Achieving Energy Proportionality In Server Clusters

Xinying Zheng zxying@mtu.edu
Ph.D Candidate /Electrical and Computer Engineering
Michigan Technological University
Houghton, 49931, US

Yu Cai cai@mtu.edu
Associate Professor /School of Technology
Michigan Technological University
Houghton, 49931, US

Abstract

Green computing is a hot issue that has received a great amount of interests in the past few years. Energy proportionality is a principal to ensure that energy consumption is proportional to the system workload. Energy proportional design can effectively improve energy efficiency of computing systems. In this paper, an energy proportional model is proposed based on queuing theory and service differentiation in server clusters, which can provide controllable and predictable quantitative control over power consumption with theoretically guaranteed service performance. Further study for the transition overhead is carried out corresponding strategy is proposed to compensate the performance degradation caused by transition overhead. The model is evaluated via extensive simulations and justified by the real workload data trace. The results show that our model can achieve satisfied service performance while still preserving energy efficiency in the system.

Keywords: green computing, energy proportional, performance differentiation, transition overhead

1. INTRODUCTION

Green computing is to support personal and business computing needs in a green and sustainable manner, such as minimizing strain and impact on resources and environment. Computing systems, particularly enterprise data centers and high-performance cluster systems consume a significant amount of energy, thus placing an increasing burden on power supply and operational cost. For example, the power consumption of enterprise data centers in the U.S. doubled between 2000 and 2005, and will likely triple again in a few years [1]. In 2005, US data centers consumed 45 billion kWH, which was roughly 1.2 percent of the total amount of US electricity consumption, resulting in utility bills of $2.7 billion [2]. In 2006, the U.S. Congress passed bills to raise the IT industry's role in energy and environmental policy to the national level [3]. Furthermore, it is estimated that servers consume 0.5 percent of the world's total electricity [4], which if current demand continues, is projected to quadruple by 2010. Some analysts predicted that IT infrastructure power usage will soon cost more than the hardware itself [5].

Many of the existing works on power management in server clusters rely heavily on heuristics or feedback control [6][7][8][9]. An important principle in green computing is to ensure energy consumption proportionality, which states that the energy consumption should be proportional to the system workload [10]. For example, when there is no or little workload, the system should consume no or little energy; when workload increases, energy consumption should increase proportionally, until the system reaches the full workload. This idea can effectively improve the energy efficiency in real-life usage. Energy proportionality is relatively hard to be achieved on a standalone server because of hardware constraints. However, it is possible to achieve energy proportionality on a server cluster, since we can control the number of active and inactive nodes in a server cluster.

In this paper, we propose an energy proportional model in a server cluster and study its performance in both single and multiple classes' scenarios. We further investigate the transition overhead based on this model. The simulation results show that the energy proportional model can provide controllable and predictable quantitative control over power consumption with theoretically guaranteed service performance.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces the energy proportional model. Performance metrics and servers allocation strategy are introduced in section 4. Section 5 evaluates the model and discusses the transition overhead influence, a strategy is also proposed to compensate the transition overhead in this section, the model is further evaluated based on the real workload data trace, and the last section concludes the paper.

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 single server and server clusters.

A. Single Server

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 [11][12][13]. 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 [14][15] proposed power control schemes based on feedback control theory. Femal et al. [16] developed an algorithm based on linear programming. In [8], 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 [17]. 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 [14] [18].

Research work on memory is often combined with processors and disks. In [19], the authors used open-loop control to shift power between processor and memory to maintain a server power budget. In [20], 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 [21][22]. Sudhanva et al. [23] presented a new approach called DRPM to modulate disk speed dynamically, and a practical implementation was provided for this mechanism.

B. Server Clusters

In recent years, power management has become one of the most important concerns on server clusters. Some methods proposed on a single server can be extended to server clusters. In [24][25], 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 [9], the authors extended the feedback control scheme to clusters. Power has been used as a tool for application-level performance requirements. Sharma et al. [26] proposed feedback control schemes to control application-level quality of service requirements. Chen et al. [27] 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 [28], the authors laid out policies for workload placement to promote uniform temperature distribution using active thermal zones.

Vary-On Vary-off (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 [29][30][28]. In [31], 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 [32]. The authors tested server provisioning and load dispatching on the MSN instant messaging framework, and evaluated various load skewing techniques to trade off energy saving and quality of service.

