Power-Saving Design in Server Farms for Multi-Tier Applications under Response Time Constraint

Shengquan Wang[1], Waqaas Munawar[2], Xue Liu[3], and Jian-Jia Chen[2]
[1]*Department of Computer and Information Science, Univ. of Michigan-Dearborn, Dearborn, USA*
[2]*Department of Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany*
[3]*School of Computer Science, McGill University, Montreal, Canada*
shqwang@umd.umich.edu, munawar@kit.edu, xueliu@cs.mcgill.ca, j.chen@kit.edu

Keywords: power saving, server farm, multi-tier, response time, Service Level Agreement (SLA), M/G/1/PS, Mean Value Analysis (MVA).

Abstract: Server farms suffer from an increasing power consumption nowadays. Power saving has become a prominent design issue in server farms. This paper presents a power-saving design in server farms under the constraint of the response time. In particular, we target on multi-tier applications, which are very typical on the web in modern days. We propose an efficient power-saving design strategy, called *PowerTier.* This strategy exploits two major techniques by using Dynamic Power management (DPM) to activate/deactivate servers and using Dynamic Voltage Scaling (DVS) to adjust the processor speed for each activated server. In addition, *PowerTier* considers two different application models: the open-queueing model and the closed-queueing model for session-less and session-based web applications respectively. With *PowerTier,* we are able to choose the number of activated servers at each tier and the processor speed for each server to minimize the overall power consumption in server farms while meeting a given mean response time guarantee for multi-tier applications. Our comprehensive simulation confirms the effectiveness and efficiency of *PowerTier.*

1 INTRODUCTION

Power has become one of the most dominant operating cost in server systems. By 2011, data centers in U.S. are expected to consume around 100 billion kW per year (U.S. Environmental Protection Agency (EPA), 2007), in which the annual power cost is around 7.4 billion US$. Moreover, given the large number of servers in use today, the worldwide expenditure on enterprise power and cooling of these servers is estimated to be in excess of 30 billion US$ (Raghavendra et al., 2008). At the same time, guaranteeing the performance-oriented Service Level Agreement (SLA) signed with the clients is critical to clients' satisfaction for online business. A common SLA is defined as (the mean value of) the response time constraint, and a delayed response to clients will have negative effects on online business including client frustrations and revenue loss. In order to meet a satisfying SLA for an increasing service demand from clients, server farm becomes a common practice in industry. A server farm could be composed of a cluster of tens to thousands of servers to provide large computing capability. However, the power consumption in server farms is tremendous.

Recently, an increasing attention has been paid on how to reduce the power consumption while maintaining a given SLA. In (Rusu et al., 2006), a timing-aware power management scheme was proposed that combines cluster-wide, server on/off scheme and local power management techniques in heterogeneous clusters. A new threshold-based approach was presented in (Wang and Lu, 2008) for efficient power management of heterogeneous soft real-time clusters in making three important design decisions on ordered server list, server activation thresholds and workload distribution. In (Guerra et al., 2008), a queueing theoretical technique was proposed to balance energy consumption and adequate application response times in heterogeneous CPU-intensive server clusters. In (Bohrer et al., 2002; Sharma et al., 2003), low-power opportunities for web servers has been utilized to reduce the energy consumption by applying Dynamic Voltage Scaling (DVS) with minimal impact on the server performance. A queueing model was used in (Gandhi et al., 2009) to predict the optimal power allocation in a variety of scenarios with DVS and Dynamic Power Management (DPM)

by activation/deactivation of servers in both closed-queueing and open-queueing models. In (Wierman et al., 2009), an optimal speed scaling was proposed to balance the mean energy consumption and mean response time under Processor Sharing (PS) scheduling.

All of these work focused on simple single-tier applications. As we know, modern web applications are usually using multi-tier architecture (Kamra et al., 2004; Liu et al., 2006; Liu et al., 2008; Urgaonkar et al., 2005; Pacifici et al., 2005; Diao et al., 2006; Liu et al., 2005; Wang et al., 2010). Each tier provides a certain specific functionality to applications. A client request will pass through a series of tiers to attain a complete service. For instance, a typical e-commerce application consists of three tiers: web server tier, application server tier, and database server tier. The multi-tier architecture follows *layered queueing models* (Rolia and Sevcik, 1995) and typically has a cross-tier dependency. The service at a tier is normally blocked while waiting for the service from its succeeding tier. Such cross-tier dependency makes the response time analysis challenging in comparison with the single-tier architecture. In (Liu et al., 2005), an analytical model was proposed for 3-tiered web service architecture. The concurrency limit was addressed in (Urgaonkar et al., 2005) for multi-tier applications. In (Pacifici et al., 2005), an architecture and underlying model of a performance management system was presented for multi-tier web applications on server clusters. In (Diao et al., 2006), a hybrid performance model for differentiated services was presented for multi-tier applications with cross-tier interaction. Among these work, only in (Diao et al., 2006) the cross-tier dependency was considered, and the rest applied a tandem-queue-like structure and ignored the cross-tier dependency. In (Wang et al., 2010), an oversimplified M/M/1 model is used to perform the queueing analysis in multi-tier architecture.

In this paper, we aim to conduct a comprehensive study on power saving in server farms for multi-tier applications requiring that a given SLA should be met. We adopt two techniques as used in server farms for power-aware design: the DVS technique with variable speeds and the DPM technique with activation/deactivation of servers. There are two specific questions that we need to address in the system design in order to achieve this goal: (i) How many servers should be activated at each tier? (ii) What is the best processor speed (corresponding to the voltage/frequency to be used) for each server? For a single-tier architecture with homogeneous servers, it is shown in (Gandhi et al., 2009) that the optimal strategy is to set all servers with the same speed for all activated servers. However, this does not hold in the multi-tier architecture due to the cross-tier dependency. We present an efficient power-saving design strategy called *PowerTier* and study how to choose the number of activated servers at each tier and the processor speed for each server to minimize the overall power consumption in server farms while meeting a given mean response time guarantee for multi-tier applications. We consider both open-queueing and closed-queueing models for applications.

The rest of this paper is organized as follows: Section 2 shows the system model. Section 3 presents a detailed power consumption and response time analysis, which is the basis for our power-saving design. Our power-saving design scheme *PowerTier* is described in Section 4. Section 5 presents detailed performance evaluation of *PowerTier* over a various of platforms. We conclude the paper in Section 6.

2 SYSTEM MODEL

In this section, we define the system model, including the power consumption model for servers, the multi-tier architecture, and the client application model.

