells	53

problems of Alg. 7.3 are convex quadratic programming which can be optimally solved in a time-efficient way. Finally, Alg. 7.3 is guaranteed for its linear convergence [40].

7.4.2 Distributed Algorithm to Implement OnCMCC

We can take the similar steps of the previous section to implement a distributed algorithm for OnCMCC. However, the corresponding Step 1 of OnCMCC becomes a non-separable optimization problem where the cost optimization for all data centers need to be performed centrally. This is because of the carbon capping constraint (6.4) for each window T. Such solution neither helps in the confidentiality of the problem, nor its scalability. We can apply decomposition techniques to decompose the constraint over data centers, which results to a complex hierarchal distributed algorithm.

7.5 Evaluation

We use the simulation testbed of Chapter 6 and Section 6.4 to evaluate OnCMCC and OnCMCCLyp. The peak power cost is typically calculated over a billing period (typically monthly), therefore we set S' to the number of slots for a month. According to [121] a typical peak power cost is 12 $/KW per averaged power over

15-minutes slots. Given our hourly basis slots we amortize β to 30 \$/KW. We also provide results with β varying from 0 \$/KW up to 40 \$/KW.

Also the data sheet of Flooded Lead Acid (FLA) batteries used in data centers is used for the simulation study (see Table 7.2). We use GNU Linear Programming Kit (GLPK) to solve the optimal solution (solution to **P5**, **P6** and the online solutions.

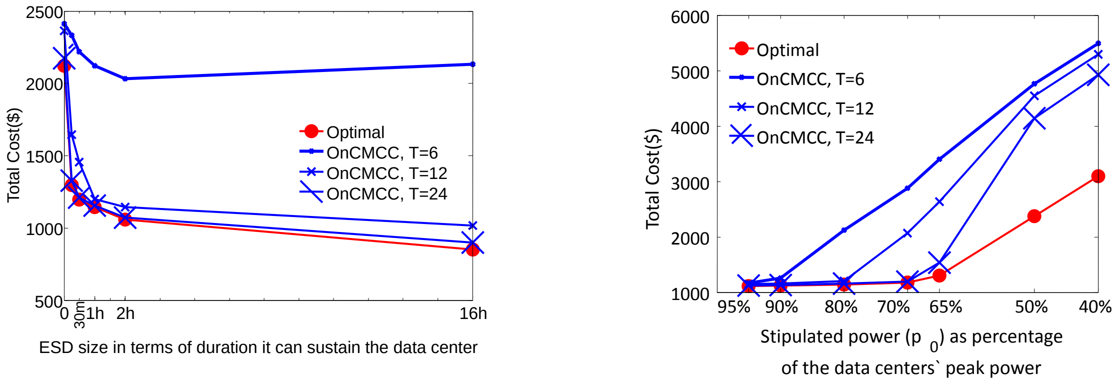
Prediction results: We use one month of training data (July traces) and build weekly and daily Seasonal Auto Regressive Integrated and Moving Average (SARIMA) prediction model to predict workload, and electricity prices and solar energy, respectively. Further, we use ARMA prediction model for wind energy. The lag one (one hour-ahead) prediction error is 14%, 12% and 18% for workload, electricity prices, solar and wind energy respectively. The error goes up to 20%, 18% and 52% for 24 lag (24 hour ahead) prediction of workload, solar and wind energy, respectively. Since wind and solar traces contain some values of zero or nearly zero, we report 90 percentile mean absolute percentage error of these two traces. For instance,lag one and lag 24 mean absolute error of the solar energy is 25%, and 40%, and lag 24 mean absolute error of wind goes up to 67%. Observe that the prediction error of both the solar and the wind energy in our data set is very high which can be typically improved using sufficient training data (using historical data of about 2-3 years []). Since, sufficient training data is not always available, We perform a pessimistic analysis on the impact of high prediction error on our solution, and the way that stochastic programming can remove its harmful impact. The prediction results of the electricity pries are very different across data centers. In particular, the electricity prices of DC3, DC4, DC5, and DC6 are predicted with relatively high accuracy, exhibiting error of 5% for lag one and 15% for lag 24. The electricity prices of DC1 and DC2, however, are predicted with low accuracy, exhibiting the error of 25% for lag one and

36% for lag 24. Note, we do not claim on the perdition models, in practice any other model suited to the data set can be used. We ignore the prediction error of the grid carbon emission intensity (i.e., ε^{grid}), due to their small intensity variation (see) and that we did not have access to their historical data). We use “forecast” library of “R” package (to build time-series based prediction models.

Experiments performed We evaluate the online solutions OnCMCC and OnCMCCLyp for various configurations of data centers including, the length of T, ESD size, E, the magnitude of the stipulated power, p_0 and the magnitude of the carbon cap Ψ. To evaluate OnCMCC and OnCMCCLyp, we use three reference solutions namely **Optimal** (optimal offline solution to **P5**), **MinCost** and **MinCarbon**. MinCost performs global workload management over the cloud to first minimize the cost and then the carbon footprint. **MinCarbon**, on the contrary, first minimizes the carbon footprint across the cloud and then the cost. MinCost and MinCarbon can be viewed as representative of the previous schemes which solely focus on either cost minimization (e.g., [12, 96, 97]) or carbon footprint minimization (e.g., an algorithm in [46]). The carbon footprint target of the cloud, i.e., Ψ, is clearly a value between the carbon emissions achieved by MinCost and MinCarbon solutions for feasibility assumption. We also perform a comprehensive study to evaluate the impact of the prediction error on the solutions and the way that stochastic optimization can help to remove its harmful impact. Finally, we evaluate the convergence of the proposed distributed algorithm of OnCMCCLyp.

7.5.1 Cost Efficiency of the Solutions

In this experiment we focus on cost minimization for the case where there is no carbon capping requirement and that the lookahead data is accurately available



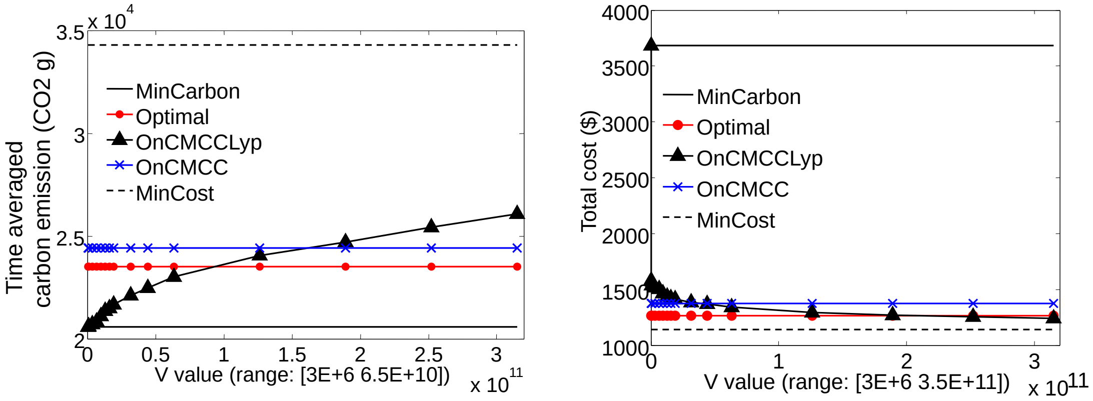
(a)

(b)

Figure 7.2: Total cost of OnCMCC versus Optimal and prediction window length (T): (a) cost versus ESD capacity, and (b) cost versus the magnitude of stipulated power, p_0 (p_0 is calculated as the percentage of the per-slot maximum power consumption of data centers).

over T. In order to run Optimal solution in a reasonable time, we use only three data centers (i.e., DC1, DC2, and DC3). First, we wish to evaluate OnCMCC against Optimal versus ESD capacity. We hypothesis that, OnCMCC, leveraging daily variabilities of data center parameters, achieves near Optimal performance in terms of cost (sum of energy cost, peak power cost and ESD operational cost) even for the case of large ESD capacities. First, we fix $p_{i,0}$ of each data center i to 80% of the its per-slot maximum power consumption. Then we vary E_i, ESD capacity of each data center i from 0 to a value that the full ESD can fully power the data center for 15 minutes, 30 minutes, an hour, two hours and 16 hours. Existing UPSes typically can power data centers for about 10-15 minutes. Data centers tend to use larger ESDs for energy cost management. However, an ESD that can power the data center for 16 hours is an extreme case and is used to evaluate the worst-case efficiency of OnCMCC. The results, shown in Fig. 7.2(a), indicates that OnCMCC for T=24 achieves a cost very close to that of Optimal. Interestingly, in agreement with our initial hypothesis, its cost increase compared to Optimal is negligible for the case of 16 hours ESD capacity. However, the cost increase of OnCMCC against Optimal is high for T=6 and T=12, as they cannot leverage the daily variabilities of the data center parameters.