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 [33]. In [34], 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. [35] used live migration of virtual machines to transfer load among the nodes on a multilayer ring-based overlay. In [4], researchers scheduled virtual machines in a computer cluster to reduce power consumption via the technique of Dynamic Voltage Frequency Scaling (DVFS). An economy driven energy and resource management framework was presented for clusters in [36]. Each service ”bids” for resources as a function of delivered performance. In [37], 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 multispeed disks for servers [23][38]. 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 [39].

3. ENERGY PROPORTIONAL MODEL

A. Energy Proportional Model on a Single Server

The energy proportional model states that energy consumption *P* should be proportional to the workload λ, while ensuring service performance.

$$P = a * \lambda + b \tag{1}$$

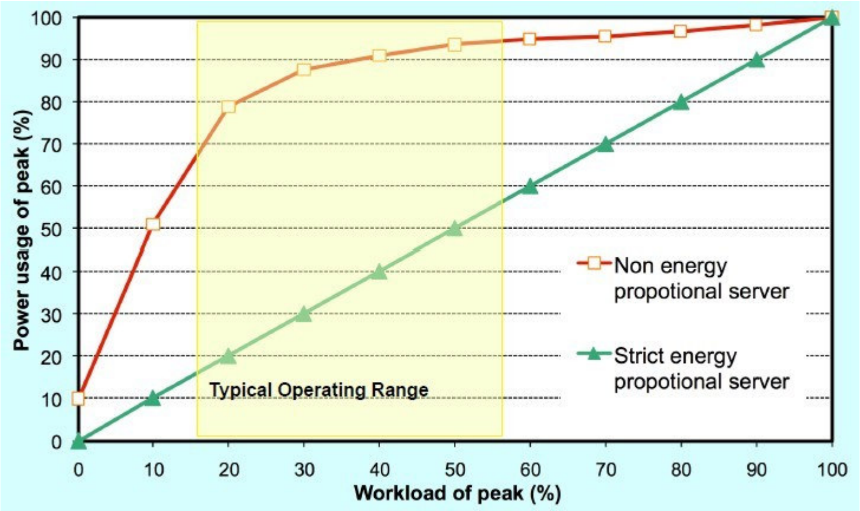

Fig. 1. The energy consumption curves of non-energy proportional server and strict energy proportional server.

Figure.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 see that in a non-energy proportional server, it still consumes about half of its full power when doing virtually no work [10]. 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. Energy-proportional designs would enable large energy savings on servers. However, most servers nowadays are CPU, memory and hard disk intensive servers. The energy consumption of CPU is almost linear to its utilization [32]. But memory and hard disks are nonlinear energy consumption components. As a result, energy proportionality is not easy to be achieved on a standalone sever because of the hardware constraints.

B. Energy Proportional Model on Server Clusters

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. It is known that to quantitatively control energy consumption, one feasible way is 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. 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 of a server cluster is:

$$P = \sum m * P_{ac} + \left(M - \sum m\right) * P_{in} \tag{2}$$

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:

$$\frac{P}{\lambda} = \frac{P_{\max}}{\Lambda} * r \tag{3}$$

where $P_{\max} = M * P_{ac}$. r is a parameter, which adjusts the energy consumption curve in Figure 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. With the help of (2), we can rewrite equation (3) as:

$$\sum m = (\frac{P_{ac}}{\Lambda / M} * r - M * P_{in}) / (P_{ac} - P_{in}) \quad (4)$$

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 paper, this assumption will not affect the performance of our model. We finally achieve that the total number of active servers $\sum m$ is determined by the system workload λ :

$$\sum m = \frac{\lambda}{\Lambda/M} * r \quad (5)$$

The number of servers may not be an integer based on (5). We will set the integer no less than $\sum m$, which is the minimal number of servers to run in fully active mode.

4. SERVERS ALLOCATION BASED ON ENERGYPROPORTIONAL MODEL

An important task of energy aware computing is to achieve energy efficiency while ensuring performance. In this section, we will describe how to allocate servers according to workload, while ensuring quality of services (QoS) metrics.