2.1 Power Consumption Model

We assume that all servers are equipped with the DVS and DPM techniques for the power management. When the server is deactivated by the DPM technique, its power consumption is negligible. So, here we focus on the power consumption when the server is activated.

With the DVS technique, we can choose a processor speed for a server (with a corresponding choice of the supply voltage). We define r as the ratio of the processor speed of the server to its maximum speed. The speed ratio r is normally bounded by a lower bound r_l. Then we have $r_l \leq r \leq 1$. When the server is activated, either it is (i) in the *idle* mode at the lowest speed ratio r_l without executing any job; or (ii) in the *running* mode executing jobs with a processor speed ratio r. The power consumption in our study is the system-level power, including the power consumed by the processor and all other components within the server such as memory and I/O devices. The power consumption depends on the mode that the server is in (idle or running) and the processor speed in use as well. In this paper, we adopt the power consumption model in (Gandhi et al., 2009). A server has the following power modes:

- *Idle power mode:* In the idle mode, the server consumes the static power P_I;
- *Running power mode:* In the running mode, the power consumption $P_R(r)$ by the server at a speed ratio r is

$$P_R(r) = \alpha[r - r_l]^\gamma + P_I, \quad (1)$$

where $\gamma \geq 1$. The cubic rule is widely suggested in the literature for the processor power-to-speed relationship in the running mode, i.e., $\gamma = 3$. However, in server farms with DVS or for some applications, the linear rule could be applied (Gandhi et al., 2009).

The different power modes provide the space for system designers to design efficient power-saving strategies.

2.2 Multi-Tier Architecture

We consider a system with M tiers, each of which consists of a server farm. We assume that all tiers run on homogeneous servers. Tier m has v_m activated servers and each runs at a speed ratio r_m. The homogeneous sever assumption was used in previous similar studies (Gandhi et al., 2009). [1] We assume a processor sharing (PS) scheduling at each server, since it approximates well the scheduling algorithms used by most commodity operating systems such as Linux.

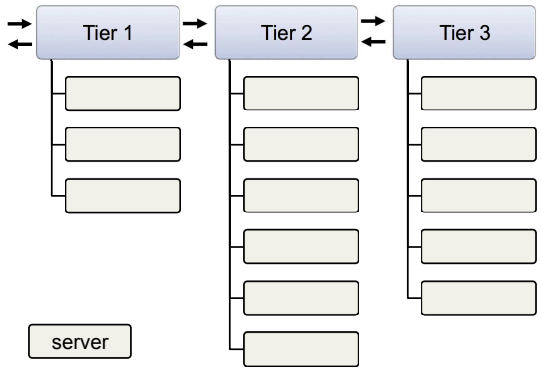


Figure 1: A three-tier architecture with 3 servers at Tiers 1, 7 servers at Tier 2, and 6 servers at Tier 3.

Figure 1 illustrates a three-tier architecture. A client starts a request at the outer tier, denoted as Tier 1, and then goes to an inner Tier m ($m \geq 2$) one by one if necessary. When a request arrives at Tier m, it triggers one or more requests at its succeeding Tier $m+1$. After some processing at Tier m, it either returns to Tier $m-1$ or proceeds to Tier $m+1$. The exceptions are the last Tier M, where all requests return to the Tier $M-1$, and the first Tier 1, where returning to the preceding queue means request completion. This model can handle multiple visits to a tier including sequential and parallel accesses (Diao et al., 2006). We denote κ_{m+1} as the average request visit ratio at Tier $m+1$ by a request at Tier m. If we define λ_m as the request arrival rate at Tier m [2], then we have

$$\kappa_{m+1} = \frac{\lambda_{m+1}}{\lambda_m}. \quad (2)$$

The system supports many clients. At each tier, a dispatcher (or load balancer) as shown in Figure 2 will distribute all incoming requests from clients to one of the servers in the server farm of this tier. We assume that they are evenly distributed to the servers at each tier. The multi-tier architecture shows a cross-tier dependency (Diao et al., 2006; Rolia and Sevcik, 1995). It could be illustrated with a nested structure in Figure 2, where Tier m includes the succeeding Tier $m+1$ for $m < M$. A request can only be completed at each tier after it has received service from the succeeding tier (if it needs service from the succeeding tier). We assume that the waiting for the outcome from the succeeding tier is non-blocking, i.e., the waiting process will not block other processes in the same server from using the resources such as CPU during its waiting.

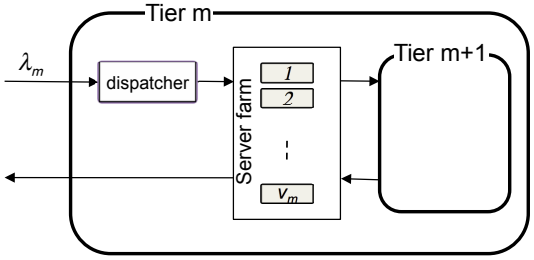


Figure 2: Cross-tier dependency.

2.3 Client Application Model

Applications for clients could be session-less or session-based. Each session-less application issues one request during its life while a session-based application usually issues more requests during its lifetime with *think times* in between and normally lasts for a while. The former can be modeled as an *open-queueing* system and the latter as a *closed-queuing* system (Jain, 1991). In an open-queueing system, applications start and end even though we could assume a fixed average arrival rate, but in a closed-queueing system, applications stay and the number of the ses-

[1] We can extend it to the heterogeneous case by taking into consideration the different characteristics of servers.

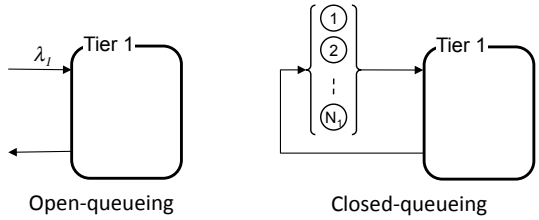


Figure 3: Open-queueing and closed-queueing models at Tier 1.

sions remains the same. Figure 3 shows open/closed-queueing models at Tier 1. For the closed-queueing model, we define N_1 delay servers, each of which corresponds to the think time for each session at Tier 1 (Liu et al., 2005).

In the multi-tier architecture, the system could be decoupled into subsystems by tiers. In either open-queueing or closed-queueing model, at Tier m ($m \geq 2$), none, one, or more requests could be generated with ratio κ_m after the proceeding tier. In the study of server performance, M/G/1/PS server model has been shown by different research studies that can model the open-queueing servers well (Kamra et al., 2004; Wang and Lu, 2008; Gandhi et al., 2009; Wierman et al., 2009; Liu et al., 2008; Liu et al., 2006; Heo et al., 2007). Mean Value Analysis (MVA) is widely used in closed-queueing servers (Lazowska et al., 1984; Jain, 1991; Reiser and Lavenberg, 1980) for response time analysis.