Next, we evaluate OnCMCC versus different magnitude of peak power (p_0). The magnitude of p_0 is typically such that such a power consumption rarely happen in data centers. Global workload management, however, utilizes data centers near to their maximum capacity whenever they offer low electricity cost. Therefore, it is important to evaluate OnCMCC under a range of p_0. We hypothesis that if p_0 is chosen reasonably large (e.g., larger than the average data center power consumption), OnCMCC can achieve near Optimal cost. We fix the ESD capacity of all data centers such that ESDs can sustain the data centers for an hour. Similar to the previous experiment, we adjust the magnitude of p_0 as percentage of data center per-slot maximum power. Results, shown in Fig.7.2(b), in agreement with the hypothesis, indicate that when p_0 is relatively large (i.e., p_0 is greater than 65% of data centers maximum power), OnCMCC for T=24 can competitively manage the cost versus Optimal. Note that the workload intensity in our simulation setup is such that at the peak all data centers are fully utilized. Also according to our data set each data center around 20% of slots need a power equal or greater than that of 65% of the data center maximum power. Given that data center operators choose the magnitude of p_0 based on how frequent such a power consumption is required, a stipulated power equal to 65% of the maximum power is a pessimistic example and is used to show the performance of OnCMCC in the worst case. When p_0 is relatively small, the Optimal solution leverages the workload variabilities across days as well as the workload variabilities within days to smoothen the power, whereas OnCMCC with T=24 is only capable of leveraging the daily variabilities. As a result, OnCMCC for



(a) (b)

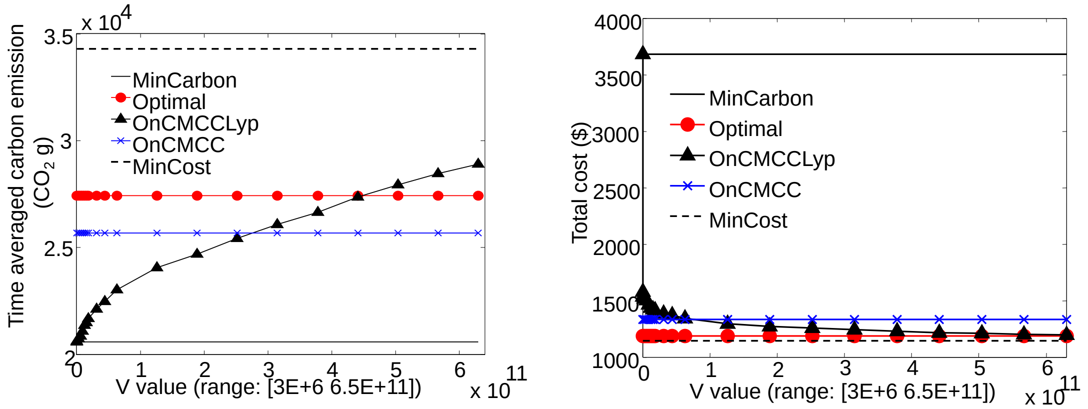
Figure 7.3: Performance of OnCMCCLyp versus Optimal and OnCMCC for Various V Values for the of Tight Cap (Ψ is small) (a) Time-averaged Carbon Footprint, (b) Total Cost.

small stipulated power, incurs a cost significantly higher than that of Optimal. For the similar reasons, OnCMCC for T=12 and T=6 incurs higher cost than Optimal even for stipulated power of greater than 80% and 70% of data center maximum power, respectively.

Summary of the Results the results show that larger the value of T, the closer the performance of OnCMCC becomes to that of Optimal. A daily basis T (T=24) can competitively manage the cost compared to Optimal even for large ESDs as long as p_0 is relatively large.

7.5.2 Cost and Carbon Footprint Efficiency of the Solutions

We evaluate our holistic solutions to jointly minimize, the electricity cost, the peak power cost while satisfying the carbon capping requirement of the cloud. Accordingly, we fix the ESD capacities such that ESDs can sustain their associated



(a) (b)

Figure 7.4: Performance of OnCMCCLyp versus Optimal and OnCMCC for Various V Values for the case where Ψ is equal to the mean carbon footprint of MinCost and MinCarbon: (a) Time-averaged Carbon Footprint, (b) Total Cost.

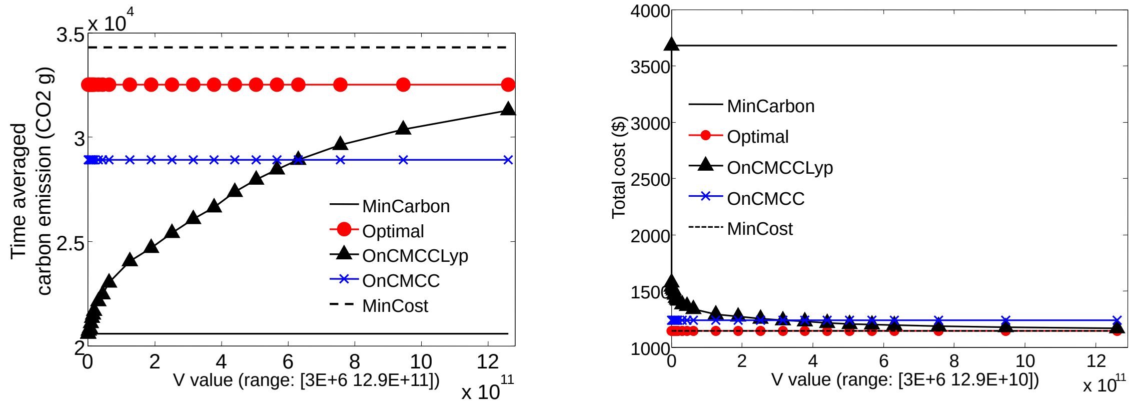
data centers for an hour, fix the stipulated power to account for 80% of the data centers per-slot maximum power consumption. Similar to the previous section we run the experiments using only three data centers, i.e., DC1, DC2, and DC3. Following the results from the previous section we also fix T as T=24. We run MinCost and MinCarbon and perform some experiments to evaluate OnCMCC and OnCMCCLyp for three values of the cap, Ψ, and various values of V, the Lyapunov control parameter (in the case of OnCMCCLyp solution). First, we set Ψ to a value very close to the carbon footprint achieved by MinCarbon. This is an example of the case where the cloud is associated with a tight cap. Results, shown in Figs. 7.3(a) and (b) show that OnCMCC fails to meet the cap, whereas OnCMCCLyp meets the cap for V values less than 1.5×10^{11} (see Fig. 7.3(a)).

Interestingly, Fig 7.3 (b) shows that for V values in the range $[0.5\times10^{11}\ 1.5\times10^{11}]$, OnCMCCLyp yields lower carbon footprint and achieves lower energy cost (up to 7.5% lower cost) than that of OnCMCC. In particular, for a V value around 1.3×10^{11}, On-

CMCCLyp performs very close to Optimal in terms of minimizing cost (sum of the electricity cost, the peak power cost and ESD cost) while satisfying the cap. Since OnCMCC independently manages the carbon footprint across T frames, it cannot opportunistically leverage the ups and downs of the cloud carbon footprint and the energy cost to optimally manage the two. OnCMCCLyp, however, takes the dynamics of the cloud carbon footprint into account and achieves a performance near to Optimal when V is appropriately adjusted. The cost competitiveness of OnCMCCLyp against Optimal also comes from the daily basis T, where OnCMCCLyp leverages the daily variability of the workload and the electricity price along with energy buffering to smoothen the peak power.

Second, we set Ψ to the mean carbon footprint of MinCost and MinCarbon. Results, shown in Fig. 7.4(a) and (b), indicate that OnCMCC, in this case, achieves a lower carbon than that of Optimal, albeit at the expense of increasing the cost by 10%. OnCMCCLyp, however, for V values less than 4.5×10^{11} meets the cap. Similar to the previous case, OnCMCCLyp, when run with appropriate V value, outperforms than OnCMCC and achieves near Optimal performance in terms of minimizing the cost (see Fig. 7.4(b) for V values in the range $[2.5\times10^{11}\ 4.5\times10^{11}]$).

Finally, we set Ψ to a value close to the carbon footprint of MinCost. This is an example of the case where the cloud's cap is loose. Results, given in Figs. 7.5(a) and (b) have similar trends to the two previous cases in the sense that OnCMCCLyp outperforms OnCMCC in minimizing cost when V is appropriately adjusted. Also in this case, expectedly, Optimal achieves a cost very close to MinCost. Further OnCMCC meets the cap, and its performance in cost minimization is close to Optimal. Note a cap equal or greater than the carbon cap of MinCost is equivalent to the case where there is no carbon capping requirement. In such a case OnCMCC is equivalent to OnCMCCLyp where the both can competitively minimize the cost against Optimal



(a) (b)

Figure 7.5: Performance of OnCMCCLyp versus Optimal and OnCMCC for Various V Values for the case of Loose Cap (Ψ is large): (a) Time-averaged Carbon Footprint, (b) Total Cost.

as shown in Section 7.5.1.

Summary of the Results The results of this section show that OnCMCCLyp achieves a near Optimal performance in minimizing cost and satisfying the cap. OnCMCC, however, independently operating on every time frame T, cannot manage the carbon footprint and achieves higher cost (up to 10% increase in the cost) than that of Optimal because of unnecessary reducing the cap. Further, it violates the cap for the cases where the cap is tight. It is worth noting that in practice, OnCMCCLyp is expected to yield higher performance against OnCMCC than what we report in this section. This is because, our experiments is for a duration of only one month, where the variability of the grid carbon intensity and the availability of the renewable energy across days is low. In practice, however, the algorithms' performance are evaluated over their long term operation where carbon intensity variation over months are huge. In this case, OnCMCCLyp, managing the dynamic of carbon footprint, is expected to significantly outperform than that of OnCMCC.

Observe that the performance of OnCMCCLyp, heavily depends on V value. In particular, as given in Theorem 7.2.1, V manages the tradeoff of cost minimization and carbon footprint capping. Appropriate V value depends on the cloud parameters e.g., electricity prices, and carbon footprint. In other words, according to Theorem 7.2.1, $\frac{B}{V}$ where $B{=}\frac{1}{2}(T^2b_{max}^2{+}T^2\psi^2)$ controls the cost optimality distance of OnCMCCLyp against Optimal. B can be estimated according to historical data, then choosing a V value comparable to the value of B gives a clue to adjust V value to both minimize the cost and meet the cap.

Although the results of Theorem 7.2.1 is based on the assumption of $T{=}S'$ (in the experiment $S'{=}S{=}168$ i.e., one month), the experimental results running for $T{=}24{<<}S'$ show that OnCMCCLyp achieves near one competitive ratio against Optimal for appropriate V value. This is because the daily basis T ($T{=}24$), leveraging the daily variabilities of the cloud can smoothen the power near to that of Optimal and competitively decrease the peak power cost (in agreement with the results of Section 7.5.1). From the results of Section 7.5.1 and this section we can conclude that OnCMCCLyp, designed based on T-slot Lyapunov optimization, is indeed effective for using as a holistic solution to manage the electricity cost, the peak power cost and the carbon capping. The results so far, however, are given for the case where the T slots future information are accurately available. Next section evaluates the solutions when using predicted data over T slots.

