Managing server energy and reducing operational cost for online service providers

Xinying Zheng
Michigan Technological University

Follow this and additional works at: https://digitalcommons.mtu.edu/etds
Part of the Electrical and Computer Engineering Commons

MANAGING SERVER ENERGY AND REDUCING OPERATIONAL COST FOR ONLINE SERVICE PROVIDERS

By

Xinying Zheng

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

(Electrical Engineering)

MICHIGAN TECHNOLOGICAL UNIVERSITY

2012

This dissertation, "Managing Server Energy and Reducing Operational Cost for Online Service Providers," is hereby approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING.

Department of Electrical and Computer Engineering

Signatures:

Dissertation Co-Advisor	____________________ Dr. Yu Cai
Dissertation Co-Advisor	____________________ Dr. Jindong Tan
Committee Member	____________________ Dr. Chunxiao Chigan
Committee Member	____________________ Dr. Jean Mayo
Department Chair	____________________ Dr. Daniel R. Fuhrmann
Date	____________________

To My Teachers and My Families and My Friends!

Contents

[1]This chapter is based on the works from X. Zheng and Y. Cai, Achieving Energy Proportionality In Server Clusters, *International Journal of Computer Networks*, CSC Press, Vol. 1, No. 1, pp. 21-35, November 2009. X. Zheng and Y. Cai, Optimal Server Provisioning and Frequency Adjustment in Server Clusters, *39th International Conference on Parallel Processing GreenCom Workshops 2010*, pp. 504-511, 2010.

[2]This chapter is based on the work from X. Zheng and Y. Cai, Markov Model Based Power Management in Server Clusters, *Proc. IEEE/ACM International Conference on Green Computing and Communications (GreenCom2010)*, Dec. 2010.

[3]This chapter is based on the works from X. Zheng and Y. Cai, Energy-aware load dispatching in geographically located Internet data centers. *Sustainable Computing Informatics and Systems*, Vol. 1, No. 4, pp.275-285, 2011. X. Zheng and Y. Cai, Reducing Electricity and Network Cost for Online Service Providers in Geographically Located Internet Data Centers, *Proc. IEEE/ACM International Conference on Green Computing and Communications (GreenCom2011)*, Aug. 2011.

List of Figures

List of Tables

Acknowledgments

First and foremost, I want to express my sincere gratitude to my advisors: Prof. Yu Cai and Jindong Tan for their guidance, motivation and support throughout my PhD study at Michigan Technological University. It is impossible for me to finish this dissertation without their encouragement, patience, and enthusiasm.

I would also like to thank my other committee members, Prof. Chunxiao Chigan and Prof. Jean Mayo for taking the time to serve on my dissertation committee and for their thoughtful comments and suggestions.

I would also like to thank my great labmates, who made my life at Michigan Tech more colorful and enjoyable. They are Lufeng Shi, Xi Chen, Shuo Huang, Fanyu Kong, Xiaolong Liu, Zhenzhou Shao, Ya Tian and Sheng Hu. My special thanks go to Xi Wang, Na Hu, Ming Ning for their friendship and help.

Last but not least, I would like to thank my parents and my husband Tao Guo, their constant love, support and encouragement, which helped me get through the challenging times. Without their love, I could not have completed this endeavor.

Abstract

The past decade has seen the energy consumption in servers and Internet Data Centers (IDCs) skyrocket. A recent survey estimated that the worldwide spending on servers and cooling have risen to above \$30 billion and is likely to exceed spending on the new server hardware [1]. The rapid rise in energy consumption has posted a serious threat to both energy resources and the environment, which makes green computing not only worthwhile but also necessary. This dissertation intends to tackle the challenges of both reducing the energy consumption of server systems and by reducing the cost for Online Service Providers (OSPs).

Two distinct subsystems account for most of IDC's power: the server system, which accounts for 56% of the total power consumption of an IDC, and the cooling and humidification systems, which accounts for about 30% of the total power consumption. The server system dominates the energy consumption of an IDC, and its power draw can vary drastically with data center utilization. In this dissertation, we propose three models to achieve energy efficiency in web server clusters: an energy proportional model, an optimal server allocation and frequency adjustment strategy, and a constrained Markov model. The proposed models have combined Dynamic Voltage/Frequency Scaling (DV/FS) and Vary-On, Vary-off (VOVF) mechanisms that work together for more energy savings. Meanwhile, corresponding strategies are proposed to deal with the transition overheads.

We further extend server energy management to the IDC's costs management, helping the OSPs to conserve, manage their own electricity cost, and lower the carbon emissions. We have developed an optimal energy-aware load dispatching strategy that periodically maps more requests to the locations with lower electricity prices. A carbon emission limit is placed, and the volatility of the carbon offset market is also considered. Two energy efficient strategies are applied to the server system and the cooling system respectively.

With the rapid development of cloud services, we also carry out research to reduce the server energy in cloud computing environments. In this work, we propose a new live virtual machine (VM) placement scheme that can effectively map VMs to Physical Machines (PMs) with substantial energy savings in a heterogeneous server cluster. A VM/PM mapping probability matrix is constructed, in which each VM request is assigned with a probability running on PMs. The VM/PM mapping probability matrix takes into account resource limitations, VM operation overheads, server reliability as well as energy efficiency.

The evolution of Internet Data Centers and the increasing demands of web services raise great challenges to improve the energy efficiency of IDCs. We also express several potential areas for future research in each chapter.

Chapter 1

Introduction

1.1 Background and Motivations

The Internet Services have provided people with convenient ways to communicate with each other, shop, and access information. It has transformed our way of life. However, the increasing demand of Internet services also brings some problems. To satisfy global user demand, more and more Internet data centers (IDCs) were built in recent years. The increased data centers require more energy supply, at the same time cause increased heat dissipation, greater cooling requirements, reduced computational density, and higher operating costs [2]. It places a heavy burden on both environment and energy resources. The power consumption of Internet data centers in the U.S. doubled between 2000 and 2005 [3]. It is estimated that servers consume 0.5 percent of the world's total electricity usage, which if current demand continues, is projected to quadruple by 2020 [4]. Meanwhile, the increasing cost for operating IDCs also becomes an issue for the online service providers (OSPs). Electricity now accounts for a large fraction of the cost for data centers [5]. In 2005, U.S. data centers consumed 45 billion kW-H; roughly 1.2% of the total amount of U.S. electricity consumption, resulting in utility bills of \$2.7 billion [6]. In 2006, the U.S. Congress passed bills to raise the IT industry's role in energy and environmental policy to the national level [7]. Some analysts predicted that IT infrastructure in IDCs will soon cost more on power consumption than the hardware itself [8].

In order to reduce the operational cost, OSPs are now focusing more on the energy efficiency improvement [9]. Two distinct subsystems account for most of an IDC's power draw: the server system, which accounts for 56% of the total power consumption of an IDC; the cooling and humidification systems, which account for 30% of total power consumption [10]. Server subsystem dominates and its power draw can vary drastically with data center utilization. It is worthwhile and effective to investigate the server system in order to improve the overall energy efficiency of IDCs. Meanwhile, electricity price exhibits both location and time diversities [11]. Therefore, the geographical distribution of data centers exposes many opportunities to reduce operational cost for OSPs.

This research work is focusing on providing strategies to reduce the energy consumption in the server system and manage the operational cost for OSPs. To achieve this particular

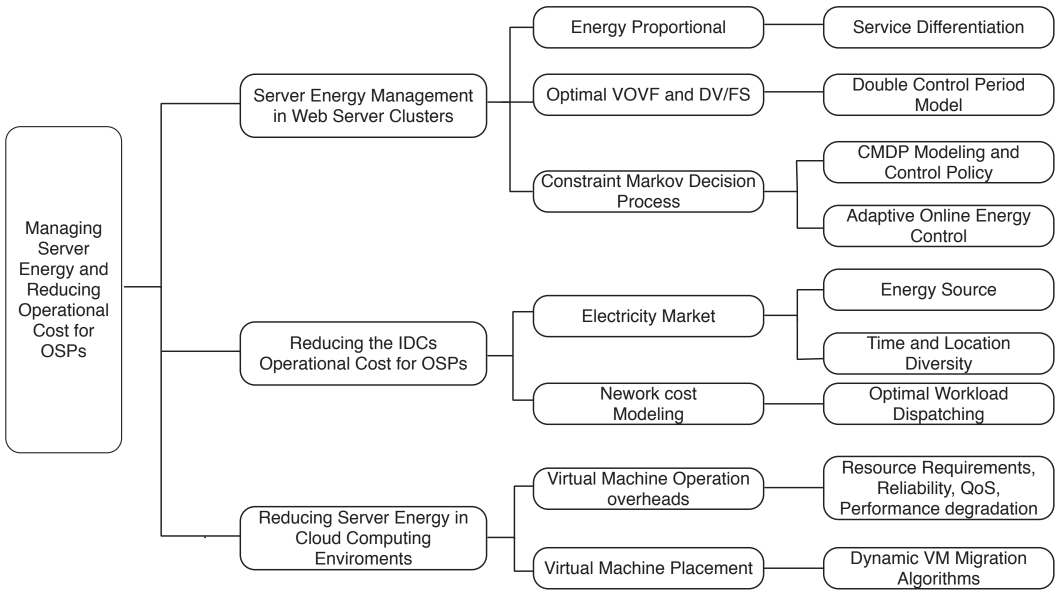

Managing Server Energy and Reducing Operational Cost for OSPs
Server Energy Management in Web Server Clusters
Energy Proportional
Service Differentiation
Optimal VOVF and DV/FS
Double Control Period Model
Constraint Markov Decision Process
CMDP Modeling and Control Policy
Adaptive Online Energy Control
Reducing the IDCs Operational Cost for OSPs
Electricity Market
Energy Source
Time and Location Diversity
Nework cost Modeling
Optimal Workload Dispatching
Reducing Server Energy in Cloud Computing Enviroments
Virtual Machine Operation overheads
Resource Requirements, Reliability, QoS, Performance degradation
Virtual Machine Placement
Dynamic VM Migration Algorithms

Figure 1.1: Research tasks overview.

research goal, there are several major challenges. The first obstacle stems from the fact that energy management involves the tradeoff between power and performance. We therefore need to evaluate the energy management policy based on its potential impact on power and performance. Moreover, workload behavior and performance metrics vary from different users, how to achieve the service differentiation under the dynamic workload is important for the power management design. The uneven electricity market, heterogeneous geographical located IDCs and the uncertainly workload make it even harder to construct a suitable model to reduce the operational cost for OSPs. Finally, with the rapidly expanding usage of the virtualization and cloud computing, effective power and resource management should be also carried out for the cloud services. Motivated by tackling the research challenges, this project identifies the following specific research tasks as shown in Figure 1.1.

1.2 Energy Management in Server Cluster System

Many previous works have been carried out to reduce power consumption in a server cluster system. Two main mechanisms are commonly used for energy savings: dynamic voltage/frequency scaling (DV/FS), dynamically adjusts the frequency and voltage of servers to produce energy savings [12, 13]; Vary-On, Vary-OFF (VOVF) used the server turn ON/OFF mechanism for energy savings [14–16]. A few work considered integrating both DV/FS and VOVF mechanisms [8, 17]. Applying DV/FS and VOVF should take careful considerations of transition overhead, which not only leads to performance degradation but also reduces the life cycle of hardware components. However, the transition overhead was not well studied in the literature [18].

In addition to server state control, another effective method for energy management is to increase the hardware utilization in server clusters by using virtualization techniques. Especially with the rapidly expanded virtualization techniques and cloud services, cloud environments require considerable investigation of techniques to improve the energy efficiency. To effectively use the virtualization techniques, the resources required by an application must be exactly determined. Reserving too much will result in wasted resources and thus wasted energy consumption, allocating fewer resources than required can lead to performance problems. Previous works use workload consolidation to vacate physical server nodes to improve system efficiency [19, 20]. However, most of them neglect the dynamic behavior of workload, which will prevent more energy savings. Moreover, the use of consolidation strategies must also consider additional factors that are of utmost importance for data centers, such as Quality of Service (QoS), reliability in addition to energy consumption. The overheads caused by VM consolidation and migration need to be investigated.

This dissertation studies the power consumption in server system and provides energy efficiency strategies in both web service environment and cloud computing environments. We design dynamic power management strategies to reduce the power consumption in server system with guaranteed performance.

1.3 IDC Operational Cost Management

Managing IDC energy consumption is complicated because of the diversity and complexity of data center infrastructure. Energy related costs, on the other hand, have become one of the most important economical factors for IDCs. Moreover, the reduction of the carbon emissions has also become a changing task for OSPs [21]. Many of the existing work on power and electricity cost management only focuse on a single data center [14, 15]. Moreover, they only consider the energy related cost in server systems. Some work on multiple data centers address the load distribution across data centers with respect to energy consumption or electricity cost. Only a few work consider the variation of the electricity market [5, 11]. The impact of energy sources is rarely addressed [22]. The complexity of the IDC energy consumption model is not well studied. The network electricity cost is not studied in the literature for reducing the total cost.

In this dissertation, we extend the server energy management to the operational cost management for OSPs. We studied the diversity of the electricity market, the network cost, the energy consumption of IDC, and designed an optimization load dispatching model to minimize the operational cost across geographically distributed IDCs.

1.4 Dissertation Outline

The main contributions of this dissertation are in the following areas:

- Design an energy proportional model and an optimization model for dynamic power management in web server clusters (Chapter 2).
- Propose an adaptive Constrained Markov Decision Process (CMDP) for power management in web server clusters which significantly reduce online computation time (Chapter 3).
- Extend the server cluster energy management to the IDC cost management, and build an energy-aware load dispatching model in geographically located IDCs to reduce the cost for OSPs (Chapter 4).

- Propose a statistical live virtual machine placement strategy to reduce the server system energy in cloud environments (Chapter 5).

A brief overview of each chapter is provided here.

Chapter 2 and 3 propose three models for the energy management in web server clusters: an energy proportional model, an optimal server allocation and frequency adjustment model, and a constrained Markov model. All three models are tested and evaluated by extensive simulations. In Chapter 4, we study the energy consumption in an IDC and propose an energy-aware load dispatching model to help OSPs to conserve, manage their electricity cost, and lower the carbon emissions. Chapter 5 extends the server power management for cloud computing services. We present an energy-efficient dynamic virtual machine placement strategy in cloud environment. Chapter 6 summaries this dissertation.

Chapter 2

Dynamic Power Management in Web Server Clusters[1]

2.1 Introduction

The server system of an Internet Data Center accounts for 56% of its total power consumption, and its power draw can vary drastically with data center utilization. Therefore, reducing the energy consumption in server systems becomes a hot topic. Energy management involves the tradeoff between power and performance. Two main

[1]This chapter is based on the works from X. Zheng and Y. Cai, Achieving Energy Proportionality In Server Clusters, *International Journal of Computer Networks*, CSC Press, Vol. 1, No. 1, pp. 21-35, November 2009. X. Zheng and Y. Cai, Optimal Server Provisioning and Frequency Adjustment in Server Clusters, *39th International Conference on Parallel Processing GreenCom Workshops 2010*, pp. 504-511, 2010.

mechanisms are commonly applied for energy savings: DV/FS dynamically changes the frequency and voltage of servers to produce energy savings [12, 13]; VOVF uses server power on/off mechanisms for power management [14–16]. Some thought is given to integrating both DV/FS and VOVF mechanisms together [8]. Applying DV/FS and VOVF simultaneously requires careful consideration due to transition overhead, which not only leads to performance degradation, but also reduces the life cycle of hardware components. However, the transition overhead was not well studied in the literature [18].

Therefore, we advocate promoting energy consumption to a first class resource constraint, in addition to performance that is well studied in the literature. Energy constraints will bring new insights and findings on how to achieve green computing in current server systems.

This chapter introduces two theoretical frameworks for dynamic power management in web server clusters. We first propose an energy proportionality model and investigate the transition overhead based on this model. We also construct an optimization power model in web server cluster by combining DV/FS and VOVF mechanisms. A novel double control periods (DCP) strategy is proposed based on the optimization model to compensate the transition overhead. The simulation results show that both models can provide controllable and predictable quantitative control over power consumption with theoretically guaranteed service performance.

2.2 Related Work

In literatures, green computing is often related to terms like green IT, sustainable computing, energy efficiency, energy saving, power aware, power saving, and energy proportional. In this section, we review relevant techniques commonly used on a single server and server clusters.

The green computing techniques for a single server focus on microprocessors, memories and disks. Current microprocessors allow power management by dynamic voltage and frequency scaling (DV/FS). DV/FS works because reducing the voltage and frequency provides substantial savings in power at the cost of slower program execution. Some researches tie the scheduler directly to DV/FS [23–25]. Most works deal exclusively with meeting real-time scheduling deadlines while conserving energy.

Traditionally, many power management solutions rely heavily on heuristics. Recently, feedback control theoretical approaches for energy efficiency have been proposed by a number of researchers. On a single server, recent works [26, 27] proposed power control schemes based on feedback control theory. Femal et al. [28] developed an algorithm based on linear programming. In [29], a control theoretical power management scheme on standalone servers was proposed. The feedback control theory is better than the traditional techniques by providing high accuracy and stability.

Thermal management is another issue in power-aware computing, since temperature is a by-product of power dissipation [30]. Recent research demonstrated that dynamic thermal management (DTM) can respond to thermal conditions by adaptively adjusting a chip power consumption profile on the according to feedback from temperature sensors [26, 31].

Research work on memory is often combined with processors and disks. In [32], the authors used open-loop control to shift power between the processor and memory to maintain a server power budget. In [33], they proposed a solution to store pages and reliability data in idle RAM instead of using slow disk. A large portion of the power budget of servers goes into the I/O subsystem, the disk array in particular. Many disk systems offer multiple power modes and can be switched to a low power mode when not in use to achieve energy saving. Such techniques had been proposed in [34, 35]. Sudhanva et al. [36] presented a new approach called DRPM to modulate disk speed dynamically, and a practical implementation was provided for this mechanism.

In recent years, power management has become one of the most urgent concerns on server clusters. Some methods proposed on a single server can be extended to server clusters. In [12, 13], the authors presented similar ways of applying DV/FS and cluster reconfiguration, using threshold values, based on the utilization of the system load to keep the processor frequencies as low as possible, with less active nodes. In [37], the authors extended the feedback control scheme to clusters. Power has been used as a tool for application-level performance requirements. Sharma et al. [38] proposed feedback control schemes to control application-level quality of service requirements. Chen et al. [39] presented a feedback controller to manage the response time in server clusters. Some researchers applied DTM on an entire data center rather than individual servers or chips. In [16], the authors laid out policies for workload placement to promote uniform temperature distribution using active thermal zones.

VOVF is a dynamic structure configuration mechanism to ensure energy-aware computing in server clusters, which turns nodes on and off to adjust the number of active servers by the workload. Other work had been carried out based on VOVF [14–16]. In [17], The authors proposed a method to reduce network energy consumption via sleeping and rate adaptation by combining VOVF and DV/FS. Another group developed power saving techniques for connection oriented servers [40]. The authors tested server provisioning and load dispatching on the MSN instant messaging platform, and evaluated their techniques in terms of energy saving and performance.

Virtualization is another key strategy to reduce power consumption in enterprise networks. With virtualization, multiple virtual servers can be hosted on less but more powerful physical servers, using less electricity [41]. In [42], researchers developed methods to efficiently manage the aggregate platform resources according to the guest virtual machines (VM) of relative importance (Class-of-Service), using both the black-box and the VM-specific approach. Hu et al. [43] used live migration of virtual machines to transfer load among the nodes on a multilayer ring-based overlay. In [44], researchers scheduled virtual machines in a computer cluster to reduce power consumption via the technique of DV/FS. An economy driven energy and resource management framework was presented for clusters in [45]. Each service "bids" for resources as a function of delivered performance. In [46], researchers formulated the problem as a cooperative game, and used game theory to find the bargaining point.

The energy-related budget has accounted for a large portion of total storage system cost of ownership. Some studies tried multi-speed disks for servers [36, 47]. Other techniques were introduced to regulate data movement. For example, the mostly used data can be transferred to specific disks or memory, thus other disks can be set to a low power mode [48].

2.3 Energy Proportional Model

2.3.1 Energy Proportionality

An important principle in green computing is to ensure energy consumption proportionality, which states that the energy consumption P should be proportional to the system workload λ [49]:

$$P = a * \lambda + b \tag{2.1}$$

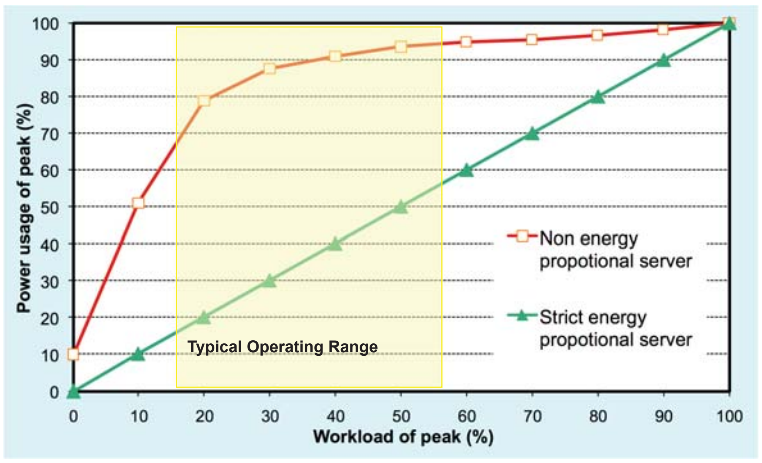


Figure 2.1: The energy consumption curves of non-energy proportional server and strict energy proportional server.

This idea can improve the energy efficiency in real-life usage. Figure 2.1 conceptually illustrates the energy consumption curve in non-energy proportional servers and energy proportional servers. The typical server operating range is between 10% - 60%. We can observe that in a non-energy proportional server, it still consumes about half of its full power when doing virtually no work [49]. Energy proportional server ideally consumes no power when idle ($b = 0$), nearly no power when very little work is performed, and gradually more power as the activity level increases. Large amount of energy savings can be achieved through the design of energy proportionality. However, most servers nowadays are CPU, memory and hard disk intensive servers. The energy consumption of CPU is almost linear to its utilization [40]. But memory and hard disks are non-linear energy consumption components. As a result, energy proportionality is not easy to be achieved on a standalone server because of the hardware constraints.

It is more feasible to achieve energy proportionality in a server cluster. Most computing systems nowadays have at least two modes of operation: an active mode when the system is working and an idle mode when the system is inactive and consumes little energy. Some researchers proposed to have finer-grained power modes, running at low speed and with lower power supply voltage. To achieve energy proportionality, it is feasible to adaptively and dynamically control the number of servers running in active and inactive modes according to system workload. For simplicity, we assume all the servers in the cluster are identical nodes. Although data center are inherently heterogeneous due to upgrading cycles and replacing of failed components, this is a reasonable assumption, since it is normally preferred that groups of servers are load balanced. On typical web servers and web clusters, system workload can be described by the request arrival rate λ. Let M be the total number of servers in the cluster, and Λ be the maximum arrival rate for the cluster. $\sum m$ is the total number of active servers. The total energy consumption P of a server cluster is:

$$P = \sum m P_{ac} + (M - \sum m) P_{in} \tag{2.2}$$

P_{ac} is the power consumption of fully active nodes, P_{in} is the power consumption of inactive nodes. Based on the energy proportional model, we have:

$$\frac{P}{\lambda} = \frac{P_{max}}{\Lambda} r \tag{2.3}$$

where $P_{max} = M * P_{ac}$ represents the maximum energy consumption in the server cluster. r is a parameter, which adjusts the energy consumption curve in Figure 2.1. The rationale of using parameter r is as follows. Ideally the r is set to $r = 1$ where energy consumption is strictly proportional to workload. However, we can adjust it to satisfy different performance constraints. We'll explain it in the next section. With the help of Equation (2.2), we can rewrite Equation (2.3) as:

$$\sum m = (\frac{\lambda P_{ac}}{\Lambda/M}r - MP_{in})/(P_{ac} - P_{in}) \tag{2.4}$$

Here Λ/M is the maximum jobs that a single cluster node can handle. Ideally $P_{in} = 0$, which indicates that a server consumes no energy when it is running on an inactive mode. For simplicity, we suppose $P_{in} = 0$ in this work, this assumption will not affect the performance of our model. We can derive that the total number of active servers $\sum m$ is determined by the system workload λ:

$$\sum m = \frac{\lambda}{\Lambda/M}r \tag{2.5}$$

The number of servers may not be an integer based on (2.5). We will set the integer no less than $\sum m$, which is the minimal number of servers to run in fully active mode.

2.3.2 Performance Metrics

An important task of energy aware computing is to achieve energy efficiency while ensuring performance. One important and commonly used QoS metric on Internet services is slowdown, which is defined as the division of waiting time by service time. Another commonly used performance metric is request time which is the sum of waiting time and service time. We choose slowdown and request time as performance metrics in our model because they are related to both waiting time and service time.

Our theoretical framework is built along the line of the previous service differentiation models presented in [50–53]. In our network model, a heavy-tailed distribution of packet size is used to describe web traffic. Here we assume that the service time is proportional to the packet size.

The packet inter-arrival time follows exponential distributed with a mean of $1/\lambda$, where λ is the arrival rate of incoming packets. A set of tasks with size following a heavy-tailed bounded Pareto distribution are characterized by three parameters: α, the shape parameter; k, the shortest possible job; p, the upper bound of jobs. The probability density function can be defined as:

$$f(x) = \frac{1}{1-(k/p)^{\alpha}} \alpha k^{\alpha} x^{-\alpha-1} \tag{2.6}$$

where, $\alpha, k > 0, k \leq x \leq p$. If we define a function:

$$K(\alpha, k, p) = \frac{\alpha k^{\alpha}}{1-(k/p)^{\alpha}} \tag{2.7}$$

then we have:

$$E[X] = \int_k^p f(x)dx = \begin{cases} \frac{K(\alpha,k,p)}{K(\alpha-1,k,p)} & \text{if } \alpha \neq 1; \\ \\ (lnp - lnk)K(\alpha,k,p) & \text{if } \alpha = 1. \end{cases} \tag{2.8}$$

Similarly, we can derive $E[X^2]$ and $E[X^{-1}]$:

$$E[X^2] = \int_k^p f(x)x^2dx = \frac{K(\alpha,k,p)}{K(\alpha-2,k,p)} \tag{2.9}$$

$$E[X^{-1}] = \int_k^p f(x)x^{-1}dx = \frac{K(\alpha,k,p)}{K(\alpha+1,k,p)} \tag{2.10}$$

According to Pollaczek-Khinchin formula, the average waiting time for the incoming packets is:

$$E[W] = \frac{\lambda E[X^2]}{2(1-\lambda E[X])} \tag{2.11}$$

We can derive a closed-form expression of the expected slowdown in a M/G/1 queue on a single server in Equation (2.12).

$$E[S] = E[W]E[X^{-1}] = \frac{\lambda E[X^2]E[X^{-1}]}{2(1-\lambda E[X])} \tag{2.12}$$

The expected request time with the incoming job rate λ is:

$$E[R] = E[W] + E[X] = \frac{\lambda E[X^2]}{2(1-\lambda E[X])} + E[X] \tag{2.13}$$

2.3.3 Servers Allocation on Service Differentiation

In a cluster system, the incoming requests are often classified into N classes. Each class may require different QoS according to its priority. We assume m_j is the number of active server nodes in class j, and λ_j is the arrival rate in class j. As it is shown in Figure 2.2. The expected slowdown of class i in a server cluster can be calculated as:

$$E[S_i] = \frac{\lambda_i E[X^2] E[X^{-1}]}{2(m_i - \lambda_i E[X])} \tag{2.14}$$

Here we choose not to use request time as a performance metric for service differentiation because of its overly complicated mathematical expression. However, each class should satisfy the request time constraint. Obviously the results presented in this work will not be affected by the selection of performance metrics.