3 POWER CONSUMPTION AND RESPONSE TIME ANALYSIS

Recall that our objective is to minimize the power consumption under the mean response time constraint. First we need to conduct the power consumption and response time analysis. We start it with single server, then we extend it to server farm in a multi-tier architecture.

3.1 Single Server

Power Consumption We consider a server that will switch in two power modes alternatively: running and idle. We define π_R and π_I as the probabilities that the server is in running and idle mode respectively, where $\pi_R + \pi_I = 1$, then the mean power consumption can be written as:

$$\mathbb{E}[P] = P_R(r)\pi_R + P_I\pi_I = \alpha[r - r_l]^\gamma \pi_R + P_I, \quad (3)$$

where $P_R(r)$ is defined in (1). According to (3), in order to calculate the mean power, we need to obtain the value of π_R:

- For an open-queueing M/G/1/PS server, if requests are with an arrival rate λ, and a generalized service time distribution with a given mean value $\mathbb{E}[S]$, then by the traditional queueing theory (Kleinrock, 1976) we have

$$\pi_R = \lambda \mathbb{E}[S]. \quad (4)$$

- For a closed-queueing server, we could use the same formula in (4) to calculate π_R. However λ in (4) should be throughput instead [3]. Assume that there are N fixed number of sessions, jobs are with a mean response time $\mathbb{E}[R]$, and a mean think time $\mathbb{E}[Z]$ in between, then by (Jain, 1991), the throughput λ can be obtained as

$$\lambda = \frac{N}{\mathbb{E}[R] + \mathbb{E}[Z]}, \quad (5)$$

where $\mathbb{E}[R]$ is to be determined.

Response Time Analysis The response time is defined as the time spent by a job in waiting in the queue and executing on the processor.

- For an open-queueing M/G/1/PS server, by the traditional queueing theory (Kleinrock, 1976), the mean response time is

$$\mathbb{E}[R] = \frac{\mathbb{E}[S]}{1 - \lambda \mathbb{E}[S]}. \quad (6)$$

- For a closed-queueing server, we use MVA to obtain the mean response time (Jain, 1991). We define $D(N)$ as the mean delay for N sessions. D can be calculated recursively in terms of N. If we denote $Q(N)$ as the mean queue length with N sessions, then with MVA we have

$$D(N) = [Q(N-1) + 1]\mathbb{E}[S], \quad (7)$$

$$Q(N) = \frac{N}{\mathbb{E}[R] + \mathbb{E}[Z]} D(N), \quad (8)$$

where $Q(0) = 0$. To reduce the computation complexity, we could use the well-accepted Schweitzer's approximation (Jain, 1991) by approximating $Q(N-1) \approx \frac{N-1}{N} Q(N)$ to avoid the recursive computation. Then $D(N)$ will be the positive solution to the following equation:

$$D(N) = \left[\frac{[N-1]D(N)}{\mathbb{E}[R] + \mathbb{E}[Z]} + 1 \right] \mathbb{E}[S], \quad (9)$$

where the response time $\mathbb{E}[R] = D(N)$ for single tier.

[2] In the closed-queueing model introduced later, λ_1 is defined as throughput since the number of sessions is fixed.

[3] The throughput is equivalent to the arrival rate in open-queueing model and so we use the same notation λ.

The analysis in single server regarding power consumption and response time will be the basis for server farms in a multi-tier architecture.

3.2 Server Farm in Multi-Tier Architecture

In the following, we consider the multi-tier architecture with each tier having multiple servers. We add a subscript m into the terms introduced above for any server at Tier m.

We assume that the average service demand is fixed in the system, which is the arrival rate λ_1 and the number of sessions N_1 at Tier 1 for open/closed-queueing models respectively. By (5), the throughput at Tier 1 in closed-queueing model can be written as

$$\lambda_1 = \frac{N_1}{\mathbb{E}[R_1] + \mathbb{E}[Z]}, \tag{10}$$

where $\mathbb{E}[R_1]$ is to be determined.

At Tier m ($m = 1, 2, \ldots, M$), given the visit ratio κ_m and the number v_m of the servers, the request arrival rate can be obtained as

$$\lambda_m = \lambda_1 [\prod_{i=1}^{m} \kappa_i]. \tag{11}$$

Recall that Tier m has v_m homogeneous servers. Since the arrival is evenly distributed to each server at each tier, the arrival rate at any server at Tier m is $\frac{\lambda_m}{v_m}$.

Power Consumption In order to calculate the mean power for servers at each tier, we need to find $\pi_{R,m}$, i.e., the probability that a server at Tier m is in the running mode. We assume that the server is in the idle power mode when it is waiting for the outcome from the succeeding tier. For either an open-queueing or closed-queueing server at Tier m, we have

$$\pi_{R,m} = \frac{\lambda_m}{v_m} \mathbb{E}[S_m], \tag{12}$$

where λ_m is defined in (11). The value of λ_1 in (11) is given for an open-queueing server and defined by (10) for a closed-queueing server respectively. Then applying (12) into (3), we can obtain the mean power consumption.

Response Time Analysis In multi-tier architecture, the response time analysis is more complex due to the cross-tier dependency. The response time at each tier also includes the waiting time for the outcome from the succeeding tier, which is $\kappa_{m+1}\mathbb{E}[R_{m+1}]$ at Tier m (Diao et al., 2006). Recall that the waiting for the outcome from the succeeding tier is non-blocking. Then applying this observation into (6) and (7), we have the following results:

- For an open-queueing M/G/1/PS server at Tier m, the response time is

$$\mathbb{E}[R_m] = \frac{\mathbb{E}[S_m]}{1 - \frac{\lambda_m}{v_m}\mathbb{E}[S_m]} + \kappa_{m+1}\mathbb{E}[R_{m+1}], \tag{13}$$

  where λ_m is defined in (11).