A. Performance Metrics

One important and commonly used QoS metric on Internet servers 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 [40][41][42][43]. 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/λ*, 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:

$$f(x) = \frac{1}{1-(k/p)^{\alpha}} \alpha k^{\alpha} x^{-\alpha-1} \quad (6)$$

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

$$K(\alpha, k, p) = \frac{\alpha k^{\alpha}}{1-(k/p)^{\alpha}} \quad (7)$$

then we have:

$$E[X]=\int_k^p f(x)dx=\begin{cases}\dfrac{K(\alpha,k,p)}{K(\alpha-1,k,p)} & if\,\alpha\neq 1;\\ (\ln p-\ln k)K(\alpha,k,p) & if\,\alpha=1.\end{cases} \tag{8}$$

Similarly, we can derive $E[X^2]$ and $E[X]$

$$E[X^2]=\int_k^p f(x)x^2dx=\frac{K(\alpha,k,p)}{K(\alpha-2,k,p)} \tag{9}$$

$$E[X^{-1}]=\int_k^p f(x)x^{-1}dx=\frac{K(\alpha,k,p)}{K(\alpha+1,k,p)} \tag{10}$$

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

$$E[W]=\frac{\lambda E[X^2]}{2(1-\lambda E[X])} \tag{11}$$

We can derive a closed-form expression of the expected slowdown in an M/G/1 queue on a single Internet server.

$$E[S]=E[W]E[X^{-1}]=\frac{\lambda E[X^2]E[X^{-1}]}{2(1-\lambda E[X])} \tag{12}$$

The expected request time with the incoming job rate is:

$$E[R]=E[W]+E[X]=\frac{\lambda E[X^2]}{2(1-\lambda E[X])}+E[X] \tag{13}$$

B. Servers Allocation for a Single Class

In this section, we assume all the incoming requests are classified into just one class. We want to ensure the QoS metrics based on different workload λ. For example, the expected request time of the incoming jobs should stay within a threshold, $E[R]<\beta$. We assume $\sum m$ is the number of active server nodes handling the incoming requests. When using a round-robin dispatching policy, the packet arrival rate of a node is $\lambda/\sum m$. The expected request time in a server cluster can be calculated as:

$$E[R]=\frac{\lambda E[X^2]}{2\left(\sum m-\lambda E[X]\right)}+E[X]<\beta \tag{14}$$

Based on the above energy proportionality (5), equation (14) can be re-written as:

$$E[R]=\frac{E[X^2]}{2\left(r*M/\Lambda-E[X]\right)}+E[X]<\beta \tag{15}$$

It is easy to observe that request time is not depending on workload, we can just adjust parameter *r* to satisfy different performance thresholds.

C. Servers Allocation on Service Differentiation

Now we study server allocation schemes for service differentiation. In a cluster system, the incoming requests are often classified into *N* classes. Each class may have different QoS requirements. We assume m_i is the number of active server nodes in class i, and λ_i is the arrival rate in class i. The expected slowdown of class i in a server cluster can be calculated as:

$$E[S_i] = \frac{\lambda_i E[X^2]E[X^{-1}]}{2\left(m_i - \lambda_i E[X]\right)} \tag{16}$$

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 paper will not be affected by the selection of performance metrics.

We adopt a relative service differentiation model where the QoS factor of slowdown between different classes are based on their predefined differentiation parameters.

$$\frac{E[S_i]}{E[S_j]} = \frac{\delta_i}{\delta_j} \tag{17}$$

Where $1 \le i, j \le N$:

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 [39].

Based on the above energy proportionality and service differentiation model, according to formula (5)(17), we can derive the server allocation scheme in a cluster system as

$$m_i = \lambda_i E[X] + \frac{\tilde{\lambda}_i \sum_{i=1}^{N} \lambda_i \left(\frac{M}{\Lambda} * r - E[X] \right)}{\sum_{i=1}^{N} \tilde{\lambda}_i} \tag{18}$$

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 formula (18) 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:

$$E[S_i] = \frac{\delta_i E[X^2]E[X^{-1}] \sum_{i=1}^{N} \tilde{\lambda}_i}{2\sum_{i=1}^{N} \lambda_i \left(\frac{M}{\Lambda} * r - E[X] \right)} \tag{19}$$

From (19) 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:

$$E[R_i] = \frac{\delta_i E[X^2] \sum_{i=1}^{N} \tilde{\lambda}_i}{2\sum_{i=1}^{N} \lambda_i \left(\frac{M}{\Lambda} * r - E[X] \right)} + E[X] \le \beta_i \tag{20}$$

β_i is request time constraint for class i . We can learn from equation (20), request time in class i is also independent of workload, but depends on both the pre-specified parameter δ_i and *r*.

The performance is controllable based on our energy proportional model, with acceptable performance degradation; large amounts of energy can be saved.