We adopt a relative service differentiation model where the QoS factor of slowdown

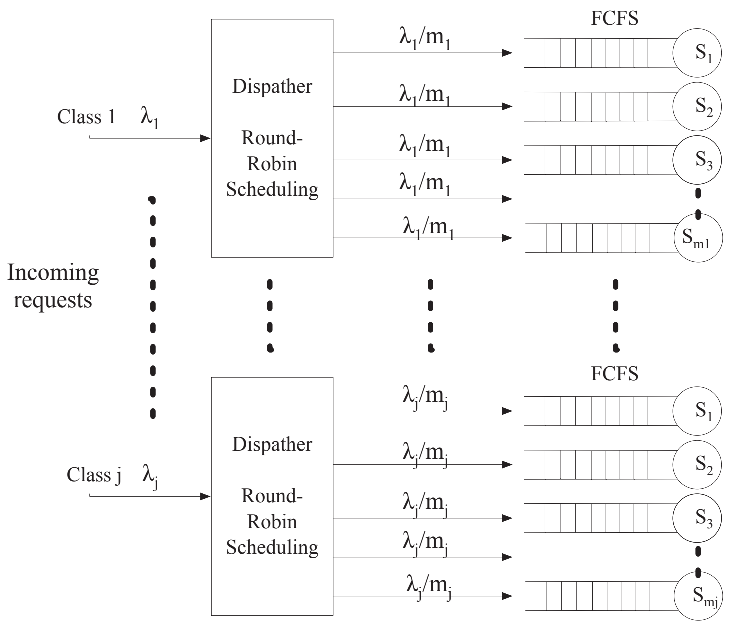


Figure 2.2: System model in multiple classes

between different classes are based on their predefined differentiation parameters.

$$\frac{E[S_i]}{E[S_j]} = \frac{\delta_i}{\delta_j} \tag{2.15}$$

where $1 \leq i, j \leq N$: We assume class 1 is the highest class and set $0 < \delta_1 < \delta_2 < \cdots < \delta_N$, then higher classes receive better service, i.e., lower slowdown.

Based on the above energy proportionality and service differentiation model, according to Equations (2.5) and (2.15), we can derive the server allocation scheme in a cluster system as

$$m_i = \lambda_i E[X] + \frac{\tilde{\lambda}_i \sum_{i=1}^{N} \lambda_i (\frac{M}{\Lambda} r - E[X])}{\sum_{i=1}^{N} \tilde{\lambda}_i} \tag{2.16}$$

Here m_i is the number of active servers in class i, and $\tilde{\lambda}_i = \lambda_i/\delta_i$ is the normalized arrival rate. The first term of Equation (2.16) ensures that the sub-cluster in class i will not be overloaded. The second term is related to arrival rates, differentiation parameters, and r.

We can also derive the expected slowdown of class i as:

$$E[S_i] = \frac{\delta_i E[X^2]E[X^{-1}]\sum_{i=1}^{N}\tilde{\lambda}_i}{2\sum_{i=1}^{N}\lambda_i(\frac{M}{\Lambda}r - E[X])} \tag{2.17}$$

From Equation (2.17) we can observe that the slowdown of class i is proportional to the pre-specified parameter δ_i, and is related to r. The slowdown ratio only depends on the pre-defined differentiation parameters.

The expected request time for class i can be calculated as:

$$E[R_i] = \frac{\delta_i E[X^2]\sum_{i=1}^{N}\tilde{\lambda}_i}{2\sum_{i=1}^{N}\lambda_i(\frac{M}{\Lambda}r - E[X])} + E[X] \leq \beta_i \tag{2.18}$$

β_i is request time constraint for class i. We can learn from Equation (2.18), request time in class i is also independent of workload, but depends on both the pre-specified parameter δ_i and r.

2.4 Optimal Server Provisioning and Frequency Adjustment

The optimal model is built in a web server cluster level. All the requests must meet a predefined quality of service (QoS). To simplify the problem, we have the following assumptions: First, all the servers in a server cluster are identical nodes, which means all the servers are same in terms of hardware. Second, in our model, when a server is switched off, it consumes no power, when the machine is turned on, it can operate at a number of discrete frequencies. With the help of adjustable frequency, it is feasible to reduce power consumption whenever possible. Last, we assume all the incoming requests are CPU bounded, in other words, CPU speed is the bottleneck of performance. This is still reasonable because Austin [54] pointed out that CPU is the largest consuming component for typical web server configuration.

2.4.1 Performance and Power Modeling

Processors today are commonly equipped with mechanisms to reduce power consumption at the expense of reduced server frequency [55]. As we mentioned before, we assume our work is built on a server cluster system with uniformed servers. Let M be the total number of servers. Each server has N levels adjustable frequency $f_i (1 \leq i \leq N)$, where $f_1 < f_2 <$

$f_3 < ... < f_N$. Since all the incoming requests are CPU bounded, higher the operating frequency leads to greater the server processing capacity, which can be represented as $c_i = af_i$.

We adopt the same queue model as we described in section 2.3.2. Given an M/G/1 queue on a server, X is service time, X' is the service time under a given capacity c, we have

$$E[X'] = \frac{1}{c}E[X] \tag{2.19}$$

$$E[X'^2] = \frac{1}{c^2}E[X^2] \tag{2.20}$$

The average waiting time for the incoming packets under capacity c in a single server can be represented as:

$$E[W'] = \frac{\lambda E[X^2]}{2c(c - \lambda E[X])} \tag{2.21}$$

When applying a round-robin dispatching policy, the packet arrival rate of a node is λ/m. The processing capacity is always proportional to the operating frequency. The expected request time for any server in a server cluster can be calculated as:

$$E[R] = \frac{\lambda E[X^2]}{2af_i(af_i m - \lambda E[X])} + \frac{1}{af_i}E[X] \tag{2.22}$$

Where $1 \leq i \leq N, 1 \leq m \leq M$.

To understand the power-to-frequency relationship of an individual server is crutial to the power aware system design. There are two widely used power model in a single server: linear and cubic.

Linear model has been widely used in related work [55, 56]. In [55], researchers found that the server power-to-frequency relationship for DV/FS is approximate linear. There are two reasons to explain this relationship. One reason is that manufacturers usually settle on a limited number of allowed voltage levels, which results in less-than-ideal relationship between power and frequency in practice. The other reason is: DV/FS is not applied to many components at the system level. In linear model, the power consumption P_{single} is set as a function of server frequency, as: $f = f_b + \alpha(P_{single} - b)$, in which b is the minimum power consumed by a fully-utilized server over the allowable range of processor frequency; f_b is the frequency of a fully utilized server running at b Watts; Coefficient α is the slope of the power-to-frequency curve.

Another commonly used model is cubic relationship between frequency and power. In cubic model, it assumes that power consumption of all other system components is essentially constant regardless of system activity. CPU power consumption depends on the CPU voltage and frequency. Furthermore, there is a linear relationship between frequency and voltage. The cubic model can be represented as: $P_{single} = c_0 + c_1 f^3$, where c_o is a constant that includes the power consumption of all components except the CPU and the the base power consumption of the CPU. The second part is the power consumption of CPU running at frequency f. This cubic model has been used in many other related work [8, 57, 58]. The chosen of power model does not affect the overall power optimization strategy proposed in this work, we use cubic model in our theoretical framework.

The energy consumption of the whole system can be calculated as:

$$\sum P = P_{active} + P_{inactive} + P_{trans} \tag{2.23}$$

Where P_{active} and $P_{inactive}$ are the power consumption of active nodes and inactive nodes respectively. P_{trans} is the power consumption when servers change between active mode and inactive mode.

2.4.2 Problem Formulation

The energy management strategy can be formulated as a minimization problem: the optimal solution is obtained by selecting the proper number of active servers running at f_i while request time is within a threshold. We formulate the problem in the following two scenarios: single class and multiple classes.

In the single class scenario, we assume all the incoming requests are classified into just one class. In other words, the same QoS should be met. Here a threshold β is set to bound the average request time. Let M be the total number of identical servers in the system. m is the number of active server nodes to handling the incoming requests. We solve the following problem:

$$\begin{aligned} &Min: \quad \sum P = \sum P_{single} + \sum P_{insingle} \\ &S.t. \quad E[R] = \frac{1}{m}\sum_{j=1}^{m}\left(\frac{\lambda E[X^2]}{2af_i(af_i m - \lambda E[X])} + \frac{1}{af_i}E[X]\right) \leq \beta \end{aligned} \tag{2.24}$$

where $1 \leq m \leq M, 1 \leq i \leq N$. $P_{insingle}$ is the power consumption of an inactive server. Here we do not consider the transition power, because it is considerably less important in typical Internet server workload, since load fluctuation occurs on a larger time scale [56].

The above optimization is non-linear and discrete in terms of the decision variables for both objective and constraint. One feasible way is to consider a finite number of frequencies and server number to determine the optimal solution. However, the complexity is $O(N^M)$ when considering M servers and N levels of adjustable frequency. It is can be reduced to $O(MN)$ after applying a coordinated voltage scaling approach, in which all active servers are assigned to equal frequency level. Previous research adopted the same strategy and showed coordinated voltage scaling approach can provide substantially higher savings [8, 57].

In a cluster system, the incoming requests are often classified into W classes. Each class may require different QoS according to its priority. We assume m_j is the number of active server nodes in class j, and λ_j is the arrival rate in class j. We adopt a relative service

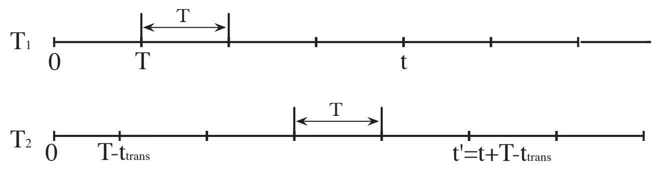


Figure 2.3: Double control periods

differentiation model where the QoS factor of request time between different classes are based on their predefined differentiation parameters.

$$\frac{E[R_i]}{E[R_j]} = \frac{\delta_i}{\delta_j} \tag{2.25}$$

where $1 \leq i, j \leq W$. We assume class 1 is the highest class and set $0 < \delta_1 < \delta_2 < \cdots < \delta_W$, then higher classes receive better service, i.e., smaller request time. We solve the optimization problem in multiple classes' scenario as following:

$$\begin{aligned} &Min: \quad \sum P = \textstyle\sum_{j=1}^{W} m_j P_{single} + (M - \sum_{j=1}^{W} m_j) P_{insingle} \\ &S.t. \quad E[R_j] = \tfrac{\lambda_j E[X^2]}{2af_{ij}(af_{ij}m - \lambda_j E[X])} + \tfrac{1}{af_{ij}} E[X] \leq \beta \delta_j \end{aligned} \tag{2.26}$$

The problem can be solved by decomposing it into W single class optimization problems as we mentioned in previous section.

2.4.3 Overhead Analysis

The model proposed in this work is a continual optimization process, where we dynamically allocate active server number and adjust their frequency levels. Here the overhead caused by frequency adjustment is ignored since it is very small [59]. We assume frequency adjustment accomplished instantly. However, the transition time when a server transfers from an inactive mode to an active mode can not be ignored, which will influence the performance greatly during the transition period. Especially when the workload is increasing in the next control period, it may lead to the increase of active servers. However, sometimes workload increase may result in decrease of active server number. The fact is that we dynamically optimize the server number and frequency together, less number of servers may not be an optimal solution even though the workload decreases in the next control period. Thus, it is necessary to estimate the cost of transition overhead. In general, transition time depends on the processor and other hardware constraints. We propose a double control period (DCP) model to compensate the transition overhead, which allows better of performance.

The basic idea of DCP model is shown in Figure 3.3. 'Double' stands for two control periods denoted as T_1 and T_2 respectively. The control interval of both periods are identical: $T_1 = T_2 = T$. Active server setting and frequency adjustment occur at the beginning of each control period of T_1. Control period T_2 helps to turn on the additional servers for the next control period of T_1 beforehand. The two control periods are designed with control time difference: $t_{diff} = T - t_{trans}$, where t_{trans} is the transition time when a server node transfers from an inactive mode to an active mode. A schematic of DCP model is shown in Figure 2.4.

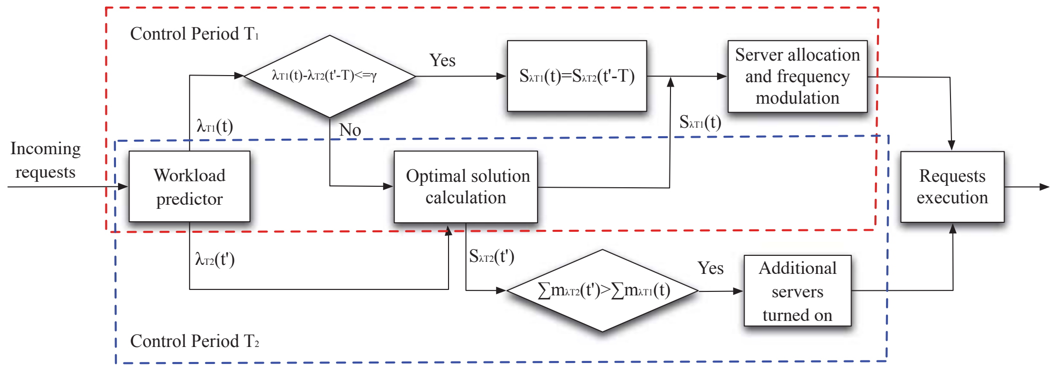


Figure 2.4: Designing of the Double Control Periods (DCP) Model

Workload predictor predicts the incoming requests arrival rate for both control periods: $\lambda_{T1}(t)$ and $\lambda_{T2}(t'-T)$. Here $t' = t + T - t_{trans}$. In each beginning of the control period T_2, optimal solution calculator computes the optimal solution based on $\lambda_{T2}(t'-T)$. To avoid redundant optimization process, we record the optimization solution $S_{T2}(t'-T)$. At each beginning of control period T_1, DCP will first check the requests' arrival rate variance of $\lambda_{T1}(t)$ and $\lambda_{T2}(t'-T)$. If $\lambda_{T1}(t) - \lambda_{T2}(t'-T) \leq \gamma$, DCP adopts solution $S_{T1}(t) = S_{T2}(t'-T)$ instead of re-calculating the server provisioning and frequency adjustment solution according to $\lambda_{T1}(t)$. This strategy enhances computational efficiency. Additional servers $\sum m_{\lambda T2}(t') - \sum m_{\lambda T1}(t)$ will be turned on at the beginning of each control period of T_2 if more servers are required for the next control period of T_1, each server will set to be the lowest frequency f_1 in order to achieve energy efficiency. Additional servers have sufficient time t_{trans} to transfer from inactive model to active mode. DCP model takes advantage of workload characteristic as we mentioned before, workload fluctuations occur on a larger time scale, which means $\lambda_{T1}(t)$ and $\lambda_{T2}(t'-T)$ are close enough for optimization prediction.

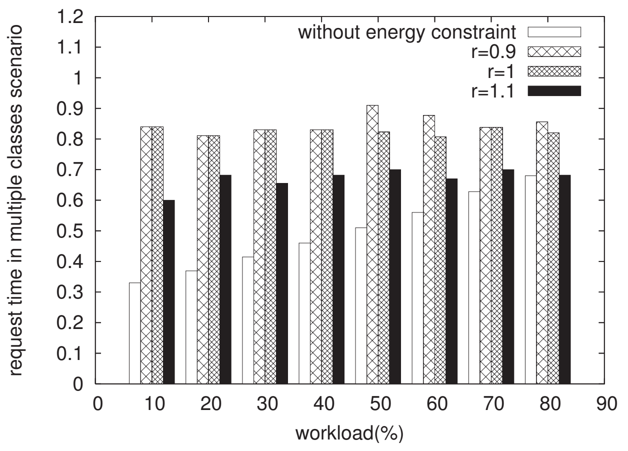


Figure 2.5: Comparison of request time in higher priority class between non-energy proportional model and energy proportional models. r is set differently according to different requirements of performance in a multiple classes scenario.

2.5 Performance Evaluation

2.5.1 Energy Propositional Model

We build a simulator which consists of a package generator, a server dispatcher, a number of waiting queues, and a number of servers. The package generator produces incoming requests with exponential inter-arrival time distribution and bounded Pareto packet size distribution. The GNU scientific library is used for stochastic simulation.

Simulation parameters are set as follows. The shape parameter α of the bounded Pareto distribution is set to 1.5. The lower bound k and upper bound p were set to 0.1 and 100, respectively [60]. The number of servers in the cluster is 20. And we set the

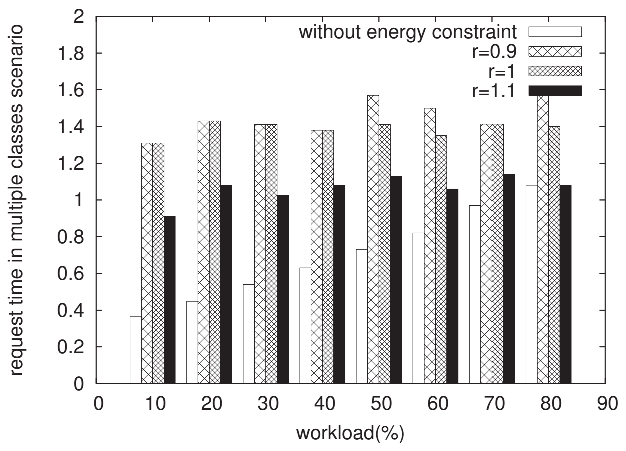


Figure 2.6: Comparison of request time in lower priority class between non-energy proportional model and energy proportional models. r is set differently according to different requirements of performance in a multiple classes scenario.

normalized maximum jobs one server can handle $\Lambda/M = 1$. We set the server active power consumption to 160W [40]. We set the request time to $\beta = 0.9$, $\beta = 1$, and $\beta = 1.3$ which correspond to adjustment parameter $r = 1.1$, $r = 1$, and $r = 0.9$ respectively. We show the simulation results in the workload range of 10% - 80% . When the workload is above 80%, the impact of energy proportionality constraint is very limited. Since the typical server operating range is between 10% - 60%, the results presented here are sufficient to test the energy proportional model.

We first compare the performance metrics as shown in Figure 2.5, 2.6,and 2.7. The number of classes is normally two or three [61, 62]. We choose two classes of incoming requests and set the target slowdown ratio to $\delta_2 : \delta_1 = 2 : 1$. The energy curve functions are set differently according to different request time constraints. Note, in a multiple classes scenario, parameter r is determined by performance requirements of all classes,

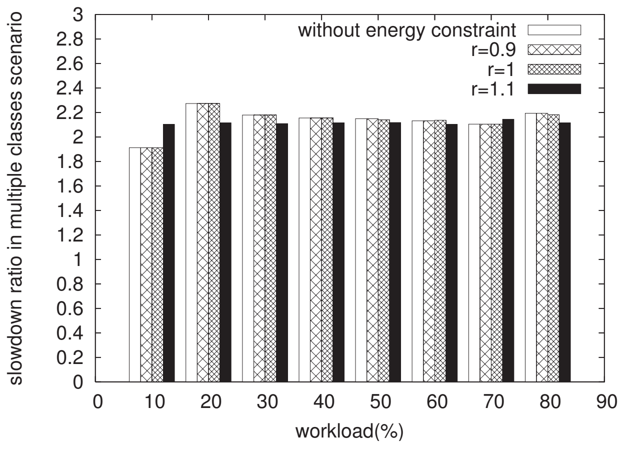


Figure 2.7: Comparison of slowdown ratio between non-energy proportional model and energy proportional models. r is set by different requirements of performance in a multiple classes scenario.

which means it should be set to be the largest value satisfying the requirements of all the classes. We observe that the model can achieve desirable proportionality of slowdown differentiation with request time constraints. Figure 2.8 also compares the energy consumptions for proportional and non-proportional models in multiple classes scenario.

The model proposed in this work is a continual optimization process, where we dynamically change the number of active servers. The transition time when a server transfers from an inactive mode to an active mode can not be ignored, which can influence the performance during the transition period. Thus, it is necessary to estimate the cost of transition overhead.

Generally speaking, the transition time for different servers is different, which depends on the processor and other hardware constraints. Therefore, we study the influence on

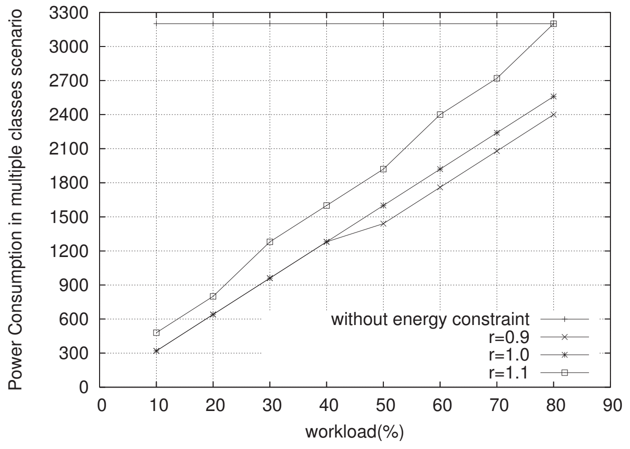


Figure 2.8: Comparison of power consumption between non-energy proportional model and energy proportional model in multiple classes scenario. r is set by different requirements of performance. We can achieve considerable energy saving compare to the non-energy proportional model.

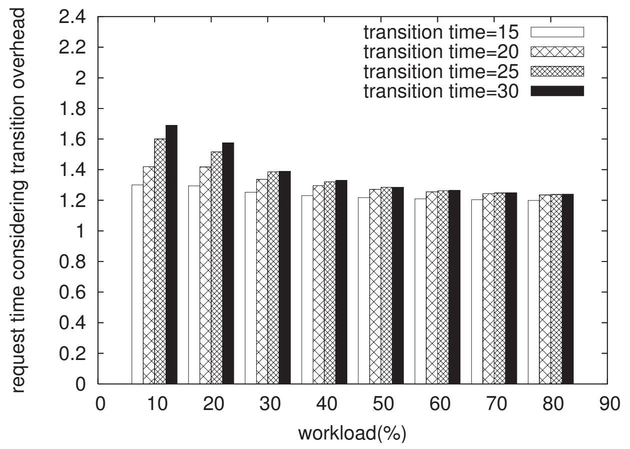


Figure 2.9: The effect to performance of transition overhead in energy proportional model, the transition time is set to be 15,20,25,30 respectively.

performance caused by transition overhead under different time. Figure 2.9 shows how the request time changes when considering transition overhead as the workload gradually changed from 0%-80% based on the energy proportional model. We only concern the situation when the workload increases, as the workload decreases, the number of active servers will decrease. Therefore it will not cause performance degradation. The y-axis

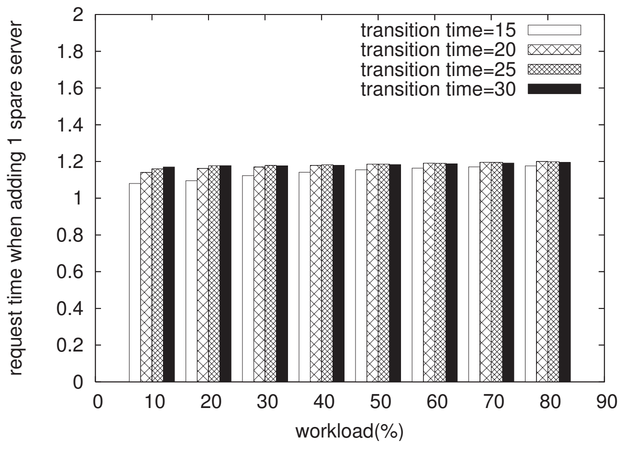


Figure 2.10: Request time after adding one spare server based on energy proportional model in a single class scenario, the transition time is set to be 15,20,25,30 respectively.

is the request time under different transition overhead. As indicated in the figure, larger transition time has more impact on performance. The performance will be affected greatly when large number of servers can not transfer to active mode on time.

To ensure satisfied QoS, spare servers are added to solve the problem of transition overhead. Figures 2.10 and 2.11 illustrate the performance after one and two spare servers are added in a single class scenario. By adding one spare server, the performance can be improved dramatically compared to the case of no spare server. Adding two spare servers, the response time can stay under the pre-defined threshold when the workload gradually changes from 0%-80%. However, in some special situations, the workload may vary significantly within two control periods. One or two spare servers are not adequate to compensate the performance degradation. More spare servers are required.

To evaluate the model on realistic traffic patterns, we use an hour's workload trace collected

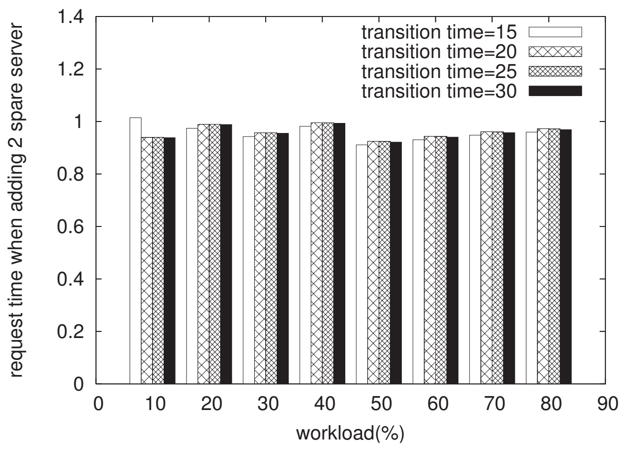


Figure 2.11: Request time when adding two spare servers based on energy proportional model in a single class scenario, the transition time is set to be 15,20,25,30 respectively.

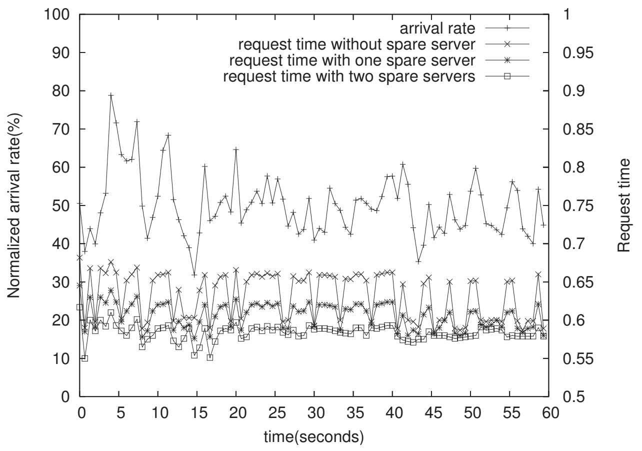


Figure 2.12: Request time when adding two spare servers based on energy proportional model in a single class scenario

by Lawrence Berkeley National Laboratory [63]. Request time threshold is set to be $\beta = 0.6$ and $r = 1$. Figure 2.12 illustrates the performance based on our model in a single class scenario. The requests arrival rate and job size are normalized. We evaluate the performance in the situations of non-spare server and spare servers respectively. As shown in the figure, when the workload decreases, there is no performance degradation, however

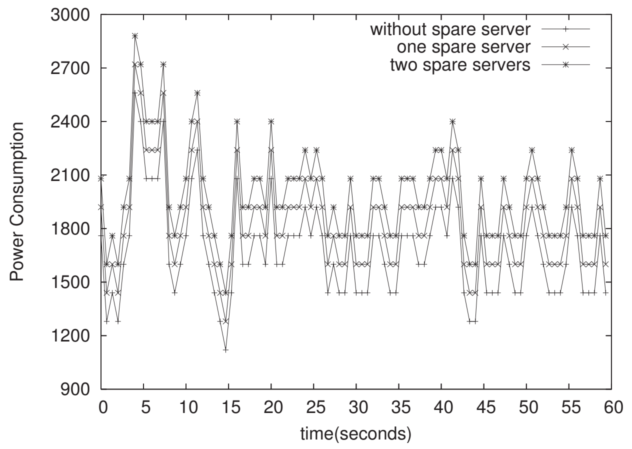


Figure 2.13: Power consumption when adding two spare servers based on energy proportional model in a single class scenario.

the performance degradation can be clearly seen as the workload increases in the case of no spare server is added. With one or two spare servers, the performance can be improved significantly. Especially, when two spare servers are always on, request time is always under pre-defined threshold. The result also indicates that as the number of spare server increases, the performance does not change dramatically. The request time tends to stay in a level, which demonstrate proper spare servers should be set to compensate the performance degradation.

Figure 2.13 evaluates the power consumption based on our model under real workload data trace. The system arrival rate is the same as shown in Figure 2.12. The power consumption is dynamically changed as the workload changed. With little more power consumption, we can achieve better performance, and eliminate the effect of transition overhead.

