A Queueing Model Based Periodical ACPI Global State Control for Web Server Cluster Energy Management

Cheng-Jen Tang[1], Miau-Ru Dai[2]

[1] Department of Electrical Engineering, Tatung University
[2] Smart Grid Group, Delta Networks, Inc. Taipei, Taiwan
ctang@ttu.edu.tw, anna.dai@delta.com.tw

Abstract

The queueing analysis of the call center staff planning is a common process for efficient resource allocation. In many ways, although at different time scales, this process is analogous to the provisioning of the nodes in a web hosting server cluster. Based on the well proven strategies learned from the call centers, this paper proposes an energy management method that (1) models a server cluster as an *M/M/m* queueing system, (2) periodically switches the ACPI (Advanced Configuration and Power Interface) global states of a server node, and (3) obtains an adequate number of active nodes constrained by a desired response time. The proposed process is simulated with a real world trace log. The simulation result shows that a significant energy saving can be achieved while maintaining a desirable response time.

Keywords: Web server cluster, Queueing model, ACPI, Energy efficiency

1 Introduction

The number of Internet users has exceeded 3 billion by 2014 [1]. To serve these users, web service providers often install additional server nodes resulting in huge energy waste during the off-peak periods. Saving energy can be done by dynamically adjusting the computing capacity to match the traffic intensity [2]. This problem is analogous to the dynamic staff sizing problem in a telephone call center. Call centers can be naturally viewed as queueing systems [3]. In a call center, the customers are callers, servers are telephone agents, and tele-queues consist of callers that await service. *M/M/m* model, known as Erlang-C [4], is widely adopted in many call centers [5-7].

The power management of earlier computer systems is often performed through Advanced Power Management (APM) protocols [8]. With APM, OS has little authority and flexibility to control the hardware activities. Therefore, ACPI specification was developed to establish the interfaces enabling OS directed motherboard device configuration and power management (OSPM) of both devices and entire systems [9].

This paper proposes an energy management solution for the web server clusters based on *M/M/m* model and ACPI global state control. This method balances the response time and the consumed energy. The simulation result shows a significant energy saving while maintaining the desired response time. This paper makes the following specific contributions.

- *M/M/m* model is adopted in the proposed approach. Although *M/M/m* model is quite restrictive, it is capable of estimating the expected node number and response time.
- This paper presents an integration of a theoretical model (*M/M/m*) with an industrial standard (ACPI). This integration enables the proposed approach to be practically realized.
- The ACPI global state control has to be operated within a reasonable short period of time, because the traffic may be implicitly smoothed by a long interval. A suspend-resume action requires enough time to complete its power-on operation for an effective state control. This study defines the Energy Saving Effectiveness (ESE) based on the ratio of the planned energy saving to the suspend-resume overhead, and finds an adequate state-control interval.
- An algorithm is designed to find the sufficient number of activated server nodes that satisfy the desired response time.

The paper is organized as follows. Section 2 discusses the general energy management processes for web server clusters. Section 3 details the system model of the proposed approach. Section 4 presents the simulation process, and discusses the results. Section 5 concludes this paper.

2 Energy Management Processes for Web Servers

The major reason of the web energy consumption

*Corresponding Author: Cheng-Jen Tang; E-mail: ctang@ttu.edu.tw
DOI: 10.3966/160792642018011901017

problem is the inefficient utilization of computing resources. It is common that server clusters do not carefully consider how best to configure the environment to maximize availability with the minimum power consumption [10].

To address the future challenges of data centers, the U.S. American Recovery and Reinvestment Act of 2009 (ARRA2009) [11] and the European Commission 7th Framework Programme for Research and Technological Development (FP7) [12] both fund a number of research and development projects.

Low and Tang [13] use *M/GI/*1 processor sharing (PS) system to model a server, and perform load balancing among multiple geographically distributed data centers. Fortune [14] proposes a collection of techniques, named as *Rate Adaption*, that match energy consumption with traffic condition. Pfeiffer and Kulali [15] develop an On-Demand operational model to reduce the energy waste through idle power consumption. Their approach also finds an optimal number of required servers. FIT4Green [16] designs an energy-aware layer of plug-ins on top of existing data center management tools to reduce energy consumption without sacrificing Service Level Agreements (SLA) and Quality of Service (QoS). GAMES [17] contains a set of methodologies and tools to manage energy efficiency in data centers. All4Green [18] explores the energy saving potential of the ecosystem where data centers operate rather than individual ICT modules. CoolEmAll [19] develops data center simulation, visualization and decision tools (SVD Toolkit) that use data center efficiency building blocks (DEBBs) to perform simulations. Dynamically adjusting the number of active servers [20], that is, Vary-On/Vary-Off (VOVO) scheme, improves energy efficiency of data centers. On the other hand, Dynamic Voltage/Frequency Scaling (DVFS) [21] can also save energy by dynamically changing the frequency and voltage of servers. Both VOVO and DVFS are commonly adopted technologies to improve energy efficiency in these projects. Wei et al. [22] present the benefits of DVFS and VOVO for energy management in data centers.

Among mentioned projects, some common energy management processes can be identified:

- Modeling the traffic patterns.
- Modeling the energy consumption patterns of the system.
- Forecasting the idle periods.
- Identifying the constraints.
- Defining the management methods.

3 System Model and Control Mechanism

M/M/m model assumes a steady-state environment, in which arrivals conform to a Poisson process, service times are exponentially distributed, and customers and servers are statistically identical and act independently of each other. Therefore, this study assumes that the number of the active server node is m; the server nodes in the cluster are identical; the arrivals of Internet requests during a period t form a Poisson process with rate λ; the inter-arrival times are exponentially distributed with the mean $\frac{1}{\lambda}$; the services times are exponentially distributed; the server rate is μ; and the requests are served in the First-Come-First-Served (FCFS) manner.