7.5.3 Performance of the Stochastic Optimization Solutions

We characterize $\zeta_{r,i}$, ζ_λ and $\zeta_{\alpha,i}$ through the prediction results. Then the marginal distribution of each of them, covering 90% confidence interval, is approximated to five samples each with equal probabilities. Due to their large differences, parameters' samples are normalized between zero and one. We use $\zeta_{r,i}$ to represent the aggregated prediction error of both the wind and the solar energy at each data center and ζ_λ to represent the aggregated prediction error of the workload for all front-ends (see Section 7.3). Given a random process at one stage, we construct the scenario tree over T and apply [54, Algorithm 2] to construct two reduced scenario sets: (i) S1 solely from the discrete marginal distribution of ζ_λ, and (ii) S2 from the discrete marginal distribution of ζ_λ, $\zeta_{r,i}$, and $\zeta_{\alpha,i}$. In order to run [54, Algorithm 2] in a reasonable time, we evolve the scenario trees of S1 over eight stages, and S2 over two stages. Fig. 7.6, shows that S1 and S2 capture the randomness of the predicted workload more accurately than that of the predicted renewable energy due to its high prediction error. We evaluate OnCMCC and OnCMCCLyp when using predicted data over T=24 (namely OnCMCC$_{\text{pred}}$ and OnCMCCLyp$_{\text{pred}}$) versus when using stochastic programming approach (namely OnCMCC$_{\text{stoch}}$ and OnCMCCLyp$_{\text{stoch}}$) and when using accurate data (namely OnCMCC$_{\text{opt}}$ and OnCMCCLyp$_{\text{opt}}$). We run stochastic solutions for different number of scenarios (OnCMCC$_{\text{stoch}}$ of one scenario is identical to OnCMCC$_{\text{pred}}$). In the figures we show the sum of the electricity cost and the battery cost as energy cost.

Number and Type of Scenarios First, we set the renewable energy of all data centers to zero and use S1. From the results of Fig. 7.7(a), it can be seen that the prediction error has a harmful effect on the peak power cost. In particular, while OnCMCC$_{\text{opt}}$ can manage grid power draw to avoid the peak power cost, OnCMCC$_{\text{pred}}$ with one scenario incurs \$2400 for the peak power, increasing the total cost by 66% compared to OnCMCC$_{\text{opt}}$. The total cost of OnCMCC$_{\text{stoch}}$ is decreased from 6% for 3 scenarios up to 24% for 15 scenarios compared to the total cost of OnCMCC$_{\text{pred}}$ (i.e., OnCMCC$_{\text{stoch}}$ of one scenario). This means that OnCMCC$_{\text{stoch}}$ yielding \$900 more

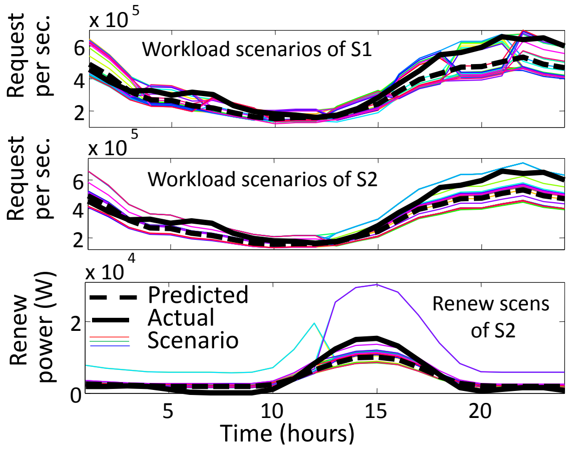
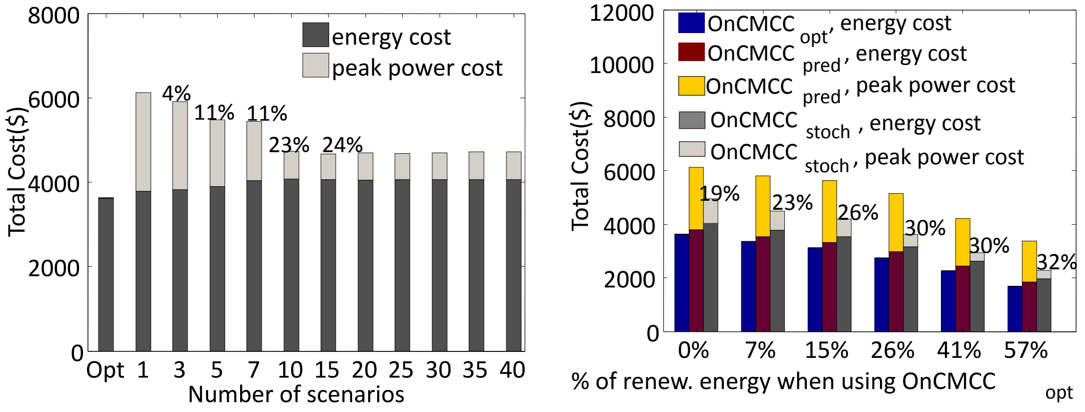


Figure 7.6: Scenario Tree of Stochastic Workload and Renewable Generation for a Sample Time Frame $T = 24$.

cost than OnCMCC$_{\text{opt}}$ (as opposed to \$2400 for OnCMCC$_{\text{pred}}$), can remove 62.5% of the harmful prediction error impact in increasing the cost. Hence, the results agree with our initial hypothesis that stochastic programming with small number of scenarios can significantly mitigate the harmful impact of the prediction error. Fig. 7.7(a) also shows that the peak power cost saving of OnCMCC$_{\text{stoch}}$ with multiple scenarios, compared to its deterministic counterpart (OnCMCC$_{\text{pred}}$), comes at the expense of a slightly increase in the energy cost. Further, the performance of OnCMCC$_{\text{stoch}}$ does not improve when number of scenarios increases beyond 15. Note that stochastic programming does not guarantee an optimal performance, and its performance heavily depends on the problem, the predicted error magnitude, and the scenarios.

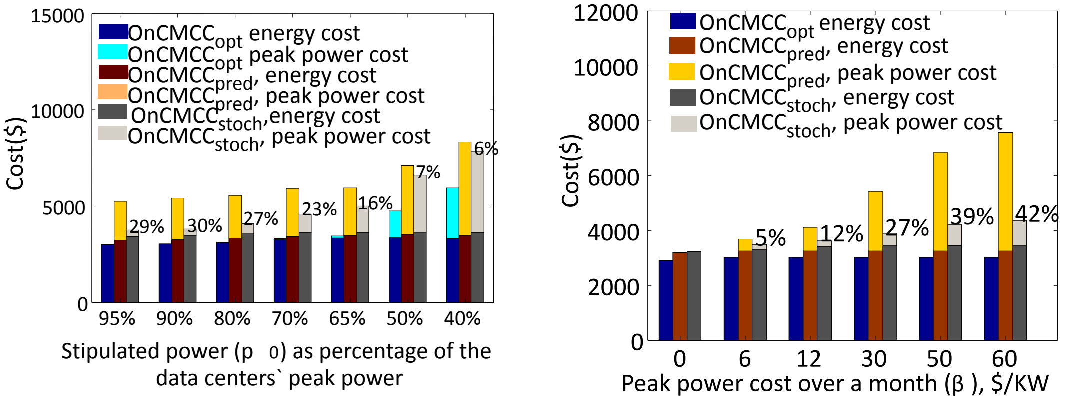
Next, we fix the number of scenarios of S2 to 15, and scale the renewable energy of DC1, DC2, and DC3 such that the total renewable energy utilization of the cloud varies from 0% to 57% when using OnCMCC$_{\text{opt}}$. Results, as shown in Fig. 7.7(b), similar to that of Fig. 7.7(a), indicates that OnCMCC$_{\text{stoch}}$ when using S2 significantly removes the impact of the prediction error of the workload, the electricity prices, and the renewables (removing 66% and 89% of the prediction error



(a)

(b)

Figure 7.7: Total cost of OnCMCC$_{\text{stoch}}$ versus OnCMCC$_{\text{opt}}$ (Opt) (the cost savings are calculated with respect to one scenario case, i.e., OnCMCC$_{\text{pred}}$) and : (a) Number of S1 Scenarios (zero renewable energy), and (b) using 15 Scenarios of S2 and Various Renewable Energy Utilization.



(a)

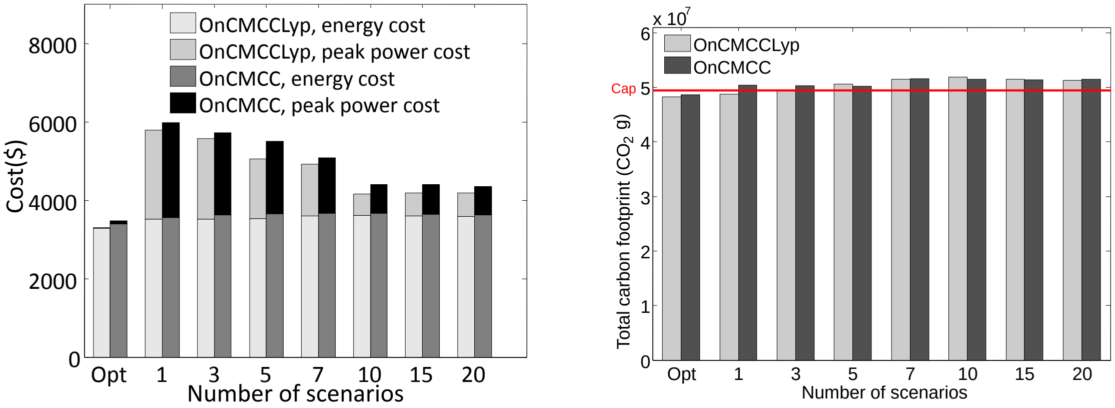
(b)

Figure 7.8: Total Cost of OnCMCC$_{\text{stoch}}$ versus OnCMCC$_{\text{opt}}$ (Opt) when using 15 Scenarios of S2 (the cost savings are calculated with respect to one scenario case, i.e., OnCMCC$_{\text{pred}}$) versus (a) Various Stipulated Power (p_0), and (b) Various Peak Power Cost (β).

impact for 15% and 57% renewable energy utilization cases, respectively). The less scenario coverage of S2 for the predicted workload compared to that of S1, causes the performance of OnCMCC$_{\text{stoch}}$ to downgrade by 5% (compare 24% cost saving of OnCMCC$_{\text{stoch}}$ in Fig. 7.7(a) with 19% in Fig. 7.7(b) for the case of 0% renewable utilization). The cost saving of OnCMCC$_{\text{stoch}}$ increases compared to its deterministic counterpart (OnCMCC$_{\text{pred}}$) with increasing the availability of the renewable energy. This is because taking the randomness of the renewable and workload prediction error into consideration results in higher utilization of the renewable energy and consequently decreasing the cost. The impact of such a management is higher for the higher availability of the renewable energy.