5. PERFORMANCE EVALUATION

A. Simulation Results

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 [44]. The number of servers in the cluster is 20. And we set the normalized maximum jobs one server can handle Λ/M = 1. We set the power consumption 160W for active nodes [32].

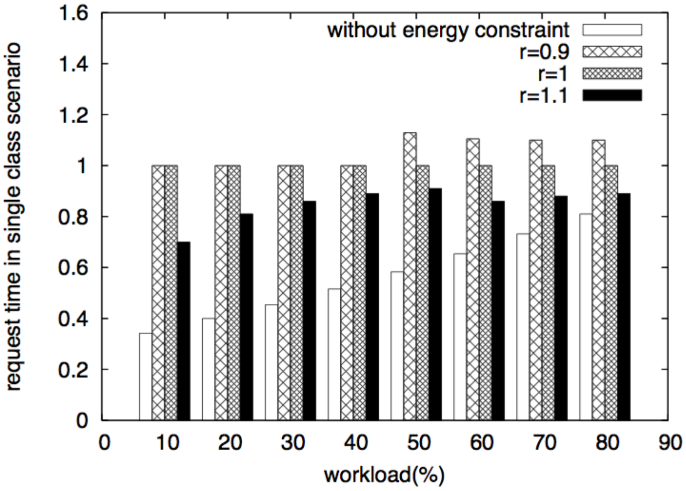

Fig. 2. Comparison of request time between non-energy proportional model and energy Proportional model. *r* is set differently according to different requirements of performance in a single class scenario.

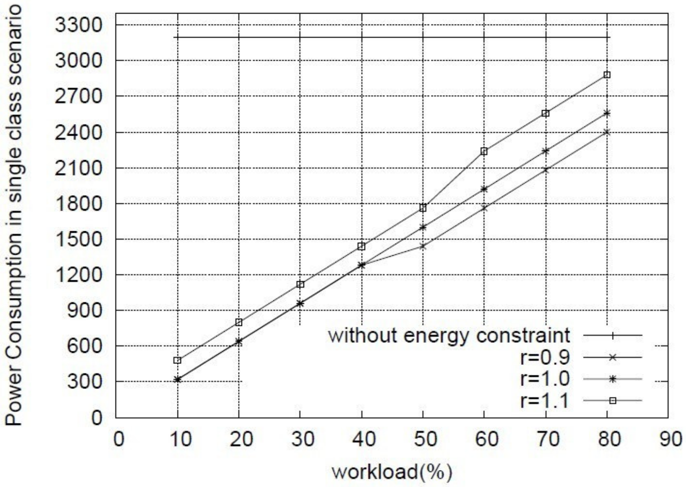

Fig. 3. Comparison of power consumption between non-energy proportional model and energy proportional model in a single class scenario. *r* is set differently according to different requirements of performance. we can achieve considerable energy saving with energy proportional model.

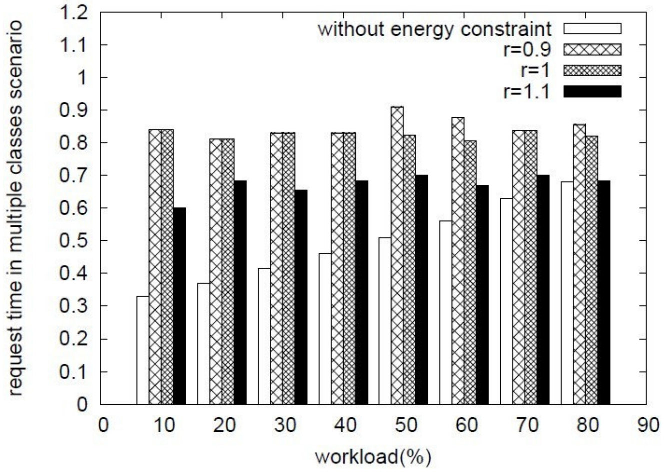

Fig. 4. Comparison of request time in higher priority class between non-energy proportional model and energy proportional models. *r* is set differently according to different requirements of performance in a multiple classes scenario.