The utilization rate of the system is defined as $\rho = \frac{\lambda}{m\mu}$. $\rho < 1$ is required for the stability of this system. Denoting k as a state of the system, the steady-state probabilities π_k are:

$$\pi_k = \begin{cases} \pi_0 \dfrac{(m\rho)^k}{k!} & ,0 < k < m \\ \pi_0 \dfrac{(m\rho)^k}{m!m^{k-m}} & ,m \le k \end{cases} \quad \textbf{(1)}$$

Since $\sum_{k=0}^{\infty} \pi_k = 1$, π_0 can be obtained as:

$$\pi_0 = \frac{1}{\sum_{k=0}^{m-1} \dfrac{(m\rho)^k}{k!} + \dfrac{(m\rho)^m}{m!(1-\rho)}} \quad \textbf{(2)}$$

Let a random variable N denote the number of processing requests. The probability of having k processing requests is the probability of the system at the state k, i.e. $P(N = k) = \pi_k$. The expected value of N is $E(N) = k\pi_k$. Therefore,

$$E(N) = \sum_{k=1}^{m-1} \frac{\pi_0 (m\rho)^k}{(k-1)!} + \frac{\pi_0 m^m}{m!} \sum_{k=m}^{\infty} k\rho^k \quad \textbf{(3)}$$

$E(N)$ can be further simplified as [23]:

$$E(N) = m\rho + \frac{\rho\pi_m}{(1-\rho)^2} \quad \textbf{(4)}$$

Let a random variable M denote the number of busy nodes. The probability of having k busy server nodes is:

$$P(M = k) = \begin{cases} P(N = k) = \pi_k & , \quad 0 \le k \le m-1 \\ P(N \ge m) = \dfrac{\pi_m}{1-\rho}, & k = m \end{cases} \quad \textbf{(5)}$$

The probability of $P(M = m) = \frac{\pi_m}{1-\rho}$ represents the probability of the incoming requests having to enter the waiting queue. The average number of the busy server nodes in an m server cluster is:

$$E[M]=\sum_{k=0}^{m-1}k\pi_k+\frac{m\pi_m}{1-\rho}=m\rho=\frac{\lambda}{\mu} \quad \textbf{(6)}$$

The calculations and proofs of Eq. (1)-(6) can be found in many textbooks, e.g. [24].

Many studies [25-29] use the linear model to represent the relationship between the CPU utilization and the power usage of a server system. Vasan et al. [30] show that a linear model of power consumption based on CPU utilization works effectively across a variety of servers. This study adopts the linear model for the system load pattern. When a server node is waiting for a job, the CPU utilization rate is usually very close to 0%. Almeida et al. [31] found that the average CPU utilization rate is approximately 90% when a server is handling HTTP requests. Although the average CPU utilization rate may vary when a management policy is applied or another type of CPU is adopted, the CPU utilization rate of a server fluctuates slightly when processing web requests. To simplify the calculation, this study uses a constant to denote the CPU utilization rate of the periods when a server node is handling web requests.

Therefore, the following assumptions are made in this study for the cluster power modeling:

(1) L_{peak} represents the peak power load of a server node,

(2) L_{idle} represents the idle power load of a server node,

(3) L_{dyna} represents the dynamic range of power load,

(4) u_i is the CPU utilization rate of the server node i,

(5) the CPU utilization rate is 0%, when idle,

(6) the CPU utilization rate is a constant U, when busy,

(7) the system is instrumented for a sufficiently long period of time, denoted by t,

(8) there are h homogeneous server nodes,

(9) m server nodes are activated, and

(10) a workload distributor distributes incoming requests to activated but idle server nodes.

The linear power model of a server node i in the server cluster is defined as:

$$L_i(u_i)=L_{dyna}u_i+L_{idle} \quad \textbf{(7)}$$

The energy consumption of a server node i can be expressed as:

$$\Xi_i=(busytime)L_{dyna}U+tL_{idle} \quad \textbf{(8)}$$

Therefore, to estimate the energy consumption of a server node needs to estimate the expected length of the busy time of a server node. Since all nodes are identical, the ratio of the busy time to the period equals to the ratio of the number of the busy nodes to the number of the activated nodes. M denotes the number of busy nodes. $E[M]$ is the expected number of the busy nodes. The estimated energy consumption of the server node i during the period t can be obtained using the following formula:

$$\Xi_i=\frac{E[M]}{m}tL_{dyna}U+tL_{idle} \quad \textbf{(19)}$$

The energy consumption of the server cluster is the summation of all activated server nodes:

$$\Xi=\sum_{i=1}^{m}\Xi_i=E[M]tL_{dyna}U+mtL_{idle} \quad \textbf{(10)}$$

The average power load is:

$$L=\frac{\Xi}{t}=E[M]L_{dyna}U+mL_{idle} \quad \textbf{(11)}$$

Based on Eq. (6), the average power load of the system can be written as:

$$L=\frac{\lambda}{\mu}L_{dyna}U+mL_{idle} \quad \textbf{(12)}$$

It is clear that m is the major factor of the overall energy consumption, if the arrival rate λ, the service rate μ, and U were known.

In $M/M/m$ model, the service rate has to be obtained beforehand. If the average service rate of a system is μ, the expected service time of a request is the reciprocal of μ, which is $\frac{1}{\mu}$. The service time is defined as the time difference between a request arrival time and the completion time of the request, and does not include the queue waiting time [32].