- For a closed-queueing server at Tier m, we denote D_m as the mean delay experienced by any server at Tier m. The hit ratio for any session at any server at Tier m is $[\prod_{j=1}^{m} \kappa_j]\frac{v_m}{v_1}$ (Urgaonkar et al., 2005). Then with the approximated MVA in (9), the response time for any request at Tier m is

$$\mathbb{E}[R_m] = D_m + \kappa_{m+1}\mathbb{E}[R_{m+1}], \tag{14}$$

  which satisfies

$$D_m = \left[\frac{[N_1 - 1][\prod_{j=1}^{m} \kappa_j]\frac{v_m}{v_1} D_m}{\mathbb{E}[R_1] + \mathbb{E}[Z]} + 1 \right] \mathbb{E}[S_m]. \tag{15}$$

We assume the mean service time for a server at Tier m is $\mathbb{E}[S_m] = \frac{1}{\mu_m}$ under the maximum speed. If the server runs at a speed ratio r_m in the running mode, we have $\mathbb{E}[S_m] = \frac{1}{\mu_m r_m}$. Applying this into the above results, we have the complete analysis of power consumption and response time for server farms in multi-tier architecture. It is summarized in the following theorem:

Theorem 1. *We consider both open/closed-queueing models for server farms in multi-tier architecture. For either an open-queueing or closed-queueing server at Tier m, the mean power consumption is*

$$\mathbb{E}[P_m] = \frac{\lambda_m}{v_m \mu_m} \frac{\alpha[r_m - r_l]^\gamma}{r_m} + P_I, \tag{16}$$

where λ_m is defined in (11). The mean response time of a job can be obtained as follows:

- *For an open-queueing M/G/1/PS server at Tier m, the mean response time of a job is*

$$\mathbb{E}[R_m] = \frac{1}{\mu_m r_m - \frac{\lambda_m}{v_m}} + \kappa_{m+1}\mathbb{E}[R_{m+1}], \tag{17}$$

  for $m = 1, 2, \ldots, M$ with $R_{M+1} = 0$.

- *For a closed-queueing server at Tier m, the mean response time of a job is*

$$\mathbb{E}[R_m] = D_m + \kappa_{m+1}\mathbb{E}[R_{m+1}], \tag{18}$$

  which satisfies

$$D_m = \left[\frac{[N_1 - 1][\prod_{j=1}^{m} \kappa_j]\frac{v_m}{v_1} D_m}{\mathbb{E}[R_1] + \mathbb{E}[Z]} + 1 \right] \frac{1}{\mu_m r_m}. \tag{19}$$

In the above formulas, λ_1 and N_1 are given for open/closed-queueing server respectively.

We notice that in closed-queueing model, D_m's depends on each other by (18) and (19). Such inter-dependency among D_m's generates an implicit formula for the response time analysis in closed-queueing model. In such case, we could use the classical *fixed point theorem* to solve it.

4 *POWERTIER*: AN EFFICIENT POWER-SAVING DESIGN

In this section, we will study our power-saving design strategy *PowerTier* for server farms in multi-tier architecture. The power-saving design could be divided into two phases:

- Planning or upgrading: In this phase, we could determine the number of servers (denoted as $\hat{v}_m$) needed for the peak service demand which is defined as the maximum arrival rate and the maximum number of sessions at Tier 1 for open/closed-queueing models respectively (denoted as $\hat{\lambda}_1$ and $\hat{N}_1$ respectively).

- Runtime: In this phase, the number of the available servers are fixed, which were obtained in the planning or upgrading phase. For any level of service demand, we will decide how many servers should be activated/deactivated and the processor speed for each server.

In each phase, we assume that the visit ratio κ_m's and the service rate μ_m's were learned through the monitored history, and so they are known in the design. The major difference between these two phases is that we can switch servers among different tiers in the planning or upgrading phase, but not practically in the runtime phase.

In our design, we aim to minimize the mean power consumption under a given mean response time constraint, where we choose a mean response time threshold $\hat{R}$. For given service demand λ_1 and N_1 (in open/closed-queueing model respectively), we need to determine the number of the activated servers at each tier and the processor speed of each activated server, i.e., the value of v_m and r_m for $m = 1,2,\ldots,M$. The optimization problem can be formulated as follows:

$$\min_{\{v_m,r_m:m=1,2,\ldots,M\}} \mathbb{E}[P] = \sum_{m=1}^{M} v_m \mathbb{E}[P_m] \tag{20a}$$

$$\text{subject to } \mathbb{E}[R] = \mathbb{E}[R_1] \le \hat{R}, \tag{20b}$$

$$\max\{r_l, \frac{\lambda_m}{v_m \mu_m}\} \le r_m \le 1, \tag{20c}$$

$$1 \le v_m \le \hat{v}_m. \tag{20d}$$

Inequality (20c) is based on the bound of r_m ($r_l \le r_m \le 1$) and the stability condition of a server ($r_m > \frac{\lambda_m}{v_m \mu_m}$). Inequality (20c) is the server availability constraint. Notice that we will only have the lower bound in the planning or upgrading phase. [4]

In order to solve the above optimization with nonlinear constraints, we first treat v_m as continuous values to deal with the integer programming. Since all $\mathbb{E}[P_m]$ and $\mathbb{E}[R_m]$ are convex functions in terms of r_m and v_m, the optimization problem can be solved with the Lagrangian method. We define the Lagrange equation L with the Lagrange multipliers ϕ and $\chi_{k,m}$ ($k = 1,2,3$ and $m = 0,1,\ldots,M$) as

$$L = \sum_{m=1}^{M} v_m \mathbb{E}[P_m] + \phi \mathbb{E}[R_1] + \sum_{m=1}^{M} [\chi_{1,m}[r_m - r_l][r_m - 1] - \chi_{2,m}[v_m - 1][v_m - \hat{v}_m] - \chi_{3,m} r_m v_m \mu_m], \tag{21}$$

where all multipliers are non-negative.

We set $\frac{\partial L}{\partial r_m} = 0$ and $\frac{\partial L}{\partial v_m} = 0$ for $m = 1,2,\ldots,M$. Together with equalities in (13) and the following equalities

$$\phi[\mathbb{E}[R_1] - \hat{R}] = 0, \tag{22}$$

$$\chi_{1,m}[r_m - r_l][r_m - 1] = 0, \tag{23}$$

$$\chi_{2,m}[v_m - 1][v_m - \hat{v}_m] = 0, \tag{24}$$

$$\chi_{3,m}[\lambda_m - r_m v_m \mu_m] = 0, \tag{25}$$

we can solve the values of the variables.

The key in the above equations is to solve $\frac{\partial L}{\partial r_m} = 0$ and $\frac{\partial L}{\partial v_m} = 0$ for $m = 1,2,\ldots,M$. Given the power consumption and response time analysis summarized in previous section for both open/closed-queueing models, we can obtain the formula.

We denote r_m^* and v_m^* as the obtained optimal r_m and v_m respectively. Since v_m^*'s are integer values, we could choose their rounded-up values as the final output. Then apply the rounded-up v_m^* into the above optimal approach with fixed server allocation and obtain the final solution.

We summarize the main result in the following theorem:

[4] In the planning or upgrading phase, if the number of the overall available servers are limited due to budget, we need to add this additional constraint too.