We also evaluate the performance of OnCMCC$_{\text{stoch}}$ (when using S2 with 15 scenarios and 15% renewable energy utilization case) for various stipulated peak power (p_0) and peak power cost (β). Fig. 7.8(a) shows that the cost saving of OnCMCC$_{\text{stoch}}$ against OnCMCC$_{\text{pred}}$ is higher for higher stipulated power where stochastic scenarios can significantly affect the decisions. Fig. 7.8(b) indicates that the cost saving of OnCMCC$_{\text{stoch}}$ against OnCMCC$_{\text{pred}}$ is higher for higher β. Generally, OnCMCC$_{\text{stoch}}$ incurs very similar expected electricity cost to that of OnCMCC$_{\text{pred}}$, this is the reason that OnCMCC$_{\text{stoch}}$ has a total cost almost equal to that of OnCMCC$_{\text{pred}}$ for the case where β=0. With increasing the peak power cost the impact of prediction error on increasing the peak power cost of OnCMCC$_{\text{pred}}$ is worsen which can be mitigated using OnCMCC$_{\text{stoch}}$.

Finally, we set the carbon cap, Ψ, to the mean carbon footprint of MinCost and MinCarbon and run OnCMCCLyp$_{\text{stoch}}$ with appropriate V value over different number of scenarios of S2. The results, as shown in Fig. 7.9(a), have a similar trend to those of the previous results (e.g., Fig. 7.7(a)) in the sense that the stochastic programming solutions (OnCMCC$_{\text{stoch}}$ and OnCMCCLyp$_{\text{stoch}}$), significantly re-

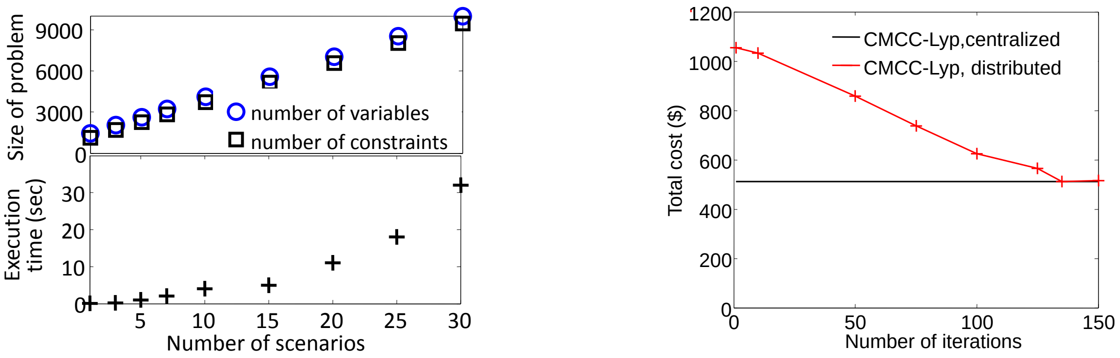


(a) (b)

Figure 7.9: Total Cost of OnCMCCLyp$_{\mathrm{stoch}}$ and OnCMCC$_{\mathrm{stoch}}$ versus OnCMCC$_{\mathrm{opt}}$ (Opt) and Number of Scenarios of S2: (a) Total Cost, and (b) Carbon Footprint.

move the impact of the prediction error, improving the cost of OnCMCC$_{\mathrm{pred}}$, and OnCMCCLyp$_{\mathrm{pred}}$ up to 30% by using ten scenarios (removing the impact of the prediction error by 66%). This cost saving comes at a slightly energy cost increase as shown in Fig. 7.9(a) and consequently a slightly carbon footprint increase as shown in Fig. 7.9(b).

Overhead of the Stochastic Solution The cost efficiency of the stochastic programming solutions comes at the expense of increasing the size of the optimization problems. As a result, the execution time of the solutions increases depending on the computing system's capability. Fig. 7.10(a) shows that the size of the optimization problem of OnCMCC$_{stoch}$ linearly increases with increasing the number of scenarios in terms of both the number of decision variables and the number of constraints. This translates into the exponential increase in the execution time of the solution in our testbed (Intel Quad core i7-3770 CPU 3.4GHz, and 8G memory). Therefore it is important to run the stochastic solutions with small number of scenarios and an



(a) (b)

Figure 7.10: (a) Solution Overhead of OnCMCC$_{stoch}$ versus the Number of Stochastic Scenarios when using S1 , and (b) Convergence Rate of OnCMCCLyp when Solved via Algorithm 7.3.

efficient implementation.

Summary of the Results The main conclusion from the results is that stochastic programming approach is indeed effective in reducing the harmful impact of the prediction error in our holistic solution. Also given large number of parameters, and uncertainties the solution run into curse of dimensionality, therefore it is needed to run the solution under a compact scenarios to represent uncertainties. The results agree with the main hypothesis that we can mitigate the impact of the prediction error by using small number of stochastic scenarios (removing the prediction error impact in increasing cost up to 66%). Further, since in practice it is required to choose only a small number of scenarios out of billions of the total scenarios, it is important to deploy some problem specific strategies to choose the right reduced scenario set. Finally, an efficient implementation, similar to our proposed distributed implementation, is required to handle the overhead of the stochastic solutions.

7.5.4 Convergence of the Distributed Algorithm

We use MATLAB quadratic programming (quadprog and interior-point-convex) to simulate Alg. 7.3. Further, we initialize the penalty term (i.e., ρ) to 0.01. Next we run OnCMCCLyp for 120 number of slots in centralized way (using linear programming solver) and in distrusted way using Alg. 7.3 (the algorithm is implemented sequentially). The results shown in Fig.7.10(b) demonstrate that the distributed ADMM based algorithm converges quickly. Since it can be implemented in parallel, it is well suited for large-scale global workload management algorithms with many data centers, front-ends and stochastic scenarios.

7.6 Summary

We proposed a holistic global workload management solution, which jointly minimizes data centers operational cost (including peak power cost), while satisfying the carbon capping requirement of the geo-distributed data centers. Data centers spend significant cost for their power infrastructure and for their peak power draw, therefore it is important to take the peak power cost into consideration when designing a global workload management. We formulated such a problem as linear programming. Peak power cost management, energy buffering and carbon capping all introduce time coupling in the solution. We developed an online algorithm OnCMCCLyp which (i) leverages (daily) predictability of data center input parameters to efficiently manage energy storage dynamics and to smoothen the power draw from the grid, and (ii) uses T slot Lyapunov optimization to manage the cost carbon footprint tradeoff. Our trace based study shows that our T-slot Lyapunov based solution, OnCMCCLyp can achieve near one competitive ratio with respect to the optimal offline solution when the Lyapunov control parameter is appropriately adjusted, T is sufficiently large and data over T is accurately available. However, the prediction error of the parameters over T slots has a very harmful impact on the peak power shaving and consequently on the cost efficiency of the solution. Our proposed stochastic programming approach is shown to remove up to 66% of such an impact as demonstrated in our trace based study. Global workload management operating across data centers need to maintain the confidentiality of data centers as much as possible. Also it becomes a large scale problem in practice depending on the number of data centers, front-ends, and scenarios of the stochastic programming based solution. We relied on the ADMM algorithm and designed a distributed algorithm which is shown to maintain the scalability and confidentiality requirements and converge in tens of steps.

Chapter 8

CONCLUSIONS AND FUTURE WORK

8.1 Discussion and Conclusions

We discuss how the solution of this dissertation help data centers to achieve sustainability, how the solutions can be utilized in practice and how effective they are.

8.1.1 Holistic View of the Solutions

This dissertation presented several server and workload management schemes at the local and the global data center level. The solutions, when integrated, result in an integrated energy and cost management architecture suited for Internet Data Centers (IDCs).

TACOMA, an energy management solution at the local data center level, distinguishes two functionalities: (i) deciding on which servers are needed during a small period to serve the predicted workload, and (ii) how to dynamically distribute the workload among the selected resources as it arrives. The first functionality is known as *server provisioning*, which is largely equivalent to *dynamic consolidation plus power management.* The second functionality is *workload distribution* (or *dispatching*).

The global cost and carbon aware workload distribution schemes determine the request redirection mechanism policies in order to distribute requests across data centers and through leveraging the inherent variabilities of several aspects (e.g., electricity cost, available renewable energy, carbon footprint) in participating data centers of a cloud. We design the global workload management schemes for different data center operational energy cost models. The solutions account for a holistic solution that combines various data center energy and power operational cost models, those being time-varying electricity cost, peak power cost and the battery cost, along with the cloud's carbon capping requirement. The holistic solution introduces new challenges which have been successfully addressed. This is performed using a combination of window based predictive approach to smoothen the peak power draw and manage the dynamics of the energy storage device in data centers, Lyapunov optimization to mange the dynamics of the carbon footprint over a long operation of the data centers in an online way, stochastic programming based optimization to remove the harmful impact of the prediction error on the peak power shaving, and distributed implementation of the algorithm to improve the scalability of the solution.

The proposed solutions at the local and the global data center levels can be integrated, similar to the model in Section 6.3. In other words, TACOMA runs locally at each individual data center, whereas solutions such as OnlineCC, run globally (in a distributed or central implementation model) and decide on the workload distribution policies taking into account the energy consumption of data centers dictated and managed by TACOMA. This integration comes at the expense of increasing the complexity of the global workload management solution, depending on the data center energy consumption model when running TACOMA (e.g., the impact of TACOMA on data center PUE).