The expected service time of a web server is determined by the capability of a server node or even the whole cluster. In general, the factors that affect the service rate μ of a web server may include web content (static or dynamic content, data size, access type, etc.), access time, user profile, hardware profile, and software profile (middleware, operating system, server system, applications, etc.). A web server can adopt a suitable estimation method to get the expected service time of every possible web request to the server. The probability of a specific web request being issued can be obtained through a long-term monitoring. With the expected service time and the issuing probability of every possible web request, the expected service time $\frac{1}{\mu}$ (as well as the service rate μ) of a web server can be easily calculated.

Based on Eq. (11), theoretically, letting m be $\lceil E[M] \rceil$ makes a stable-state server cluster consume the least energy. However, this solution may not result in a desirable response time. Suppose the desired response time is r. Since all server nodes have the same service rate μ. The desired response time r can be modeled as:

$$\frac{1}{\mu} < r = \frac{1}{\mu} + \varepsilon, \text{where } \varepsilon > 0 \quad \textbf{(13)}$$

Let the random variable *R* denotes the response time in the stable state of this system. According to *Little's Theorem* [24] and Eq. (4), the average response time *E[R]* is:

$$E[R] = \frac{E[N]}{\lambda} = \frac{1}{\mu} + \frac{\pi_m}{m\mu(1-\rho)^2} \quad \textbf{(14)}$$

Since the mean response time *E[R]* is the summation of the mean waiting time (queue time) and the mean service time $\frac{1}{\mu}$, the mean waiting time is $\frac{\pi_m}{m\mu(1-\rho)^2}$.

Let m' be a non-negative integer value that satisfies the following conditions:

$$\begin{cases} E[R \mid m = m'-1] > r \\ E[R \mid m = m'] \le r \end{cases} \quad \textbf{(15)}$$

m' is the minimum number of server nodes required to satisfy the desired response time.

Based on Eq. (15), the following computing algorithm, as shown in Figure 1, is developed to find m'. In the algorithm, *r* is defined as the desired response time.

1: m' Finder(λ, μ, r)

2: $\varepsilon \leftarrow r - \frac{1}{\mu}$

3: $m' \leftarrow \left\lceil \frac{\lambda}{\mu} \right\rceil$

4: **repeat**

5: $m' \leftarrow m' + 1$

6: $\rho \leftarrow \frac{\lambda}{m'\mu}$

7: $\pi_0 \leftarrow (\sum_{k=0}^{m'-1} (\frac{(m'\rho)^k}{k!} + \frac{(m'\rho)^{m'}}{m'!(1-\rho)}))^{-1}$

8: $\pi_m \leftarrow \frac{\pi_0 (m'\rho)^{m'}}{m'!}$

9: $d \leftarrow \frac{\pi_m}{m'\mu(1-\rho)^2}$

10: **until** $d < \varepsilon$

11: **return** m'

Figure 1. Algorithm for Finding m'

Due to the fluctuation nature of the web request traffic, the energy management of a server cluster is handled dynamically. The request arrival rate is also an important parameter for *M/M/m* model. Therefore, traffic forecasting is required for the dynamic provisioning of server node.

Many time-series forecasting methods [33] have been developed for Internet traffic, including: Naïve Forecasting [34], Neural Networks [35-36], ARMA [37], ARIMA [38], Support Vector Regression (SVR) [39], etc. Finding the most suitable forecasting model is another interesting research topic and beyond the scope of this study. This study just adopts the Naïve forecasting for its simplicity, and as a benchmark basis for future studies and estimations [40].

The proposed energy management solution adopts OSPM/ACPI control for the web server clusters. Switching a computer from an active state to an inactive state can be instructed by CPU. This study also uses Wake-On-LAN (WOL) to switch a node from an inactive state to an active state. WOL [41] is an Ethernet computer networking standard that allows a computer to wake from Sleep or Standby when directed by a network request, i.e. Magic Packet [42]. Energy Star Program [43] requires that computers with Ethernet capability shall offer the option of enabling WOL.

Transitioning server nodes into a low-power state during idle times is a widely used approach for improving the energy efficiency of server cluster [44]. The effectiveness of an activation policy depends largely on the latency of state transitions [45]. The power state transition from the working state to the sleep state and then back to the working state can be roughly depicted as Figure 2.

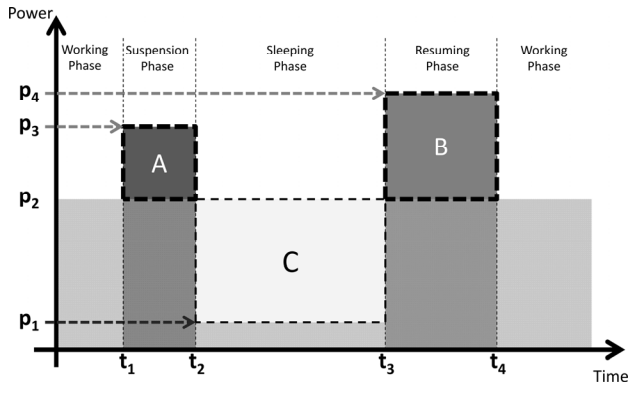

Figure 2. Working-Sleep-Working power state transition

The Working-Sleep-Working state transition can be distinguished as 4 consecutive phases that are (1) Working phase; (2) Suspension phase; (3) Sleeping phase; and (4) Resuming phase.