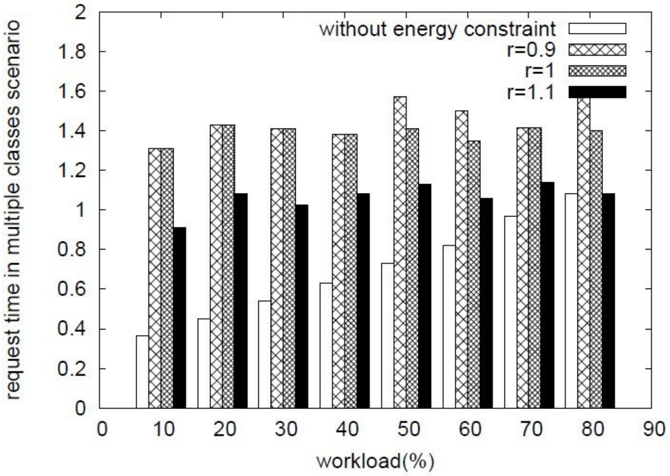

Fig. 5. Comparison of request time in lower priority class between Non-energy proportional model and energy proportional models. *r* is set differently according to different requirements of performance in a multiple classes scenario.

We first evaluate the energy proportional model for the single class scenario. We set the request time β= 0.9, β= 1.1 and β= 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.

As Figure 2 indicates that the request time is always around the pre-defined performance parameter under different workload. The request time increases as the value of *r* decreases. The results show that with adjustable parameter *r* desirable service performance can be achieved. Figure 3 compares the energy consumption of energy proportional model and non-energy proportion model for a single class scenario. We can achieve better energy efficiency under low workload, which leads to large amounts of energy saving in a server cluster.

Next, we compare the performance metrics in a multiple classes' scenario, as shown in Figure 4, 5, 6. The number of classes is normally two or three [45][46]. In this paper we choose two classes of incoming requests. We set the target slowdown ratio $\delta_2 : \delta_1$ = 2 : 1. The energy curve parameters 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, 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 7 also compares the energy consumptions for proportional and non-proportional models in multiple classes scenario.

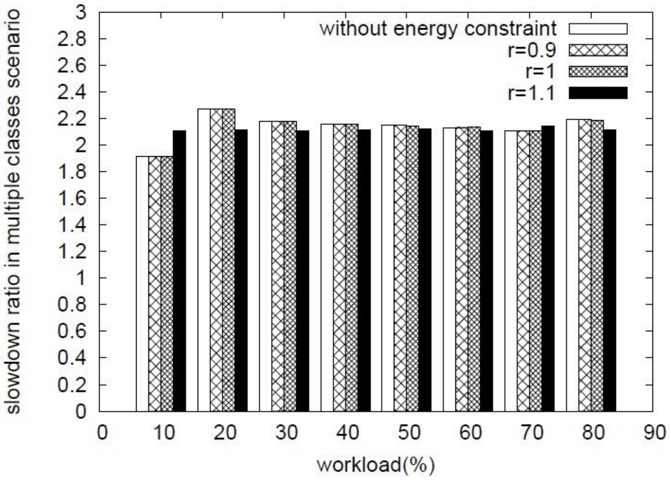

Fig. 6. Comparison of slowdown ratio between non-energy proportional model and energy proportional models. *r* is set by different requirements of performance in a multiple classes scenario.

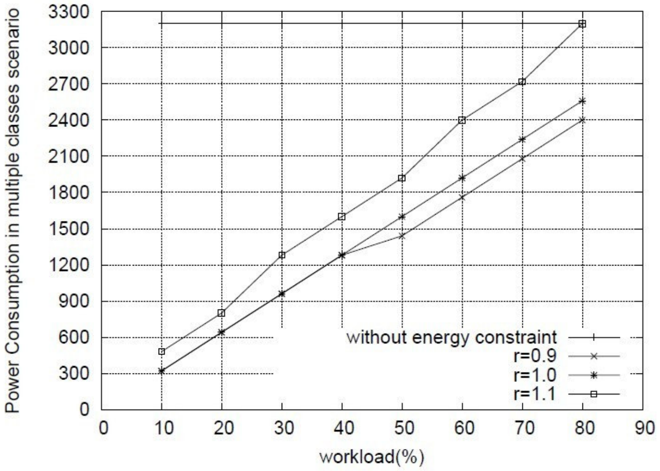

Fig. 7. Comparison of power consumption between non-energy proportional model and energy proportional model in multiple classes scenario. *r* is set by different requirements of performance. We can achieve considerable energy saving compare to the non-energy proportional model.

B. Transition Overhead Analysis

The model proposed in this paper is a continual allocation 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, this 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 performance caused by transition overhead under different time. Figure 8 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, since as the workload decreases, the number of active servers will decline, which will not cause performance degradation. The y-axis 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.