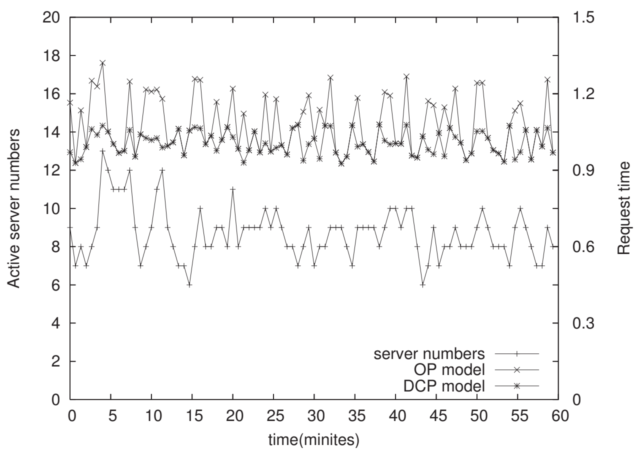


Figure 2.14: Comparison of request time of real workload trace data between OP model and DCP model in a single class scenario.

2.5.2 Optimal Server Provisioning and Frequency Adjustment

In the simulation, the optimization model is evaluated using both stochastic workload and real workload trace data. Only the simulation results with real workload data are shown. Figure 2.14 illustrates the performance based on an optimization (OP) model and a DCP model in a single class scenario. The request arrival rate and job size are normalized. As shown in Figure 2.14, performance degradation is clearly seen in the OP model; this is caused by increasing the number of active servers. The DCP model improves performance significantly. Figure 2.15 compares the power consumption of our model under real workload trace data. The OP model and DCP model are further evaluated in a multiple class scenario as shown in Figure 2.16 and 2.17 under real workload trace data.

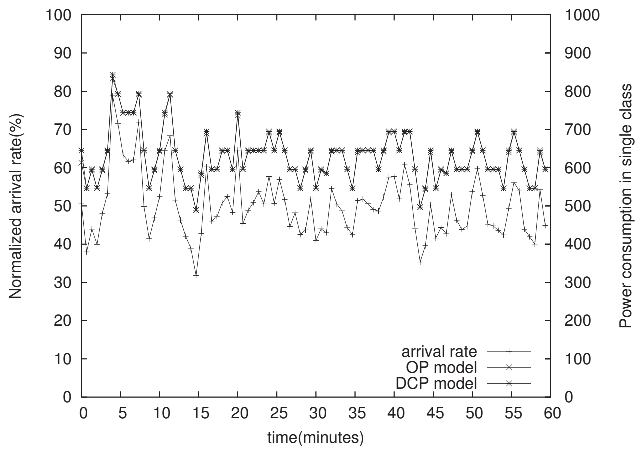


Figure 2.15: Comparison of power consumption between OP model and DCP model in a single class scenario.

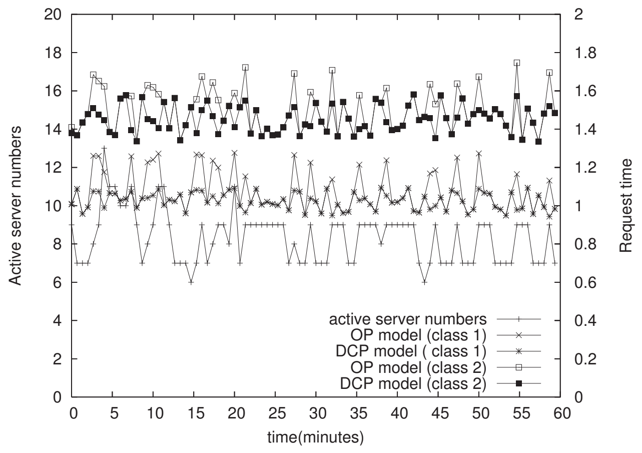


Figure 2.16: Comparison of request time between OP model and DCP model in a in a multiple classes scenario.

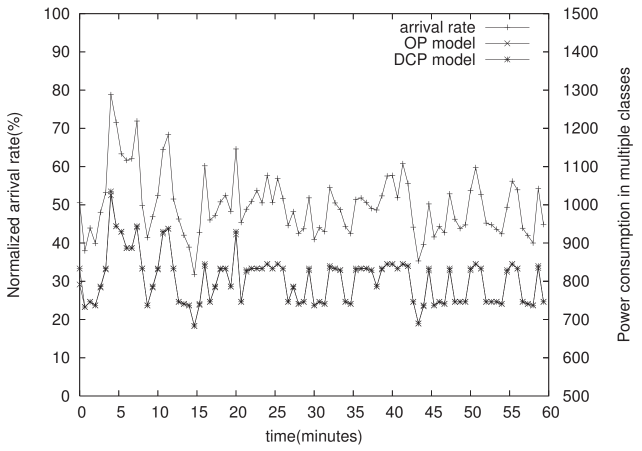


Figure 2.17: Comparison of power consumption between OP model and DCP model in a multiple classes scenario.

Chapter 3

CMDP Based Adaptive Power Management in Server Clusters[2]

3.1 Introduction and Related Work

The primary principle in green computing is to achieve highest possible energy efficiency with guaranteed performance. The tradeoff between power consumption and performance is often resolved by an optimization process, in which power consumption is minimized with the constraint of performance. An on-line controller periodically carries out the optimization problem and then allocates resources accordingly.

[2]This chapter is based on the work from X. Zheng and Y. Cai, Markov Model Based Power Management in Server Clusters, *Proc. IEEE/ACM International Conference on Green Computing and Communications (GreenCom2010)*, Dec. 2010.

Constrained Markov Decision Processes (CMDPs) provides a mathematical framework for modeling decision-making with multiple objectives. CMDP is useful for studying a wide range of optimization problems. In this chapter, we construct a Constrained Markov Decision Process (CMDP) model and propose a CMDP based adaptive power management strategy in web server clusters. Our proposed strategy can greatly reduce the online computation time through an offline initialization process. The online adaptive server adjustment process with further improve system performance and deal with the dynamic changes of workload. The constructed CMDP takes advantages of both DV/FS and VOVF mechanisms to achieve energy efficiency with guaranteed response times in a web server cluster. We take careful consideration of transition overhead in modeling the CMDP in order to obtain more precise power and performance control.

In recent years, power consumption has become one of the most important concerns in computing systems. Prior work addressed the power management issue on both single server and server clusters systems. In this section, we only review work related to power reduction in server clusters since it is more closely related to this work.

There are two main strategies for power reduction: Dynamic Voltage/Frequency Scaling (DV/FS) and server number controlling: Vary-On Vary-Off (VOVF). DV/FS works by reducing the voltage and frequency, consequently saving power at the cost of slower program execution. Researchers have developed various DV/FS scheduling algorithms to save energy under timing deadlines [12, 13]. Some researchers also utilized feedback control to dynamically adjust servers frequency [37]. In these works, control variables can be either servers frequency or application-level quality of service requirements [37–39]. The feedback control theory performs better than traditional heuristic techniques by providing higher accuracy and stability in DV/FS.

VOVF is a major mechanism for power reduction applied in server clusters [14–16]. VOVF dynamically turns idle servers off when the system experiences a light workload, and turns the appropriate servers on when the system encounters a heavy workload. VOVF dramatically improves the system energy efficiency by reducing the idle servers' power consumption. Virtualization as a key strategy to reduce power consumption for application services is another way of VOVF. When virtualization being applied, multiple virtual servers can be hosted on a smaller number of more powerful physical servers, using less electricity [41]. In [42], researchers demonstrated a method to efficiently manage the aggregate platform resources according to the guest virtual machines (VM) relative importance (Class-of-Service), for both the black-box and the VM-specific approach. Recently, researchers pointed out that combining DV/FS and VOVF potentially provided higher energy savings [59]. Other researchers developed power saving techniques for connection oriented servers [40]. Although DV/FS and VOVF are commonly applied for power reduction, most previous works are formulated and solved on-line. Our CMDP model is formulated and solved offline, greatly reducing on-line computation time.

The Markov model was first applied for power management in a battery-based system [64, 65]. Researchers proposed a Markov stochastic model for power control and explored the tradeoff between power and performance. Recently, the Markov model was further studied in server clusters for energy savings. In [66], a three-speed disk Markov power model is formulated in disk systems, and prediction schemes were proposed for achieving disk energy savings. A CMDP is constructed for power and performance control in web server clusters in [67]. This work is similar to the work discussed in this paper, however the DV/FS mechanism was not applied to reduce power consumption. Furthermore, we also propose an adaptive power management strategy to deal with the dynamic workload changes, which is not studied in previous work.

3.2 Problem Formulation

Power and performance are of the most importance for designing a power aware computing system. One of the most effective methods to resolve the tradeoff between power and performance is to formulate an optimization problem. A power controller can allocate resources in the computing system according to the optimal solution. This is often accomplished by a periodical online optimization process. If a combined DV/FS and VOVF strategy is applied for obtaining the optimal power conservation solution in a server cluster, the problem is NP-complete to solve exactly. Although by applying a coordinated voltage scaling approach [8], the online computation complexity can be reduced to $O(MF)$ given that a server cluster has M homogeneous servers and F adjustable frequency levels. It is still time-consuming because of the computation complexity.

In this section, a Constrained Markov Decision Process (CMDP) is constructed to achieve energy savings. The significant advantage of our CMDP model is that our optimization solution is computed offline which greatly reduces the online computation time. The online computation complexity is reduced to $O(1)$ after applying a deterministic CMDP policy.

3.2.1 State Space X and Action Space A

Our CMDP model is built on a homogeneous web server cluster, each server in the cluster can operate at several discrete frequency levels. With the help of adjustable frequency, it is feasible to reduce power consumption whenever possible. In a single queue cluster system, let M be the total number of servers. Each server has F levels adjustable frequency $f_i(1 \leq i \leq F)$, where $f_1 < f_2 < f_3 < ... < f_F$. We define the state of the CMDP model with a tuple $x \in X$:

$$x = \{s, m, f_i\} \tag{3.1}$$

In (3.1), s is the number of jobs waiting in a queue. $s \in \{0, 1, 2, \cdots, S\}$. S is the maximum queue length. If more than S jobs arrive in the queue, the queue will be blocked , and some jobs will be lost. The decision maker should try to avoid queue blocking as much as possible. We assume that there is at least one server in active mode to handle the incoming requests. $m \in \{1, \cdots, M\}, f_i \in \{f_1, \cdots, f_F\}$. m and f_i represent the number and running frequency level of servers in active mode. f_1 and f_F are the minimum and maximum frequency levels respectively for a server in an active mode.

A coordinated voltage scaling approach is applied in our model, by which all active servers are assigned to equal frequencies. Previous researches adopted the same strategy and proved that the coordinated voltage scaling approach can provide substantially higher energy savings [8, 57]. A set of composite actions are defined as $a \in A$:

$$\begin{aligned} &a = \{(M_a, F_a); \\ &M_a \in \{-(M-1), -(M-2), \cdots, 0, 1, \cdots, M-1\}; \\ &F_a \in \{-(F-1), -(F-2), \cdots, 0, 1, \cdots, F-1\}\}. \end{aligned} \tag{3.2}$$

where M_a is the number of server adjustment for the next control period, and F_a is the frequency adjustment for active servers. For example, $F_a = -(F-3)$ corresponds that all the servers in active modes adjust their frequency from f_i to f_{i-F+3}, for $i > F+3$. We denote by $A(x) \subset A$ actions that are available at state x. Set $K = \{(x,a) : x \in X, a \in A(x)\}$ is the set of state-action pairs. A sequential state and action transition scheme is shown in Figure 3.1.

To reduce the state space in large size server clusters, we model the system with multiple queues. New arrival jobs will be distributed to each queue with round-robin discipline waiting for processing. We then divide the system into several subsystems by the number

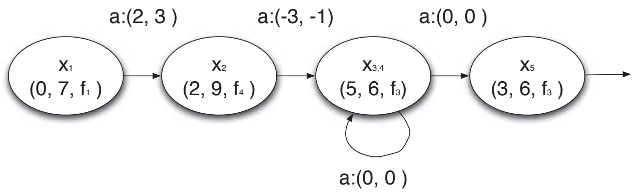


Figure 3.1: Sequential state transiton

of queues in the system and construct a CMDP model separately in each sub-system. Considering a system with W queues, s is the number of jobs waiting in a single queue, S is the maximum queue length for each queue. The maximum number of jobs that can be stored in the system is $W \cdot S$. The composite state of the CMDP in each subsystem can be defined as: $x = \{s, m_j, f_i\}$, where $m_j \in \{1, \cdots, M_j\}, f_i \in \{f_1, \cdots, f_F\}$. m_j and f_i represent the number and running frequency level of servers in active mode. M_j is the number of servers to process the jobs in each sub-queue. Because of the construction of the CMDP model is the same with the single queue system. We only present how to construct a CMDP model in a single queue system as an instance.

3.2.2 Transtion Probability P_{xay}

P_{xay} is the probability in CMDP of moving from state $x = \{s, m, f_i\}$ to $y = \{s', m', f_i'\}$ if action $a = \{M_a, F_a\}$ is taken. Given a composite action $a = \{M_a, F_a\}$ and current state $x = \{s, m, f_i\}$, the active server number m' and frequency level f_i' in next state are deterministic, since they can be easily determined: $m' = m + M_a$, $f_i' = f_{i+F_a}$. However, the queue length is non-deterministic. Consider a single queue with a buffer of finite size S. In each control period, the probability of i jobs arriving and waiting in the queue is defined as $P_i(i)$; d jobs leave and finish processing with a probability of $P_d(d)$. The probability $P_{s,s'}$ when the queue length changes from s to s' can be obtained by:

$$P_{s,s'}(a) = \begin{cases} \sum_{d=0}^{max(m,m')} P_d(d) \cdot P_i(d+s'-s) & s' \geq s; \\ \sum_{i=0}^{I_{max}} P_i(i) \cdot P_d(i+s-s') & s' < s. \end{cases} \tag{3.3}$$

where I_{max} is the maximum number of jobs that can arrive in a control period. We'll explain how to obtain the probability $P_i(i)$ and $P_d(d)$ separately.

In our network model, a set of tasks, whose size z follows a heavy-tailed Bounded Pareto distribution are characterized by three parameters: α, the shape parameter; k, the shortest possible job; p, the upper bound of jobs [50, 68]. The probability density function can be defined as:

$$f(z) = \frac{1}{1-(k/p)^{\alpha}} \alpha k^{\alpha} z^{-\alpha-1} \tag{3.4}$$

where, $\alpha, k > 0, k \leq z \leq p$. If we define a function as:

$$K(\alpha,k,p) = \frac{\alpha k^{\alpha}}{1-(k/p)^{\alpha}} \tag{3.5}$$

Then we have expectation:

$$E[z] = \int_k^p z f(z) dz = \begin{cases} \frac{K(\alpha,k,p)}{K(\alpha-1,k,p)} & \text{if } \alpha \neq 1; \\ \\ (lnp - lnk) K(\alpha, k, p) & \text{if } \alpha = 1. \end{cases} \tag{3.6}$$

We define $E[Service]$ as the average service time. Job size follows Bounded Pareto distribution with the average of $E[z]$. Here, we assume that the service time is proportional to the job size and inversely proportional to the server processing capacity $c = f_i/f_F$ [68]. Higher processing capacity means faster processing speed. So the average processing capacity is $c_{ea} = f_{\lfloor \frac{1+F}{2} \rfloor}/f_F$. The average service time for the incoming requests can be obtained from:

$$E[service] = \frac{E[z]}{c_{ea}} \tag{3.7}$$

The probability that a server finishes processing a job in one control period T is calculated as follows:

$$\beta = \begin{cases} 1 \quad if \ \dfrac{T}{E[service]} \geq 1; \\ \dfrac{T}{E[service]} \quad otherwise. \end{cases} \tag{3.8}$$

The job inter-arrival time follows exponential distribution with a mean of $1/\lambda$, where λ is the average arrival rate of incoming jobs. The probability of i jobs arriving in a control period is:

$$P_i(i) = \frac{e^{-\lambda T}(\lambda T)^i}{i!} \tag{3.9}$$

The model proposed in this work is a continuous controlling process, where we dynamically allocate the active server number and adjust their frequency. The transition time for server allocation and frequency adjustment cannot be ignored, which will influence the performance greatly during the transition period. Let T_r and T_f denote the transition time of server mode change and frequency adjustment respectively. Given s jobs are in the queue and m to m' servers are in active mode for $m, m' \in \{1, 2, \cdots, M\}$, the probability that d jobs leave the system in a control period when action $a = \{M_a, F_a\}$ is taken can be obtained from:

$$P_d(d) = \begin{cases} \frac{T_r}{T} \begin{pmatrix} m \\ d \end{pmatrix} (\beta)^d (1-\beta)^{m-d} + \frac{T-T_r}{T} \begin{pmatrix} m' \\ d \end{pmatrix} (\beta)^d (1-\beta)^{m'-d} & M_a \geq 0; \\ \begin{pmatrix} m' \\ d \end{pmatrix} (\beta)^d (1-\beta)^{m'-d} & M_a < 0. \end{cases} \tag{3.10}$$

where $d \in \{0, 1, 2, \cdots, min(S, m')\}$.

The transition probability P_{xay} can be finally summarized as:

$$P_{xay} = \begin{cases} P_{s,s'}(a) & if\ m' = m + M_a \ \ and \ \ f'_i = f_{i+F_a}; \\ 0 & otherwise. \end{cases} \tag{3.11}$$

3.2.3 Objective and Constraints

Our model is built with the objective of minimizing energy consumption under Quality of Service (QoS) constraints. Both are the functions of state X and action A. We denote $C(x_T, a_T)$ as objective and $R(x_T, a_T)$ and $Pb(x_T, a_T)$ as performance constraints.

Power modeling. Understanding the power-to-frequency relationship of an individual server is critical for designing a power aware system. There are two widely used power models in a single server: linear and cubic. Both have been widely studied in related work [8, 55–58]. We adopt a cubic power model in our theoretical framework. The power consumption for a single node at frequency f is: $P_{act} = c_0 + c_1 f^3$, where c_o is a constant that includes the power consumption of all components except the CPU, and the base power consumption of CPU. The second term is the power consumption of CPU running at frequency f. Note: the chosen of power model does not affect the overall power model proposed in this work. We denote P_{ftrans} and T_f as the power consumption and transition time for frequency adjustment. So the power consumption for a single server can be expressed as shown in Equation (3.12). In this definition, the first term is the power before the frequency adjustment is completed; the second term is the power after the server frequency transfers from f_i to f_{i+F_a}; the last term is the frequency transition power.

$$P_{act} = \frac{T_f}{T}(c_0 + c_1 f_i^3) + \frac{T - T_f}{T}(c_0 + c_1 f_{i+F_a}^3) + \frac{T_f}{T} P_{ftrans} \tag{3.12}$$

We set the power consumption as the objective measure in our CMDP model. The immediate power consumption is consisted of three parts: active server power consumption, inactive server power consumption and transition power caused by server number allocation. The power consumption of the whole system in a control period T can be expressed as follows:

$$C(x_T, a_T) = \begin{cases} P_{act}(mT + M_a(T - T_r)) + (M - m - M_a)P_{in}T + M_a P_{trans} T_r & M_a \geq 0. \\ (m - |M_a|)TP_{act} + ((M - m)T + |M_a|(T - T_r))P_{in} + |M_a| P_{trans} T_r & M_a < 0. \end{cases} \tag{3.13}$$

where P_{act} and P_{in} are the power consumption of an active node and an inactive node respectively. P_{trans} is the power consumption when a server change between active mode and inactive mode. T_r is the transition time. It is clear that transition overhead is taken into account for power modeling in Equation (4.3).

The long term objective measure is defined as follows:

$$C_{ea}^{n}(x_T, a_T) = \frac{1}{n} \sum_{T=1}^{n} C(x_T, a_T) \tag{3.14}$$

where $C_{ea}^{n}(x_T, a_T)$ is the average power consumption in n control periods.

Performance constraints. We choose request time as the performance metric in our model because it is of the most concern from the users' perspective. We also ensure job blocking probability to be within a threshold to obtain high quality of service.

Little's law relates two important measurement: average waiting time and average number of jobs waiting in a queue in a service system [69]. It states that the average number of jobs L waiting for processing in a system is the product of the long-term average arrival rate λ and the long-term average waiting time of a job in the system W: $L = \lambda W$. With the help of it, we can derive the immediate waiting time from the long-term waiting time. After applying a round-robin dispatching method, the immediate request time considering the server busy probability can be expressed as follows [70]:

$$R(x_T, a_T) = \begin{cases} \dfrac{s-\rho}{\sum_{i=1}^{Imax} i \cdot P_i(i)} + \dfrac{E[service]}{c} & s-\rho > 0; \\ 0 & Otherwise. \end{cases} \tag{3.15}$$

In the top of (3.15), the first term is waiting time according to Little's Law, where $\rho = \lambda E[service]/mc$ is the server busy probability; the second term is job service time. The performance measurement involves all the state information. Let's denote $R^n_{ea}(x_T, a_T)$ as the average request time in n control periods. The performance measure in the long term can be defined as follows:

$$R^n_{ea}(x_T, a_T) = \frac{1}{n}\sum_{t=1}^{n} R(x_T, a_T) \tag{3.16}$$

For any state $x = \{s, m, f_i\}$, job blocking could occur when there are more than S jobs that need to be stored in the queue, i.e. $i + s - d \geq S$. The job blocking probability Pb can be derived as:

$$Pb(x_T, a_T) = \begin{cases} 0 & if \quad S - s + 1 \geq I_{max}; \\ \sum\limits_{d=0}^{max(m,m')} P_d(d) \cdot P_i(S + d - s + 1) & for \;\; S + d - s + 1 \leq I_{max}. \end{cases} \tag{3.17}$$

The blocking probability in long term is defined as:

$$Pb^n_{ea}(x_T, a_T) = \frac{1}{n}\sum_{T=1}^{n} Pb(x_T, a_T) \tag{3.18}$$

Thus, the optimization problem can be formulated as minimizing the power consumption with constraints of request time and job blocking, all the objective and constraints are related to state and actions.

$$\begin{aligned} Min: &\quad C^{n}_{ea}(x_T, a_T) \\ S.t. &\quad R^{n}_{ea}(x_T, a_T) \leq R_{max} \\ &\quad Pb^{n}_{ea}(x_T, a_T) \leq B_{max} \end{aligned} \tag{3.19}$$

where R_{max} is the maximum average request time, B_{max} is the maximum average blocking probability.

3.3 Markov Control Policy and Adaptive Power Management

3.3.1 Markov Control Policy

The most critical part of CMDP is to specify the policy by which the controller chooses action at different states. In our CMDP model, we first obtain an optimal stationary policy, and then the stationary policy is converted to a stationary deterministic policy by applying a maximal action probability strategy. We will first explain how to obtain the optimal stationary policy.

Let $f(a)$ be the probability distribution which determines the probability of taking action a

at state x under a stationary policy $a = \pi(x)$. The optimization problem presented in (3.19) is then converted to obtaining the optimal stationary policy $a = \pi^*(x)$:

$$\begin{aligned} Min: \quad & C_{ea}^{n}(\pi) \\ S.t. \quad & R_{ea}^{n}(\pi) \leq R_{max} \\ & Pb_{ea}^{n}(\pi) \leq B_{max} \end{aligned} \tag{3.20}$$

Linear programming (LP) is an optimization technique of a linear objective function, subject to linear equality and linear inequality constraints [71]. The optimal stationary policy of CMDP can be obtained with the help of LP [71]. Let $\rho(x,a)$ denote the steady state probability over the set of state-action pairs corresponding to the optimal stationary policy π^*. It is proven that the objective and constraint in (3.20) corresponding to the optimal policy are determined by the immediate power and performance cost with respect to the probability $\rho(x,a)$. The CMDP then can be converted to a LP problem as follows:

$$\begin{aligned} Min: \quad & \sum_{x \in X} \sum_{a \in A(x)} C(x,a)\rho(x,a) \\ S.t. \quad & \sum_{x \in X} \sum_{a \in A(x)} R(x,a)\rho(x,a) \leq R_{max} \\ & \sum_{x \in X} \sum_{a \in A(x)} Pb(x,a)\rho(x,a) \leq B_{max} \\ & \sum_{a \in A(y)} \rho(y,a) = \sum_{x \in X} \sum_{a \in A(x)} P_{xay}\rho(x,a) \\ & \sum_{x \in X} \sum_{a \in A(x)} \rho(x,a) = 1 \end{aligned} \tag{3.21}$$

Let $\rho^*(x,a)$ be the optimal stationary solution of the LP in (3.21). The corresponding optimal stationary policy of CMDP can be expressed as [71]:

$$f(a=\pi^*(x)) = \frac{\rho^*(x,a)}{\sum\limits_{a\in A(x)} \rho^*(x,a)} \tag{3.22}$$

for $\sum_{a\in A(x)} \rho^*(x,a) > 0$. Otherwise, we specify a performance guaranteed first policy:

$$f(a=\pi^*(x)) = \begin{cases} 1 & for\ \ M_a = max(0, s-m)\ \ and\ F_a = F-i; \\ 0 & otherwise. \end{cases} \tag{3.23}$$

The stationary deterministic policy $f_d(a)$ can be obtained from the optimal stationary policy. We set our stationary deterministic policy $f_d(a=\pi_d^*(x)) = 1$ if $f(a=\pi^*(x))$ as the maximal probability for all $a \in A(x)$, otherwise $f_d(a=\pi_d^*(x)) = 0$. In each control period, our online power controller can easily make an action decision according to the current state and the deterministic policy. This is a one to one mapping with the online computation complexity of $O(1)$.

3.3.2 CMDP Based Adaptive Online Power Control.

Workload often fluctuates over time with dynamic characteristics. To deal with the changes of the computing workload, we propose a CMDP based adaptive power control algorithm as shown in Algorithm 1. This algorithm starts with an offline system initialization process by analyzing the workload arrival rate and creating a set of CMDP control tables.

Workload typically exhibits periodic pattern, which makes workload prediction possible from analyzing the history data. In this work, we do not focus on workload prediction. We assume that the workload arrival rate is the pre-knowledge and take the workload trace as input. Given a workload trace data, its arrival rate varies between λ_{min} and λ_{max}. We first discrete the workload arrival rate from a continuous space to a discrete space and create a set of CMDP control tables $\Pi_d^*(X,\Lambda)$. Each element λ_i in the discrete space corresponds to a CMDP control table $\pi_d^*(x,\lambda_i)$. At the beginning of each control period, the power controller observes the system state and the system workload arrival rate. It checks the CMDP control table and makes corresponding action $a = \pi_d^*(x,\lambda_i)$. By this way, the power controller can make dynamic action decisions with the changes of workload arrival rate.

However, it is not enough to just make an initial decision at the beginning of each control period. To ensure high performance and system reliability, we propose an online server adjustment strategy. The basic idea is that when the number of jobs in the queue approaches the maximum, there is a higher probability of system performance degradation. For example, the increase of waiting time or queue blocking probability. Therefore, we increase the system computation capacity before the queue length reaches its maximum. A parameter $S_{threshold}$ is set to determine when to trigger the online server status adjustment process. For each new arrival job stores in the queue, the power controller checks if the number of jobs in the queue is larger than $S - S_{threshold}$, if so, the power controller tries to increase the system computation capacity by frequency adjustment. Frequency adjustment can be accomplished in a short period of time. It involves small transition overhead. If all the servers in the system are currently running at the highest frequency level, the power controller will switch a server from inactive to active mode.

Algorithm 1 CMDP Based Adaptive Power Control Algorithm

```
Offline system initialization
1. Workload arrival rate $\lambda$ analysis: $\lambda_{min} \leq \lambda_i \leq \lambda_{max}$.
2. Discrete the workload arrival rate range from continuous space to discrete space.
3. Create a set of CMDP control tables $\Pi_d^*(X,\Lambda) = \{\pi_d^*(x,\lambda_{min}),\cdots,\pi_d^*(x,\lambda_i),\cdots,\pi_d^*(x,\lambda_{max})\}$
for Each control period $T$
  1. Workload arrival rate $\lambda_i$ prediction.
  2. Choose an action according to CMDP control tables based on current system state $x$ and workload $\lambda_i$: $a = \pi_d^*(x,\lambda_i)$.
  3. Turn servers ON/OFF and adjust their frequency levels according to step 2.
  4. Send requests to the servers with Round-robin policy.
  for each new arrival job arrives at the queue
    if Queue length $s$ is larger than $S - S_{threshold}$.
      if All active servers are running at the highest frequency.
      Switch an inactive server to active mode and set its frequency to the lowest level.
      else
      Select one server and adjust its frequency to a higher level.
      end if
    end if
  end for
  accumulate system energy consumption
end for
report system performance and energy consumption
```

3.4 Evaluation

We built a simulator to evaluate the performance of the CMDP based adaptive power management strategy. It consists of a job generator, a power controller and several homogeneous servers. The simulator takes workload trace and the CMDP control tables as inputs and outputs the system energy consumption and performance.