Let L_{susp}, L_{slep}, and L_{resu} denote the power load during the Suspension phase, Sleeping phase, and Resuming phase, respectively. Let U_{susp}, and U_{resu} denote the CPU utilization rate the Suspension phase, and Resuming phase, respectively. L_{susp}, and L_{resu} can be expressed as:

$$L_{susp} = L_{idle} + L_{dyna} U_{susp} \quad \textbf{(16)}$$

$$L_{resu} = L_{idle} + L_{dyna}U_{resu} \quad \textbf{(17)}$$

In order to grantee the power state transition results in energy saving, the area denoted by C must be larger than the summation of the area A and the area B in Figure 2. Let $L_C = L_{idle} - L_{slep}$, $L_B = L_{resu} - L_{idle}$, $L_A = L_{susp} - L_{idle}$, $t_{slep} = t_3 - t_2$, $t_{susp} = t_2 - t_1$, and $t_{resu} = t_4 - t_3$. It can be easily found that: $L_B = L_{dyna}U_{resu}$, and $L_A = L_{dyna}U_{susp}$. Let ESE (*Energy Saving Effectiveness*) denote by S, and $S = \dfrac{L_C t_{slep}}{L_B t_{resu} + L_A t_{susp}}$. Therefore,

$$S = \frac{(L_{idle} - L_{slep})t_{slep}}{L_{dyna}(U_{susp}t_{susp} + U_{resu}t_{resu})} \quad \textbf{(18)}$$

If $S > 1$, then the preformed power state control is effective. A cluster can define its expected ESE to obtain the targeted sleep time t_{slep} for a power state control by:

$$t_{slep} = \frac{SL_{dyna}(U_{susp}t_{susp} + U_{resu}t_{resu})}{L_{idle} - L_{slep}} \quad \textbf{(19)}$$

If the state control instructions are issued periodically, the control interval t_{ctrl} can be obtained by:

$$\begin{aligned} t_{ctrl} &= t_{susp} + t_{slep} \\ &= t_{susp} + \frac{SL_{dyna}(U_{susp}t_{susp} + U_{resu}t_{resu})}{L_{idle} - L_{slep}} \end{aligned} \quad \textbf{(20)}$$

The system applying the proposed control mechanism is shown in Figure 3. This system includes the following components: (1) Job Queue, (2) Workload Distributor, (3) Traffic Forecasting Unit, (4) State Controller, (5) Web Server Cluster, and (6) Server Node.

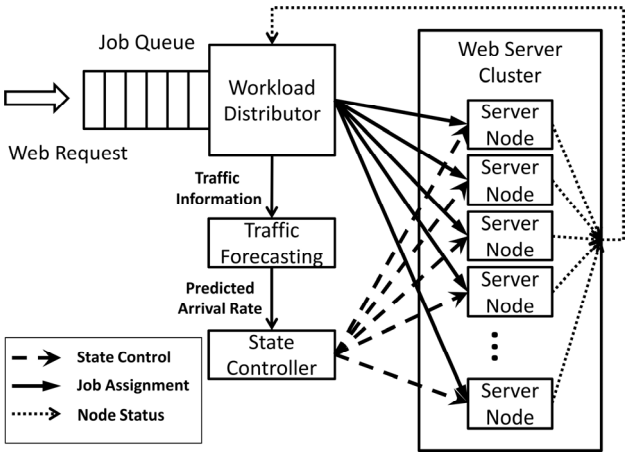

Figure 3. Block diagram of a managed web server cluster

The proposed control mechanism consists of 4 stages: (1) the node resuming stage, (2) the job arrival stage, (3) the job dispatching and processing stage, and (4) the node suspending stage. The messages exchanged among the involved components and the sequences of the related processes are shown in Figure 4.

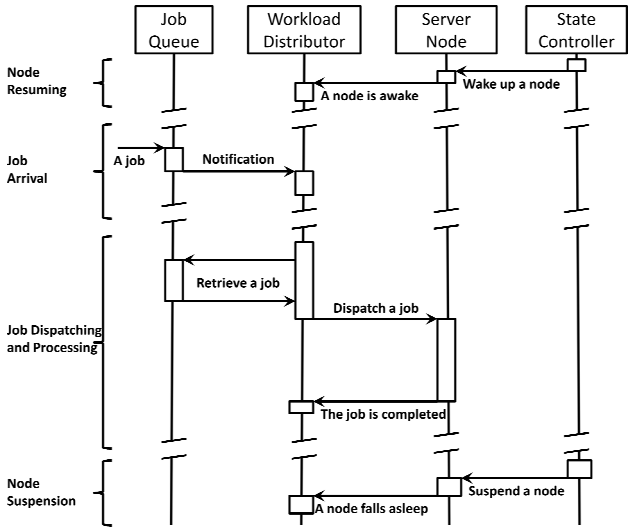

Figure 4. Sequence diagram of the proposed energy-management mechanism

The instructions of node resuming and suspension are periodically issued by the state controller. A node notifies the workload distributor when it is about to sleep, and just awake. If a sleep command is issued to a busy node, the node will complete the processing job before falling asleep. Every incoming job request is queued, and a notification is issued to the workload distributor upon the arrival of the job request. The workload distributor then dispatches each job to an available node.

The sleep states in ACPI specification are adopted. There are 5 ACPI sleep states, S1 to S5. However, this study does not consider S1 and S2 states, since they are not generally used [46].

4 Real World Case Simulation and Discussions

The real-world workload trace log is acquired from The Internet Traffic Archive [47]. The characteristics of this trace log are well documented by Arlitt and Jin [48]. This log consists of all the requests made to the 1998 World Cup Web site with 1,352,804,107 requests. This is one of few publicly available logs that contain the number of the working servers, which is required in this study.