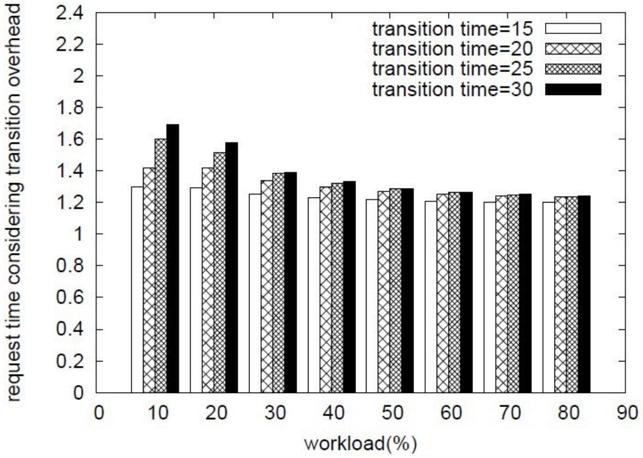

Fig. 8. The effect to performance of transition overhead in energy proportional model, the transition time is set to be 15,20,25,30 respectively

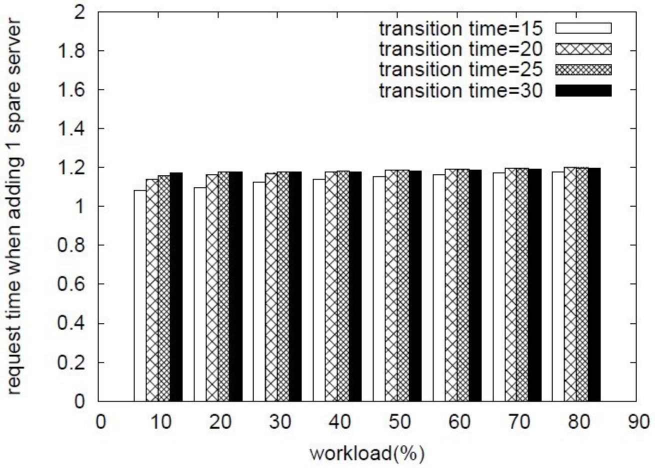

Fig. 9. Request time after adding one spare server based on energy proportional model in a single class scenario, the transition time is set to be 15,20,25,30 respectively.

It is important to make sure that the QoS is not sacrificed excessively in favor of power and energy savings. Spare servers are added to solve the problem of transition overhead. Figure 9, 10 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.

C. Performance Evaluation Based on Real Workload Data Trace

To evaluate the model on realistic traffic patterns, we use an hour's workload trace collected by Lawrence Berkeley National Laboratory [47]. Request time threshold is set to be β= 0.6 and $r = 1$. Figure 11 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 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 predefined 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 demonstrates proper spare servers should be set to compensate the performance degradation.

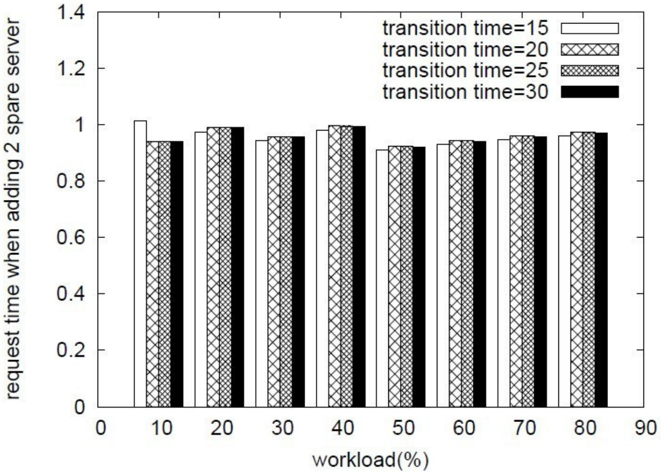

Fig. 10. Request time when adding two spare servers based on energy proportional model in a single class scenario, the transition time is set to be 15,20,25,30 respectively.