Theorem 2. *Given κ_m's, μ_m's, λ_1 (in the open-queueing model) and N_1 (in the closed-queueing model), with the PowerTier design we could find the (heuristically) optimal values r_m^* and v_m^* ($m = 1,2,\ldots,M$) such that the power consumption can be minimized as $\mathbb{E}[P^*]$ while the mean response time $\mathbb{E}[R_1]$ is below the mean response time threshold $\hat{R}$.*

With the *PowerTier* design, we determine server allocation at each tier for the planning or upgrading phase for the peak service demand. Also, for the running phase, for different service demands and the visit ratios, we also determine the number of activated/deactivated servers at each tier and the processor speed for each activated server. With the *PowerTier* design, we are able to minimize the mean power consumption under the given mean response time constraint.

5 PERFORMANCE EVALUATION

In this section, we first verify the queueing analysis in multi-tier architecture in both open/closed-queueing model as shown in Section 3. Then we present performance evaluation of our proposed *PowerTier* design.

5.1 Verifying Queueing Analysis in Multi-Tier Architecture

We employed the RUBiS (Cecchet et al., 2002) system to generate and analyze the traffic for measuring the timing parameters. The RUBiS system offers an eBay like web service with an associated client to generate the test traffic. It is a three-tier hierarchical system with a possibility of having more than one server per tier. First tier consists of Apache load balancer, the second includes the JBoss application server and the third consists of MySQL database server. We allocated one physical machine per tier. For the purpose of evaluation, we modified two aspects of RUBiS' test traffic generator: (i) it was altered to record a complete trace of the requests and wrote an additional tool to replay the saved trace. This helps negate the effects of any probabilistic fluctuations arising from differences in the traces. (ii) Secondly, we instrumented the RUBiS server side code to measure the inter tier request ratios and per tier response times. Overall, this architecture to measure the empirical results, insured accurate emulation of an actual typical multi-tier web service.

We obtained the mean service time at the maximum speed at each tier as $(\frac{1}{\mu_1}, \frac{1}{\mu_2}, \frac{1}{\mu_3}) = (1.2, 6.86, 5.43)$ ms, and the visit ratio as $(\kappa_1, \kappa_2, \kappa_3) = (1, 1, 1.24074)$, and adopted the think time 0.035 sec in the closed-queueing model, which is used in (Liu et al., 2005). The available frequencies for each server are: the Apache server – 3.0GHz and 2.8GHz; the JBoss server – $3.1, 3.0, \ldots, 1.6$GHz; the MySQL server – 2.13GHz, 1.87GHz, and 1.60GHz. We consider two kinds of configurations of frequencies at tiers: $(3.0, 3.1, 2.13)$GHz and $(2.8, 2.5, 1.6)$GHz.

Table 1: Comparison of Measured/Modeled Response Time in Open-Queueing Model.

(a) $\lambda_1 = 8.862$ and $(r_1, r_2, r_3) = (1, 1, 1)$

	$\mathbb{E}[R_1]$	$\mathbb{E}[R_2]$	$\mathbb{E}[R_3]$
Measured	16.50 ms	14.72 ms	5.83 ms
Modeled	15.93 ms	14.55 ms	5.83 ms
Error	3.6%	1.2%	0.0%

(b) $\lambda_1 = 17.723$ and $(r_1, r_2, r_3) = (1, 1, 1)$

	$\mathbb{E}[R_1]$	$\mathbb{E}[R_2]$	$\mathbb{E}[R_3]$
Measured	18.27 ms	16.49 ms	6.71 ms
Modeled	17.72 ms	16.14 ms	6.71 ms
Error	3.1%	2.2%	0.0%

(c) $\lambda_1 = 8.862$ and $(r_1, r_2, r_3) = (0.90, 0.81, 0.75)$

	$\mathbb{E}[R_1]$	$\mathbb{E}[R_2]$	$\mathbb{E}[R_3]$
Measured	18.27 ms	16.49 ms	6.71 ms
Modeled	17.71 ms	16.37 ms	6.71 ms
Error	3.2%	0.8%	0.0%

(d) $\lambda_1 = 17.723$ and $(r_1, r_2, r_3) = (0.90, 0.81, 0.75)$

	$\mathbb{E}[R_1]$	$\mathbb{E}[R_2]$	$\mathbb{E}[R_3]$
Measured	21.38 ms	19.63 ms	8.24 ms
Modeled	20.86 ms	18.89 ms	8.24 ms
Error	2.5%	4.0%	0.0%

We have run over $5,000$ requests and fixed the arrival rate $\lambda_1 = 8.862, 17.723$ per second for the open-queueing model. We measure the mean response time for each request at each tier and compare it with the modeled one. The results are shown in Table 1. In all cases, the maximum error of measured response time compared with the modeled one is 4% and most of them are pretty small. It shows that the modeling is pretty accurate.

5.2 Evaluating *PowerTier*

Servers We use three types of servers to evaluate the performance of *PowerTier*: the one as a JBoss server in the previous experiment, and the other two from (Gandhi et al., 2009). The one in the previous experiment is an Intel i3 dual core processor based server with 6GB of RAM. The server is equipped

Table 2: Server profiles.

	γ	r_l	α (Watt)	P_l (Watt)
Type-A server	1	0.4	100	180
Type-B server	1.2137	0.5161	12.3977	36.7040
Type-C server	3	0.4	455	150

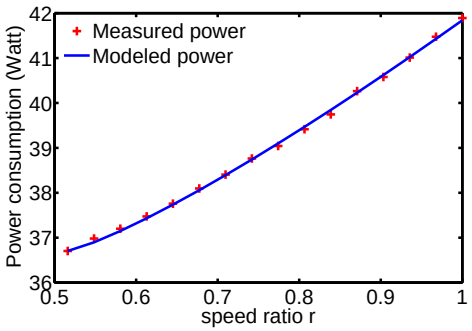


Figure 4: Measured and modeled power.

with DVFS capability with option to switch between 16 frequencies between 1.6GHz and 3.1GHz. We enable one core of each server. We measure the power consumption for the one in the previous experiment. Based on the measured power, we use *Least Squares Fitting* to obtain the modeled power as shown in Figure 4. The power profiles of all three types of servers in Table 2, where Type-B is the one in the previous experiment. We consider the same three-tier architecture used in the previous experiment. We adopt from (Liu et al., 2005) the mean service time at the maximum speed at each tier with $(\frac{1}{\mu_1}, \frac{1}{\mu_2}, \frac{1}{\mu_3}) = (1.2, 15.1, 36.7)$ ms, the visit ratio $(\kappa_1, \kappa_2, \kappa_3) = (1, 0.998, 1.603)$, and the think time with 0.035 sec in the closed-queueing model.