The solutions at both the local and the global data center levels aim to infuse an elastic behavior to IDCs, providing mechanisms to adjust the energy consumption, electricity cost, carbon footprint, peak power to the given workload at a given time. The achievable energy and cost savings of the solutions come from the synergy between predictive modeling (e.g., TACOMA and OnlineCC work based on one slot ahead predicted future information, whereas OnlineCC-T and OnCMCCLyp work based on T slot ahead predicted information), the ability to individually shut down unused resources, the ability to redirect the requests, and the ability to charge and discharge energy storage devices in a data center.

Carbon capping: OnlineCC is guaranteed to achieve a near optimal performance (i.e., optimal offline solution) in minimizing energy cost while bounding the carbon cap violation. When integrated with TACOMA, OnlineCC can further decrease the cost without affecting the total carbon footprint due to the increased energy efficiency by TACOMA.

8.1.2 Practical Considerations

All of the solutions are sensitive to the *predictive accuracy* of the models, and on the *ability* to sense and implement their functionality.

1. *Predictive accuracy of models.* The proposed solutions such as TACOMA heavily rely on several models and methods, including traffic prediction, performance estimation, heat recirculation modeling and power consumption modeling. As the solutions needs to make dynamic, regular and frequent decisions, the models and prediction methods have to be very lightweight. The models presented in this research (e.g. Kalman filtering for traffic average, linear scaling of average to estimate mean, use of heat recirculation matrix to compute the temperature distribution at the servers' air inlet, and queuing theory to estimate the average response time) have been carefully selected to meet the accuracy and speed objectives. Nevertheless, if the solutions are viewed as an abstract architecture, it is easy to see that several models, e.g., prediction models can be flexibly replaced with others that meet stricter criteria (on either time or accuracy). Although high curacy of the prediction techniques generally improve the performance of the solutions, its impact on the peak power shaving is huge. Since in practice, it

is inevitable to have non-zero prediction error for parameters such as workload and renewable energy, we design a solution based on stochastic programming to mitigate and remove the impact of the prediction error (see Chapter 7). The performance of this solution heavily depends on the stochastic scenarios which represent the possible future realization of the prediction error. Therefore, it is important to build and use prediction models with high accuracy.

2. *Implications on implementation.* These implications are mainly on the impact of frequent switching servers on and off, CRAC control and monitoring, energy storage charging/discharging and request redirection mechanism.

    - *Feasibility of sensing and actuating*: All solutions require the ability to sense the various cyber parameters (e.g., workload intensity and utilization levels) and physical parameters (e.g., temperatures and power consumption), and also to actuate the various decisions, e.g., to power down certain servers. Modern computing servers are equipped with built-in temperature and power sensors, and provide control interfaces, e.g., ACPI (`http://www.acpi.info/` and IPMI (`http://www.intel.com/design/servers/ipmi/index.htm`); modern CRACs have optional modules for remove monitoring and control over the network (network thermostats); modern power distribution units (PDUs) have similar networked capabilities for individual monitoring and control of their power plugs. Further, OnCMCCLyp requires to dynamically adjust the charging/discharging of the energy storage devices such as (UPSes), which has been throughly studied and tested in the related work [51, 52]. Other cyber related parameters such as workload, and performance models have to be collected using system resource utilization monitoring tools.

- *Cost of switching*: Generally, suspending the servers can cause overhead in two ways: (i) affect the response time of workload due to resume time, and (ii) incur extra power consumption due to frequent switching. The proposed solutions are proactive scheme that activates the required servers before they are needed, thusly eliminating the performance concerns. Switching a server is an action that happens once in an epoch ($\sim$1hr); modern systems are manufactured to support "deep sleep" modes that can go back to an active state rather quickly without adversely affecting the expected remaining lifetime of the equipment.
- *Compatibility with performance metrics*: The performance (QoS) of IDCs is typically defined as some metric on delay, e.g. 95^{th}-percentile delay. We believe that the delay metric used in this research, i.e., the expected delay at peak traffic, is compatible with the ones widely used. This is because traffic peaks account for less than 5% of the traffic, thus the presented optimization satisfies the 95th-percentile. Moreover, it is possible to introduce performance margins to the active server set calculation in order to satisfy arbitrary performance constraints.
- *Request redirection mechanism*: The load balancing policies of the global workload management solutions such as OnlineCC can be achieved with protocol-level mechanisms which are in use today such as dynamically generated DNS responses, HTTP redirection and the forwarding of HTTP requests. All of these have been evaluated thoroughly [33, 79, 95].

8.1.3 Value of the Solutions

The value of the proposed solutions are governed by three criteria: *usability*, *validity* and *effectiveness*.

1. **Usability**. Much of the usability has to do with the *implementation limitations*, as discussed in section "Practical considerations", and with low running time. The time complexity analysis of the solutions i.e., greedy solutions to TACOMA, and DAHM and the online solutions of OnlineCC, OnCMCCLyp, suggests a polynomial ruining time for all of the solutions. Further, the simulations running on MATLAB and a desktop computer shows a running time of at most a few seconds for all of the solutions. OnCMCCLyp when implemented using stochastic programming, however, run into "curse of dimensionality" problem with increasing number of stochastic scenarios (see results of Section 7.5.3 and Fig. 7.10). Therefore, it is important to use small number of scenarios and implement the solution using distributed algorithm as proposed in Algorithm 7.4 in Section 7.3.

2. **Validity**. To show the validity of the solutions we perform exhaustive simulation studies using real-world traces (e.g., Microsoft hotmail and NASA workload traces, realistic electricity prices, and renewable energy traces). We also perform a small scale experimental study to further validate the results of OnlineCC solution.

3. **Effectiveness**. The solutions are effective if they *yield considerable energy/cost/carbon footprint savings* for realistic parameters.

    - Energy management at the local data center level, TACOMA: As we saw in Section 4.1.2, the effectiveness of TACOMA with respect to conventional methods depends on the thermal efficiency of the data center and on the energy proportionality of its servers. In Fig. 2.4, we observe that servers have low IPR, but no less than 0.2, and they have a varying LDR. Low IPRs favor the occurrence of cooling-computing power trade-off, whereas a varying LDR pushes toward heterogeneity of system's peak efficiency; both of these factors necessitate the use of thermal-aware management, even at low data center PUEs. For example, in Fig. 4.11, the data center has an initial PUE of about 1.4; this is yielded from dividing the leftmost column's height (∼19 GJ) over the computing-only height (∼14 GJ). For that kind of data center energy efficiency, TACOMA manages to yield over 31% energy savings, compared to 23% savings of non-thermal aware sever consolidation (our reference solution, CPSP). These results are complementary to our analytical results which ensure the performance of TACOMA under general cases of data center conditions and technology trend. Our analytical results proves the necessity of thermal awareness depending on PUE and IPR of data centers, derives an easy-to-use mathematical method to check the necessity of thermal awareness, proves NP hardness of the thermal aware server management schemes and the performance of the developed greedy solution with respect to the optimal solution.

- Operational energy cost and carbon footprint management at the global data center level (DAHM, OnlineCC, and OnCMCCLyp): The electricity cost saving of the schemes depend on the variability of electricity cost over the data centers' locations. Our study considers the electricity variation of 3-6 states inside USA and showed an electricity saving of around 30% against a performance orientated cost oblivions workload distribution (e.g., see results in Section 5.3). Further, OnlineCC, is shown to outperform a heuristic predictive scheme in cost shaving by 18% while resulting in an equal or a smaller carbon footprint (Section 6.4). Furthermore, our holistic global workload management (OnCMCCLyp) achieve near offline optimal performance in minimizing the peak power cost, and the electricity cost while satisfying the clouds carbon capping , when the lookahead information is accurately available. Prediction error, however, increases the cost up to 45%, which can be removed (about 66% improvement) when using our proposed stochastic programming based solution (Section 7.5). These trace-based simulations results complement the analytical results ensuring the performance of the solutions with respect to the optimal solutions. In summary, our analytical study accounts for the NP-hardness proof of the cost minimization problem and a greedy solution for solving it (Section 5.1.1 and Section 5.2). Our results on joint optimization of cost minimization and carbon footprint capping proves the performance of the online solutions (e.g., OnlineCC and OnlineCC-T) for stationary and non-stationary parameters of data centers, estimates maximum carbon capping violation of the online solutions (Section 6.2), and suggests a distributed implementation of online solutions with proven linearly convergence (Section 7.4).

8.2 Future Work

Future work improves the research results by relaxing the assumptions under which the study is performed, and adopting new approaches for ease of implementation and improving the effectiveness of the solutions in practice.

8.2.1 Relaxing the Assumptions

The research is performed under some assumptions on the applications characteristics and requirements, underlying infrastructure, and models to characterize cyber and physical aspects of the data centers. Future work would extend and enhance the proposed work by relaxing such assumptions to improve their usability, functionality and effectiveness in practice.

Heterogeneous applications

The solutions of this research solely account for delay-sensitive Internet applications. The assumption is that either the entire data center or some clusters in data centers dedicatedly provide Internet services. However, in data centers, there exist a vast amount of delay-tolerant jobs, such as background/maintenance jobs. The existing energy and cost management solutions also mainly focus either solely on Internet applications or solely on delay-tolerant batch applications. In particular, a thermal aware scheduling for a data center with mix of delay tolerant and delay sensitive applications has yet to be addressed throughly. Similarly, a global cost and carbon aware workload management accounting for both delay-sensitive and delay-tolerant jobs needs to be studied. The potential effectiveness of such schemes are as follows. First, the proposed server management solutions performs workload-proportional resource provisioning schemes, where servers are turned on/off according to the load of requests. However, delay-tolerant jobs can be opportunistically scheduled to fill the extra capacity of data centers, when doing so is more cost efficient (e.g., avoiding frequent switching of the servers). Second, such solutions are consistent with the functionality of some data centers that do not dedicatedly allocate some clusters to Internet applications. Due to their high migration overhead, data centers unlikely to consider global scheduling for bath jobs as a cost efficient option. However, batch jobs when jointly managed with Internet applications, creates many carbon and cost management possibilities in global workload management. The idea is to scale up and down the available data center capacity considering its local background jobs and its global Internet requests.