3.4.1 Methodology

To better evaluate the CMDP based adaptive power control algorithm, the evaluations are performed in two simulation environments: small server clusters and large server clusters. The evaluation in small server clusters is aimed at verifying the correctness of the constructed CMDP model, while the large one is aimed at evaluating its robustness to workload. Both simulation environments are sharing the following parameter settings: each server has $F = 4$ levels adjustable frequency $f_i \in \{2.08, 2.25, 2.42, 2.6\}$. We adopt the power model of $P_{act} = 52.69 + 2.66f^3$ [8] in a single server. The inactive server power consumption is $P_{in} = 10$ watts. The server ON/OFF transition power P_{trans} and frequency adjustment transition power P_{ftrans} are set to $P_{trans} = 50$ watts and $P_{ftrans} = 10$ watts, respectively. The length of each control period is set to: $T = 60s$. The transition time is $T_r = \frac{T}{2} = 30s$ for the server switching between active mode and inactive mode. The transition time for frequency adjustment is set to $T_f = \frac{T}{10} = 10s$. Each server has a processing rate P_{rate} proportional to its frequency level: $P_{rate} = 6 \cdot \frac{f_i}{f_F}$ seconds per unit job. The maximum average request time is $R_{MAX} = 50s$. The job blocking probability could not exceed $B_{max} = 1\%$. At the beginning of each control period, the power controller observes the current system status and makes the control decision according to the deterministic CMDP policy. Each server will change its active or inactive modes and frequency levels according to the power controller's decision. The new arrival jobs will first be stored in the queue or queues waiting for processing. A round-robin dispatching discipline is applied to distribute the jobs in the queue for processing.

3.4.2 CMDP Model Evaluation

Parameter setting and baselines : The experiments illustrated in this subsection are performed in a small server cluster environment. Workload traces are generated by the job generator with an average arrival rate of $\lambda = 3 jobs/minute$. The incoming jobs followed Bounded Pareto distribution with an average job size of $E[z] = 5$. The maximum queue length is set to $S = 9$. We first evaluate the correctness and effectiveness of our CMDP model by varying the number of servers in the cluster. We compared the CMDP model with two baseline models with respect to performance, power consumption and online computation time. The first baseline model is the VOVF based CMDP model as describe in [67], in which only the VOVF strategy is applied in constructing the CMDP model; the second one is an online optimization model in [72] where the optimal controller dynamically changes the number of active servers and adjusts their frequency levels in each control period.

Sensitivity to server numbers: We show the average request time and the power consumption as the number of servers change from 2 to 9 respectively in Figures 3.2

and 3.3. As shown in Figure 3.2, all three models can achieve desirable performance. Specifically, the performance is not affected by the change of total server number in the server clusters, which proves the effectiveness of CMDP model. In Figure 3.3, the online optimization model contributes the most energy savings, followed by our CMDP model with an average of 3% more energy consumption, and the VOVF based CMDP model with an average of 8% more energy consumption. Although our CMDP model requires slightly more energy consumption compared to the online optimal model, it significantly reduces the online computation time as indicated in Figure 3.4. Figure 3.4 illustrates the runtime ratios between our CMDP model and the online optimal model. The waiting and service time are excluded. According to the results, the CMDP model requires 17% of the runtime required by the online optimal model when the server number is set to 2, and the ratio reduces to 3.67% when the server number is set to 9. The runtime ratio greatly decreases with the increase of server numbers. Based on those results, we can claim that our CMDP model can significantly reduce online computation time, especially when more servers are involved.

Sensitivity to job size: To further evaluate the proposed CMDP model, we vary the average job size. The power consumption and performance are given in Figures 3.5 and 3.6 respectively with respect to job size. Given the same average job size, the power consumption increases as the number of servers in the server cluster increases, which can be explained easily: inactive servers still consume a certain amount of energy. In Figure 3.6, the performance cannot be met when the average job size is set to $E[z] = 6$, and the server

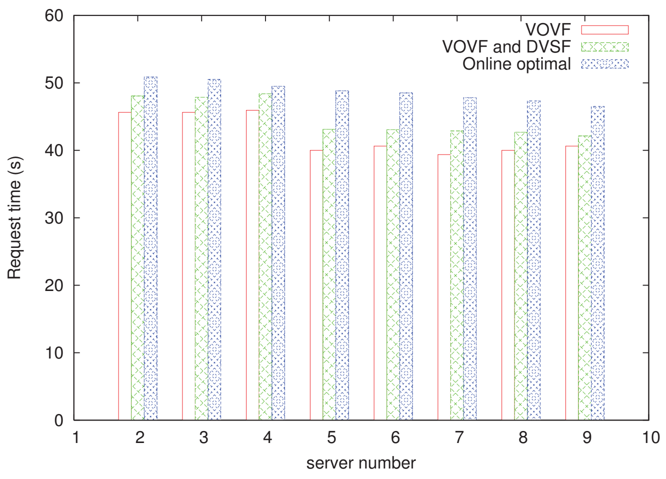


Figure 3.2: Request time comparison, the number of servers in the server cluster is varied between 2 to 9.

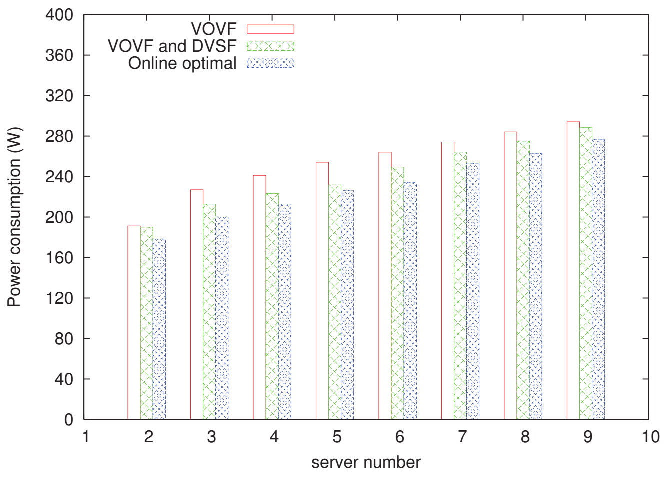


Figure 3.3: Power consumption comparison, the number of servers in the server cluster is varied between 2 to 9.

number is set to $M = 2$. The reason is that when the cluster only has two servers, there is not sufficient computation capacity to meet the performance constraint. So the server number in a cluster must be appropriate designed, not too large in order to achieve more energy savings, sufficient enough to satisfy the performance. Except the above situation, the performance constraint can always be met as we vary the server number and the average

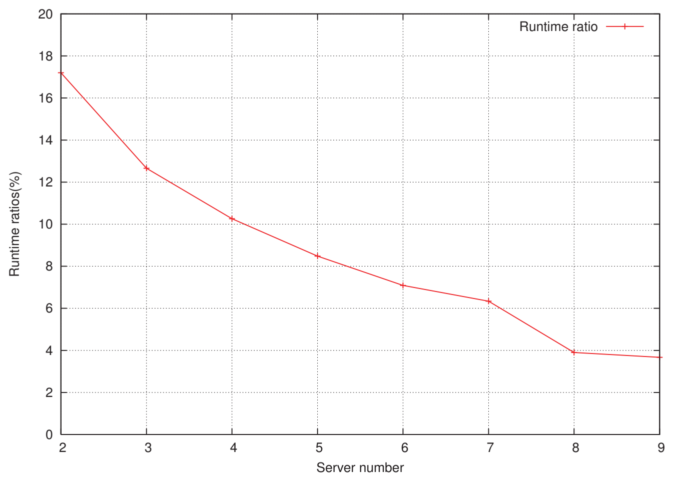


Figure 3.4: Runtime ratios between the CMDP model and Online optimal model.(%)

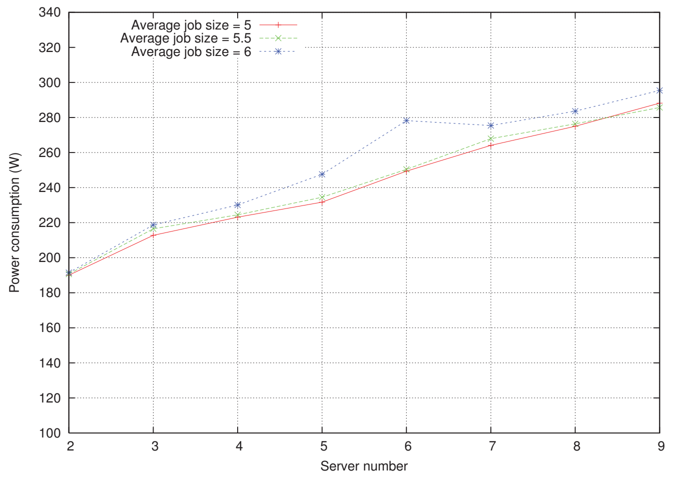


Figure 3.5: Comparison of power consumption when the average job size is set to be 5, 5.5, 6 respectively.

job size. However, larger job size requires more energy to achieve the desired performance. Larger job size will cause an increase in waiting time and queue length, in order to achieve the same performance, more servers and higher frequency levels are required to process the incoming jobs.

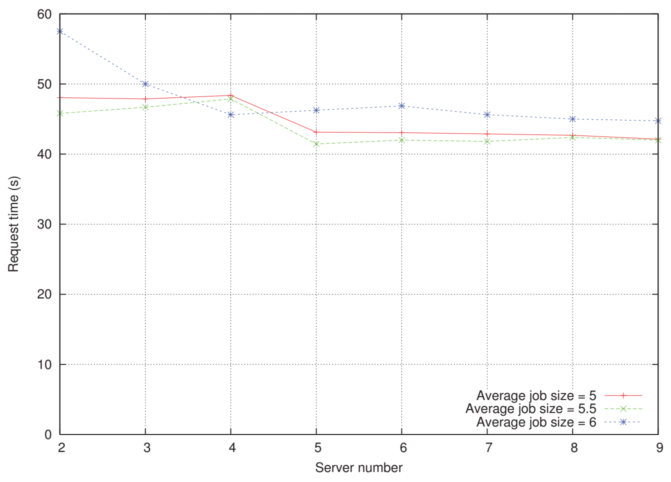


Figure 3.6: Comparison of request time when the average job size is set to be 5, 5.5, 6 respectively.

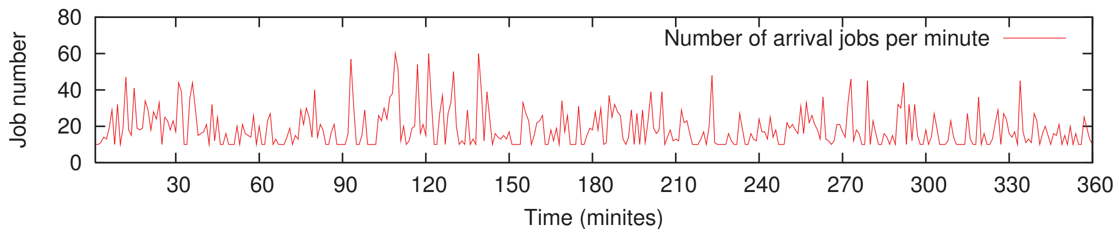


Figure 3.7: Number of arrival jobs per minute.

3.4.3 Workload Sensitivity Study

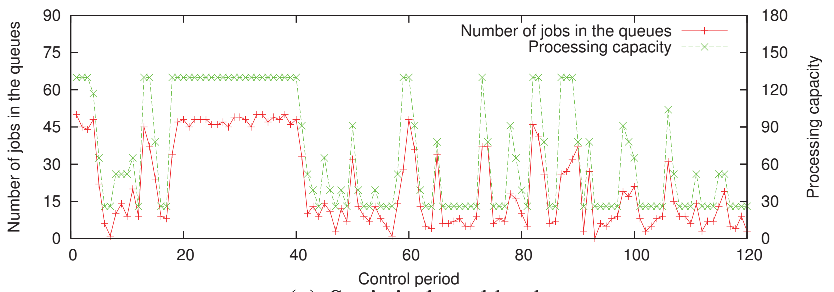
Parameter settings: The CMDP model is built based on a statistical distributed workload. To further evaluate our model, we also perform workload sensitivity study. In this simulation, we compare the power consumption and performance between real workload and statistical workload. The real workload trace is collected by Lawrence Berkeley National Laboratory [63]. The real workload data trace is normalized with an average job size of $E[z] = 5$, and the incoming arrival rate λ is varied with time as shown in Figure 3.7.

The maximum jobs arrival rate is $\lambda_{max} = 60 jobs/minite$, and the minimum jobs arrival rate is $\lambda_{min} = 10 jobs/minites$. The statistical workload is generated with the same arrival rate and job size. The simulation environment is consisted of 50 homogeneous servers and a power controller. There are 10 queues in the system, and each queue can store the maximum of 10 jobs. We discrete the workload arrival rate by the step of 0.5 and create 100 CMDP power control tables. The online server adjustment threshold $S_{threshold}$ is set to $S_{threshold} = 2$.

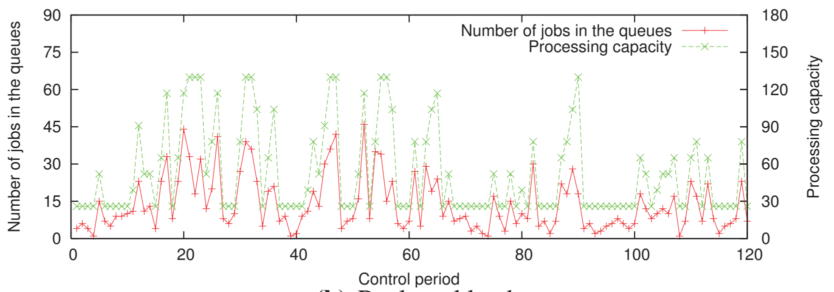
Statistical workload V.S. real workload: In order to make the results more apparent, we define the processing capacity of the server clusters as the accumulated processing capacity of each active server. The processing capacity of a single active server in a control period is defined as:$Capacity = \sum f_i \frac{T_i}{T}$, where T_i is the time for a server running at frequency f_i. Higher processing capacity means faster processing ability but more energy consumption. As we mentioned before, for each state $x = \{s, m, f_i\}$, m and f_i are deterministic given an action a. We only present the non-deterministic factor s to verify the correctness of the proposed control algorithm.

Figure 3.8 illustrates the number of jobs in the queue and corresponding processing capacity in two hours for both statistical workload and real workload. As illustrated in the figure, the processing capacities increase as the number of jobs in the queue increases, which means the power controller chooses an action to increase the processing capacity for achieving quality of service as the number of jobs in the queue increases. On the contrary, the power controller reduces the active server numbers and frequency levels if the number of jobs in the queue is small for more energy savings. In each control period, the power controller is trying to minimize the energy consumption and guarantee QoS at the same time. Figure 3.9 illustrates the instant power in two hours. As shown in Figure 3.9, our algorithm can achieve significant energy savings under low workload in both statistical workload and real workload, which will contribute to significant energy savings in a server cluster.

Figures 3.10 and 3.11 compare the hourly power consumption and the average request time between real workload and statistical workload. The red line in the figure represents the number of arrival jobs in an hour. The energy consumption is approximately proportional with the system workload, which proves that our CMDP based adaptive power control algorithm can effectively control the system energy consumption and performance with the change of system workload. The real workload requires more energy consumption when compared to the statistical workload. However, the performance constraint can still be met. This can be explained by the dynamic behavior as shown in Figure 3.8 and Figure 3.9. The real workload is more volatile, with more high-frequency variation of changing server modes in order to satisfy performance. It requires transition time and transition power and ultimately leads to a longer request time and more energy consumption compared to the statistical workload. Overall, our model can achieve desirable performance with real workload.

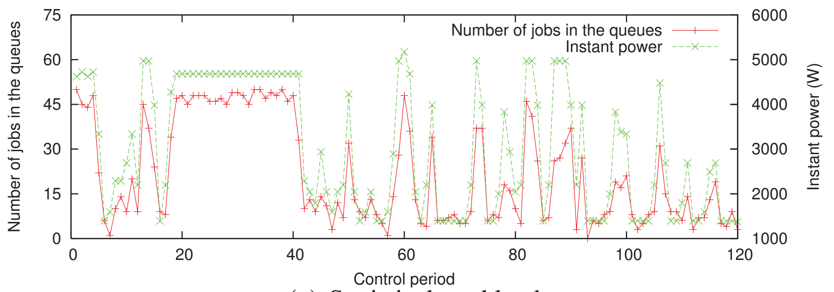


(a) Statistical workload

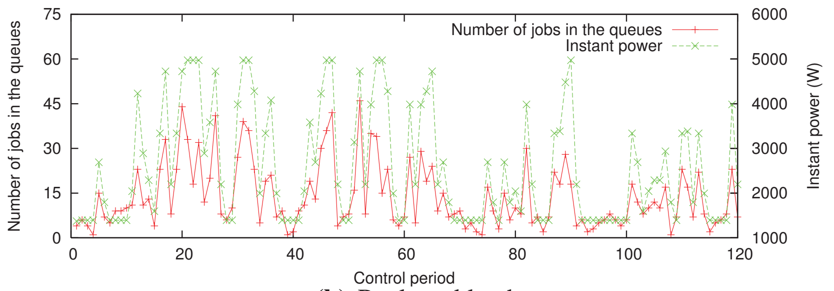


(b) Real workload

Figure 3.8: Instant processing capacity and queue length in two hours.



(a) Statistical workload



(b) Real workload

Figure 3.9: Instant power consumption and queue length in two hours.

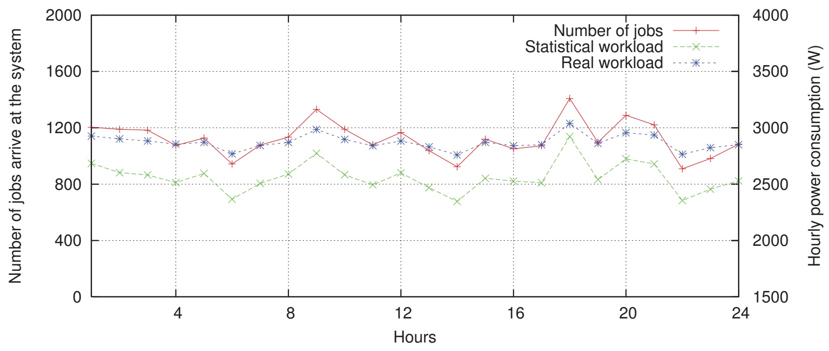


Figure 3.10: Comparison of power consumption between statistical workload and real workload.

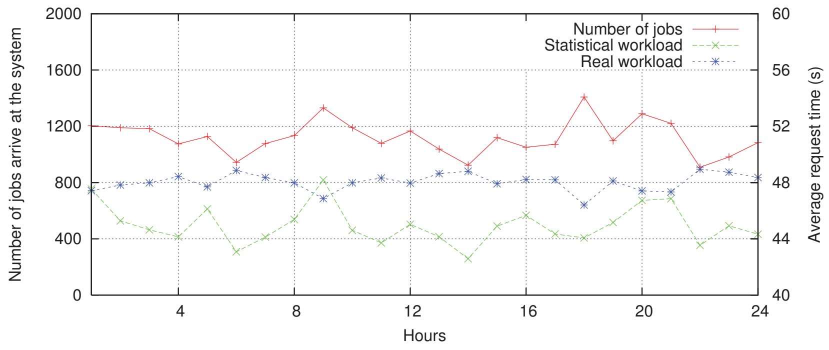


Figure 3.11: Comparison of request time between statistical workload and real workload.

Sensitivity to parameter: Next, we study the parameter for online server status adjustment. We compare the power consumption and the average request time by varying the control parameter between 0 to 3. As shown in Figures 3.12 and 3.13, the red lines represent the results when the threshold is set to $S_{threshold} = 0$. In this scenario, there is no further server status adjustment process after the initial CMDP action decision has been made. The QoS violations in Figure 3.13 indicate that the CMDP control policy did not allocate enough system processing capacity to achieve the QoS target. It verifies that the online server status adjustment process is not only necessary but important in order to ensure the QoS. As we increase the threshold $S_{threshold}$ to 1, there are no significant

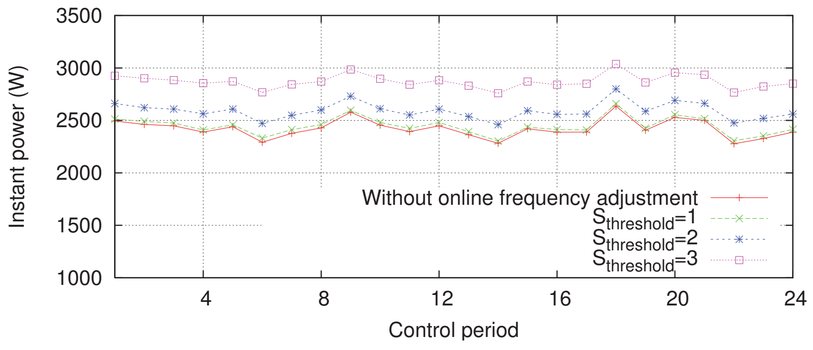


Figure 3.12: Comparison of power consumption between statistical workload and real workload.

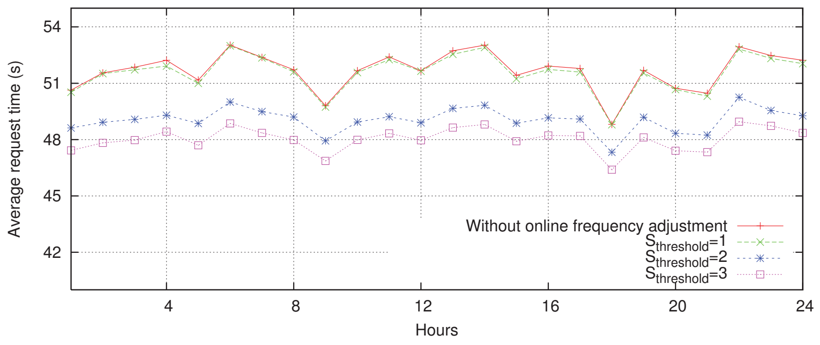


Figure 3.13: Comparison of request time between statistical workload and real workload.

changes for both power consumption and performance. When the threshold $S_{threshold}$ is set to 2 and 3, we can obtain the satisfied performance. However, more energy is required when increasing the threshold. From the above observations, we conclude that the online server status adjustment process can effectively improve the system performance with the dynamic workload. However, the control parameter should be carefully set for both energy savings and performance constraint. The best parameter setting can be obtained by the online tanning.

3.5 Future Work

Hardware integration. A data center is a highly complex system with complex relationships between hardware and software. The computing capacity of IDCs are usually over-provided for the sake of reliability and availability. The software based power management strategies can effectively reduce the unnecessary energy consumptions by allocating the least possible computing resources with the change of workload. However, software energy management strategy is ultimately limited by hardware. A recent study also shows that choosing the right hardware would save more energy than state-of-art power management software [73]. For example, a single desktop with a low-power embedded computer sleep proxy, can keep machines in sleep for up to 50% of the time while providing uninterrupted network access [73]. Moreover, software techniques add more complexities on top of the computing systems. The complexities can be reduced by redesigning the hardware itself. Therefore, software and hardware integration will bring more opportunities to reduce the data center energy consumption.

Chapter 4

Reducing Operational Costs in Geographically Located Internet Data Centers[3]

4.1 Introduction and Related Work

Online service providers(OSPs) have Internet data centers in multiple geographical locations in order to satisfy global user demand; however, the dynamic variation of the

[3]This chapter is based on the works from X. Zheng and Y. Cai, Energy-aware load dispatching in geographically located Internet data centers. *Sustainable Computing Informatics and Systems*, Vol. 1, No. 4, pp.275-285, 2011. X. Zheng and Y. Cai, Reducing Electricity and Network Cost for Online Service Providers in Geographically Located Internet Data Centers, *Proc. IEEE/ACM International Conference on Green Computing and Communications (GreenCom2011)*, Aug. 2011.

electricity market is ignored. Furthermore, increased data centers require more energy supply, more network usage, and at the same time causes increased heat dissipation, greater cooling requirements, reduced computational density, and higher operating costs [2]. It places a heavy burden on both environment and energy resources.

Energy related costs, on the other hand, have become one of the most important economical factors for IDCs. Moreover, the reduction of the carbon emissions has also become a changing task for OSPs [21]. Electricity is produced via a variety of energy sources, which can be categorized into renewable and nonrenewable sources. Electricity generated by nonrenewable sources carry large carbon footprints, which we refer to as Brown energy. Renewable energy sources can be replenished in a short period of time and also have less environmental impact, which is we refer to as Green energy [74]. Congress is working on establishing an emission limit for large carbon emission firms including OSPs to deal with the carbon footprint problem [75]. The dilemma is left to the OSPs to determine how to purchase their power supplies in an economic and sustainable manner. Also, electricity price varies with both location and time because of the energy sources in different regions. The geographical distribution of data centers therefore exposes many opportunities to reduce the electricity cost. Besides electricity cost, increased Internet services require more network usage, which also accounts for a large portion of operation cost for OSPs. Much of the existing work on power management focused only on a single data center [14, 15]. A few work considered the variation of the electricity market and addressed load distribution across data centers with respect to energy consumption or energy cost [5, 76]. However they only focus on controlling the energy consumption of the server sub-systems, less attention has been paid to the dynamic behavior of cooling systems. The impact of energy sources is rarely addressed [22]. The network cost is not well studied in the literature for reducing the total cost.

With the above observations, this work develops an optimal energy-aware load dispatching strategy that periodically maps more request to the locations with lower electricity price. Energy proportional and chiller ON/OFF strategies are applied to save energy consumption in each IDC. We also place a carbon emission limit and consider the volatility of the carbon offset market, which encourages OSPs to control their carbon emission from the financial perspective.

In order to better present the research work in this work, the topics of power management and cost management in data centers will be reviewed.

Power management in dater centers There are two main strategies for power reduction in server system: Dynamic Voltage/Frequency Scaling (DV/FS) and server number controlling: Vary-On Vary-Off (VOVF). DV/FS works by reducing the voltage and frequency, consequently saving power at the cost of slower program execution. Researchers have developed various DV/FS scheduling algorithms to save energy under timing deadlines [12, 13]. Some researchers also utilized feedback control to dynamically adjust server frequency [37]. In these works, control variables can be either server frequency or application-level quality of service requirements [37–39]. VOVF is a key mechanism for power reduction applied in server clusters [14–16]. VOVF dynamically turns idle servers off when the system experiences a light workload, and turns the appropriate servers on when the system encounters a heavy workload. VOVF dramatically improves the system energy efficiency by reducing the idle server power consumption. Virtualization as a key strategy to reduce power consumption for application services, which is another way of VOVF. When applying virtualization, multiple virtual servers can be hosted on a smaller number of more powerful physical servers, using less electricity [41]. In [42], researchers demonstrated a method to efficiently manage the aggregate platform resources according to the guest virtual machines (VM) relative importance (Class-of-Service), for both the black-box and the VM-specific approach.

Increasing computation capabilities in IDCs results in higher cooling energy requirements [77]. There are several works attempting to reduce the energy consumption in the cooling sub-system. In [78], the authors explored the physics of heat transfer, and presented methods for integrating it into batch schedulers. It reduced the amount of heat recirculation in the data center and improved the cooling sub-system efficiency. A mathematical scheduling problem is formulated in [79] to minimize the data center cooling cost, they also provided two heuristic methods XInt-GA and XInt-SQP to solve the problem. In [80], researchers present a unified, coordinated, thermal-computational approach to the the problem of IDCs energy management. Another group of researchers formulated an optimization problem to reduce the power consumption in servers and cooling system by selecting frequency level and cold air supply [81]. An integer linear programming was applied to solve the problem.

This work differs from these efforts: all of the above work focus on a single data center, none of them considered multi-mirror services and their request distribution, and how it influence on the total cost for OSPs.

Leveraging variability electricity price in reducing cost In [5], the researcher first considered the variable electricity prices for data centers and proposed a scheme to shut down the data center when the electricity price is high. Qureshi et al [82] proposed a load dispatching strategy to reduce total electricity cost. An optimization problem was formulated in to minimize the electricity cost in geographical located IDCs [76]. In [22], researchers considered the problem of capping the brown energy consumption and interacting with the carbon market. However, existing efforts focus narrowly on electricity usage of the server sub-system, without considering the dynamic behavior of the cooling system and how to leverage it to reduce cost. Also, the network cost is not well studied in relation to reducing the total operational cost. The contribution of this work is that we provide a precise modeling of electricity usage in IDCs and provide efficiency strategies in both server and cooling systems in addition to leveraging variability of electricity price. The network cost is also considered to obtain the optimal load dispatching among IDCs.