The implemented simulator includes 7 classes of objects including web requests (*WR*), job queue (*JQ*), workload distributor (*WD*), traffic forecasting unit (*TFU*), state controller (*SC*), web server cluster (*WSC*), and server nodes (*SN*). In order to speed up the simulation process, the activated *SN*s are sorted by

their next available moments, and *WD* picks the *SN* with the earliest available moment to serve the present *WR* as shown in Figure 5.

```
1: Simulation(trace, t_ctrl, state)
2: cur ← −1
3: nxt ← −1
4: while count(trace) > 0 do
5:    WR ← retrieve( front(trace))
6:    if cur < 0 then
7:       cur ← WR.arrival − mod(WR.arrival, t_ctrl)
8:       nxt ← cur + t_ctrl
9:    else if WR.arrival ≥ nxt then
10:      while WR.arrival ≥ nxt do
11:         statstic(WSC)
12:         arrival_rate ← forecast(nxt)
13:         state _ control(WSC, arrival _ rate)
14:         cur ← nxt
15:         nxt ← cur + t_ctrl
16:      end while
17:   end if
18:   SN ← earliest _ avil(WSC)
19:   waiting_time ← get_waiting(SN, WR)
20:   service_time ← get_service(SN, WR)
21:   response_time ← get_response(SN, WR)
22:    breach ← get_breach(SN, WR)
23:   energy+ = get_energy(SN, WR)
24: end while
```

Figure 5. Simulation process

In the adopted workload trace, 99.98% of the requests are static [48]. For static content, the data size is a dominant factor of the service rate in a network application [49]. In addition to the information gathered from the workload trace, This study summarizes the specifications from [50] and the experiment results from [45] and [51] to determine the parameters for the simulated machine. The required physical parameters of the simulated server node is based on IBM BladeCenter HC10 [51].

ACPI S3, S4, and S5 states [9] have been simulated. The experiment result from [51] is adopted to define the resume and suspension latencies. Based on the responsiveness classes offered by [52], the desired response time is set to 1 second. All simulations record the breach rate (denoted as b), which is the ratio of responses exceeding the desired response time. The complete simulation parameters are listed in Figure 6.

$m = 33$, Number of installed Servers
$U = 90\%$, Average CPU rate of a busy node
$\mu = 87.88\ req/\sec$, Service rate
$L_{peak} = 105\ Watt$, Peak power load
$L_{idle} = 45\ Watt$, Idle power load
$L_{slep} = 8.4, 7.35, 4.2 Watt$, Power Load on S3, S4, S5
$t_{susp} = 3, 20, 40 \sec$, Suspension latencies of S3, S4,S5
$t_{resu} = 18, 60, 65 \sec$, Resuming latencies of S3, S4, S5
$U_{susp} = 3\%, 20\%, 25\%$, CPU % when suspending to S3,S4,S5
$U_{resu} = 5\%, 35\%, 50\%$, CPU % when resuming from S3,S4,S5
$r = 1 \sec$, Desired response time

Figure 6. Simulation parameters

In addition to the simulation run with the settings from the workload traces, there are two sets of simulation runs that are (1) ESE-oriented simulations, in which t_{ctrl} are calculated from Eq. (20), and (2) t_{ctrl}-oriented simulations. ESE-oriented simulations include 24 simulation runs that are conducted with smaller ESEs, i.e. $S = 1, 2, 3..., 24$ (denoted as *S-ESE*), and 8 simulation runs that are with larger ESEs, i.e. $S = 25, 50, 75..., 200$ (denoted as *L-ESE*). t_{ctrl}-oriented simulations include 20 simulation runs that are conducted with t_{ctrl} set from 30 seconds to 600 seconds stepping by 30 seconds. The primary difference between the ACPI controlled results and the original data is the average number of the activated server nodes, as shown in Figure 7.

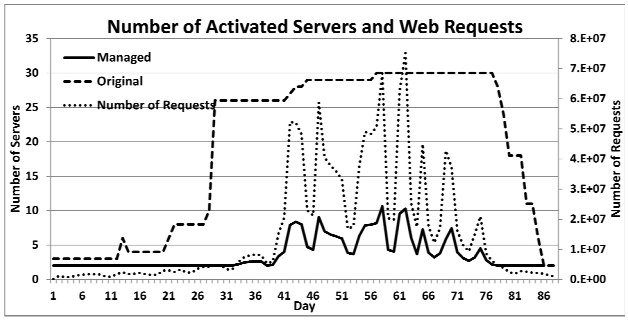

Figure 7. Daily average number of activated servers

Table 1 shows the summerized result of ESE-oriented simulation runs. From the result, the proposed method can save approximately 36%, 47%, and 62% of originally consumed energy by switching the unused nodes to ACPI S3, S4, and S5 state, respectively.

As shown in Table 1, R tends to go up as the ESE increases. When all servers are busy, a web request has to stay in the service queue waiting for an available serve. Sometimes, this situation can only be resolved by increasing m'. As ESE increases, t_{ctrl} and the waiting time both increase, which leads to the increment of R. Such phenomena can be clearly observed in all L-ESE tests. Therefore, R has a direct

correlation with t_{ctrl}. Figure 8, Figure 9, and Figure 10 show the results of t_{ctrl}-oriented simulations.