Also, this research assumes a single Internet application type. In practice, data centers host different Internet applications. In this regard, energy and cost management solutions would be designed to consolidate different applications over the cloud to optimize energy and cost. Applications can be assigned to VMs, which can be placed at the most cost efficient data center at a give time. However, consolidation of different set of applications comes with interferences amongst them. Existing work suggest that consolidation of applications in a single server increases the contention on the shared resources such as on-chip caches, buses, main memory, CPUs and network [80, 84]. This contention results in performance degradation of applications. The performance overhead due to contention depends on the workload type of applications. The contention can also cause energy consumption overhead due to increase in the runtime. Modeling such an effect and incorporating into energy management solution is not an easy task. The interference effect depends on the workload type of applications and the workload intensity which are not easy to quantify and model for the scale of data centers [80].

Implementation and Underlying Infrastructure

This research is performed under the assumption that the underlying infrastructure is virtualized, meaneing that every server is capable of running the application. Such assumption is consistent with the large-scale Internet applications infrastructure. To enhance the functionality of the proposed schemes, they need to be implemented in widely used data center software management tools and should be provided with solutions to deal with practical challenges associated with every management kit. While there exist plenty of software management frameworks for big data analysis in data centers, e.g., Hadoop and cloud management e.g., OpenStack, their scheduling algorithms are usually preliminary in the sense that they trade energy consumption for achieving guaranteed quality of service, and swiftness.

Evaluation

The evaluation of the solutions are performed using small scale experiments and some assumption on data center power infrastructure (e.g., any node can be turned off at any time without considering the physical data center power infrastructure). The small scale experiments performed in this dissertation only provides proof of the concept, and needs further investigation (implementing and testing in moderate-large scale experiments) in order to build elegant solutions for real world problems. For further validation, the future work becomes to implement and test the solutions on experimental data centers such as BlueCenter, a small data center which is part of the BlueTool[1] [57], a project with the purpose to provide an infrastructure for research on data centers.

Models

This study incorporated steady state models of data centers' dynamics i.e., workload, renewable energy sources, and data center thermal conditions over an epoch/slot. However, all of these parameters exhibit temporal fluctuation. Ignoring such temporal aspects can potentially affect the energy savings projected by the proposed solutions. Due to the management overhead of the proposed solutions, it is infeasible to choose very short decision time interval. However, initial transient analysis of data center parameters can help to optimally choose the solution control parameters such as slot length and prediction window to minimize such problems. Further, most of the theoretical results are derived under the assumptions of linear servers power consumption models and linear performance models. The future work would extend the results by incorporating more general models.

[1] http://impact.asu.edu/BlueTool/

8.2.2 Adopting New Approaches

The future work would explore new approaches to improve the effectiveness of the solutions in practice. New approaches would address the followings.

Thermal-aware Algorithms

This study provides approximation algorithm with provable guarantee for thermal aware scheduling of the homogeneous data center case i.e., a data center with homogeneous servers in terms of computing capabilities and power consumption (Section 4.4.1). However, the theoretical approximation ratio is in the order of number of servers in the cluster which can be a large number. Although the simulation study using real-world traces showed a near optimal performance of the proposed solution (i.e., approximation ratio of 1.18), future work is required to improve the theoretical results and to ensure the performance of the solution for any instance of data center parameters. For the heterogeneous data center case (i.e., a data center with heterogeneous servers in terms of computing capabilities and power consumption), the proposed scheme is a heuristic solution (Section 4.4.2). Devising a solution with provable guarantee for the heterogeneous data center case is left for future work.

Online Carbon Capping

Online carbon capping is a crucial management scheme which helps data centers to achieve carbon neutrality in a cost efficient way. Our exhaustive trace based simulation study shows that, the proposed online carbon capping solution, OnlineCC, achieves near one competitive ratio in optimizing cost and capping carbon footprint with respect to the offline optimal solution when the Lyapunov control parameter is optimally adjusted. In particular, it achieves within $O(1/V)$ of the optimal solution cost and $O(V)$ of the optimal solution carbon footprint. Although the proposed heuristic solution significantly reduces the search space to adjust the Lyapunov control parameter (Section 6.2.1), its optimal adjustment requires a number of trials. Further, the efficiency of the proposed heuristic solutions depends on how big the ratio of the peak electricity price over the average electricity price is. Therefore, in general, due to the variability of input data parameters, the task of deciding the right value for the Lyapunov control parameter is tedious and demands more investigation. One potential solution is to extend Lyapunov optimization to automatically and dynamically decide on the value of its control parameter. This has to be done without compromising its performance both in theory and in application.

Further, the future work would improve the competitiveness of the Lyapunov optimization solution for cost and carbon aware workload management problem. Particularly, the existing work devised Lyapunov optimization solution that manages the tradeoffs of the competitive factors for wireless network data transfer (i.e., energy delay tradeoff) in logarithmic order [91] (e.g., achieving $O(V)$ in optimizing energy and $O(logV)$ in optimizing delay with respect to offline optimal solution). A cost and carbon aware workload management algorithm which provides a tight competitive bound in optimizing the electricity cost and the carbon footprint, results a tight bound on the algorithm's worst case carbon capping violation.

Other future work is to adopt other online algorithms which are easy to implement while providing provable competitiveness against the offline solution.

Online Peak Power Management

Peak power management of data centers, as discussed in Chapter 7, is very sensitives to the parameters' prediction error. This study adopted stochastic programing in order to remove the harmful impact of the prediction error. The experimental results show that the proposed solution removes upto 66% of the harmful prediction error impact in increasing the cost. Designing a solution to entirely remove the impact of the prediction error is left for future work. Future work would also adopt new approaches to design competitive online algorithms which do not require the predicted data.

Renewable Energy Prediction

The results of this study show that the peak power shaving and the energy buffering, being effective in reducing the data centers' operational cost, can be best managed when leveraging the future information within a time frame (e.g., 24 hours). However, the performance of such a solution heavily depends on the prediction accuracy of data centers' input parameters, and in particular the available on-site renewable energy sources. The proposed stochastic programming solution is shown to remove the harmful impact of the prediction error. However, its effectives heavily depends on the prediction accuracy. The number of data centers, installing on-site renewable energy sources, is expected to grow in future. In such data centers energy buffering is essential in order to smoothen the available renewable energy and maximize their utilization. The results of this study show that while workload can be predicted with reasonable accuracy (e.g., using seasonal time series), the available solar and wind energy, particularly wind energy, cannot be predicted when using a small training data set (e.g., one month training data). The related work highlighted that the prediction accuracy of both solar and wind energy significantly increases when using a very large training data set (e.g., data set of up to three years) [36]. However, such a scheme is not always an option for data centers, particularly for those who recently installed their renewable infrastructure. One potential solution would be implementing wind and solar prediction services using the available weather information for data centers.

The services should be consistent with the nature of workload management in data centers in terms of time granularity of decision intervals and the prediction window.

REFERENCES

[1] http://ita.ee.lbl.gov/html/traces.html.

[2] http://www.nrel.gov/midc/.

[3] http://gigaom.com/2012/05/08/microsoft-pledges-to-be-carbon-neutral-by-the-summer/.

[4] http://www.google.com/about/datacenters/renewable/index.html.

[5] http://www.solarbuzz.com/.

[6] http://www.dsireusa.org/.

[7] http://arstechnica.com/science/2014/09/price-slowly-rising-on-carbon-emissions-in-us-cap-and-trade-states/.

[8] Quick start guide to increase data center energy efficiency. Technical report, General Services Administration (GSA) and the Federal Energy Management Program (FEMP)., September 2010.

[9] About the carbon pricing mechanism, 2013. Clean Energy Regulator.

[10] Effects of a carbon tax on the economy and the environment, 2013. Congressional Budjet Office.

[11] Z. Abbasi, , and S. K. Gupta. Operational cost minimization and carbon footprint capping for geo-distributed data centers: Predictive approach. 2014.

[12] Z. Abbasi, T. Mukherjee, G. Varsamopoulos, and S. K. S. Gupta. DAHM: A green and dynamic web application hosting manager across geographically distributed data centers. *ACM Journal on emerging technology (JETC)*, 8(4):34:1–34:22, Nov. 2012.

[13] Z. Abbasi, M. Pore, and S. K. Gupta. Impact of workload and renewable prediction on the value of geographical workload management. In *Second International Workshop on Energy Efficient Data Centers (E2DC), held as a part of ACM eEnergy*, 2013.

[14] Z. Abbasi, M. Pore, and S. K. Gupta. Cloud load balancing towards carbon neutrality. 2014.

[15] Z. Abbasi, M. Pore, and S. K. Gupta. Online server and workload management for joint optimization of electricity cost and carbon footprint across data centers. May 2014.

[16] Z. Abbasi, G. Varsamopoulos, and S. K. S. Gupta. Thermal aware server provisioning and workload distribution for internet data centers. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 130–141, New York, NY, USA, 2010. ACM.

[17] Z. Abbasi, G. Varsamopoulos, and S. K. S. Gupta. TACOMA: Server and workload management in internet data centers considering cooling-computing power trade-off and energy proportionality. *ACM Trans. Archit. Code Optim.*, 9(2):11:1–11:37, June 2012.

[18] H. Aissi, C. Bazgan, and D. Vanderpooten. Approximating min-max (regret) versions of some polynomial problems. In *Proceedings of the 12th International Computing and Combinatorics Conference COCOON*, pages 428–438, 2006.

[19] S. Akoush, R. Sohan, A. Rice, A. W. Moore, and A. Hopper. Free lunch: exploiting renewable energy for computing. In *Proceedings of HotOS*, 2011.

[20] A. O. Allen. *Probability, statistics and queuing theory with computer science applications.* Academic Press Inc., 1990.

[21] M. F. Arlitt and C. L. Williamson. Web server workload characterization: The search for invariants. In *ACM SIGMETRICS Perf. Eval. Rev.*, volume 24, pages 126–137, 1996.

[22] A. Bar-Noy, M. P. Johnson, and O. Liu. Peak shaving through resource buffering. In *Approximation and Online Algorithms*, pages 147–159. Springer, 2009.

[23] P. Barford and M. Crovella. Generating representative web workloads for network and server performance evaluation. *SIGMETRICS Perform. Eval. Rev.*, 26(1):151–160, 1998.