We scale the job service time for all three type servers so that the mean service time at the maximum speed at each tier follows the above specification.

Baseline Design Strategies We consider two baseline design strategies: (i) *EvenDist*: All servers are evenly assigned to 3 tiers. All tiers will almost have the same number of servers with at most 1-server difference. (ii) *PropDist*: All servers are proportionally assigned to 3 tiers according to the absolute utilization of the tier, i.e., $\frac{\lambda_m}{\mu_m}$. In each baseline, all servers are always activated with the maximum processor speed, i.e., $r_m = 1$. An optimization approach similar to previous section to find the minimum required servers for each baseline. For *PowerTier*, we use the result in Section 4 to determine the v_m^* at each tier and r_m^* for each server.

Evaluation in the Planning or Upgrading Phase First we determine the optimal number of servers allocated to each tier for the peak service demand in the planning or upgrading phase. We consider the following peak service demand: in the open-queueing model, the maximum arrival $\hat{\lambda}_1$ at Tier 1 is 800 per second; in the closed-queueing model, the maximum number of sessions $\hat{N}_1$ at Tier 1 is 200. The response time constraint is fixed as $\hat{R} = \frac{200}{\mu_1} = 0.24$ sec. Table 3 shows the resulting server allocation and the corresponding speed assignment for each design strategy.

We observe that *EvenDist* needs significantly more processors than the others in all cases. The server assignment at each tier for *PowerTier* and *PropDist* are pretty similar in all cases. In other words, the optimal design under peak demand may adopt proportional server allocation at tiers. *PowerTier* might need extra more servers than *PropDist* (as shown in Type-C Server) since the enable DVFS in *PowerTier* could use more servers with lower speed to reduce power consumption while *PropDist* always uses the highest speed.

Evaluation in the Running Phase Second we compare the performance of *PowerTier* with the baseline design strategies for the running phase in the following two scenarios:

5.2.1 Fixed Mean Response Time Constraint

In this experiment, we fix the mean response time constraint as $\hat{R} = \frac{200}{\mu_1} = 0.24$ second. We conduct evaluation in both open/closed-queueing models.

In the open-queueing model, we vary λ_1 from 0 to 800 per second and choose different types of servers. The results are shown in Figure 5. In the closed-queueing model, we vary N_1 from 0 to 200 and choose different types of servers. Similarly, we obtain the results as shown in Figure 6.

The subfigures in the first, second, and third rows are the optimal power consumption for all design strategies, and the corresponding optimal r_m^* and v_m^* for *PowerTier* respectively. The different columns are the cases for all three type servers. All the sawtooth-shaped processor curves are due to the re-optimization with the consideration of the integer value of v_m^* in *PowerTier*.

Both open/closed-queueing models reveal the similar phenomenon. In all the cases *PowerTier* out-

Table 3: Server allocation and processor speed assignment in the planning or upgrading phase.

(a) Open-Queueing Model

Strategies		Type-A Server	Type-B Server	Type-C Server
PowerTier	(v_1, v_2, v_3)	$(2, 18, 69)$	$(2, 18, 69)$	$(2, 19, 73)$
	(r_1, r_2, r_3)	$(0.7, 0.98, 1)$	$(0.74, 0.98, 0.99)$	$(0.84, 0.95, 0.95)$
EvenDist	(v_1, v_2, v_3)	$(65, 65, 65)$	$(65, 65, 65)$	$(65, 65, 65)$
	(r_1, r_2, r_3)	$(1, 1, 1)$	$(1, 1, 1)$	$(1, 1, 1)$
PropDist	(v_1, v_2, v_3)	$(2, 18, 69)$	$(2, 18, 69)$	$(2, 18, 69)$
	(r_1, r_2, r_3)	$(1, 1, 1)$	$(1, 1, 1)$	$(1, 1, 1)$

(b) Closed-Queueing Model

Strategies		Type-A Server	Type-B Server	Type-C Server
PowerTier	(v_1, v_2, v_3)	$(2, 16, 63)$	$(2, 16, 63)$	$(2, 17, 67)$
	(r_1, r_2, r_3)	$(0.63, 1, 0.99)$	$(0.69, 1, 0.99)$	$(0.81, 0.95, 0.95)$
EvenDist	(v_1, v_2, v_3)	$(59, 59, 59)$	$(59, 59, 59)$	$(59, 59, 59)$
	(r_1, r_2, r_3)	$(1, 1, 1)$	$(1, 1, 1)$	$(1, 1, 1)$
PropDist	(v_1, v_2, v_3)	$(2, 16, 63)$	$(2, 16, 63)$	$(2, 16, 63)$
	(r_1, r_2, r_3)	$(1, 1, 1)$	$(1, 1, 1)$	$(1, 1, 1)$

performs the others. Both the power consumption under *PowerTier* and *EvenDist* seem linearly proportionally to the traffic arrival rate (in the open-queueing model) or the number of sessions (in the closed-queueing model). But the increasing slope for *EvenDist* is larger. For *PropDist*, when the traffic is not heavy, it consumes significant amount of power due to the constraint of the proportional distribution with at least one server at each tier. When the traffic is heavy, *PropDist* approaches to the optimal design *PowerTier*.

5.2.2 Fixed Service Demand

In this experiment, we fix the service demand and vary the response time constraint. We also conduct evaluation in both open/closed-queueing models. We fix the service demand intensity as 25% with respect to the peak one, and vary $\hat{R}$ from $\frac{100}{\mu_1} = 0.12$ second to $\frac{400}{\mu_1} = 0.48$ second.