Table 1. Result of ESE-oriented simulations

		m'	R	b	Ξ
Original		18.7	0.25	0.3%	2017
$S3$	S – ESE	3.63	0.47	6.5%	1289
	L – ESE	3.63	0.57	6.4%	1289
$S4$	S – ESE	3.63	1.13	7.7%	1074
	L – ESE	3.63	271	29.1%	1074
$S5$	S – ESE	3.63	2.93	9.4%	768
	L – ESE	3.63	687	35.8%	768

m' : Number of Activated Server Nodes

R:Response Time (seconds)

b : Breach Rate

Ξ : Energy Consumption(kWh)

From Figure 8 and Figure 9, it is obvious that t_{ctrl} is the dominant factor for the average response time and the breach rate. The sleep state decides the level of energy consumption as shown in Figure 10. Using the feature scaling to normalize the simulation results, a t_{ctrl} having a shorter response time and a lower breach rate while consuming less energy can be found, which is approximately between 120 seconds and 150 seconds.

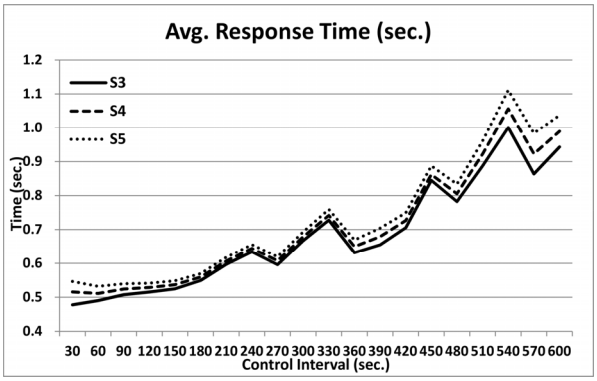

Figure 8. Average response time of t_{ctrl}-oriented simulations

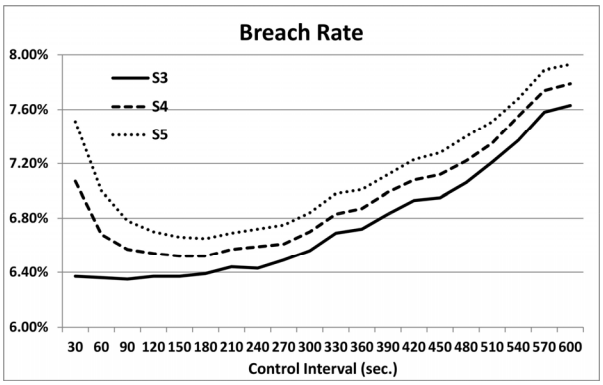

Figure 9. Breach rate of t_{ctrl}-oriented simulations

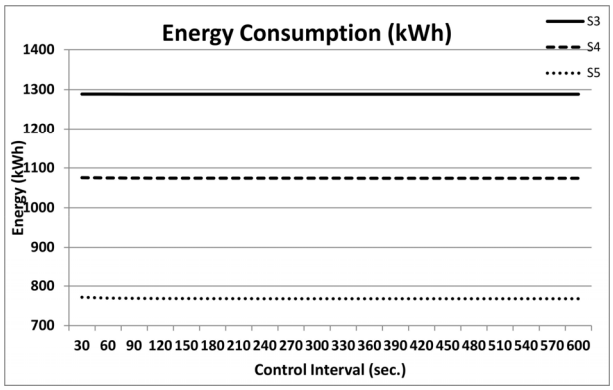

Figure 10. Energy consumption of t_{ctrl}-oriented simulations

The impacts of ESE drastically degrad as ESE increases. As ESE increases, the dominant factor is transferred from the ESE to t_{ctrl}, which is shown in Figure 8 and Figure 9. Surprisingly, very little performance-energy trade-off is observed by adopting different sleep states. In other words, the performance enhancement of adopting S3 rather than S4 or S5 is limited. This finding implies that a server cluster system may only need just one sleep state.

5 Conclusion

The conducted simulations periodically resume or suspend a number of server nodes. The number of the server nodes is acquired by the algorithm shown in Figure 1. For an effective management, a cluster is usually grouped by a number of homogeneous servers while servers may be different from one cluster to another [53]. This study aims at the cluster level management, where *M/M/m* model can be reasonably applied and is well-suited. The proposed approach achieves:

(1) adjusting the number of servers to adapt the fluctuation of request flow, as shown in Figure 7,

(2) calculating the number of servers to satisfy the desired response time, as shown in Figure 1,

(3) reducing a considerable amount of energy consumption, and

(4) adopting different ACPI global state controls to manage energy consumption.

The simulation results show that this approach can save a significant amount of energy. The developed mechanism helps the adopted cluster consuming only approximately 64%, 53%, and 38% of original energy usage by switching the unused nodes to the S3, S4, and S5 state, respectively. The amount of energy consumption is mainly decided by the number of the working nodes. Although this approach takes a longer average response time, the simulated system is able to satisfy the desired response time with an adequate control interval.

Acknowledgement

This study is funded by the Ministry of Science and Technology of the Republic of China (Taiwan) under grant NSC 101-2632-E-036-001-MY3 for the project "A Study of Applications and Examinations on the Smart Meter Enabled Electricity Grid".

References

[1] Miniwatts Marketing Group, *Internet world stats: Usage and population statistics*, http://www.internetworldstats.com.

[2] G. Varsamopoulos, Z. Abbasi, S. Gupta, Trends and Effects of Energy Proportionality on Server Provisioning in Data Centers, *Proc. of the International Conference on High Performance Computing*, Goa, India, 2010, pp. 1-11.

[3] G. Koole, A. Mandelbaum, Queueing Models of Call Centers: An Introduction, *Annals of Operations Research*, Vol. 113, No. 1-4, pp. 41-59, July, 2002.

[4] E. Brockmeyer, H. L. Halstrøm, A. Jensen, *The Life and Works of A. K. Erlang*, Danish Academy of Technical Sciences, 1948.