[24] L. A. Barroso and U. Hoelzle. The case for energy-proportional computing. *Computer*, 40:33–37, December 2007.

[25] L. Bent, M. Rabinovich, G. M. Voelker, and Z. Xiao. Characterization of a large web site population with implications for content delivery. In *Int. conf. on World Wide Web*, pages 522–533. ACM, 2004.

[26] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 1995.

[27] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[28] N. Buchbinder, N. Jain, and I. Menache. Online job-migration for reducing the electricity bill in the cloud. *NETWORKING 2011*, 6640:172–185, 2011.

[29] R. D. Carr, L. K. Fleischer, V. J. Leung, and C. A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proceedings of the eleventh annual ACM-SIAM symposium on discrete algorithms*, pages 106–115. Society for Industrial and Applied Mathematics, January 2000.

[30] J. S. Chase, D. Anderson, P. Thakar, A. Vahdat, and R. Doyle. Managing energy and server resources in hosting centers. In *Proc. of the eighteenth ACM symposium on Operating systems principles (SOSP)*, pages 103–116, 2001.

[31] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao. Energy-aware server provisioning and load dispatching for connection-intensive Internet services. In *NSDI'08: Proc. of the 5th USENIX Symposium on Networked Systems Design and Implementation*, pages 337–350, Berkeley, CA, USA, 2008. USENIX Association.

[32] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. Managing server energy and operational costs in hosting centers. *SIGMETRICS Performance Evaluation Review*, 33(1):303–314, 2005.

[33] M. Conti, E. Gregori, and F. Panzieri. Load distribution among replicated web servers: A qos-based approach. *ACM SIGMETRICS Performance Evaluation Review*, 27(4):12–19, 2000.

[34] G. Cook and J. Van Hor. How dirty is your data? a look at the energy choices that power cloud computing, 2011.

[35] A. M. Costa. A survey on benders decomposition applied to fixed-charge network design problems. *Computers & operations research*, 32(6):1429–1450, 2005.

[36] M. G. De Giorgi, A. Ficarella, and M. Tarantino. Error analysis of short term wind power prediction models. *Applied Energy*, 88(4):1298–1311, 2011.

[37] N. Deng, C. Stewart, D. Gmach, M. Arlitt, and J. Kelley. Adaptive green hosting. In *ACM ICAC*, pages 135–144, 2012.

[38] T. T. DiCaprio. How microsoft is striving to become leaner, greener, and more accountable. *Microsoft, white paper*, june 2012.

[39] J. Doyle, R. Shorten, and D. O'Mahony. Stratus: Load balancing the cloud for carbon emissions control. *IEEE Transactions on Cloud Computing*, page 1, 2013.

[40] J. Eckstein. *Splitting methods for monotone operators with applications to parallel optimization.* PhD thesis, Massachusetts Institute of Technology, 1989.

[41] M. L. Emens, D. A. Ford, R. Kraft, and G. Tewari. Method of automatically selecting a mirror server for web-based client-host interaction, Aug. 12 2003.

[42] U. S. EPA. Report to congress on server and data center energy efficiency. `http://www.energystar.gov/ia/partners/prod\_development/downloads/EPA\_Datacenter\_Report_Congress\_Final1.pdf`, Aug. 2007.

[43] M. Etinski, M. Martonosi, K. Le, R. Bianchini, and T. D. Nguyen. Optimizing the use of request distribution and stored energy for cost reduction in multi-site internet services. In *Sustainable Internet and ICT for Sustainability (SustainIT), 2012*, pages 1–10. IEEE, 2012.

[44] A. Faraz and V. T.N. Joint optimization of idle and cooling power in data centers while maintaining response time. In *Proceedings of the fifteenth edition of ASPLOS on Architectural support for programming languages and operating systems*, ASPLOS XV, pages 243–256, New York, NY, USA, 2010. ACM, ACM.

[45] G. Forman and C. E. Bash. Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center. Technical Report HPL-2007-62, HP Laboratories Palo Alto, aug 2007.

[46] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav. It's not easy being green. *ACM SIGCOMM Computer Communication Review*, 42(4):211–222, 2012.

[47] Í. Goiri, R. Beauchea, K. Le, T. D. Nguyen, M. E. Haque, J. Guitart, J. Torres, and R. Bianchini. Greenslot: scheduling energy consumption in green datacenters. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 20:1–20:11, New York, NY, USA, 2011. ACM.

[48] Í. Goiri, K. Le, T. D. Nguyen, J. Guitart, J. Torres, and R. Bianchini. Greenhadoop: leveraging green energy in data-processing frameworks. In *Proceedings of the 7th ACM european conference on Computer Systems*, pages 57–70. ACM, 2012.

[49] L. H. Goulder and A. R. Schein. Carbon taxes versus cap and trade: A critical review. *Climate Change Economics*, 4(03), 2013.

[50] S. Govindan, A. Sivasubramaniam, and B. Urgaonkar. Benefits and limitations of tapping into stored energy for datacenters. In *Proc. The 38th International Symposium on Computer Architecture (ISCA)*, San Jose, CA, USA, June 2011.

[51] S. Govindan, D. Wang, Anand, Sivasubramaniam, and B. Urgaonkar. Leveraging stored energy for handling power emergencies in aggressively provisioned datacenters. In *Proceedings of the seventeenth international conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 75–86. ACM, 2012.

[52] S. Govindan, D. Wang, A. Sivasubramaniam, and B. Urgaonkar. Aggressive datacenter power provisioning with batteries. *ACM Transactions on Computer Systems (TOCS)*, 31(1):2, 2013.

[53] greenpeace.org. Greenpeace activists project messages on apple headquarters from supporters asking for cleaner cloud, cited Jan 2013.

[54] N. Growe-Kuska, H. Heitsch, and W. Romisch. Scenario reduction and scenario tree construction for power management problems. In *Power Tech Conference Proceedings, 2003 IEEE Bologna*, volume 3, pages 7–pp. IEEE, 2003.

[55] B. Guenter, N. Jain, and C. Williams. Managing cost, performance, and reliability tradeoffs for energy-aware server provisioning. In *Proc. IEEE INFOCOM, Shanghai, China*, pages 702–710, 2011.

[56] S. K. Gupta, A. Banerjee, Z. Abbasi, G. Varsamopoulos, M. Jonas, J. Ferguson, R. R. Gilbert, and T. Mukherjee. GDCSim - a simulator for green data center design and analysis. *ACM Transactions on Modeling and Computer Simulation*, 2014.

[57] S. K. S. Gupta, G. Varsamopoulos, A. Haywood, P. E. Phelan, and T. Mukherjee. BlueTool: Using a computing systems research infrastructure tool to design and test green and sustainable data centers. In *Handbook of Energy-Aware and Green Computing*. 1st edition, 2012.

[58] T. Jin and M. Arlitt. Workload characterization of the 1998 world. Technical report, Hewlett-Packard Labs, Sept. 1999.

[59] M. Kaut and S. W. Wallace. Evaluation of scenario-generation methods for stochastic programming. 2003.

[60] R. Kohavi, R. M. Henne, and D. Sommerfield. Practical guide to controlled experiments on the web: listen to your customers not to the hippo. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 959–967. ACM, 2007.

[61] V. Kontorinis, L. E. Zhang, B. Aksanli, J. Sampson, H. Homayoun, E. Pettis, D. M. Tullsen, and T. S. Rosing. Managing distributed UPS energy for effective power capping in data centers. In *Proceedings of the 39th International Symposium on Computer Architecture*, ISCA '12, pages 488–499, Piscataway, NJ, USA, 2012. IEEE Press.

[62] J. Koomey. Growth in data center electricity use 2005 to 2010. *The New York Times*, 49(3), 2011.

[63] J. Koomey. Growth in data center electricity use 2005 to 2010, August 1 2011.

[64] J. G. Koomey, C. Belady, M. Patterson, A. Santos, and K.-D. Lange. Assessing trends over time in performance, costs, and energy use for servers. Technical report, Microsoft Corp. and Intel Corp., August 2009.

[65] A. Krioukov, P. Mohan, S. Alspaugh, L. Keys, D. Culler, and R. Katz. NapSAC: design and implementation of a power-proportional web cluster. In *Proc.of the first SIGCOMM workshop on Green networking*, pages 15–22. ACM, 2010.

[66] S. O. Krumke, H. Noltemeier, S. Schwarz, H.-C. Wirth, and R. Ravi. Flow improvementand network flows with fixed costs. *OR 98*, 1998.

[67] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang. Power and performance management of virtualized computing environments via lookahead control. *Cluster Computing*, 12:1–15, 2009.

[68] S. Lacey. Data center efficiency may be getting worse. http://www.greentechmedia.com/articles/read/are-data-centers-getting-less-energy-efficient, April 2013.

[69] K. Le, R. Bianchini, T. D. Nguyen, O. Bilgir, and M. Martonosi. Capping the brown energy consumption of internet services at low cost. In *IEEE IGCC*, 2010.

[70] K. Le, O. Bilgir, R. Bianchini, M. Martonosi, and T. D. Nguyen. Managing the cost, energy consumption, and carbon footprint of internet services. In *Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '10, pages 357–358, New York, NY, USA, 2010. ACM.

[71] K. Le, Ricardo, B. Margaret, Martonosi, and T. Nguyen. Cost-and energy-aware load distribution across data centers. *Proceedings of HotPower*, 2009.

[72] C. Li, A. Qouneh, and T. Li. Characterizing and analyzing renewable energy driven data centers. *SIGMETRICS Perform. Eval. Rev.*, 39(1):323–324, June 2011.

[73] M. Lin, Z. Liu, A. Wierman, and L. L. H. Andrew. Online algorithms for geographical load balancing. In *Proc. of International Green Computing Conference (IGCC11)*. IEEE, june 2012.

[74] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. In *INFOCOM, 2011 Proceedings IEEE*, pages 1098 –1106, Apr. 2011.

[75] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser. Renewable and cooling aware workload management for sustainable data centers. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '12, pages 175–186, New York, NY, USA, 2012. ACM.