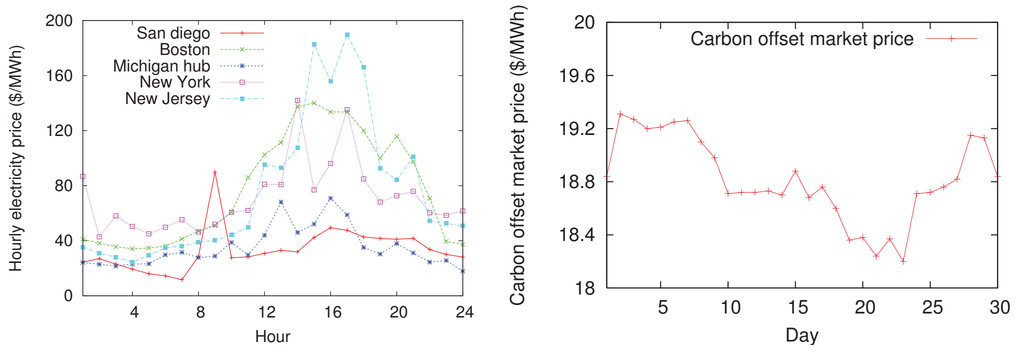
Table 4.1
Electricity generation by energy source(%) from 2008 to 2010.

Period	Coal	Petroleum	Natural Gas	Nuclear	Hydroelectric	Other Renewable	Others
2008	49.5	1.1	19.5	19.7	6.7	3.2	0.3
2009	45.2	1.2	20.8	21.1	7.5	3.8	0.4
2010	46.3	0.9	21.2	20.3	6.7	4.2	0.4

4.2 Background

Electricity Generation Although electricity is a relatively clean and safe form of energy to use, the production and transmission of electricity do have environmental impacts [74]. Electricity is produced via a variety of energy sources, which can be categorized into renewable and nonrenewable sources. Electricity generated by nonrenewable sources carry large carbon footprints, which we refer to as Brown energy. Renewable energy sources can be replenished in a short period of time and also have less environmental impact, which is we refer to as Green energy [22]. Table 4.1 gives the detailed information of electricity generation sources in the U.S. from 2008 to 2010. As shown in the table, nearly 50% of the electricity is generated by coal. Other "Brown" energy sources like natural gas and nuclear also place an important role of electricity production, which is about 35%. About 10% of electricity is generated by "Green" energy sources. Investment in and use of "Green" energy brings benefits to both our planet and the next generation [83].

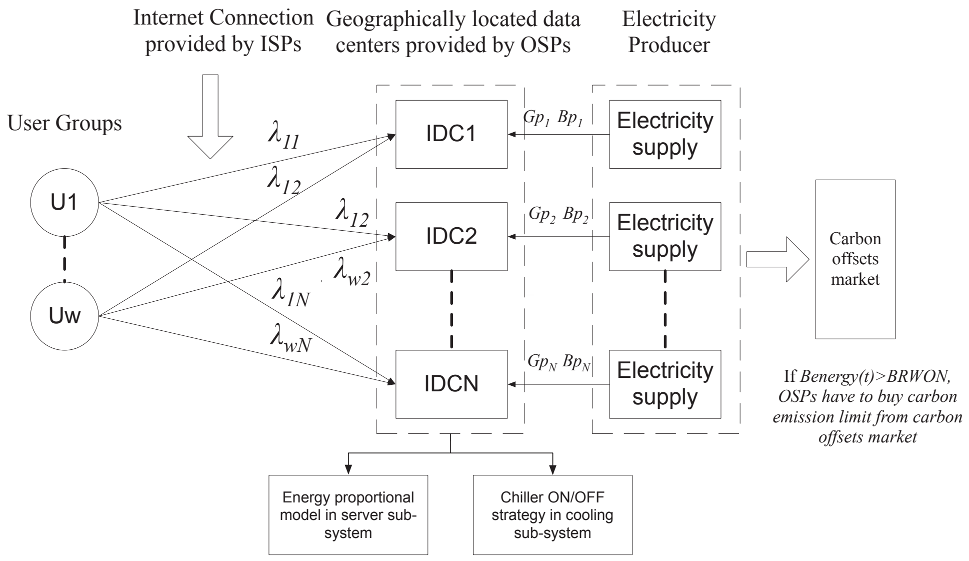
Electricity and Carbon offsets Markets. There are ten electricity markets with varying degrees of inter-connectivity in the United States [74]. Electricity price exhibits both location and time diversities because of the electricity generation, price regulation and other factors. Figure 4.1(a) reflects the real-time hourly electricity price in five selected market in different regions on Sep.1st 2010 [83]. As shown in the figure, the electricity price is higher in the afternoon and early evening, which means the demand is higher in those time periods [83].



(a) Hourly real-time electricity price from different electricity markets on Sep.1st 2010. **(b)** Daily real-time carbon offset market price in Sep. 2010.

Figure 4.1: Electricity and carbon offset market prices

Carbon offset market is used to transfer permits of emissions, which provided economic incentives to reduce the emissions of large emissions pollutants like OSPs. A limit or cap on the amount of carbon emission for a pollutant that can be emitted is set by a central authority (e.g. government). Any pollutant has to hold the permits, if more emission permits are required besides the cap, pollutant has to buy permits from the carbon offset market. The carbon offset trade is intended to encourage the pollutants to reduce emissions from the economic perspective. Figure 4.1(b) illustrate how this market behaves [84](data are normalized from €/ton to $/MWh). Although the carbon market price shows less



Internet Connection provided by ISPs
Geographically located data centers provided by OSPs
Electricity Producer
User Groups
U1
Uw
λ_{11}
λ_{12}
λ_{12}
λ_{w2}
λ_{1N}
λ_{wN}
IDC1
IDC2
IDCN
Gp_1 Bp_1
Gp_2 Bp_2
Gp_N Bp_N
Electricity supply
Electricity supply
Electricity supply
Carbon offsets market
If Benergy(t)>BRWON, OSPs have to buy carbon emission limit from carbon offsets market
Energy proportional model in server sub-system
Chiller ON/OFF strategy in cooling sub-system

Figure 4.2: Internet data centers architecture.

variability compared to electricity price, however it is still changed appreciably over time. The observation suggests that it is worthwhile to consider the volatilities of both electricity price and carbon market price in our optimization modeling.

4.3 Problem Overview

4.3.1 Notations

For better understanding of our model, Table 4.2 summarizes the notations and definitions which will be used throughout this chapter.

Table 4.2
Notations

Notation	Definition
N	Total number of Internet data centers
Λ_j	Maximum workload arrival rate for data center j
M_j	Total number of servers in data center j
m_j	Active server numbers in data center j
a_j	Energy proportional slope (KWh/requests) of data center j
b_j	Energy proportional coefficient (KWh) of data center j
Pac_j	An active server power consumption (KWh) in data center j
Pin_j	An inactive server power consumption (KWh) in data center j
$Pnet_j$	Network equipment power consumption(MWh) in data center j
$PUPS_j$	UPS sub-system power consumption(MWh) in data center$j$
P_j	IT equipment sub-system power consumption(MWh) in data center j
U_j	System utilization (%) in data center j
$Pcooling_j$	Cooling sub-system power consumption(MWh) in data center j
$c1_j, c2_j, c3_j$	chiller power consumption coefficients in data center j
$PCARC_j$	CARC system power consumption in data center j
$Green_j(t)$	Green supplied electricity price ($) in data center j at time t
$Brown_j(t)$	Brown supplied electricity price ($/MWh) in data center j at time t
$Bp_j(t)$	Brown power supply mix in data center j at time t
$Gp_j(t)$	Brown power supply mix in data center j at time t
$Market(t)$	Carbon offset market price ($/MWh) at time t
$Benergy(t)$	Accumulated brown energy consumption ($/MWh) until time t
$BROWN$	Carbon emission cap
$Cost_j$	Total electricity cost($) of data center j in a control period.
W	Number of user groups
λ_i	Request arrival rate from user group i
Dis_{ij}	Distance (km) between user group i and data center j
$Delay_{ij}$	Round trip delay (ms) between user group i and data center j
C_{ij}	Network cost ($/request) for dispatching per unit request from user group i to data center j
$Ncost$	Total network cost ($) in a control period
λ_{ij}	The portion of workload dispatched from user group i to data center j

4.3.2 Problem Formulation

Figure 4.2 illustrates a typical IDCs network architecture of large OSPs. To satisfy global user demand, OSPs have Internet data centers in multiple geographic locations. They are fully inter-connected by Internet Service Providers (ISPs or their own backbone network) to carry traffic between the IDCs and their millions of users. In each data center j, it hosts a large number of servers M_j, the electricity supply is the mix of brown energy $Bp_j(t)$ and green energy $Gp_j(t)$ from local electricity provider. A limited carbon emission permit $BRWON$ is placed. If the total carbon emissions $Benergy(t)$ exceed the limit, OSPs have to buy the permit from the carbon offset market based on market price $Market(t)$. In order to simplify our network model, the users from the same state are grouped together as a single user i, previous work also utilized the same model [82]. Our optimization problem formulated as minimizing the overall electricity cost and network cost $Ncost$ as shown in Equation (4.1). The electricity cost is the summation of each IDC electricity cost $Cost_j$, which is a dynamic function of several factors: $Cost_j(t) = f(Bp_j(t), Gp_j(t), \lambda_{ij}(t), Market(t))$. In each IDC, two energy efficiency strategies are applied: energy proportional model in the server sub-system and chillier ON/OFF strategy in the cooling sub-system. For our optimization model to be practical, we guarantee high performance and availability for end users in addition to minimizing the total cost for OSPs. We evaluate performance by using the average end-to-end response time $AveDelay$, since it is the most concern from the users perspective.

$$\begin{aligned}
Min: \quad & \sum_{j=1}^{N} Cost_j + Ncost \\
S.t. \quad & \sum_{j=1}^{N} \lambda_{ij}(t) = \lambda_i \\
& \sum_{i=1}^{W} \lambda_{ij}(t) \leq \Lambda_j \qquad for\ any\ \ (1 \leq i \leq W, 1 \leq j \leq N) \\
& AveDelay \leq DMAX
\end{aligned} \tag{4.1}$$

We assume the optimization takes place in a centralized location, the optimal solution then delivers to the distributed IDCs. Our optimization controller dynamically makes the following decisions for the next hour in order to minimize the total cost for OSP: (1) determine the power mix portion: brown energy supply $Bp_j(t)$ and green energy supply $Gp_j(t)$ in each IDC; (2) distribute workload λ_{ij} from user groups i to data centers j; (3) calculate active server numbers m_j in each IDC; (4) determine chiller ON/OFF based on workload distribution; (5) monitor the average end-to-end response time; (6) prevent data center overload.

4.4 Cost and Performance Modeling

4.4.1 Electricity Consumption in IDCs

Construct a suitable model for data center power consumption is a challenging task because of the diversity and complexity of data center infrastructure. Our modeling and analysis is focused on the three sub-systems that account for more than 90% of the power consumption in an IDC [10]. The three sub-systems on which we focus are: IT equipment system, cooling system and power distribution system. Two external factors that primarily affect data center power usage are: the aggregate workload presented to the computing infrastructure and the outside air temperature (which mainly affects cooling sub-systems) [10]. We do not discuss the dynamic behavior of outside temperature in this work for simplicity. We construct our model by composing models for the individual data center sub-systems: $Ptotal_j = P_j + Pcooling_j + P_{UPS}$. The electricity cost for a single IDC j in an hour can be expressed as in Equation (4.2):

$$Cost_j = \begin{cases} (Gp_j(t)Green_j(t) + Bp_j(t)Brown_j(t)) * Ptotal_j \\ \qquad (if\ Benergy(t) \leq BROWN) \\ (Gp_j(t)Green_j(t) + Bp_j(t)(BROWN_j(t) \\ \quad + Market_j(t))) * Ptotal_j \ \ (Otherwise) \end{cases} \tag{4.2}$$
$$where\ \ Benergy(t) = \sum_{t=1}^{t} \sum_{j=1}^{N} Ptotal_j$$

Where $Brown_j(t)$ and $Green_j(t)$ are the brown and green electricity market prices for data center j. If the total brown energy consumption exceeds the emission limit $BROWN$, the OPSs has to buy carbon emission permits from the carbon offset market.

IT equipment energy consumption modeling. The UPS system is always on to supply power for servers and other IT components. Therefore, the UPS sub-system can be modeled by a fixed power draw [10]. We first detail the IT equipment sub-systems followed by cooling sub-system. The power consumption in the IT equipment sub-system mainly comes from two aspects: the power consumption of servers and networking hardware components. Unlike servers, the networking hardware has a fixed power draw and typically accounts for less than 6% of the IT power [85]. The total power consumption in the IT sub-system can be significantly reduced by applying energy efficient strategies in the server system. An important principle to improve energy efficiency is to ensure energy consumption proportionality, which states that the energy consumption of server system P should be proportional to the system workload λ [49, 68]: $P = a\lambda$. When considering the network equipment power consumption, the power consumption in IT equipment sub-system should be linearly increased with system workload: $P = a\lambda + b$. For simplicity, we assume all the servers in the data center are identical nodes. On a typical web server and web clusters, system workload can be described by the request arrival rate λ. Let M_j be the total number of servers in a data center j, and Λ_j be the maximum arrival rate for the data center to achieve desirable QoS. m_j is the total number of active servers. The total energy consumption of a data center can be expressed as follows:

$$P_j = m_j(Pac_j - Pin_j) + M_j Pin_j + Pnet_j \tag{4.3}$$

Pac_j is the power consumption of a fully active node, Pin_j is the power consumption of an inactive nodes. $Pnet_j$ is the network equipment power consumption. Based on the linear model: $b_j = M_j * Pin_j + Pnet_j$, which is the power consumption when there is no system workload. While the system reaches its maximal workload Λ_j, the power consumption can be represented as:

$$Pmax_j = M_j(Pac_j - Pin_j) + M_j Pin_j + Pnet_j = \Lambda_j a_j + b_j \tag{4.4}$$

According to equation 4.3 and 4.4, we can derive the energy consumption model of IT equipment sub-system as:

$$P_j = \frac{M_j(Pac_j - Pin_j)}{\Lambda_j} \sum_{i=1}^{W} \lambda_{ij} + M_j Pin_j + Pnet_j \tag{4.5}$$

With the help of 4.3 and 4.5, we can derive the total number of active servers in a data center as:

$$m_j = \frac{\sum_{i=1}^{W} \lambda_{ij}}{\Lambda_j / M_j} = \frac{U_j}{M_j} \tag{4.6}$$

Where U_j is the system utilization in data center j. The number of servers may not be an integer based on Equation (4.6). We will set the integer no less than m_j, which is the minimal number of servers to run in fully active mode.

Cooling sub-system power consumption modeling. The cooling sub-system evacuates large amount of heat produced by an IDC. Computer Room Air Conditioners (CRACs) and fans are used to remove hot air from servers on the data center floor and bring in fresh cooler air [77]. Conventional CRAC transfers heat from the air to fluid coolant that is then pumped to large chillers or cooling towers in another part of the facility [10]. The heat is expelled into the external atmosphere, and the cooled fluid is circulated back to the CRACs.

Generally speaking, the cooling sub-system electricity consumption increases with the amount of heat it needs to evacuate. Modern data centers use variable speed drive chillers and variable speed fans, which gives their cooling systems a large dynamic range. Chillers are the dominant consumers in the cooling system, so they can require more than three times as much power as the other cooling components [86]. In our power model, the power consumption for CRAC systems is simplified as a fixed power consumption *PCARC*. Given an outside temperature and a data center utilization level, the chillers can be turned off to ensure reliable operation with the most energy-efficient manner. We adopt a quadratic power model for the chiller power consumption as in [82]. So the total power consumption of cooling system is summarized as:

$$Pcooling_j = \begin{cases} c1_j U_j^2 + c2_j U_j + c3_j + PCARC_j & (if\ U_j \leq 25\%) \\ PCARC_j & (Otherwise) \end{cases} \tag{4.7}$$

4.4.2 Networking Cost Modeling

Network usage costs are a significant component of the total operating costs for OSPs [87, 88]. The cost of network link connected to ISPs is a function of traffic volume, i.e., $F(v)$, where F is non-decreasing cost function, and v is the charging volume of traffic. The cost function F is commonly of the form $price * v$, where $price$ is the unit traffic volume price of a link [88]. The charging volume v is based on actual traffic volume. The links between end users and data centers may be interconnected by multiple ISPs. So the cost varies among different paths. For simplicity, we define the total network cost as the sum of individual network usage costs. In each path, the cost is linearly increased with the traffic volume as in [82], the total network cost in an hour can be expressed as:

$$Ncost = \sum_{i=1}^{W} \sum_{j=1}^{N} \lambda_{ij}(t) C_{ij} \tag{4.8}$$

Although this is a highly simplified model of reality, there is evidence that such model results in a reasonable approximation to a proper network cost optimization [88].

4.4.3 Performance Modeling

Degraded system performance is known to result in lost revenue. In our optimization model, we select end-to-end response time as the metric of performance, which consists of network delay and response time inside an IDC.

We use geographic distance as a rough measure of network latency, which has been tested and applied as a simple model for modeling network delay [82]. The round trip time for a request from user group i to data center j is a linear function of its distance Dis_{ij}: $Delay_{ij} = c * Dis_{ij} + d$.

In a single data center, the incoming workload is coming from multiple user groups based on our optimization dispatching discipline, which is $\sum_{i=1}^{W} \lambda_{ij}$ for any data center j. The response time inside a data center is the summation of service time and waiting time. We obtain the average response time based on the queuing models presented in [50–53]. The average service time for the incoming packets is $E[X]$. According to Pollaczek-Khinchin formula, the average response time for the incoming packets in data center j with m_j active servers is:

$$E[R]_j = \frac{\sum_{i=1}^{W} \lambda_{ij} E[X^2]}{2(m_j - \sum_{i=1}^{W} \lambda_{ij} E[X])} + E[X] \tag{4.9}$$

Based on the above energy proportionality 4.6, equation 4.9 can be re-written as:

$$E[R]_j = \frac{E[X^2]}{2(M_j/\Lambda_j - E[X])} + E[X] \tag{4.10}$$

After we take network delay into account, the average end-to-end userâĂŹs delay can be summarized as:

$$AveDelay = \frac{\sum_{i=1}^{W} \sum_{j=1}^{N} (\lambda_{ij}(Delay_{ij} + E[R]_j))}{\sum_{i=1}^{W} \sum_{j=1}^{N} \lambda_{ij}} \tag{4.11}$$

4.5 Optimization Problem Solution

Workload and electricity price predictions are beyond the scope of our work. There are many tools and online services which can provide precise predictions. We only take the workload trace and electricity trace as inputs in our optimization process. However, the optimization problem cannot be solved linearly, since the optimization controller has to periodically determine both the power mix and workload fractions. In order to solve the optimization problem, we propose an adaptive optimization algorithm as shown in Algorithm 2. The dynamic optimization process starts by dividing the carbon emission cap into 12 pieces, one piece per month. The carbon emission cap for each month is then weighted by monthly workload intensity. The power mix is determined monthly based on an average monthly workload and electricity price instead of hourly as in workload fractions calculation. We claim it as a more practical solution, because an electricity purchase contract is usually set for a long period of time. At the beginning of each month, the power mix for each data centers is calculated from half portion of brown energy and half portion of green energy. The brown energy and the green energy portion are gradually increased and decreased the by 10% in the optimization process. If the total workload is less than 25% of the total IDCs processing capacity, all the chillers can be turned off to save cooling system energy. In this case, the workload fractions can be calculated by Linear Programming (LP) after the power mix is computed. Otherwise, the optimal controller will search for possible chiller ON/OFF combinations, and calculate workload distribution fractions using Quadratic Programming (QP). The minimal cost combination is the optimal solution for load distribution. We record the brown energy consumption at the end of each month and update the monthly carbon emission caps for the rest of the months.

Algorithm 2 Adaptive optimization Algorithm

```
system initialization
1.divide carbon emission cap into 12 pieces
2.weighted by monthly workload intensity
for each month
    calculate the power mix for each data center according to monthly input
    for each hour $i$
        if total workload is less than 25% system workload
            1. turn off the chillers in all the data centers
            2. calculate the fractions of workload distribution using LP
        else
            1. search for possible chiller ON/OFF combination in all the data centers according to total workload
            2. for each possible combination, calculate the fractions of workload distribution and total costs using QP
            3. search for the minimal cost workload distribution solution
            4. turn off chiller according to the minimal cost solution in step 3.
        send requests to data centers for processing
        accumulate brown energy consumption
    end for
    update carbon emission cap for the rest of months
end for
report system performance and total cost
```

4.6 Simulation Results

We built a simulator to evaluate our energy-aware optimization dispatching policy. Our simulator takes workload traces, electricity, carbon offset market price traces, and a carbon emission cap as inputs. It simulates the request distribution policy and outputs the performance and the total electricity and network cost.

Table 4.3
Data centers power consumption parameters setting

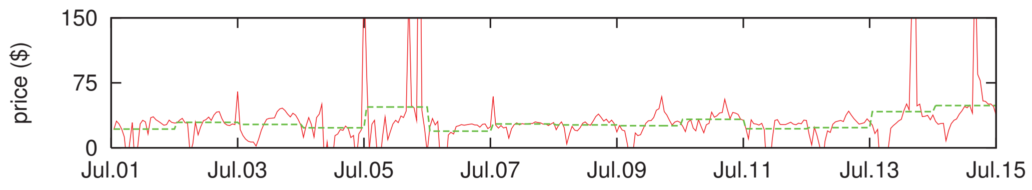
Data center	Location	Power supply market	Power consumption chillers on(KW/h)	Power consumption chillers off(KW/h)
1	South California	San Diego	$2.03\lambda^2+158.18\lambda+1166.7$	$100.5\lambda+940.36$
2	Central Michigan	Michigan hub	$0.96\lambda^2+132.8\lambda+1209.3$	$96\lambda+1036$
3	North New Jersey	New Jersey	$0.77\lambda^2+115.15\lambda+1423$	$78.5\lambda+1206$

4.6.1 Parameters Setting

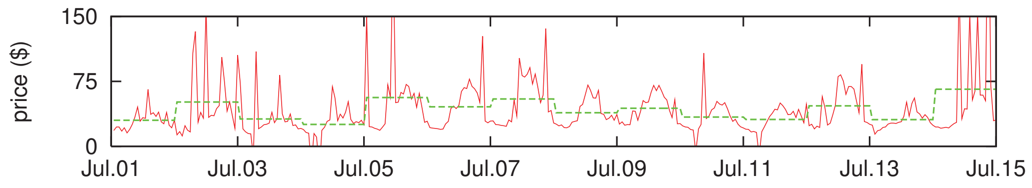
We simulated two user groups from the States of Texas and Georgia in the U.S. and three data centers. We denote the requests from Texas as user group 1 and those from Georgia as user group 2. The detailed information and parameter settings of the three data centers are shown in table 4.3. The maximal processing capacities are set as: $\Lambda_1 = 16k/s, \Lambda_2 = 18k/sand\Lambda_3 = 20k/s$ in each data center . The carbon emission cap is set to 75% of the dynamic energy required to process the trace. The electricity and carbon offset market prices are collected from the real markets.

4.6.2 Influences of Electricity Price and Network Cost

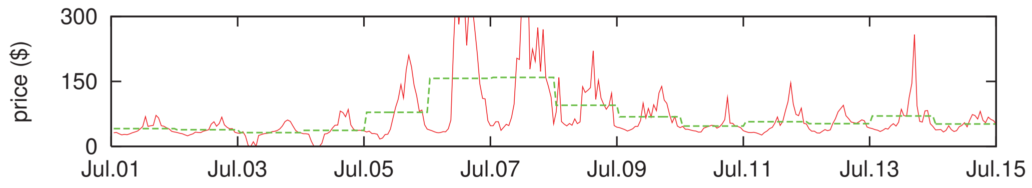
Online Service Providers often sign long-term contracts with power producers with a fixed electricity price over the duration of the contract. By implying our energy-aware optimal load dispatching policy, the OSPs need to work with power producers negotiating the contracts. In our model, the optimal controller will dynamically distribute the workload



(a) San Diego.



(b) Michigan hub

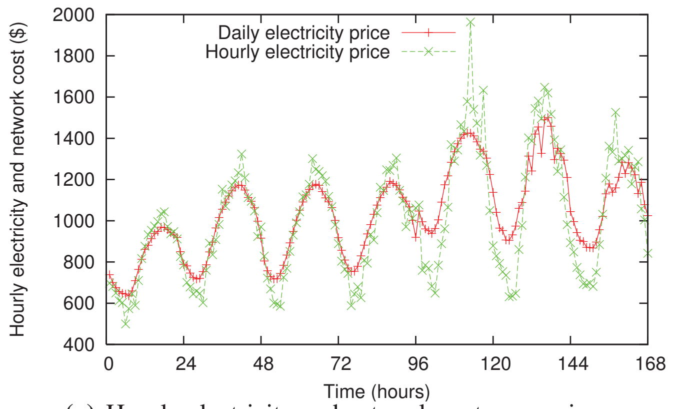


(c) New Jersey

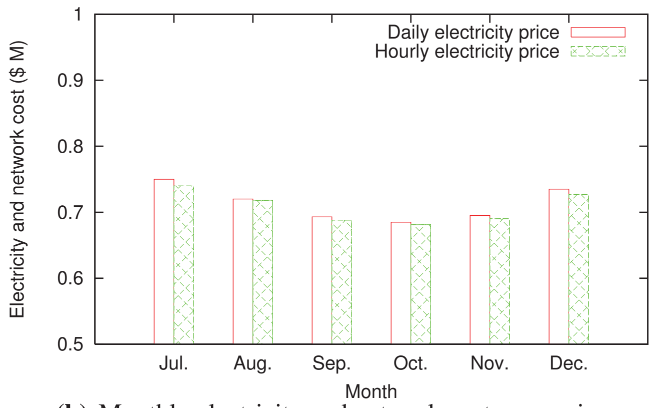
Figure 4.3: 14 days hourly electricity price and daily electricity price comparison.

hourly, so it would be preferred if power producers can accept an hourly spot price on the contract. However, there are two problems we must consider: first, most power producers will not like to provide the off-peak (the lower hourly prices in a day) spot price, since sometimes the price can be negative; second, our policy may not contribute the most cost savings if the spot hourly price shows large fluctuations overtime.

In our evaluation, we first study two kinds of electricity prices: spot hourly electricity price and average daily price. We also compare the total electricity and network cost of OSPs between using daily electricity price and hourly spot electricity price. Figure 4.3 shows the hourly spot electricity prices and average daily prices at three locations for two weeks (Jun.



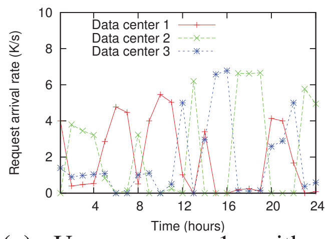
(a) Hourly electricity and network cost comparison



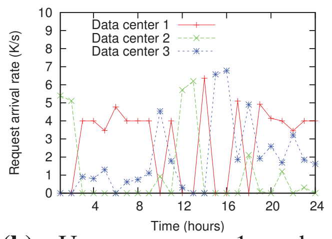
(b) Monthly electricity and network cost comparison

Figure 4.4: Electricity and network cost comparison using hourly and daily electricity price.

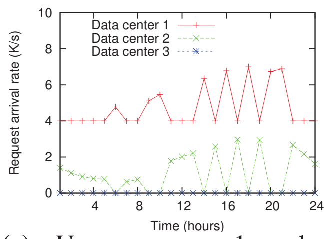
1st, 2010-Jun. 14th 2010). The red lines illustrate the hourly spot electricity prices, and the green lines illustrate the average daily prices. On certain days, the spot hourly electricity prices show large fluctuations, however the standard deviations of spot hourly price are normally below 10 (we study the standard deviations of spot hourly prices in a half year in three locations). Figure 4.4(a) illustrates the dynamic hourly total cost for a week, and Figure 4.4(b) shows the monthly total cost for a half year. As we can see from the figures, the hourly electricity price results in slightly more savings than the daily electricity price,



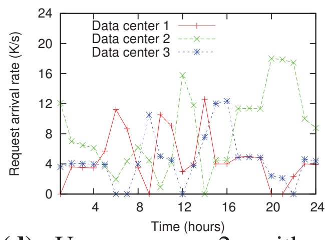
(a) User group 1 without considering network cost.