[5] L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, L. Zhao, Statistical Analysis of a Telephone Call Center: A Queueing-science Perspective, *Journal of the American Statistical Association*, Vol. 100, No. 469, pp. 36-50, March, 2005.

[6] N. Gans, G. Koole, A. Mandelbaum, Telephone Call Centers: Tutorial, Review, and Research Prospects, *Manufacturing & Service Operations Management*, Vol. 5, No. 2, pp. 79-141, April, 2003.

[7] M. Yu, J. Gong, J. Tang, H. Zhu, The Method of Staffing a Call Center with Delay Information Considering the Customers' Behavior, *25th Chinese Control and Decision Conference*, Guiyang, China, 2013, pp. 4723-4727.

[8] Intel Corporation & Microsoft Corporation, *Advanced Power Management BIOS Interface Specification Revision 1.2*, APM Std, 1996.

[9] Hewlett-Packard, Intel, Microsoft, Phoenix, Toshiba Corporation, *Advanced Configuration and Power Interface Specification, Rev. 5*, ACPI Std, 2011.

[10] W. E. Smith, K. S. Trivedi, L. A. Tomek, J. Ackaret, Availability Analysis of Blade Server Systems, *IBM Systems Journal*, Vol. 47, No. 4, pp. 621-640, October, 2008.

[11] C. D. Strobel, American Recovery and Reinvestment Act of 2009, *Journal of Corporate Accounting & Finance*, Vol. 20, No. 5, pp. 83-85, July/August, 2009.

[12] European Commission and Others, *FP7 in Brief-How to Get Involved in the EU 7th Framework Program for Research*, Official Publications of the European Communities, 2007.

[13] S. Low, K. Tang, *Power Minimization Techniques for Networked Data Centers*, California Institute of Technology, 2011.

[14] S. Fortune, *Energy Efficiency of Data Networks through Rate Adaptation*, U.S. Department of Energy, 2011.

[15] C. Pfeiffer, E. Kulali, *Recovery Act: Data Center Transfer from "Always On" to "Always Available" to Reduce Power*, U.S. Department of Energy, 2012.

[16] S. Klingert, R. Basmadjian, C. Dupont, A. Somov, V. Georgiadou, M. Di Girolamo, *FIT4Green Reader's Digest Technical Aspects*, 2012, Federated IT for a Sustainable Environment Impact FP7-ICT-2009-4.

[17] B. Pernici, C. Cappiello, M. G. Fugini, P. Plebani, M. Vitali, I. Salomie, T. Cioara, I. Anghel, E. Henis, R. Kat, D. Chen, G. Goldberg, M. vor dem Berge, W. Christmann, A. Kipp, T. Jiang, J. Liu, M. Bertoncini, D. Arnone, A. Rossi, Setting Energy Efficiency Goals in Data Centers: The GAMES Approach, *Energy Efficient Data Centers, Lecture Notes in Computer Science*, Vol. 7396, pp. 1-12, May, 2012.

[18] R. Basmadjian, F. Niedermeier, A. Fischer, T. Ortmeier, S. Dambeck, T. Skolnik-Korff, G. Lovász, G. Giuliani, S. Klingert, M. Kessel, *All4Green: Final Description of the Energy Provider/Data Centre Subecosystem Components*, 2013, FP7 Project No. 288674.

[19] E. Volk, D. Rathgeb, A. Oleksiak, CoolEmAll - Optimising Cooling Efficiency in Data Centres, *Computer Science - Research and Development*, Vol. 29, No. 3-4, pp. 253-261, August, 2013.

[20] J.-L. Yen, M.-J. Yang, C.-T. Yang, Power-Saving Management for Energy-Efficient Green Clouds, *Journal of Internet Technology*, Vol. 15, No. 3, pp. 381-390, May, 2014.

[21] F. D. Rossi, M. Storch, I. de Oliveira, C. A. F. De Rose, Modeling Power Consumption for DVFS Policies, *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, Lisbon, Portugal, 2015, pp. 1879-1882.

[22] W. Wang, J. Luo, A. Song, F. Dong, Energy-Aware Dynamic Server Provisioning and Frequency Adjustment in Multi-Tier Data Centers, *Journal of Internet Technology*, Vol. 14, No. 4, pp. 609-618, July, 2013.

[23] H. C. Tijms, *A First Course in Stochastic Models*, John Wiley & Sons, 2003.

[24] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, Prentice-Hall, 2001.

[25] C.-H. Lien, Y.-W. Bai, M.-B. Lin, Estimation by Software for the Power Consumption of Streaming-Media Servers, *IEEE Transactions on Instrumentation and Measurement*, Vol. 56, No. 5, pp. 1859-1870, October, 2007.

[26] L. A. Barroso, U. Hölzle, The Case for Energy-Proportional Computing, *Computer*, Vol. 40, No. 12, pp. 33-37, December, 2007.

[27] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, F. Zhao, Energy Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services, *Proc. 5th USENIX Symposium on Networked Systems Design and Implementation*, Seattle, WA, 2008, pp. 337-350.

[28] D. Economou, S. Rivoire, C. Kozyrakis, P. Ranganathan, Full System Power Analysis and Modeling for Server Environments, *Proc. Workshop on Modeling, Benchmarking, and Simulation*, Boston, MA, 2006, pp. 70-77.

[29] M. Elnozahy, M. Kistler, R. Rajamony, Energy-Efficient Server Clusters, *Lecture Notes in Computer Science*, Vol.

2325, G. Goos, J. Hartmanis, J. van Leeuwen, eds., 2003, pp. 179-197.