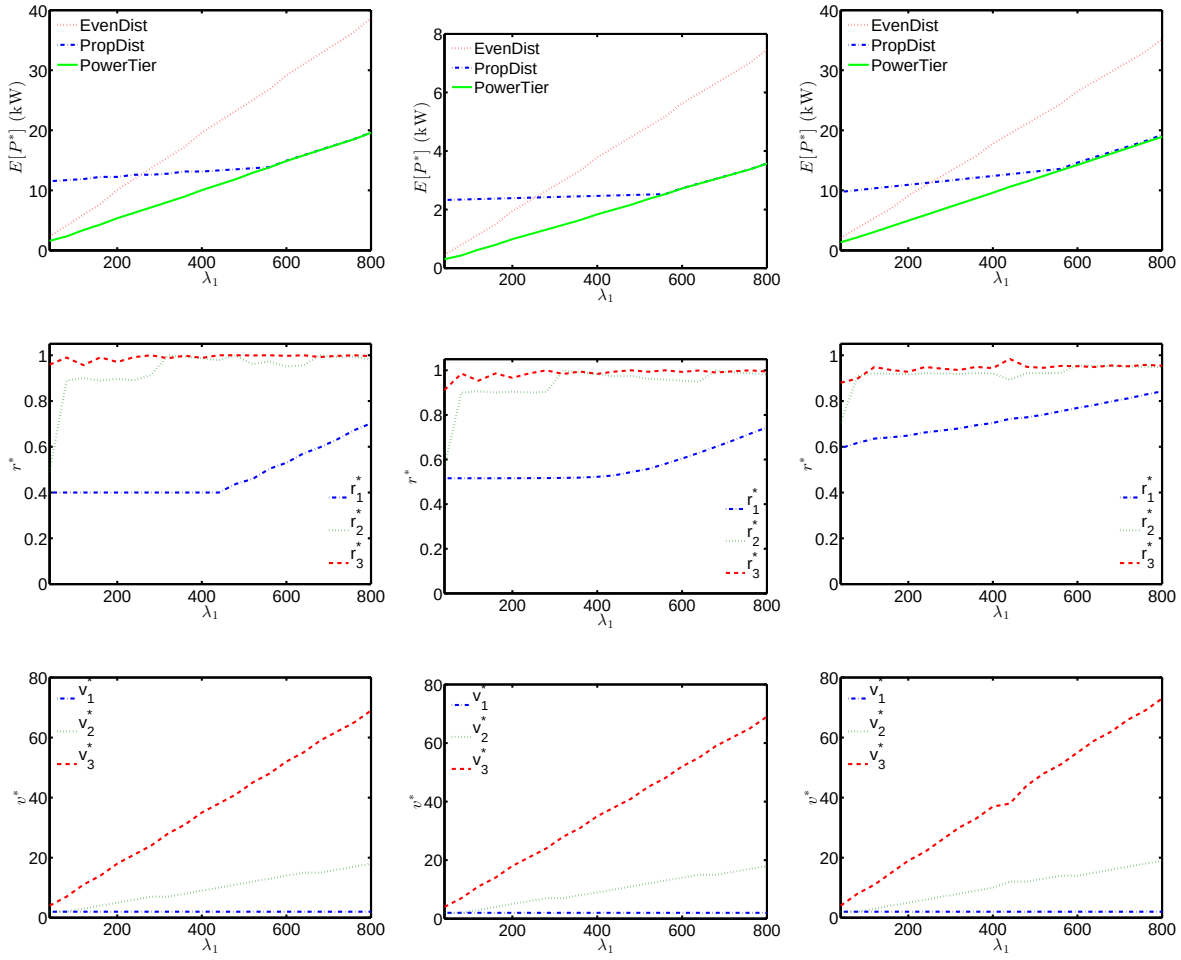
In the open-queueing model, we choose the request arrival rate as $\lambda_1 = 200$ per second. Figure 7 shows the power consumption comparison. In the closed-queueing model, we choose the number of sessions as $N_1 = 50$. Similarly, we obtain the results as shown in Figure 8.

In all design strategies, as $\hat{R}$ increases, the power consumption reduces. When $\hat{R}$ is very small, i.e., the timing requirement is more stringent, then more servers will be activated, which consumes more power. *PowerTier* alway outperforms the others. For larger response time threshold $\hat{R}$, *EvenDist* outforms *PropDist*. For smaller response time threshold $\hat{R}$, *PropDist* outforms *EvenDist*.

6 CONCLUSION AND FUTURE WORK

In this paper, we have explored power-saving design on server farms running multi-tier web applications under a given SLA. We proposed an efficient power-saving design strategy *PowerTier*, which jointly considers both DVS and DPM power-saving techniques. Specifically, we have considered two application models: the open-queueing model and the closed-queueing model for session-less and session-based web applications respectively. With *PowerTier*, we are able to optimally determine the number of servers needed at each tier for the peak service demand in the planning or upgrading phase. And also in the running phase, for different service demands and request visit ratios, we are able to optimally determine the number of activated/deactivated servers at each tier and the processor speed for each activated server. The simulation results showed that *PowerTier* is able to efficiently save the power consumption of server farms while meeting the response time constraint for multi-tier applications. Our simulation has also showed that the optimal server allocation under the open-queueing and closed-queueing models are quite different due to the different application behaviors.

This paper focused on homogeneous servers at each tier. One of potential future work is to extend the current work to heterogeneous servers by taking into consideration the different characteristics of servers in the power consumption and response analysis. However, the optimal design is much more complex and challenging in this case. This paper also targets on the



(a) Type-A Servers (b) Type-B Servers (c) Type-C Servers

Figure 5: Comparison in the open-queueing model for $\hat{R} = 0.24$ second.

stable workload. In the future work, we would like to exploit a control mechanism to activate and deactivate servers dynamically. The control period is assumed to be sufficiently long to pay off the energy and timing overhead of the activation and deactivation of servers.

ACKNOWLEDGMENT

This work is sponsored in part by NSF CAREER Grant No. CNS-0746906, Baden Wuerttemberg MWK Juniorprofessoren-Programms, NSERC Discovery Grant 341823, FQRNT grant 2010-NC-131844, CFI Leaders Opportunity Fund 23090, and National Science Foundation Award 1116606 and 1117664.

REFERENCES

Bohrer, P., Elnozahy, E., Keller, T., Kistler, M., Lefurgy, C., McDowell, C., and Rajamony, R. (2002). The case for power management in web servers. *Power Aware Computing*, pages 261–289.