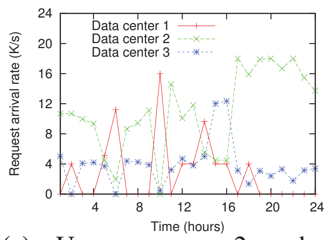
(b) User group 1 when considering network cost (same ISP).



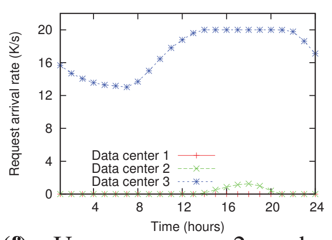
(c) User group 1 when considering network cost (multiple ISP).



(d) User group 2 without considering network cost.



(e) User group 2 when considering network cost (same ISP).



(f) User group 2 when considering network cost (multiple ISP).

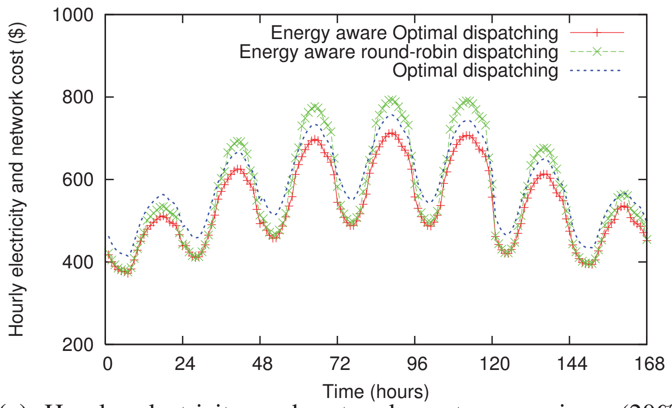
Figure 4.5: One day(June 1st, 2010) load dispatching comparison

with the average savings of 1%. However, the daily electricity price is sufficient enough for cost savings if the power producers do not accept hourly spot price on the contract. Even if a fixed price contract is applied to each data center, our policy will still work once price differences exist between different locations. For the rest of the simulation, we use an average daily electricity price to evaluate our energy-aware optimal dispatching policy.

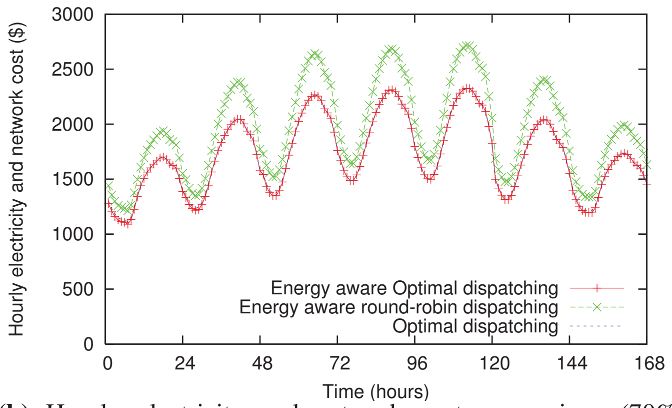
For network cost, we evaluate our optimal energy aware load dispatching policy with varied network costs given a fixed brown and green power mix in a day. We are not aware of any public disclosures that provide specific information about the geographic and temporal variation in network prices [86]. We assume network services are provided by the same ISP or multiple ISPs through direct links for simplicity. Figures 4.5(a) and 4.5(d) illustrate the workload distribution for two user groups when we do not consider the network cost. Figures 4.5(b) and 4.5(e) illustrate the workload distribution for two user groups when the Internet connections are provided by the same ISP, which means the cost per unit job between each user group and data center is the same. Figures 4.5(c) and 4.5(f) illustrate the workload distribution for two user groups when the Internet connections are provided by multiple ISPs, which means the network cost for transferring per unit job between each user group and data center is different. As shown in Figure 4.5, the workload distribution is quite different in the above three scenarios, which means network cost plays an important role in load distribution, so our work can provide a more accurate load dispatching strategy to reduce the total cost.

4.6.3 Performance Evaluation

To further evaluate our model, we compare our energy-aware load dispatching policy with two baseline policies. The first baseline policy is energy-aware round-robin load dispatching policy, in which the workloads for each user groupsăare evenly distributed to three data centers, energy efficiency strategies are applied in both server and cooling sub-systems; The second baseline policy is optimal load-dispatching policy, in which each data center ignores the dynamic power consumption of cooling the system, chillers will always be on. Figure 4.6 reflects the hourly total electricity and network cost of the three dispatching policies when the workload is normalized to 20% and 70% respectively. With



(a) Hourly electricity and network cost comparison (20% workload)



(b) Hourly electricity and network cost comparison (70% workload)

Figure 4.6: Electricity and network cost comparison with different load dispatching disciplines.

light workload data centers, the optimal load dispatching policy contributes the highest cost because of the unnecessary cooling power consumptions. On the other hand, with heavy workload data centers, the round-robin load dispatching policy contributes the highest total cost because of ignoring dynamic electricity prices. In either situation, our optimal energy-aware load dispatching policy achieves the most cost savings. We further evaluate our model for six months with varied workloads as shown in Figure 4.7, and we observe

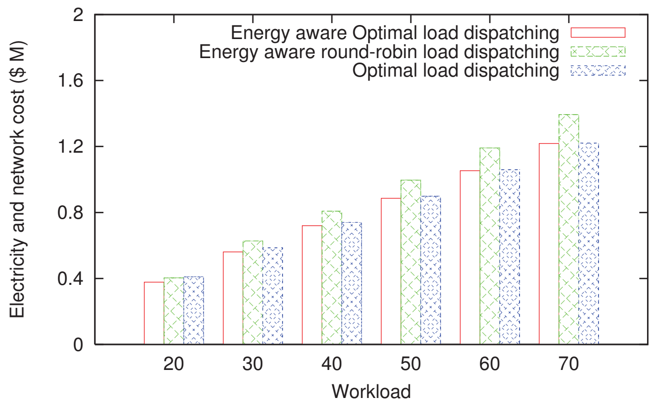


Figure 4.7: Monthly electricity and network cost comparison

that our model can significantly reduce the total cost for OSPs.

Next, we study the carbon emission caps and chiller ON/OFF threshold and how they influence total cost. The carbon emission limits are set to 50%-90% of dynamic energy required to process the trace, and the chiller ON/OFF threshold varies between 10%-50%. We study the total cost for different workload intensities in a month. The results are shown in Figure 4.7. Under different workload intensities, we can summarize the following conclusions. First, the total cost gradually decreases with the increase of carbon emission limit, since the green energy price is higher than brown energy. Our controller provides an accurate power mix decision under different carbon emission limits. Second, the total cost gradually decreases with the increase of chiller ON/OFF threshold, our optimal controller helps to save more cooling power by distributing the workload and turning off chillers whenever possible. Also, there is not a significant cost change under different workload intensities, which proves that our energy-aware load dispatching is not influenced by the chiller ON/OFF parameter. Finally, the total cost is higher under higher workload intensity, which means our energy-saving strategies applied in the sever sub-system and cooling sub-system together work to save the total electricity cost.

Finally, we study the brown energy consumption and total cost in a half year as shown in Figure 4.8. The carbon emission caps are set to 60%, 75% and 90% of the dynamic energy required to process the trace. The total cost is decreases as the carbon emission cap increases. It further proves that our optimal controller effectively selected the power mix in order to save total cost for OSPs.

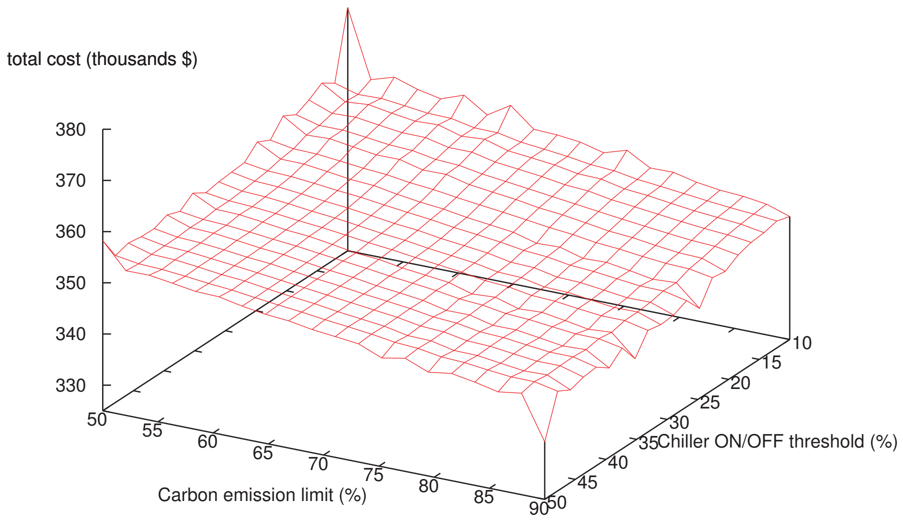
Figure 4.9 shows the hourly average end-to-end response time in a week. In our simulation, the maximum average response time is set to 65*ms*. We can see from the figure that the average response time is always below the threshold, which means our model can achieve desirable quality of service.

4.7 Future Work

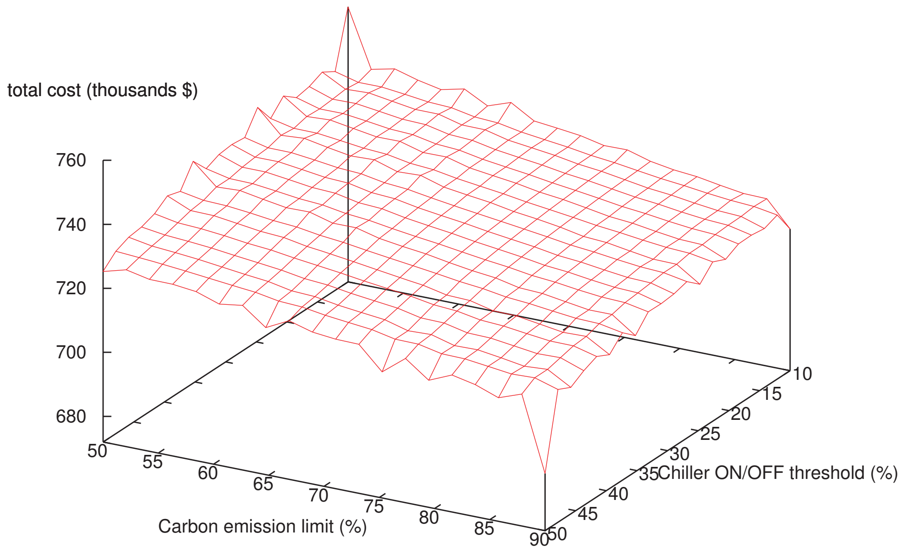
Cooling system energy efficiency. Considering the fact that cooling system takes 30% of IDC total power consumption, we argue that the efficiency improvement of cooling system is part of green computing. Our previous work only used a simplified chiller ON/OFF strategy to reduce the energy consumption of cooling system. The CRAC fan speed, which determines the airflow rate through the data center, can also be dynamically adjusted with

IDC workload. Moreover, with the newer highly modular container based designs, the dynamic behavior of outside temperatures, cooling system therefore exhibits opportunities in reducing the total energy consumption of an IDC. Our future work will exploit those opportunities to improve the energy efficient of cooling system together with the server system.

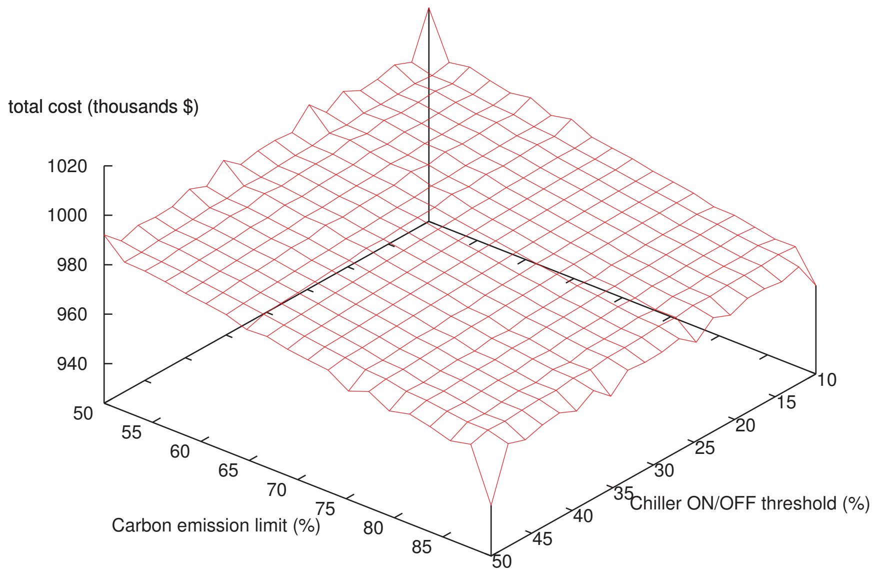
Complex network model. Network cost is an important contributor to OSPs costs, and there could be large differences on the costs levied by different network providers, and even on the same network provider over time [82]. In our previous work, we modeled the network cost as a linear function of the traffic volume. This model significantly simplified the problem, however, in realistic, the network cost is a non-linear function of the network traffic volume. Moreover, the network charges for a link are levied at the end of a billing period and are a function of the traffic volume during that entire period instead hourly. Therefore, we plan to investigate how to estimate the hourly charges of network. For the complex network cost, how to construct the network model and solve the cost problem are worthwhile for future researches.



(a) 20% workload intensity



(b) 50% workload intensity



(c) 70% workload intensity

Figure 4.7: Monthly electricity and network cost with varied carbon emission limit and chiller ON/OFF threshold.

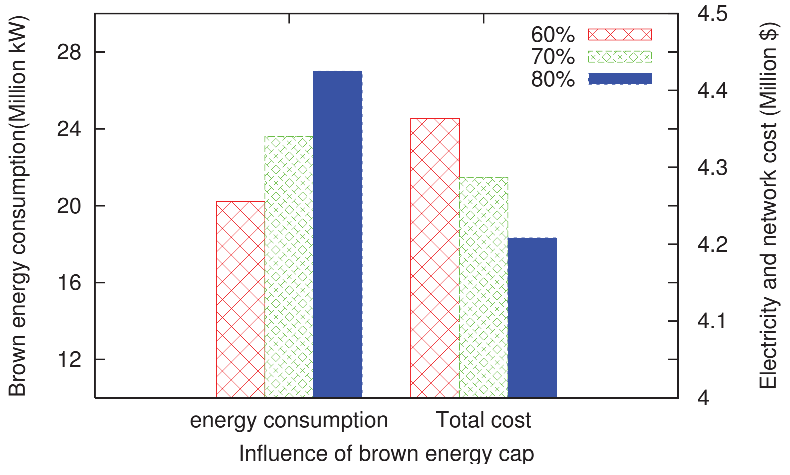


Figure 4.8: Brown energy consumption and total cost with different brown energy caps.

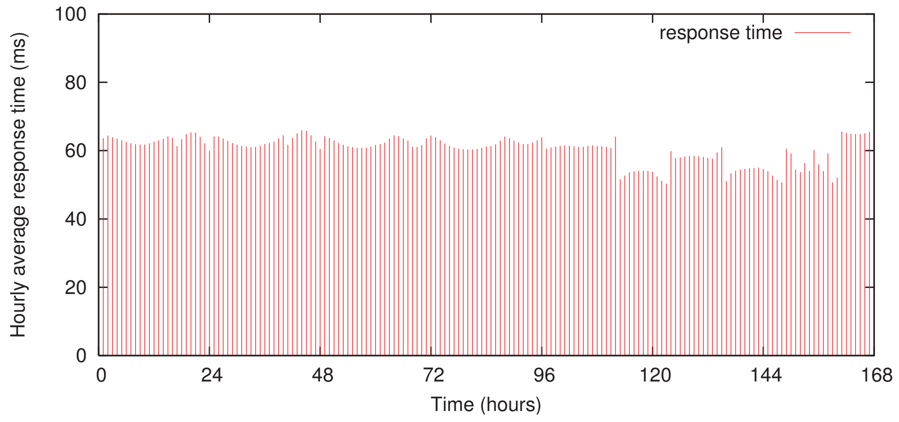


Figure 4.9: Average hourly end-to-end response time (ms)

Chapter 5

Energy-efficient Statistical Live Virtual Machine Placement in Cloud Computing Environments

5.1 Background and Related Work

Cloud computing, a new computing platform in which users can acquire and release the resources on demand from a Web browser, becomes more and more popular recently. One of the most important technology making cloud computing possible is the use of virtualization technology, such as VMware [89], Xen [90, 91]. Virtualization provides a method for on demand migration and dynamic allocation of virtual machines(VMs). It allows users to achieve the same level of performance in a more flexible and secure way. It also enables workloads to consolidate to less physical machines(PMs), thus reducing overall data center power consumption.

The use of workload consolidation to vacate PM to improve system efficiency has already been presented in many other works. A key issue in workload consolidation is to map the VMs to PMs [92]. Many previous works have formulated the VM/PM mapping problem as a multidimensional bin-packing problem. Each dimension represents a particularly resource type of a VM request. The goal is to use the least possible bins to fulfill all the VM requests [19, 20]. The problem is NP-hard and can be solved by some heuristic methods such as first-fit or best-fit. However, there are several limitations of bin-packing based methods: first, most of them formulated the problem in a homogeneous platform with identical PMs. However, IDCs are inherently heterogeneous because of upgrading cycles and replacement of failed components, the heterogeneous of PMs makes the problem even harder than the NP hard bin-packing problem; second, the dynamic behavior of workload is neglected. When a VM arrives and departs, the VMs consolidation will be violated. Therefore, it is not only necessary but essentical to deal with the dynamic workload for energy savings; third, the use of consolidation strategies have to consider additional factors that are of utmost importance for data centers, such as QoS, reliability in addition to energy consumption. The overheads caused by VM consolidation and migration need to be investigated.

In this work, we propose a new live VM placement scheme that can dynamically and effectively map the VM requests to PMs with substantial energy savings in a heterogeneous server clusters. We construct a VM/PM mapping probability matrix, in which each VM request is assigned with a probability running on a specific PM. The VM/PM mapping probability matrix takes into account resource limitations, VM operation overheads, server reliability as well as energy efficiency. Our scheme then decides where to execute a new job, and whether to move existing jobs in order to improve global system efficiency. Furthermore, the proposed scheme is able to extend for more considerations in the light of users' demand. This work discusses the entire proposed scheme and evaluates its effectiveness via extensive simulations.

There is an expansion in research on energy efficiency in large scale data center or server clusters in the past few years. In this section, we only review the work related to VM management and cloud computing since they are more closely related to this work.

One most influential technology making cloud computing possible is the use of virtualization [90, 93, 94]. Virtualization allows user to achieve the same level of performance and security with lower energy consumption by consolidating multiple VMs into a larger PM [20].

A significant amount of works are focused on VM scheduling and consolidation planning. The goal is to map VMs to fewest possible PMs without degradation of performance in a data center or a server cluster. These efforts can be divided into two categories: static

VM consolidation and dynamic VM consolidation. Under static consolidation, strategies were proposed to allocate system resources for the incoming VMs with both energy and QoS considerations [95]. In [96], researchers proposed a statistical static capacity management in virtualized data centers with guaranteed QoS. In [97], a multi-capacity aware bin-packing algorithm is proposed to make use of the information in the additional capacities for resource allocations. However, VM loads often change over time, it is not sufficient to make good initial placement choices only using static consolidation approach. It is necessary to dynamically alter placements as conditions change in a data center [1].

The technique of live migration is to reallocate an executing VM between two PMs without significant interruption of the VM [93, 98]. In [91], the authors proposed a new consolidation approach in a homogeneous cluster environment that considered both VMs allocation and VMs migrations. Bo Li [99] investigated the live placement of applications dynamically in a cloud platform. Another group of researchers presented a planning tool named ReCon to control dynamic consolidations in an IDC [100]. However, these works were either formulated in a homogeneous platform or did not treat live migration overhead carefully. In [9], the authors proposed a score-based live migration mechanism that carefully addressed virtualization overheads. However, in their work, the active number of physical servers did not depend on the dynamic VM mapping results, but depended on two workload intensity thresholds, which will not lead to the most energy savings. Our work is built on a heterogeneous platform. It has taken into considerations of all the VM operation overheads, system reliability in addition to energy efficiency. Specifically, the dynamic VM placement scheme and the spare server strategy will together determine the total number of active servers. From one hand, the dynamic VM placement scheme will minimize the number of servers, from the other hand, the spare server number will be determined by workload intensity. Therefore, our work will effectively reduce the system power consumption through dynamic consolidation scheme and also be capable of dealing with workload spike.

There are also some researchers made efforts to reduce the power consumption or computational cost in a cloud platform. In [101], a GreenCloud architecture is proposed, which reported significant energy saving in cloud computing environment. In [102], the authors proposed a dynamic load distribution policy that addressed all electricity-related costs as well as transient cooling effects in a cloud platform. Michele Mazzucco [103] formulated a queuing model to maximize the average revenue for the cloud providers.

5.2 Statistical Dynamic Virtual Machine Migration

5.2.1 Problem Statement

The primary goal of VM management in a visualized data center is to minimize the PMs needed for all the VM requests. Mapping a VM "correctly" into PMs requires knowing the capacity of each PM and the resource requirements of the VM. It must also take into accounts VM operation overheads, reliability of the PMs, QoS in addition to energy efficiency.

An example is shown in Figure 5.1 for better understanding the problem. In the example, we have three PMs in the system, each of them has a limited resource of 10, 11 and 11 respectively. We only consider a single resource type for simplicity in the example. There are three jobs are currently running on PM_1 and PM_2. When two new jobs arrive sequentially, static methods require an additional PM. However, if we firstly migrate VM_1 from PM_1 to PM_2, the new arrival jobs can be both allocated to PM_1, while PM_3 can remain off to save energy. Therefore, the consolidation strategy need to find an energy efficient VM/PM mapping dynamically with the change of system status. The problem is different from the multidimensional bin packing problems from two perspectives: first, in traditional bin-packing problem, the bin size is the same. However, in our problem, the resource capacity of each PM is different; second, the VM operation overheads, the reliability of PMs and QoS must be considered when packing the VMs into PMs. So our problem is much harder than the NP-hard bin-packing problem.

In this work, we build a statistical framework for VM management in a heterogeneous server cluster. A new dynamic VM placement process can be triggered by three different kinds of events: new VM arrival, VM departure, and the changes of PM reliability. Then, the best VM/PM mapping is determined with the help of a probability matrix that is constructed with several considerations.

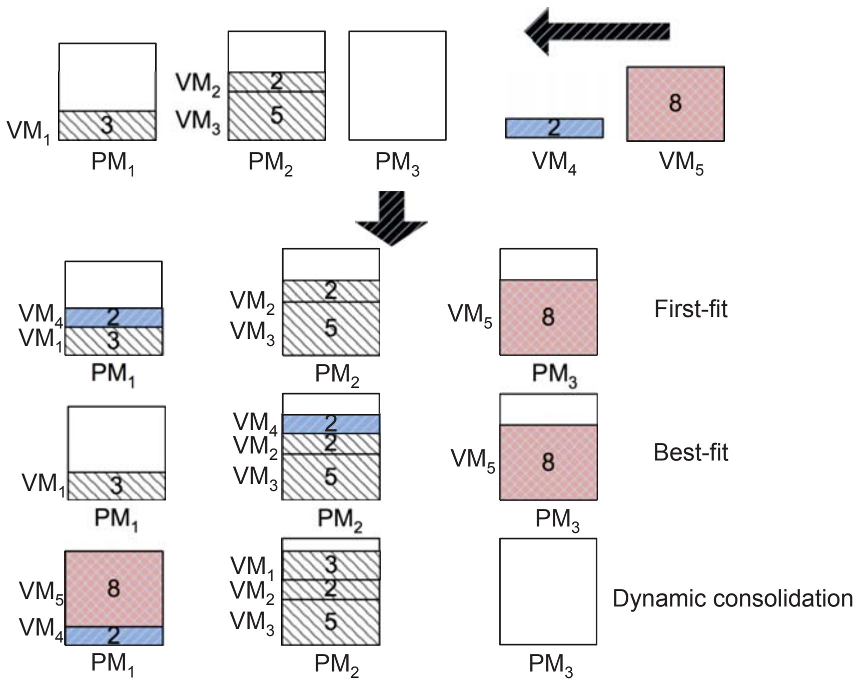


Figure 5.1: Dynamic consolidation *V.S.* static consolidation.

For better understanding of our scheme, Table 5.1 summarizes the notations and definitions which will be used throughout this chapter.

5.2.2 Transition Probability Matrix Formulation

The probability matrix P consists of N rows and M columns. Each column represents an active PM in the system, each row represents a VM that is currently running in the system. The elements in the matrix p_{ij} is the normalized probability of hosting a VM i in a PM j with several constraints. We firstly study the constraints on an individual PM and obtain the joint probability $p'_{ij} = p^{res}_{ij} * p^{vir}_{ij} * p^{rel}_{ij} * p^{eff}_{ij}$ with resource requirements, virtualization

Table 5.1
Notations

Notation	Definition
N	Number of VMs running in the system.
M	Number of active PMs.
VM^{MAP}	VM and PM mapping vector, $\|VM^{MAP}\| = N$.
p'_{ij}	The normalized joint probability of hosting VM i in PM j.
p_{ij}	The joint probability of hosting VM i in PM j.
p_{ij}^{xxx}	The probability of hosting VM i in PM j with only consideration of condition xxx. $xxx = res$: resource limitations, $xxx = ope$: VM operation overheads, $xxx = rel$: server reliability, $xxx = eff$: energy efficiency.
R_i	Resource requirement vector of VM i, $\|R_i\| = K+1$.
R^{MIN}	VM minimum resource requirement vector, $\|R^{MIN}\| = K$.
C_j^{MAX}	Maximum resource capacity vector of PM j.
C_j	Current resource occupation vector of PM j, $\|C_j\| = K$.
T_j^{cre}	VM creation time in PM j.
T_i^{mig}	VM i migration time to any destination PM.
$Delay_i^{MAX}$	maximum performance delay of VM i.
$Delay_i(t)$	VM i performance delay at time t.
U_j	Resource utilization (%) of PM j.
U_j^{MIN}	Resource utilization (%) with one mini VM hosted in PM j.
w_j	Resource utilization level of PM j.
W_j	The maximum number of mini VMs can be hosted in PM j.
$Ustatus_j$	Resource utilization status of PM j.
α_k	Type k resource intensive parameter.
$power_j$	Per VM power consumption of PM j.
eff_j	Relative power efficiency parameter of PM j.
$MIG_{threshold}$	Migration threshold.
$n_{arrival}(t, t+T)$	Number of VMs arrival in the next control period T.
$n_{departure}(t, t+T)$	Number of VMs departure in the next control period T.
$n_{spare}(t, t+T)$	Number of spare PMs in the next control period T.
$n_{idle}(t)$	Number of non-idle PMs at time t.
$n_{AVE}(t)$	Average number of PM required by a VM until time t.

overhead, the physical server reliability and energy efficiency considerations. After that, we normalize the joint probabilities in each row to obtain the elements in the transition probability matrix.

$$\begin{array}{c|cccc|}
 & PM_1 & PM_2 & \cdots & PM_M \\
VM_1 & p_{11} & p_{12} & \cdots & p_{1M} \\
VM_2 & p_{21} & p_{12} & \cdots & p_{1M} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
VM_N & p_{N1} & p_{N2} & \cdots & p_{NM}
\end{array} \tag{5.1}$$

5.2.2.1 Resource Limitations

Each VM has a specific resource demand from PMs, such as the number of CPUs, memory size or disc space. We firstly check if a VM can be hosted in the PMs. We define a VM request i as a $K+1$ dimensional vector: $|R_i| = K+1$. The first K components represent the resource demand of a VM. The last component is the estimated running time of the VM request. It is easy to specify the amount of needed resources; this is typically under the direct control of the user. Users often run the same applications many times and can predict runtime based on experience [102]. The resource capacity of a PM j is a K dimensional vector: $|C_j^{MAX}| = K$, each component $C_j^{MAX}(k)(for\, k = 1, 2, ...K)$ represents the maximum capacity of the resource type k in PM j. A K dimensional vector C_j represents the current resource occupations of PM j. We define p_{ij}^{res} is the probability of VM i hosted in PM j by only considering the resource requirements:

$$p_{ij}^{res} = \begin{cases} 1; \quad \forall \ R_i(k) + C_j(k) \leqslant C_j^{MAX}(k) \ (for\ k = 1, 2, \ldots, K) \\ 0 \quad otherwise \end{cases} \tag{5.2}$$

The rationale of the above definition is quit straightforward, if there is sufficient resource for VM i hosted in PM j, the probability $p_{ij}^{res} = 1$, otherwise, $p_{ij}^{res} = 0$.