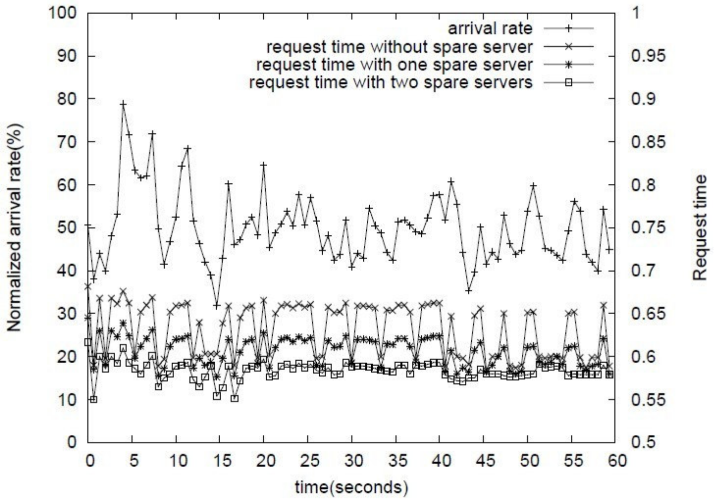

Fig. 11. Request time when adding two spare servers based on energy proportional model in a single class scenario.

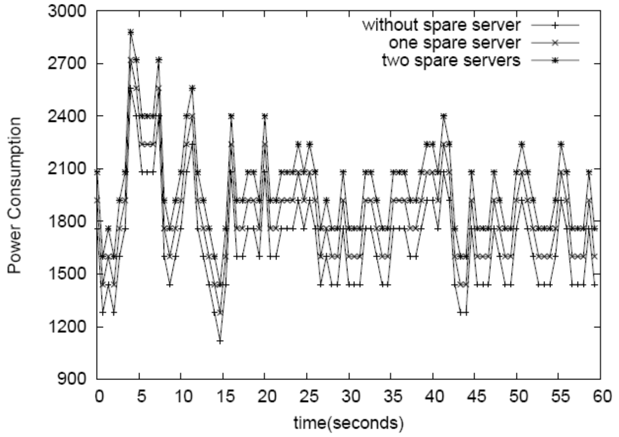

Fig. 12. Power consumption when adding two spare servers based on energy proportional model in a single class scenario.

Figure 12 evaluates the power consumption based on our model under real workload data trace. The system arrival rate is the same as shown in figure 11. 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.

6. CONCLUSION AND FUTURE WORK

Energy management becomes a key issue in server clusters and data centers. This paper aims at providing effective strategies to reduce power consumption and reduce the impact of performance degradation. We summarize out work as follows: first, the energy proportional model based on queuing theory can provide accurate, controllable and predictable quantitative control over power consumption; second, we analyze the effect of transition overhead and propose a strategy to improve the performance efficiency. Finally we evaluate the energy proportional model via simulation. Simulation results show that the energy proportional model can achieve predictable and controllable proportional energy consumption and desirable performance in a server cluster.

Future work would include studying the effect on performance when applying different dispatching strategies in our model. We are still trying to extend the server states to solve the problem of non-integer number of nodes, which will further enhance the energy efficiency. Eventually our goal is to apply our model to the real Internet web servers in the future.

7. REFERENCES

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

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

[3] US Congress. House bill 5646. To study and promote the use of energy efcient computer servers in the united states.http://www.govtrack.us/congress/bill.xpd?bill=h109-5646. Retrieved: 02-14-2008.

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

[5] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, and Q. Wang, “Managing server energy and operational costs in hosting centers,” Proceedings of the 2005 ACM SIGMETRICS international, Jan 2005.