[30] A. Vasan, A. Sivasubramaniam, V. Shimpi, T. Sivabalan, R. Subbiah, Worth Their Watts? - An Empirical Study of Datacenter Servers, *IEEE High Performance Computer Architecture*, Chennai, India, 2010, pp. 1-10.

[31] J. M. Almeida, V. Almeida, D. J. Yates, Measuring the Behavior of a World-Wide Web Server, *Proc. TC6 Seventh Informational of the IFIP Conference on High Performation Networking*, White Plains, NY, 1997, pp. 57-72.

[32] D. A. Menasce, Web Server Software Architectures, *IEEE Internet Computing*, Vol. 7, No. 6, pp. 78-81, November/December, 2003.

[33] S. Basu, A. Mukherjee, S. Klivansky, Time Series Models for Internet Traffic, *Proceedings of IEEE International Symposium on Industrial Electronics*, Vol. 2, pp. 611-620, June, 1996.

[34] W. A. Sherden, *The Fortune Sellers: The Big Business of Selling and Buying Predictions*, John Wiley, 1998.

[35] P. Cortez, M. Rio, M. Rocha, P. Sousa, Internet Traffic Forecasting Using Neural Networks, *Proc. of the International Joint Conference on Neural Network*, Vancouver, Canada, 2006, pp. 2635-2642.

[36] C. Wang, X. Zhang, H. Yan, L. Zheng, An Internet Traffic Forecasting Model Adopting Radical Based on Function Neural Network Optimized by Genetic Algorithm, *Proc. of IEEE Workshopon Knowledge Discovery and Data Mining*, Adelaide, Australia, 2008, pp. 367-370.

[37] P. K. Hoong, I. K. Tan, C. Y. Keong, Bittorrent Network Traffic Forecasting with ARMA, *International Journal of Computer Networks & Communications*, Vol. 4, No. 4, pp. 143-156, July, 2012.

[38] X. Li, S. A. Reza Zekavat, *Traffic Pattern Prediction Based Spectrum Sharing for Cognitive Radios, Cognitive Radio Systems*, INTECH Open Access Publisher, 2009.

[39] G.-F. Fan, S. Qing, H. Wang, W.-C. Hong, H.-J. Li, Support Vector Regression Model Based on Empirical Mode Decomposition and Auto Regression for Electric Load Forecasting, *Energies*, Vol. 6, No. 4, pp. 1887-1901, April, 2013.

[40] P. Andres, M. Spiwoks, Forecast Quality Matrix: A Methodological Survey of Judging Forecast Quality of Capital Market Forecasts, *Journal of Economics and Statistics*, Vol. 219, No. 5-6, pp. 513-542, December, 2002.

[41] Intel, *Energy Star System Implementation*, 2007, Intel Corp. No:316478-001.

[42] AMD, *Magic Packet Technology*, 1995, AMD Corp. No. 20213Rev:A.

[43] ENERGY STAR Program, *Energy Star Program Requirements for Computers, Eligibility Criteria Version 5.2*, US Environmental Protection Agency, 2010.

[44] A. Gandhi, M. Harchol-Balter, M. A. Kozuch, Are Sleep States Effective in Data Centers? *Green Computing Conference*, San Jose, CA, 2012, pp. 1-10.

[45] S. L. Xi, M. Guevara, J. Nelson, P. Pensabene, B. C. Lee, Understanding the Critical Path in Power State Transition Latencies, *IEEE Low Power Electronics and Design*, Beijing, China, 2013, pp. 317-322.

[46] C. Gough, I. Steiner, W. Saunders, *Platform Power Management, Energy Efficient Servers*, Springer, 2015.

[47] The Internet Traffic Archive, *Traces Available in the Internet Traffic Archive*, 2008.

[48] M. Arlitt, T. Jin, A Workload Characterization Study of the 1998 World Cup Web Site, *IEEE Network*, Vol. 14, No. 3, pp. 30-37, May, 2000.

[49] D. Meliksetian, F. F.-K. Yu, C.-Y. R. Chen, Methodologies for Designing Video Servers, *IEEE Transactions on Multimedia*, Vol. 2, No. 1, pp. 62-69, March, 2000.

[50] Standard Performance Evaluation Corporation, *SPEC's Benchmarks and Published Results*, 2013.

[51] C. Isci, S. McIntosh, J. Kephart, R. Das, J. Hanson, S. Piper, R. Wolford, T. Brey, R. Kantner, A. Ng, J. Norris, A. Traore, M. Frissora, Agile, Efficient Virtualization Power Management with Low-Latency Server Power States, *ACM SIGARCH Computer Architecture News*, Vol. 41, No. 3, pp. 96-107, June, 2013.

[52] S. C. Seow, *Designing and Engineering Time: The Psychology of Time Perception in Software*, Addison-Wesley Professional, 2008.

[53] H. Qian, F. Li, D. Medhi, On Energy-Aware Aggregation of Dynamic Temporal Demand in Cloud Computing, *4th International Conference on Communication Systems and Networks*, Bangalore, India, 2012, pp. 1-6.

Biographies

Cheng-Jen Tang received his B.Sc. ('88) degree in Computer Engineering from National Chiao-Tung University, Taiwan, and M.Sc. ('97) and Ph.D. ('01) degrees in Electrical and Computer Engineering from Syracuse University, USA. Currently, he is with Tatung University. Energy efficiency and Smart grid related issues are his research interests.

Miau-Ru Dai is currently with Delta Network Inc. She received her B.S.('06), M.S.('07), and Ph.D.('14) degrees in Electrical Engineering and Communication Engineering, all from Tatung University, Taipei, Taiwan. Her research interests include power system analysis, energy efficiency, and computer networks.