5.2.2.2 VM Operation Overheads

To avoid losing customers, our scheme never violates the QoS. We carefully consider two VM operation overheads from the performance perspective in our framework: VM migration time and VM creation time.

The VM migration time reflects the time duration from migration start to finish, which we refer to as migration overhead. The migration time is related to the amount of memory used by the migrated VM i [91, 104]. It has been tested linearly increased with the requested VM memory size. We use T_i^{mig} to denote the VM i migration time to any physical machine j. Two metrics are usually used to quantify the performance of migration. One is downtime, which reflects the time of the migrated VM with no response. Since this time has been tested in milliseconds, it is negligible compared to the total runtime [91, 104]. The other one is performance overhead, which is the performance degradation of co-hosted VMs because of extra computation resources consumed to perform the migration. It reduces the computation capacity, at the same time increases the runtime of co-hosted VMs and the migrated VM. In the worst case, the performance loss is about the same as the migration time [91]. Although the performance overhead is fairly short, the numbers of migrations should be kept to least possible for QoS.

The VM creation time mostly depends on the processing capacity of the PMs. We use T_j^{cre} to denote any VM creation time on PM j. The incoming request cannot be served until the creation of the VM is finished. The VM creation time involves extra waiting time for each incoming request, therefore it affects the overall performance.

In our framework, the QoS is set to complete each job within 105% of the job's total runtime plus 30 seconds as in previous work [102]. The latter part of the slack is to avoid missing the QoS for short running jobs. Let $Delay_i^{MAX}$ be the maximum acceptable delay for the VM request i, in which $Delay_i^{MAX} = 5\% * R_i(K+1) + 30$ [102]. The VM i current delay is caused by queuing delay, VM creation and VM migration, we use $Delay_i(t)$ to denote it.

We define p_{ij}^{ope} as the probability of VM i migrated to PM j under operation overheads consideration. Before migrating the VM i, we firstly check if the migration will cause a violated performance for any co-hosted VMs, or cause a violated performance for the VM i itself. Suppose VM i is currently hosted in PM m, let VM_i^{MAP} represents the VM/PM mapping vector, so $VM_i^{MAP} = m$. The VM i is going to be migrated from PM m to PM j. The migration will be forbidden under two circumstances: first, there exists a co-hosted

VM w that the QoS will be violated because of the migration: $Delay_w(t) + T_i^{mig} > Delay_w^{MAX},\ for\ any\ VM_w^{MAP} = m$; second, the violated QoS is detected for the VM i itself: $Delay_i(t) + T_i^{mig} > Delay_i^{MAX}$. We then set the p_{ij}^{ope} to be 0. Otherwise, we set $p_{ij}^{ope} = 1$, if the migration will not result in unacceptable performance loss or the VM i is already hosted in PM j.

5.2.2.3 Server Reliability

We also consider physical server reliability in the dynamic consolidation process. Each physical machine is given a probability of reliability p_j^{rel} according to its life time, chance of failure and so on. The higher the probability is, the more reliable of the physical machine. We use this reliability of the physical machine j as the probability of any VM i hosted in PM j while only considering the reliability issue, so $p_{ij}^{rel} = p_j^{rel}$. If a physical machine fails, all the VMs that are running in it will be reallocated.

5.2.3 Energy Efficiency

The primary goal of our work is to improve the overall energy efficiency. The basic idea is to make the best use of high energy efficiency PMs. In order to achieve this objective, we design a strategy, in which the utilization status and power efficiency will combine to determine the probability p_{ij}^{eff}, so p_{ij}^{eff} is the product of two parts. (Note: the energy efficiency probability p_{ij}^{res} will be only calculated when the resource limitations can be met: $p_{ij}^{res} \neq 0$)

5.2.3.1 Server Utilization Status

From the utilization perspective, we want to ensure each PM is fully or nearly fully utilized to minimize the number of required PMs. It is important to consider multiple resource types instead of a single resource type to better evaluate the PM utilization status. For example, if 100% CPU of a PM is utilized while only 20% memory is utilized. In this case, no more jobs can be allocated in the physical machine, wasting 80% of memory.

Our work considers multiple resource types. We define two utility functions to evaluate the resource utilization of each PM as described below.

Joint resources utilization: For each PM j, the resource utilization is a joint product of several resource utilizations, $U_j = \prod \frac{C_j(k)+R_i(k)}{C_j^{MAX}(k)}$, for $k \in \{1,2,3..K\}$, each $\frac{C_j(k)+R_i(k)}{C_j^{MAX}(k)}$ represents the utilization of resource type k. The PM j with U_j closer to 1 means better utilization, while closer to 0 means poorer utilization. In this definition, each resource plays an important role to evaluate the PM utilization, the PM will only be considered well utilized when each resource is fully or nearly fully utilized.

Resource intensive utilization: For each PM j, the resource utilization is the summation of the weighted resource utilization for each resource type, $U_j = \sum \alpha_k \frac{C_j(k)+R_i(k)}{C_j^{MAX}(k)}$, for $k \in \{1,2,3..K\}$ and $\sum \alpha_k = 1$, in which α_k is the intensive parameter of resource type k. This definition is extremely useful when certain resource type dominates resource demand. For example, VMs require a large amount of CPUs and a small amount of other resources. Also, the PM j with U_j closer to 1 means better utilization, while closer to 0 means poorer utilization.

After obtaining the resource utilization of each PM j. We non-evenly partition the resource utilization interval into several sub-intervals for each PM j, and each sub-interval represents a resource utilization level. We define a minimum resource requirement vector as $|R^{MIN}| = K$ in order to partition the resource utilization for each PM. Each component $R^{MIN}(k)$ represents the minimum requirement of the resource type k. This can be treated as a small instance or a mini instance type in cloud services [105]. Assuming the PM j has sufficient resources for a maximum number of W_j mini VMs, we then partition the resource utilization into $W_j + 1$ levels. Because lack of space, we only show how to partition the resource utilization through **Joint resources utilization** method in Equation (5.3).

$$
\begin{aligned}
&L_0 = [0, U_j^{MIN}) \\
&L_1 = [U_j^{MIN}, 2^k U_j^{MIN}) \\
&L_2 = [2^k U_j^{MIN}, 3^k U_j^{MIN}) \\
&\vdots \\
&L_{w_j} = [(w_j)^k U_j^{MIN}, (w_j+1)^k U_j^{MIN}) \\
&\vdots \\
&L_{W_j-1} = [(W_j-1)^k U_j^{MIN}, W_j^k U_j^{MIN}) \\
&L_{W_j} = [W_j^k U_j^{MIN}, 1]
\end{aligned}
\tag{5.3}
$$

In the above equation, $U_j^{MIN} = \prod \frac{R^{MIN}(k)}{C_j^{MAX}(k)}$, for $k \in \{1,2,3..K\}$, which represents the resource utilization when only one mini VM hosted in the PM j. Level L_0 is an impossible state, since there is no possibility that the resource utilization U_j with VM i hosted in is less than the minimum resource utilization U_j^{MIN}. L_1 means that there is no more than one VM hosted in PM j. As the level increases, the utilization of PM j increases. At the last level, PM j is fully utilized or nearly fully utilized; thus no further VM requests can be accepted, for which we consider that the PMj has achieved its maximal energy efficiency.

By partitioning the resource utilization, the resource utilization status is defined to be proportional to its utilization level w_j as shown in Equation (5.4). VM i has a higher probability to be hosted in the PM j with higher utilization level. Note, we formulate the problem in an inhomogeneous server system, so each server may have different computation capacity, that is why we separately partition the resource utilizations into different groups for different PMs. If all the PMs are identical nodes: $W_1 = W_2 = \cdots = W_j = \cdots = W_M$.

$$Ustatus_j = \frac{w_j}{W_j};\ \ if\ \ U_j \in L_{w_j}(w_j = 0, 1, \ldots, W_j) \tag{5.4}$$

An example is shown below to illustrate how to obtain the utilization status. We only consider two resource types: CPU and memory in the example. Assuming a PM has the computational capacity with 4 cores and 8G memory. The minimum VM resource requirements are 1 core and 1.5 G memory, so 4 mini VMs can be hosted in the PM. We then partition the resource utilization into 5 sub-intervals as: $L_0 = [0, 3/64)$, $L_1 = [3/64, 3/16)$, $L_2 = [3/16, 27/64)$, $L_3 = [27/64, 3/4)$, $L_4 = [3/4, 1]$, each sub-interval represents a utilization level. If $U_j = 0.7$, since 0.7$\in L_3$, the utilization level of the PM j is $U_j = 3/5$.

5.2.3.2 Server Energy Efficiency

It is not enough to just evaluate the server utilization for energy efficiency. For example, two servers have the same computation capacity, both of them can host a maximum of $W_1 = W_2 = 4$ mini VMs. However, the power consumption of PM 1 is 300 W/h, the power consumption of PM 2 is 400 W/h. Suppose their current utilization levels are the same.

Only considering the utilization level, we can allocate a VM to any of them. However, server 1 should have a high probability because of its lower power consumption.

A relative power efficiency parameter is defined to compare the energy efficiency between each PM. It will help to best use of energy efficient servers. We define: $eff_j = \frac{min:\{power_j\}}{power_j}$(for $j = 1, 2, \ldots, M$), where $power_j$ is the active power consumption of PM j divided by W_j. In other words, it is the per mini VM power consumption of PM j. The PM j with the minimum per VM power consumption will have a relative power efficiency parameter equal to 1. The PM with higher per unit power consumption will be assigned with a smaller value. In the above example, the relative power efficiency parameter for PM 1 and PM 2 are $eff_1 = 1$ and $eff_2 = 3/4$, respectively.

The energy efficiency probability is defined to be proportional to its utilization status and its relative power efficiency as shown in Equation (5.5):

$$p_{ij}^{eff} = Ustatus_j \cdot eff_j \tag{5.5}$$

5.2.4 Dynamic Consolidation Process

The dynamic consolidation process will be triggered by three different events: new job arrival, job departure and the failed of PMs. We firstly identify which event causes a new VM migration process. All three events will bring changes to the probability matrix. For a new VM request, we only calculate the probability in the new VM row and allocate it to the PM with the highest probability. We then update the probabilities column where the new VM is allocated. For job departure, the departing VM row will be removed, and the corresponding PM column will be updated in the probability matrix. If a PM fails, all the VMs hosted in that PM will be treated as new VM requests.

After detecting an event, our VM controller initializes the probability matrix and gets ready for the dynamic consolidation process. It starts with probability matrix normalization. The normalization process is important and necessary, because it is aimed at finding a better improvement instead of higher probability. We divide each p_{ij} with $p_{i(current)}$ (which is the probability the VM i currently allocated in) in each row to check if there is a better VM/PM mapping compared to the current one, and then obtain the normalized matrix D. Numbers in the matrix with values greater than 1 indicate efficiency improvement. Numbers which are less than 1 correspond to degradation while 1 indicates that the VM is currently hosted in the PM.

The dynamic consolidation algorithm is shown in Algorithm 3, it consists of several migration rounds. In each migration round, we select the largest value in the normalized probability matrix, and move corresponding VM to the new PM, having the VM/PM mapping updated. After that, we release the PM resources in which the VM moves from, update the resource occupation of PM which the VM moves to, and refresh the probability matrix. We only need to recalculate the corresponding PM columns with migration involved instead of all the probabilities, and prepare for the next migration round. Each migration round will bring a more efficient and reliable VM/PM mapping until the migration is terminated.

To minimize unnecessary migrations and terminate migration rounds, we set a parameter migration threshold $MIG_{threshold}$ to restrict the dynamic migration rounds. It ensures that only those migrations resulting in improvement will be counted. For example, if we set $MIG_{threshold} = 1.05$, the migration process will stop if there is no number larger than 1.05 in the normalized matrix.

Algorithm 3 Dynamic VM consolidation algorithm

```
Probability matrix $P$ initialization
for each row
  Normalize the $P$ by dividing the probability of the current hosted VM and obtain
  normalized matrix $D$
end for
While there is values larger than $MIG_{threshold}$ in $D$ do
    1. Select the largest value in the matrix $d_{ij}$
    2. Move VM $i$ from the current PM $m$ to PM $j$
    3. Update matrix $P$ in columns $m$ and $j$
    4. Update wmatrix $D$ in columns $m$, $j$ and row $i$.
End while
```

An example is shown below to illustrate the whole process for better understanding the dynamic migration process. In the example, there are 4 VMs currently running in three PMs, in which initially VM_1 is running in PM_2, VM_2 in PM_1, VM_3 in PM_1 and VM_4 in PM_3. We first obtain a probability matrix as shown in the probability matrix P, each element represents the probability of VM i hosted on PM j. We then normalize the probability matrix in each row to obtain the normalized matrix D. For example, in VM_1 row, each element is divided by 0.8, since VM_1 is currently running in PM_2, same for the rest of the rows. In the normalized matrix below, we observe that 1.28 is the largest value, so we migrate VM_2 to PM_2, release the resource of PM_1, refresh the PM_1 and PM_2 columns in the probability matrix and be prepared for the next migration round.

$$\text{Probability matrix:} \quad \boldsymbol{P}$$

$$\begin{array}{c c} & \begin{array}{ccc} PM_1 & PM_2 & PM_3 \end{array} \\ \begin{array}{c} VM_1 \\ VM_2 \\ VM_3 \\ VM_4 \end{array} & \begin{bmatrix} 0.32 & 0.42 & 0.26 \\ 0.29 & 0.38 & 0.33 \\ 0.40 & 0.18 & 0.41 \\ 0.28 & 0.37 & 0.35 \end{bmatrix} \end{array}$$

$$\text{Normalized matrix:} \quad \boldsymbol{D}$$

$$\begin{array}{c c} & \begin{array}{ccc} PM_1 & PM_2 & PM_3 \end{array} \\ \begin{array}{c} VM_1 \\ VM_2 \\ VM_3 \\ VM_4 \end{array} & \begin{bmatrix} 0.75 & 1.00 & 0.63 \\ 1.00 & 1.28 & 1.14 \\ 1.00 & 0.44 & 1.02 \\ 0.80 & 1.04 & 1.00 \end{bmatrix} \end{array}$$

To improve the dynamic consolidation efficiency, we modified our dynamic consolidation algorithms to two parallel dynamic consolidation strategies, named **Top k improvements(TKI)** and **Top k migration time (TKMT)**. Both strategies can migrate multiple VMs in one migration rounds, which substantially improve the consolidation efficiency.

Top k improvements (TKI) : Top k improvements also consists of several migration rounds. However, in each migration round, we migrate multiple VMs at the same time. We select no more than k largest values that are larger than $MIG_{threshold}$ in the normalized probability matrix and sort them in a decreasing order. To ensure simultaneous migrations, only the VMs with unique destination PMs are selected as candidates. We then move VMs to the new PMs and release the PMs resources in which the VMs move from, update the resource occupation of PMs which the VMs move to, having probability matrix refreshed. The next migration round will not start until all the migrations in the current rounds are finished.

Top k migration time (TKMT) : Top k migration time is similar to top k improvements algorithm. The difference is that we select no more than k VMs that having the shortest migration time from the normalization matrix with values larger than $MIG_{threshold}$, and sort them in an increasing order. Also, the VMs with the same destination PMs will be filtered out.

5.3 Spare Server Controlling

In addition to dynamic VM consolidation, another key decision is to determine the amount of active physical servers in the system. The number of active physical machine should be large enough to handle the unexpected workload spike and avoid performance degradation. It should also be remained at minimum to achieve energy efficiency in the system.

We periodically determine the active PMs from two aspects. In a time slot T, the total number of active PMs $n_{ac}(t,t+T)$ is the sum of non-idle PMs $n_{nidle}(t)$ and spare servers $n_{spare}(t,t+T)$. The non-idle PMs is the number of PMs hosting VM requests, which can be easily derived. In order to derive the spare servers $n_{spare}(t,t+T)$ for the next control period, we define another parameter $N_{Ave}(t)$, which is the average number of PM required by a VM request. This number can be computed by the non-idle physical servers $n_{nidle}(t)$ divided by the number of VM requests running in the system. To best estimate this parameter, $N_{Ave}(t)$ is dynamically updated after each dynamic VM migration process. The spare server is determined by the following equation:

$$n_{spare}(t,t+T) = \begin{cases} 0; \quad (if\ n_{arrival}(t,t+T) - n_{departure}(t,t+T) \leqslant 0) \\ \\ \dfrac{n_{arrival}(t,t+T) - n_{departure}(t,t+T)}{N_{Ave}(t)} \quad (Otherwise) \end{cases} \tag{5.6}$$

In equation(5.6), $n_{departure}(t,t+T)$ is the number of VM requests that will finish their execution and depart from the system in the next control period. It can be easily derived, since each VM request is submitted with an estimated running time. $n_{arrival}(t, t+T)$ is the number of VM requests arriving in the system in the next time slot T. Workload prediction is beyond the scope of our work, since there are many tools and methods which can provide precise predictions [106]. We use a simple workload prediction method to estimate the number of arrival VMs $n_{arrival}(t, t+T)$. If more VM requests depart the system, there is no need to keep spare servers. The number of active servers in the current time $n_{nidle}(t)$ is sufficient to handle the incoming request. Idle server will be turned off during the dynamic consolidation process. However, we will have no less than $N_{nidle}(t) + \frac{n_{arrival}(t, t+T) - n_{departure}(t, t+T)}{N_{Ave}(t)}$ servers in the active mode; on the contrary, if there are more VMs arriving in the system, we will keep $n_{spare}(t, t+T)$ spare servers active.

5.4 Evaluation

5.4.1 Parameter Settings

We build a simulator using real workload trace data to evaluate our dynamic VM migration scheme. It takes workload trace as input and outputs the performance and the power consumption. The datacenter is configured to have 100 nodes, including 40 fast nodes

Table 5.2
Data center parameter settings

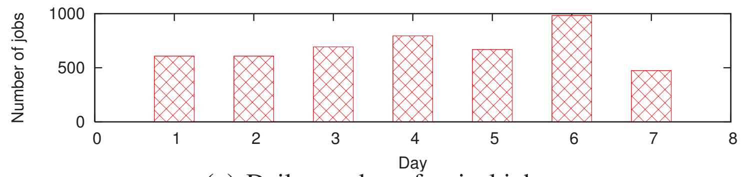
Nodes	Fast	Slow
Number	40	60
VM creation time(seconds)	15	20
ON/OFF overhead (seconds)	40	50
Number of processors	2	2
Cores per processor	4	2
Memory (G)	8	4
Active power consumption (W)	400	300
Idle power consumption (W)	240	180

Table 5.3
Average migration time and power consumption comparison.

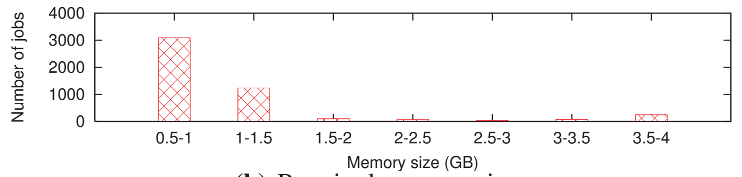
	TKI-J	TKMT-J	TKI-R	TKMT-R
Power consumption (Kw)	973.8	966.8	976.56	966.4
Average migration time (ms)	12.21	14.24	9.79	11.06

and 60 slow nodes. The detailed information and parameter settings of the virtualized data center is shown in Table 5.2.

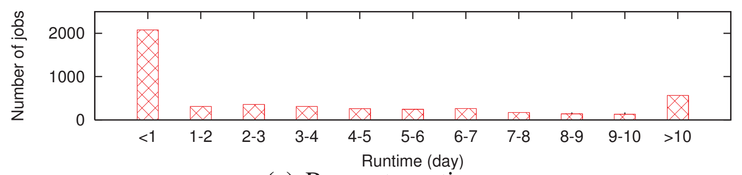
Because lack of Cloud Computing traces, we use a slightly modified trace of jobs submitted to a HPC cluster available from the Parallel Workloads Archive [107]. This trace contains approximately 10 months (August 2004 through May 2005) of data. The trace, LPC Log, contains a record for each job serviced by the cluster, with each record containing a job ID, submitted time, waiting time, actual run-time, and maximum number of cores and amount of memory used. We randomly extract a week long workload from this trace, and filter out the canceled jobs, jobs with small memory requirements, then use it as the workload for all the simulations discussed below.



(a) Daily number of arrival jobs.



(b) Required memory size.



(c) Request run time.

Figure 5.2: Workload characteristics.

Figure 5.2 (a) shows the number of arrival jobs per day. This week long workload contains 4574 jobs, with a peak demand of 982 VM requests per day. We only consider two resource types in the simulation: CPU and memory. We have normalized the memory required by each job by equally dividing its number of cores required. So each VM request requires a single core, a specific memory size with an estimate of its run-time. The required memory and runtime for the week long trace are shown in Figure 5.2 (b)(c). We noticed that most jobs require the memories less than 1GB. There are 2077 jobs with a runtime less than a day. The workload indicates that jobs arrive and leave frequently, which makes dynamic consolidation necessary to improve system efficiency.

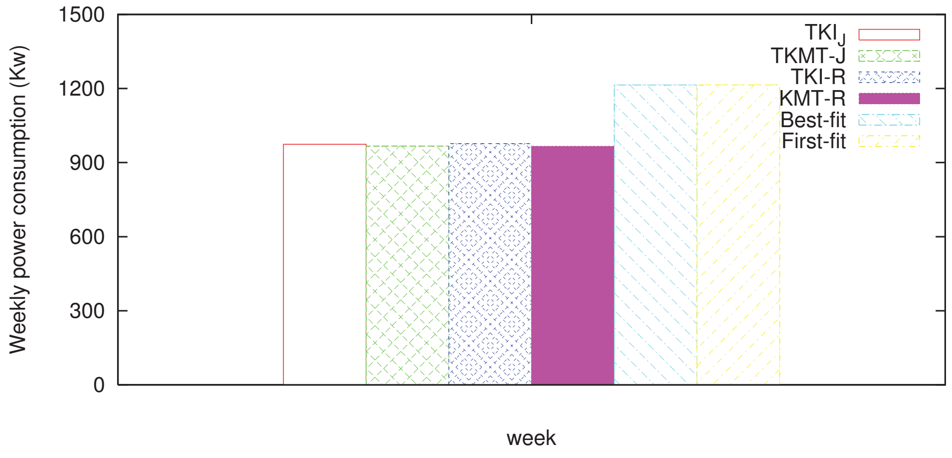


Figure 5.3: Weekly power consumption comparison.

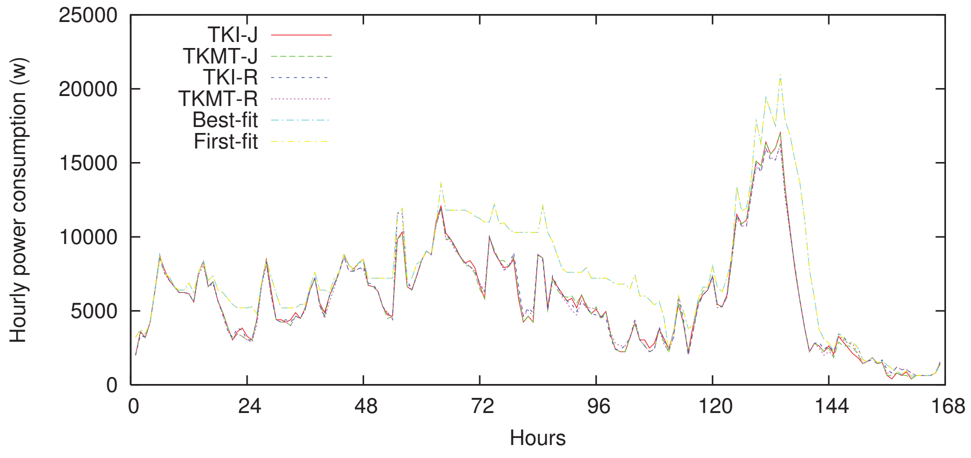


Figure 5.4: Hourly power consumption in a week.

5.4.2 Performance Evaluation

We use the week long workload to evaluate our dynamic VM consolidation algorithms under different resource utilization functions: top k improvements under joint resource utilization (TKI-J), top k migration time under joint resource utilization (TKMT-J), top k improvements under resource intensive utilization (TKI-R) and top k migration time under resource intensive utilization (TKMT-R). The migration threshold is set to $MIG_{threshold} = 1$. The resource intensive parameters are set to $\alpha_1 = \alpha_2 = 0.5$. We compare our proposed algorithms with two static schemes: the first one is the first-fit scheme, in which the new arrival VM request will be placed to the first PM with available computation resources; the second one is the best-fit scheme, in which the new arrival VM request will be placed to the PM that can achieve its maximum utilization.

We first compare the weekly power consumption as shown in Figures 5.3. As illustrated in the figure, our proposed dynamic VM consolidation algorithms consume similar energy consumptions under both of the two proposed utilization functions. They can all contribute more energy savings compared to statistic schemes. We do not show the system performance in detail, because all of them can achieve desirable performance with less than 1% performance loss. To better illustrate our results, we also record the hourly power consumptions as shown in Figures 5.4. We can observe that our proposed algorithms always require less energy consumption compared to static schemes. Based on those results, we can claim that our dynamic VM placement algorithms can significantly reduce power consumption in the cloud computing environment.

Next, we study the average time required for the system arriving at a stable state in each dynamic migration event. Here, a stable state means that there is no value in the normalization matrix larger than the migration threshold $MIG_{threshold}$. The average migration time evaluates the speed and efficiency of our proposed dynamic consolidation algorithms. We compare the weekly power consumption and the average migration time as shown in Table 5.3. As illustrated in the table, when using the resource intensive utilization function, both TKI-R and TKMT-R require less migration time but with similar energy consumptions compared to TKI-J and TKMT-J. Therefore, resource intensive utilization method outperforms the joint utilization method by eliminating unnecessary VM migrations and helping the system to achieve the stable state more faster.

5.4.3 Sensitivity to Parameters

To better evaluate our dynamic consolidation algorithms, we also perform sensitivity study. We first compare the energy consumption by changing the resource intensive parameters. In this simulation, two resource types are considered: CPU and memory. We use α_1 to denote the CPU resource parameter and α_2 denotes the memory resource parameter. The daily energy consumption is shown in Figure 5.5 by varying α_1 from 0.1 to 0.9. As shown the figure, the energy consumption first decreases and then increases as the CPU resource parameter increases. The most energy savings are achieved when α_1 and α_2 are equally set. It can be explained by the workload characteristics and the PMs compaction capacity. In this simulation, if a VM is hosted in a PM, it occupies similar percentage of the total compaction capacity of the PM, because most of the VMs require a single core and a memory size less than 1GB. So the resource intensive utilization method is partially useful when most of the VMs require a large amount of certain resource type than other resource

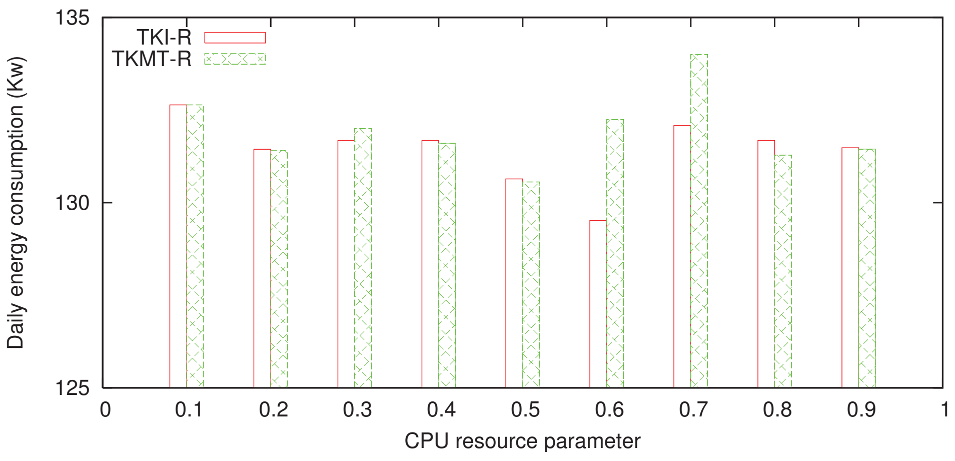


Figure 5.5: Daily power consumption with the change of CPU resource parameter.

types. The resource intensive parameter can be used to balance the resource requirements in order to better evaluate the energy efficiency of the PMs. However, when the workload is uncertain, the joint utilization method is more effective. We do not show the results here because of the space limitations.