[6] C. Lefurgy, K. Rajamani, F. Rawson, and W. Felter, “Energy management for commercial servers,” Computer, Jan 2003.
[7] Y. Lu and G. D. Micheli, “Operating-system directed power reduction,” In proc. of international symposium on Low power electronics and design, Jan 2000.
[8] 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.
[9] X. Wang and M. Chen, “Cluster-level feedback power control for performance optimization,” In Proc. of Symposium on High-Performance Computer Architecture, Jan 2008.
[10] L. Barroso and U. Holzle, “The case for energy-proportional computing,” Computer, vol. 40, pp. 33 – 37, Dec 2007.
[11] G. Quan and X. Hu, “Energy efficient fixed-priority scheduling for real-time systems on variable voltage processors,” Design Automation Conference, Jan 2001.
[12] M. Elnozahy, M. Kistler, and R. Rajamony, “Energy conservation policies for web servers,” Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems, Jan 2003.
[13] J. Pouwelse, K. Langendoen, and H. Sips, “Energy priority scheduling for variable voltage processors,” Proceedings of the 2001 international symposium on Low power, Jan 2001.
[14] 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.
[15] Q. Wu, P. Juang, M. Martonosi, L. Peh, and D. Clark, “Formal control techniques for power-performance management,” IEEE Micro, Jan 2005.
[16] M. Femal and V. Freeh, “Boosting data center performance through non-uniform power allocation,” Autonomic Computing, Jan 2005.
[17] R. Graybill and R. Melhem, “Power aware computing,” books.google.com, Jan 2002.
[18] D. Brooks and M. Martonosi, “Dynamic thermal management for high-performance microprocessors,” High-Performance Computer Architecture, Jan 2001.
[19] W. Felter, K. Rajamani, T. Keller, and C. Rusu, “A performanceconserving approach for reducing peak power consumption in server systems,” Proceedings of the 19th annual international conference on Supercomputing, Jan 2005.
[20] T. Newhall, D. Amato, and A. Pshenichkin, “Reliable adaptable network ram,” 2008 IEEE International Conference on Cluster Computing, Jan 2008.
[21] A. Weissel, B. Beutel, and F. Bellosa, “Cooperative I/O–a novel I/O semantics for energy-aware applications,” usenix.org.
[22] D. Helmbold, D. Long, T. Sconyers, and B. Sherrod, “Adaptive disk spindown for mobile computers,” Mobile Networks and Applications, Jan 2000.
[23] S. Gurumurthi, A. Sivasubramaniam, and M. Kandemir, “DRPM: dynamic speed control for power management in server class disks,” Computer Architecture, Jan 2003.
[24] 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.
[25] E. Pinheiro, R. Bianchini, E. Carrera, and T. Heath, “Dynamic cluster reconfiguration for power and performance,” Compilers and operating systems for low power, Jan 2001.
[26] V. Sharma, A. Thomas, T. Abdelzaher, K. Skadron, and Z. Lu, “Poweraware qos management in web servers,” 24th IEEE Real-Time Systems Symposium, Jan 2003.
[27] C. Dovrolis, D. Stiliadis, and P. Ramanathan, “Proportional differentiated services: Delay differentiation and packet scheduling,” Proceedings of the conference on Applications, Jan 1999.
[28] R. Sharma, C. Bash, C. Patel, and R. Friedrich, “Balance of power: Dynamic thermal management for internet data centers,” IEEE Internet Computing, Jan 2005.
[29] X. Fan, W. Weber, and L. Barroso, “Power provisioning for a warehouse-sized computer,” Proceedings of the 34th annual international conference on architecture, Jan 2007. B-2-2-2.
[30] R. Guerra, J. Leite, and G. Fohler, “Attaining soft real-time constraint and energy-efficiency in web servers,” Proceedings of the 2008 ACM symposium on Applied computing, Jan 2008.
[31] S. Nedevschi, L. Popa, G. Iannaccone, and S. Ratnasamy, “Reducing network energy consumption via sleeping and rate-adaptation,” NSDI, Jan 2008.
[32] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, and L. Xiao, “Energy-aware server provisioning and load dispatching for connection-intensive internet services,” Proceedings of the 5th USENIX

Symposium on Networked Systems Design and Implementation, Jan 2008.

[33] S. Murugesan, “Harnessing green it: Principles and practices,” IT Professional, Jan 2008.

[34] 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.

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

[36] J. Chase, D. Anderson, P. Thakar, and A. Vahdat, “Managing energy and server resources in hosting centers,” Proceedings of the eighteenth ACM symposium on Operating Operating System Principles, Jan 2001.

[37] 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.

[38] E. Carrera, E. Pinheiro, and R. Bianchini, “Conserving disk energy in network servers,” Proceedings of the 17th annual international conference on Supercomputing, Jan 2003.

[39] M. Song, “Energy-aware data prefetching for multi-speed disks in video servers,” Proceedings of the 15th international conference on Supercomputing, Jan 2007.

[40] X. Zhou, Y. Cai, C. Chow, and M. Augusteijn, “Two-tier resource allocation for slowdown differentiation on server clusters,” Parallel Processing, Jan 2005.

[41] 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.

[42] C. Dovrolis and P. Ramanathan, “A case for relative differentiated services and the proportionaldifferentiation model,” Network, Jan 1999.

[43] 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.

[44] M. Harchol-Balter and C. U. PITTSBURGH, “Task assignment with unknown duration,” doi.ieeecomputersociety.org, Jan 1999.

[45] 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.

[46] L. Zhang, “A two-bit differentiated services architecture for the internet,” Request for Comments (Informational), Jan 1999.

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