[76] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. H. Andrew. Geographical load balancing with renewables. *ACM SIGMETRICS Performance Evaluation Review*, 39(3):62–66, 2011.

[77] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. H. Andrew. Greening geographical load balancing. In *Proc. ACM SIGMETRICS*, pages 233–244, june 2011.

[78] A. H. Mahmud and S. Ren. Online capacity provisioning for carbon-neutral data center with demand-responsive electricity prices. *ACM SIGMETRICS Perf. Eval. Rev.*, 41(2):26–37, 2013.

[79] Z. M. Mao, C. D. Cranor, F. Douglis, M. Rabinovich, O. Spatscheck, and J. Wang. A precise and efficient evaluation of the proximity between web clients and their local dns servers. In *USENIX Annual Technical Conference, General Track*, pages 229–242, 2002.

[80] J. Mars, L. Tang, R. Hundt, K. Skadron, and M. L. Soffa. Bubble-up: Increasing utilization in modern warehouse scale computers via sensible co-locations. In *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 248–259. ACM, 2011.

[81] D. Meisner, B. T. Gold, and T. F. Wenisch. Powernap: eliminating server idle power. *SIGPLAN Not.*, 44:205–216, March 2009.

[82] D. Meisner, C. M. Sadler, L. A. Barroso, W.-D. Weber, and T. F. Wenisch. Power management of online data-intensive services. In R. Iyer, Q. Yang, and A. González, editors, *ISCA*, pages 319–330. ACM, 2011.

[83] P. Mell and T. Grance. The nist definition of cloud computing (draft). *NIST special publication*, 800:145, 2011.

[84] A. Merkel, J. Stoess, and F. Bellosa. Resource-conscious scheduling for energy efficiency on multicore processors. In *Proceedings of the 5th European conference on Computer systems*, EuroSys '10, pages 153–166, New York, NY, USA, 2010. ACM.

[85] R. Miller. Facebook installs solar panels at new data center. White Paper, april 2011.

[86] D. C. Montgomery, E. A. Peck, and G. G. Vining. *Introduction to Linear Regression Analysis*. John Wiley and Sons Inc, 2001.

[87] J. Moore, J. S. Chase, and P. Ranganathan. Weatherman: Automated, online, and predictive thermal mapping and management for data centers. In *IEEE International Conference on Autonomic Computing (ICAC)*, pages 155–164, jun 2006.

[88] J. Moore, J. S. Chase, P. Ranganathan, and R. K. Sharma. Making scheduling "cool": temperature-aware workload placement in data centers. In *ATEC '05: Proc. of the annual conference on USENIX Annual Technical Conference*, pages 5–5, Berkeley, CA, USA, 2005. USENIX Association.

[89] T. Mukherjee, A. Banerjee, G. Varsamopoulos, S. K. S. Gupta, and S. Rungta. Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers. *Computer Networks*, 53(17):2888 – 2904, Dec. 2009.

[90] M. J. Neely. Energy optimal control for time-varying wireless networks. *Information Theory, IEEE Transactions on*, 52(7):2915–2934, 2006.

[91] M. J. Neely. Intelligent packet dropping for optimal energy-delay tradeoffs in wireless downlinks. *Automatic Control, IEEE Transactions on*, 54(3):565–579, 2009.

[92] D. S. Palasamudram, R. K. Sitaramanz, B. Urgaonkar, and R. Urgaonkar. Using batteries to reduce the power costs of internet-scale distributed networks. In *Proceedings of 2012 ACM Symposium on Cloud Computing*. ACM, Oct. 2012.

[93] V. S. Pappala, I. Erlich, K. Rohrig, and J. Dobschinski. A stochastic model for the optimal operation of a wind-thermal power system. *Power Systems, IEEE Transactions on*, 24(2):940–950, 2009.

[94] L. Paroliniy, N. Tolia, B. Sinopoliy, and B. H. Krogh. A cyber-physical systems approach to energy management in data centers. In *ACM ICCPS '10*, April 2010.

[95] M. Pathan, C. Vecchiola, and R. Buyya. Load and proximity aware request-redirection for dynamic load distribution in peering cdns. In *On the Move to Meaningful Internet Systems: OTM 2008*, pages 62–81. Springer, 2008.

[96] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the electric bill for Internet-scale systems. In *Proc. ACM SIGCOMM*, pages 123–134, 2009.

[97] L. Rao, X. Liu, L. Xie, and W. Liu. Minimizing electricity cost: optimization of distributed Internet data centers in a multi-electricity-market environment. In *Proc. IEEE INFOCOM*, pages 1–9, 2010.

[98] L. Rao, L. Xue, and I. Marija. Mec-idc: joint load balancing and power control for distributed internet data centers. In *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, pages 188–197. ACM, 2010.

[99] A. Rawson, J. Pfleuger, T. Cader, and C. Belady. Green grid data center power efficiency metrics: Pue and dcie. Technical report, The green grid, 2008.

[100] S. Ren and Y. He. COCA: Online distributed resource management for cost minimization and carbon neutrality in data centers. In *Super Computing*, 2013.

[101] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy. An analysis of Internet content delivery systems. *ACM SIGOPS Operating Systems Review*, 36(SI):315–327, 2002.

[102] A. Shapiro, D. Dentcheva, et al. *Lectures on stochastic programming: modeling and theory*, volume 9. SIAM, 2009.

[103] N. Sharma, S. Barker, D. Irwin, and P. Shenoy. Blink: managing server clusters on intermittent power. *ACM SIGPLAN Notices*, 46(3):185–198, 2011.

[104] R. K. Sharma, C. E. Bash, C. D. Patel, R. J. Friedrich, and J. S. Chase. Balance of power: Dynamic thermal management for internet data centers. *IEEE Internet Computing*, 9(1):42–49, 2005.

[105] J. Si. *Handbook of learning and approximate dynamic programming*, volume 2. John Wiley & Sons, 2004.

[106] M. Stansberry. Uptime institute 2013 data center industry survey. *Uptime Institute Survey*, 2013.

[107] M. Stansberry and J. Kudritzki. Uptime institute 2012 data center industry survey. *Uptime Institute Survey*, 2012.

[108] C. Stewart and K. Shen. Some joules are more precious than others: Managing renewable energy in the datacenter. In *Proceedings of the Workshop on Power Aware Computing and Systems*, 2009.

[109] B. T OGRAPH and Y. R. MORGENS. Cloud computing. *Communications of the ACM*, 51(7), 2008.

[110] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos. Thermal-aware task scheduling for data centers through minimizing heat recirculation. In *IEEE Cluster*, pages 129–138, Sept. 2007.

[111] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos. Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach. *IEEE Transactions on Parallel and Distributed Systems, Special Issue on Power-Aware Parallel and Distributed Systems,* conditionally accepted, 19(11):1458–1472, Nov. 2008.

[112] S. P. Technologies. Sp1000w (1500w max) domestic wind turbine - datasheet. `http://www.allsmallwindturbines.com/files/SP1000W_July_2007.pdf`, 2007.

[113] E. Thereska, A. Donnelly, and D. Narayanan. Sierra: a power-proportional, distributed storage system. Technical report, Microsoft Research, 2009.

[114] B. Thrash. The green data center opportunity. http://www.datacenterjournal.com/facilities/the-green-data-center-opportunity/, Feb. 2012.

[115] B. Urgaonkar, P. Shenoy, A. Chandray, and P. Goyal. Agile dynamic provisioning of multi-tier internet applications. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 3(1):1–39, 2008.

[116] R. Urgaonkar, B. Urgaonkar, M. J. Neely, and A. Sivasubramanian. Optimal power cost management using stored energy in data centers. In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, pages 221–232. ACM, 2011.

[117] G. Varsamopoulos, Z. Abbasi, and S. K. S. Gupta. Trends and effects of energy proportionality on server provisioning in data centers. In *International Conference on High performance Computing Conference (HiPC2010)*, pages 1–11, Dec. 2010.

[118] G. Varsamopoulos and S. K. S. Gupta. Energy proportionality and the future: Metrics and directions. In *Parallel Processing Workshops (ICPPW), 2010 39th International Conference on*, pages 461–467. IEEE, 2010.

[119] R. Vokoun. Renewable energy in today's data center. White Paper, april 2012.

[120] R. Vokoun. Renewable energy in today's data center. White Paper, May 2012.

[121] D. Wang, C. Ren, A. Sivasubramaniam, B. Urgaonkar, and H. Fathy. Energy storage in datacenters: what, where, and how much? In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '12, pages 187–198, New York, NY, USA, 2012. ACM.

[122] A. Weidlich and D. Veit. A critical survey of agent-based wholesale electricity market models. *Energy Economics*, 30(4):1728–1759, 2008.

[123] D. Xu and X. Liu. Geographic trough filling for internet datacenters. In *INFOCOM, 2012 Proceedings IEEE*, pages 2881–2885. IEEE, 2012.

[124] G. Yu and P. Kouvels. On min-max optimization of a collection of classical discrete optimization problems. *Optimization Theory and Applications*, 98(1):221–242, December 1998.

[125] Y. Zhang, Y. Wang, and X. Wang. Greenware: greening cloud-scale data centers to maximize the use of renewable energy. *Middleware 2011*, pages 143–164, 2011.

[126] X. Zhou, J. Yang, M. Chrobak, and Y. Zhang. Performance-aware thermal management via task scheduling. *ACM Transactions on Architecture and Code Optimization (TACO)*, 7(1):1–31, 2010.

[127] Z. Zhou, F. Liu, Y. Xu, R. Zou, H. Xu, J. C. Lui, and H. Jin. Carbon-aware load balancing for geo-distributed cloud services. In *IEEE MASCOTS*, 2013.

BIOGRAPHICAL SKETCH

Zahra Abbasi joined the School of Computing, Informatics, and Decision Systems Engineering at Arizona State University in Spring 2009 and began her Ph.D. in Computer Science under Dr. Sandeep Gupta. She received her Bachelor's degree in Computer Engineering from Shiraz University, Iran, in 2000, and her Master's degree in computer engineering from Iran University of Science and Technology, Iran, in 2003. Her research interests include green and sustainable cloud computing, combinatorial algorithms and optimization specially in designing scheduling and resource management solutions.