Next, we study the migration threshold $MIG_{threshold}$, and how they influence total power consumption. The resource intensive parameters are equally set: $\alpha_1 = \alpha_2 = 0.5$. The migration threshold varies between 1.0 and 2.0. We compare the weekly power consumption and the total number of migrations as shown in Figures 5.6 and 5.7. We notice that both TKI and TKMT consolidation algorithms consume approximately the same amount of energy under the same migration threshold. However, both algorithms (TKI-R and TKMT-R) require less number of migrations when using the resource intensive utilization method. We also noticed that overall the total number of migrations gradually decreases with the increase of migration threshold. However, the energy consumption only

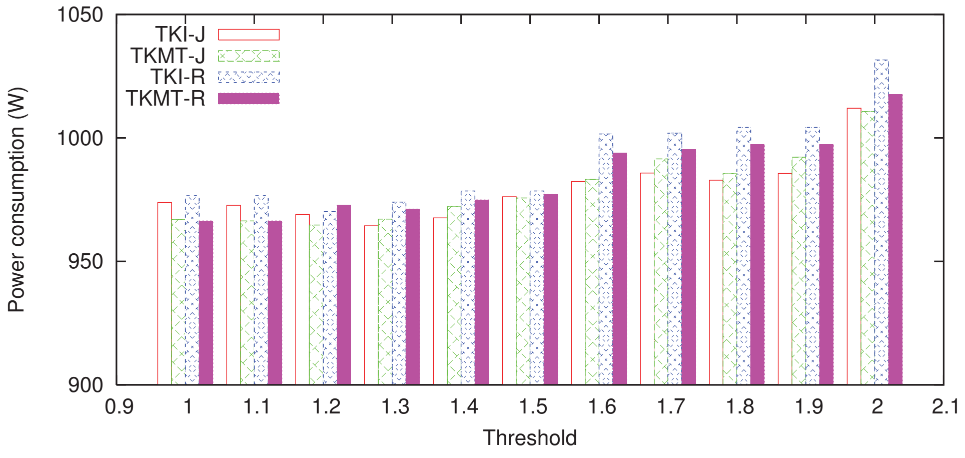


Figure 5.6: Weekly power consumption comparison

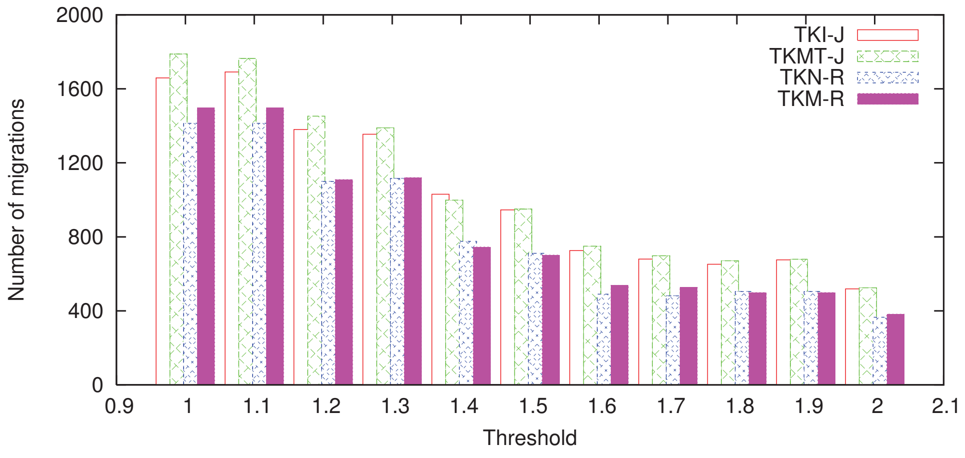


Figure 5.7: Total number of migrations

gradually increases when the migration threshold is larger than 1.5. As the migration threshold is less than 1.4, the total energy consumption remains at the same level. It indicates that too strict migration parameter will not significantly improve the system efficiency, so we can slightly relax the migration parameter to reduce the total number of migrations.

5.5 Future Work

Dynamic VM migration across IDCs. Considering geographically located IDCs, the cost of executing a VM at each data center can be estimated. As the change of electricity price and other factors, VM can be migrated to another IDC in which the operational cost is lower. The VM migrations across IDCs bring many possibilities to reduce the operational cost for OSPs. However, most of the VM images are large, the live migration of a VM across IDCs through WAN may bring extra latencies. How to design a good replica migration algorithm in order to minimize the VM migration latencies is challenging. In my future research, I will address this issue to deal with the problem by combining VM scheduling strategies with VM replication strategies. The replica placement strategies will minimize the long term OSPs cost with the constraints of live migration latencies.

Chapter 6

Conclusion

In this dissertation, we have presented four topics about server cluster energy management and IDC cost management. We summarize the contributions for each of the four focused areas as follows:

In Chapter 2, we propose two dynamic power management strategies to reduce the energy consumption in server clusters. The contributions can be described as follows: we firstly propose an energy proportional model and a DCP model that can provide accurate, controllable and predictable quantitative control over power consumption; second, we discuss our models based on queuing theory in multiple classes scenario, which can provide service differentiation; third, we analyze the effect of transition overhead and propose strategies to improve the performance. Finally, we evaluate our models via simulation. Simulation results show that our models can achieve predictable and controllable proportional energy consumption and desirable performance in a server cluster.

In Chapter 3, we design an adaptive CMDP based power management strategy to reduce power consumption. The contributions can be described as follows: first, a CMDP model is formulated to minimize the power consumption with guaranteed QoS; second, the DV/FS and VOVF mechanisms are combined for substantial more energy savings; third, transition overhead is carefully considered in modeling the CMDP, which provides more precise power and performance control; fourth, the proposed CMDP based adaptive power management strategy can greatly reduce computation time and also be capable of dealing with the dynamic changes of the system workload; finally, our model is evaluated by extensive simulations. Simulation results show that our strategy can minimize energy consumption under performance constraint.

Chapter 4 aims to provide an optimal energy aware load dispatching strategy to reduce overall electricity and network cost in geographically located Internet data centers. We summarise our contributions as follows: In this chapter, an electricity cost model and a network cost model are formulated in geographically located Internet data centers. We formulated an optimization problem to minimize the total cost for OSPs; second, we consider dynamic behavior of electricity and carbon offset market when solving the optimization model; third, we propose an adaptive optimization algorithm to solve the optimal energy-aware load dispatching problem which can achieve desirable performance;

fourth, two energy efficient strategies are applied in each data center, which greatly reduces the energy consumptions in IDCs; Finally, our optimal dispatching discipline is evaluated by extensive simulations. Simulation results show that our dispatching strategy can minimize electricity cost under performance constraint.

Chapter 5 proposes a statistical live VM placement scheme in a cloud computing environment. The contributions can be summarized as follows: first, a statistical VM/PM mapping matrix is constructed by considering the resource requirements, virtualization overheads, server reliability and energy efficiency; second, we propose a nonuniform partition strategy together with server efficiency to evaluate the VM/PM mapping energy efficiency; third, we propose three dynamic VM migrations algorithms to improve system energy efficiency; fourth, we propose a spare server strategy to determine the number of active servers in each control period. The proposed strategy can efficiently reduce the system power consumption and also be capable of dealing with workload spike; finally, our proposed dynamic VM migration algorithms are evaluated by extensive simulations with real workload. Simulation results show that our algorithms can save significant energy compared to static scheme with satisfied performance.

References

[1] C. Hyser, B. McKee, R. Gardner, and B. J. Watson, “Autonomic Virtual Machine Placement in Data Center,” in *HP Technical report, http://www.hpl.hp.com/techreports/2007/HPL-2007-189.pdf*, Feb. 2008.

[2] C. Weddle, M. Oldham, J. Qian, A.-I. A. Wang, P. Reiher, and G. Kuenning, “PARAID: A gear-shifting power-aware RAID,” *Trans. Storage*, vol. 3, Oct. 2007.

[3] U.S. Environmental Protection Agency. "Report to Congress on Server and Data Center Energy Efficiency", August 2007.

[4] M. Tedre, B. Chachage, and J. Faida, “Integrating environmental issues in IT education in Tanzania,” *Frontiers in Education Conference, 2009. FIE ’09. 39th IEEE*, pp. 1 – 7, Oct 2009.

[5] A. Qureshi, “Plugging Into Energy Market Diversity,” in *7th ACM Workshop on Hot Topics in Networks (HotNets)*, Oct 2008.

[6] J. S. Aronson, “Making it a positive force in environmental change,” *IT Professional*, vol. 10, pp. 43 – 45, Jan 2008.

[7] U. Congress, “ To study and promote the use of energy efficient computer servers in the united states.,” in *House bill 5646.*, Feb 2008.

[8] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, “Managing server energy and operational costs in hosting centers,” *SIGMETRICS Perform. Eval. Rev.*, vol. 33, pp. 303–314, June 2005.

[9] I. Goiri, F. Julia, R. Nou, J. L. Berral, J. Guitart, and J. Torres, “Energy-Aware Scheduling in Virtualized Datacenters,” in *Cluster Computing (CLUSTER), 2010 IEEE International Conference on*, pp. 58 –67, Sep. 2010.

[10] S. Pelley, D. Meisner, T. F. Wenisch, and J. W. VanGilder, “ Understanding and Abstracting Total Data Center Power,” in *Proc. of the 2009 Workshop on Energy Efficient Design (WEED).*, Jun 2009.

[11] 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 *Proceedings of IEEE INFOCOM*, 2010.

[12] M. Vasic, O. Garcia, J. Oliver, P. Alou, and J. Cobos, “A DVS system based on the trade-off between energy savings and execution time,” *Control and Modeling for Power Electronics, 2008. COMPEL 2008. 11th Workshop on*, pp. 1 – 6, Jul 2008.

[13] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath, "Dynamic cluster reconfiguration for power and performance," in *Compilers and operating systems for low power*, pp. 75–93, 2003.

[14] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *SIGARCH Comput. Archit. News*, vol. 35, pp. 13–23, Jun 2007.

[15] R. Guerra, J. Leite, and G. Fohler, "Attaining soft real-time constraint and energy-efficiency in web servers," in *Proceedings of the 2008 ACM symposium on Applied computing*, SAC '08, pp. 2085–2089, ACM, 2008.

[16] R. Sharma, C. Bash, C. Patel, R. Friedrich, and J. Chase, "Balance of power: dynamic thermal management for internet data centers," *Internet Computing, IEEE*, vol. 9, pp. 42 – 49, Jan 2005.

[17] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, pp. 323–336, USENIX Association, Aug 2008.

[18] X. Zheng and Y. Cai, "Optimal Server Allocation and Frequency Modulation on Multi-Core Based Server Clusters," *International Journal of Green Computing*, vol. 1, no. 2, pp. 18–31, 2010.

[19] R. Nathuji and K. Schwan, "VirtualPower: coordinated power management

in virtualized enterprise systems," in *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, SOSP '07, pp. 265–278, ACM, 2007.

[20] A. Verma, P. Ahuja, and A. Neogi, "Power-aware dynamic placement of HPC applications," in *Proceedings of the 22nd annual international conference on Supercomputing*, ICS '08, pp. 175–184, ACM, 2008.

[21] J. L. Berral, I. n. Goiri, R. Nou, F. Julià, J. Guitart, R. Gavaldà, and J. Torres, "Towards energy-aware scheduling in data centers using machine learning," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, e-Energy '10, pp. 215–224, ACM, 2010.

[22] K. Le, R. Bianchini, T. Nguyen, O. Bilgir, and M. Martonosi, "Capping the brown energy consumption of internet services at low cost," in *Green Computing Conference, 2010 International*, pp. 3 –14, Aug 2010.

[23] G. Quan and X. Hu, "Energy efficient fixed-priority scheduling for real-time systems on variable voltage processors," in *Design Automation Conference, Proceedings*, pp. 828 – 833, 2001.

[24] M. Elnozahy, M. Kistler, and R. Rajamony, "Energy conservation policies for web servers," in *Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems - Volume 4*, USITS'03, pp. 8–8, USENIX Association, 2003.

[25] J. Pouwelse, K. Langendoen, and H. Sips, "Energy priority scheduling for variable voltage processors," in *Proceedings of the 2001 international symposium on Low power electronics and design*, ISLPED '01, pp. 28–33, ACM, 2001.

[26] K. Skadron, T. Abdelzaher, and M. Stan, "Control-theoretic techniques and thermal-rc modeling for accurate and localized dynamic thermal management," pp. 17–28, Feb. 2002.

[27] Q. Wu, P. Juang, M. Martonosi, L. Peh, and D. Clark, "Formal control techniques for power-performance management," *Micro, IEEE*, vol. 25, pp. 52 – 62, Sep 2005.

[28] M. Femal and V. Freeh, "Boosting data center performance through non-uniform power allocation," in *Autonomic Computing, 2005. ICAC 2005. Proceedings. Second International Conference on*, pp. 250 –261, Jun 2005.

[29] C. Lefurgy, X. Wang, and M. Ware, "Server-level power control," *Autonomic Computing, 2007. ICAC '07. Fourth International Conference on*, pp. 4 – 4, May 2007.

[30] R. Graybill and R. Melhem. Power aware computing. Jan 2002. Available: books.google.com.

[31] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," *High-Performance Computer Architecture, 2001. HPCA. The Seventh International Symposium on*, pp. 171 – 182, Dec 2000.

[32] W. Felter, K. Rajamani, T. Keller, and C. Rusu, "A performance-conserving approach for reducing peak power consumption in server systems," in *Proceedings of the 19th annual international conference on Supercomputing*, ICS '05, pp. 293–302, ACM, 2005.

[33] T. Newhall, D. Amato, and A. Pshenichkin, "Reliable adaptable Network RAM," in *Cluster Computing, 2008 IEEE International Conference on*, pp. 2 –12, Oct. 2008.

[34] A. Weissel, B. Beutel, and F. Bellosa, "Cooperative I/O: a novel I/O semantics for energy-aware applications," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 117–129, 2002.

[35] D. P. Helmbold, D. D. E. Long, T. L. Sconyers, and B. Sherrod, "Adaptive disk spin-down for mobile computers," *Mob. Netw. Appl.*, vol. 5, pp. 285–297, December 2000.

[36] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke, "DRPM: dynamic speed control for power management in server class disks," pp. 169 – 179, Jun. 2003.

[37] X. Wang and M. Chen, "Cluster-level feedback power control for performance optimization," in *High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on*, pp. 101 –110, Feb. 2008.

[38] V. Sharma, A. Thomas, T. Abdelzaher, K. Skadron, and Z. Lu, "Power-aware QoS Management in Web Servers," in *Proceedings of the 24th IEEE International*

Real-Time Systems Symposium, RTSS ’03, pp. 63–73, IEEE Computer Society, 2003.

[39] C. Dovrolis, D. Stiliadis, and P. Ramanathan, “Proportional differentiated services: delay differentiation and packet scheduling,” *Networking, IEEE/ACM Transactions on*, vol. 10, pp. 12 – 26, Feb 2002.

[40] 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 *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI’08, pp. 337–350, USENIX Association, 2008.

[41] S. Murugesan, “Harnessing Green IT: Principles and Practices,” *IT Professional*, vol. 10, pp. 24–33, January 2008.

[42] M. Kesavan, A. Ranadive, A. Gavrilovska, and K. Schwan, “Active CoordinaTion (ACT)–Toward Effectively Managing Virtualized Multicore Clouds,” *2008 IEEE International Conference on Cluster Computing*, Jan 2008.

[43] L. Hu, H. Jin, X. Liao, X. Xiong, and H. Liu, “Magnet: A novel scheduling policy for power reduction in cluster with virtual machines,” *Cluster Computing, 2008 IEEE International Conference on*, pp. 13 – 22, Jan 2008.

[44] G. von Laszewski, L. Wang, A. J. Younge, and X. He, “Power-aware scheduling of virtual machines in dvfs enabled clusters,” *Cluster Computing and Workshops, 2009. CLUSTER ’09. IEEE International Conference on*, pp. 1 – 10, Jan 2009.

[45] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle, “Managing energy and server resources in hosting centers,” *SIGOPS Oper. Syst. Rev.*, vol. 35, pp. 103–116, October 2001.

[46] I. Ahmad, S. Ranka, and S. Khan, “Using game theory for scheduling tasks on multi-core processors for simultaneous optimization of performance and energy,” pp. 1–6, April 2008.

[47] E. V. Carrera, E. Pinheiro, and R. Bianchini, “Conserving disk energy in network servers,” in *Proceedings of the 17th annual international conference on Supercomputing*, ICS ’03, pp. 86–97, ACM, 2003.

[48] M. Song, “Energy-aware data prefetching for multi-speed disks in video servers,” in *Proceedings of the 15th international conference on Multimedia*, MULTIMEDIA ’07, pp. 755–758, ACM, 2007.

[49] L. Barroso and U. Holzle, “The Case for Energy-Proportional Computing,” in *Computer*, vol. 40, pp. 33 – 37, Dec 2007.

[50] X. Zhou, Y. Cai, C. Chow, and M. Augusteijn, “Two-tier resource allocation for slowdown differentiation on server clusters,” in *Parallel Processing, 2005. ICPP 2005. International Conference on*, pp. 31 – 38, Jun. 2005.

[51] X. Zhou, Y. Cai, G. Godavari, and C. Chow, “An adaptive process allocation strategy for proportional responsiveness differentiation on Web servers,” *Web Services, 2004. Proceedings. IEEE International Conference on*, pp. 142 – 149, Jun 2004.

[52] C. Dovrolis and P. Ramanathan, “A case for relative differentiated services and the proportional differentiation model,” *Network, IEEE*, vol. 13, pp. 26 – 34, Sep 1999.

[53] X. Zhou and C. Xu, “Harmonic proportional bandwidth allocation and scheduling for service differentiation on streaming servers,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 15, pp. 835 – 848, Sep 2004.

[54] P. Bohrer, E. N. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, C. McDowell, and R. Rajamony, “The case for power management in web servers,” in *Power aware computing*, pp. 261–289, Kluwer Academic Publishers, 2002.

[55] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy, “Optimal power allocation in server farms,” in *SIGMETRICS ’09: Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, pp. 157–168, ACM, 2009.

[56] T. Horvath and K. Skadron, “Multi-mode energy management for multi-tier server clusters,” in *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, PACT ’08, pp. 270–279, ACM, 2008.

[57] E. Elnozahy, M. Kistler, and R. Rajamony, “Energy-efficient server clusters,” *Lecture Notes in Computer Science*, Jan 2003.

[58] M. S. Floyd, S. Ghiasi, T. W. Keller, K. Rajamani, F. L. Rawson, J. C. Rubio, and M. S. Ware, “System power management support in the IBM POWER6

microprocessor," *IBM Journal of Research and Development*, vol. 51, pp. 733 –746, nov. 2007.

[59] T. Imada, M. Sato, Y. Hotta, and H. Kimura, "Power management of distributed web savers by controlling server power state and traffic prediction for QoS," *Parallel and Distributed Processing Symposium, International*, pp. 1–8, 2008.

[60] M. Harchol-Balter, "Task assignment with unknown duration," vol. 49, pp. 260–288, ACM, Mar. 2002.

[61] H. Zhu, H. Tang, and T. Yang;, "Demand-driven service differentiation in cluster-based network servers," *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 679 – 688 vol.2, Mar 2001.

[62] L. Zhang, "A two-bit differentiated services architecture for the internet," *Request for Comments (Informational)*, Jan 1999.

[63] NASA Kennedy Space Center Server Traces. Available: http://ita.ee.lbl.gov/html/traces.html.

[64] L. Benini, A. Bogliolo, G. A. Paleologo, R. Bogliolo, S. Member, and G. D. Micheli, "Policy Optimization for Dynamic Power Management," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, pp. 813–833, 1999.

[65] Q. Qiu and M. Pedram, “Dynamic Power Management Based on Continuous-Time Markov Decision Processes,” in *In Design Automation Conference*, pp. 555–561, 1999.

[66] R. Garg, S. W. Son, M. Kandemir, P. Raghavan, and R. Prabhakar, “Markov Model Based Disk Power Management for Data Intensive Workloads,” in *CCGRID ’09: Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pp. 76–83, IEEE Computer Society, 2009.

[67] D. Niyato, S. Chaisiri, and L. B. Sung, “Optimal Power Management for Server Farm to Support Green Computing,” in *CCGRID ’09: Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pp. 84–91, IEEE Computer Society, 2009.

[68] X. Zheng and Y. Cai, “Achieving Energy Proportionality In Server Clusters,” *International Journal of Computer Networks*, vol. 1, no. 1, pp. 21–35, 2009.

[69] C. Lee and M. Andersland, “Average queueing delays in ATM buffers under cell dropping,” in *Global Telecommunications Conference, 1998. GLOBECOM 98. The Bridge to Global Integration. IEEE*, vol. 4, pp. 2440 –2445 vol.4, 1998.

[70] H. W. Lee, S. H. Cheon, and W. J. Seo, “Queue length and waiting time of the M/G/1 queue under the D-policy and multiple vacations,” *Queueing Syst. Theory Appl.*, vol. 54, no. 4, pp. 261–280, 2006.

[71] Eitan Altman. Constrained Markov Decision Processes. 1999.

[72] X. Zheng and Y. Cai, “Optimal Server Provisioning and Frequency Adjustment in Server Clusters,” *39th International Conference on Parallel Processing GreenCom Workshops*, pp. 504–511, 2010.

[73] M. Kazandjieva, B. Heller, O. Gnawali, W. Hofer, P. Levis, and C. Kozyrakis., “Software or Hardware: The Future of Green Enterprise Computing,” in *Stanford University Technical Report CSTR*, july 2011.

[74] Department of Energy. United States Energy Information Administration. http://www.eia.gov/.

[75] Regional Greenhouse Gas Initiative: An Initiative of the Northeast and Mid-Atlantic States of the U.S. 2008. Avilable: http://www.rggi.org/.

[76] 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 *INFOCOM, 2010 Proceedings IEEE*, pp. 1 –9, march 2010.

[77] M. K. Patterson and D. Fenwick, “The State of Data Center Cooling. A review of current air and liquid cooling solutions,” in *Intel White paper*, 2009.

[78] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, “Making scheduling cool: temperature-aware workload placement in data centers,” in *Proceedings of the annual conference on USENIX Annual Technical Conference*, ATEC ’05, pp. 5–5, USENIX Association, 2005.

[79] Q. Tang, S. Gupta, and G. Varsamopoulos, “Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, pp. 1458 –1472, nov. 2008.

[80] A. Banerjee, T. Mukherjee, G. Varsamopoulos, and S. K. S. Gupta, “Cooling-aware and thermal-aware workload placement for green HPC data centers,” in *Proceedings of the International Conference on Green Computing*, GREENCOMP ’10, pp. 245–256, IEEE Computer Society, 2010.

[81] E. Pakbaznia and M. Pedram, “Minimizing data center cooling and server power costs,” in *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design*, ISLPED ’09, pp. 145–150, ACM, 2009.

[82] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, “Cutting the electric bill for internet-scale systems,” in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, SIGCOMM ’09, pp. 123–134, ACM, 2009.

[83] Federal energy regulatory commission. www.ferc.gov.

[84] Point Carbon, http://www.pointcarbon.com/.

[85] K. Lim, P. Ranganathan, J. Chang, C. Patel, T. Mudge, and S. Reinhardt, “Understanding and designing new server architectures for emerging warehouse-computing environments,” in *Proceedings of the 35th Annual*

International Symposium on Computer Architecture, ISCA '08, pp. 315–326, IEEE Computer Society, 2008.

[86] A. Qureshi, "Power-Demand Routing in Massive Geo-Distributed Systems ," Sep. 2010.

[87] Z. Zhang, M. Zhang, A. Greenberg, Y. C. Hu, R. Mahajan, and B. Christian, "Optimizing cost and performance in online service provider networks," in *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, NSDI'10, pp. 3–3, USENIX Association, 2010.

[88] D. K. Goldenberg, L. Qiuy, H. Xie, Y. R. Yang, and Y. Zhang, "Optimizing cost and performance for multihoming," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '04, pp. 79–92, ACM, 2004.

[89] C. A. Waldspurger, "Memory resource management in VMware ESX server," *SIGOPS Oper. Syst. Rev.*, vol. 36, pp. 181–194, December 2002.

[90] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *Proceedings of the nineteenth ACM symposium on Operating systems principles*, SOSP '03, pp. 164–177, ACM, 2003.

[91] F. Hermenier, X. Lorca, J.-M. Menaud, G. Muller, and J. Lawall, "Entropy: a consolidation manager for clusters," in *Proceedings of the 2009 ACM*

SIGPLAN/SIGOPS international conference on Virtual execution environments, VEE '09, pp. 41–50, ACM, 2009.

[92] V. Petrucci, O. Loques, and D. Mossé, "A dynamic optimization model for power and performance management of virtualizedLive migration of virtual machines clusters," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, e-Energy '10, pp. 225–233, ACM, 2010.

[93] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, NSDI'05, pp. 273–286, USENIX Association, 2005.

[94] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-box and gray-box strategies for virtual machine migration," in *Proceedings of the 4th USENIX conference on Networked systems design and implementation*, NSDI'07, pp. 17–17, USENIX Association, 2007.

[95] J. Shahabuddin, A. Chrungoo, V. Gupta, S. Juneja, S. Kapoor, and A. Kumar, "Stream-Packing: Resource Allocation in Web Server Farms with a QoS Guarantee," in *Proceedings of the 8th International Conference on High Performance Computing*, HiPC '01, pp. 182–191, Springer-Verlag, 2001.

[96] M. Hoyer, K. Schröder, and W. Nebel, "Statistical static capacity management in virtualized data centers supporting fine grained QoS specification," in *Proceedings*

of the 1st International Conference on Energy-Efficient Computing and Networking, e-Energy '10, pp. 51–60, ACM, 2010.

[97] W. Leinberger, G. Karypis, and V. Kumar, "Multi-Capacity Bin Packing Algorithms with Applications to Job Scheduling under Multiple Constraints," in *Proceedings of the 1999 International Conference on Parallel Processing*, ICPP '99, pp. 404–, IEEE Computer Society, 1999.

[98] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No "power" struggles: coordinated multi-level power management for the data center," *SIGARCH Comput. Archit. News*, vol. 36, pp. 48–59, March 2008.

[99] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, "EnaCloud: An Energy-Saving Application Live Placement Approach for Cloud Computing Environments," in *Cloud Computing, 2009. CLOUD '09. IEEE International Conference on*, pp. 17 –24, Sep. 2009.

[100] S. Mehta and A. Neogi, "ReCon: A tool to Recommend dynamic server Consolidation in multi-cluster data centers," in *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*, pp. 363 –370, Apr. 2008.

[101] L. Liu, H. Wang, X. Liu, X. Jin, W. B. He, Q. B. Wang, and Y. Chen, "GreenCloud: a new architecture for green data center," in *Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session*, ICAC-INDST '09, pp. 29–38, ACM, 2009.

[102] K. Le, R. Bianchini, J. Zhang, Y. Jaluria, J. Meng, and T. D. Nguyen, "Reducing electricity cost through virtual machine placement in high performance computing clouds," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pp. 22:1–22:12, ACM, 2011.

[103] M. Mazzucco, D. Dyachuk, and R. Deters, "Maximizing Cloud Providers' Revenues via Energy Aware Allocation Policies," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pp. 131 –138, july 2010.

[104] K. Ye, X. Jiang, D. Huang, J. Chen, and B. Wang, "Live Migration of Multiple Virtual Machines with Resource Reservation in Cloud Computing Environments," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pp. 267 –274, july 2011.

[105] Amazon EC2. http://aws.amazon.com/ec2/.

[106] L. M. Leemis, "Nonparametric Estimation of the Cumulative Intensity Function for a Nonhomogeneous Poisson Process," *Management Science*, vol. 37, no. 7, pp. 886–900, 1991.

[107] Parallel Workloads Archive. http://www.cs.huji.ac.il/labs/parallel/workload/.