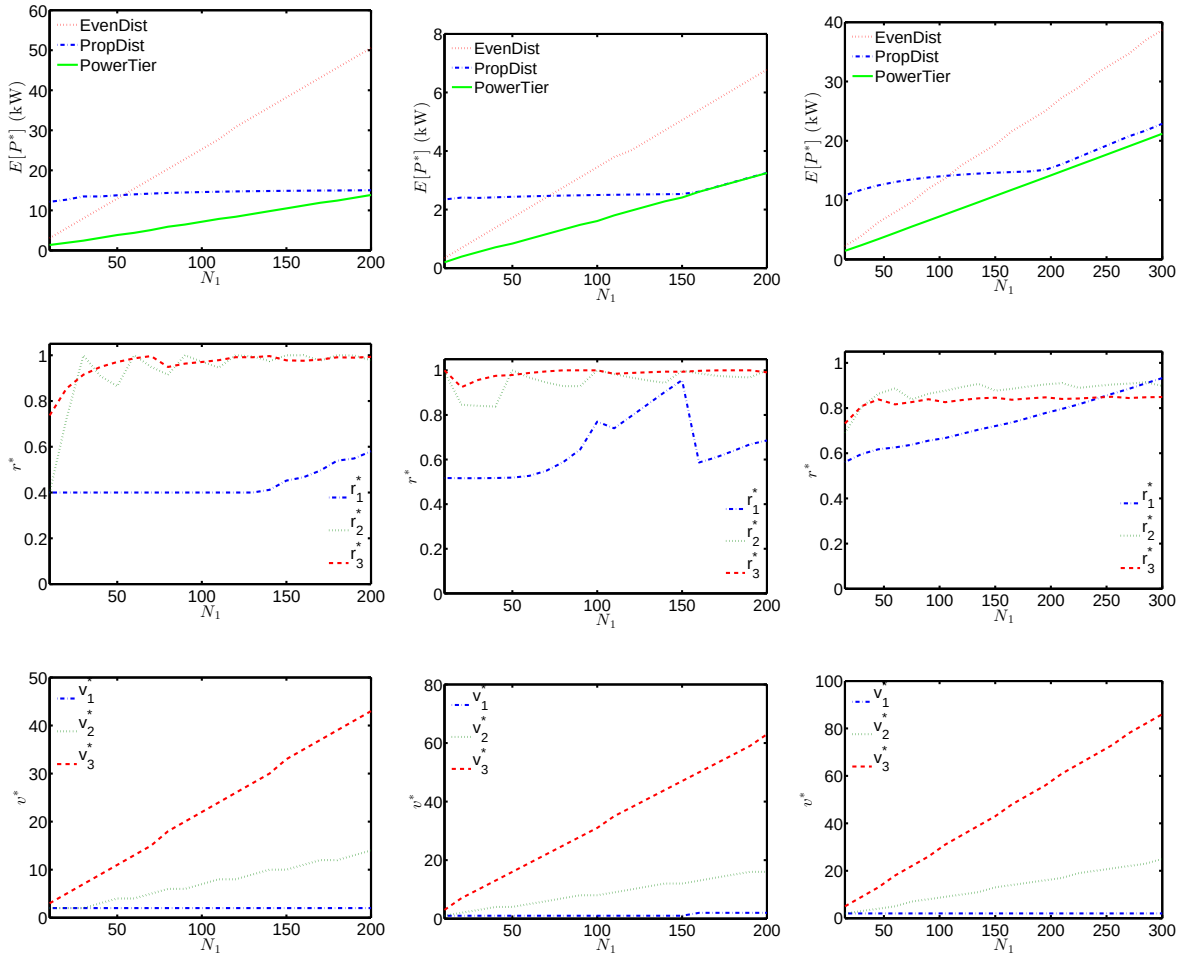
Cecchet, E., Marguerite, J., and Zwaenepoel, W. (2002). Performance and scalability of EJB applications. *ACM Sigplan Notices*, 37:246–261.

Diao, Y., Hellerstein, J., Parekh, S., Shaikh, H., Surendra, M., and Tantawi, A. (2006). Modeling differentiated services of multi-tier web applications. In *IEEE International Symposium on Modeling, Analysis, and Simulation*.

Gandhi, A., Harchol-Balter, M., Das, R., and Lefurgy, C. (2009). Optimal power allocation in server farms. In *ACM SIGMETRICS*.

Guerra, R., Leite, J., and Fohler, G. (2008). Attaining soft real-time constraint and energy-efficiency in web servers. In *ACM symposium on Applied computing*.

Heo, J., Henriksson, D., Liu, X., and Abdelzaher, T. (2007).



(a) Type-A Servers (b) Type-B Servers (c) Type-C Servers

Figure 6: Comparison in the closed-queueing model for $\hat{R} = 0.24$ second.

Integrating adaptive components: An emerging challenge in performance-adaptive systems and a server farm case-study. In *IEEE Real-Time Systems Symposium*.

Jain, R. (1991). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons Inc.

Kamra, A., Misra, V., and Nahum, E. (2004). Yaksha: a self-tuning controller for managing the performance of 3-tiered web sites. In *IEEE International Workshop on Quality of Service*.

Kleinrock, L. (1976). *Queueing Systems Volume II: Computer applications*. Wiley Interscience.

Lazowska, E., Zahorjan, J., Graham, G., and Sevcik, K. (1984). *Quantitative system performance: computer system analysis using queueing network models*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

Liu, X., Heo, J., and Sha, L. (2005). Modeling 3-tiered web applications. In *IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2005*, pages 307–310.

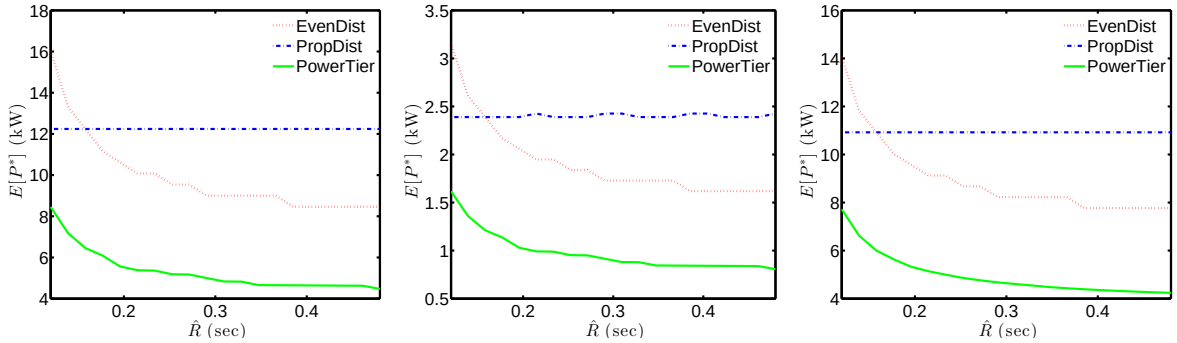
Liu, X., Heo, J., Sha, L., and Zhu, X. (2006). Adaptive control of multi-tiered web applications using queueing predictor. In *IEEE/IFIP Network Operations and Management Symposium*.

Liu, X., Heo, J., Sha, L., and Zhu, X. (2008). Queueing-model-based adaptive control of multi-tiered web applications. *IEEE Transactions on Network and Service Management*, 5(3):157–167.

Pacifici, G., Segmuller, W., Spreitzer, M., Steinder, M., Tantawi, A., and Youssef, A. (2005). Managing the response time for multi-tiered web applications. Technical Report RC 23651, IBM.

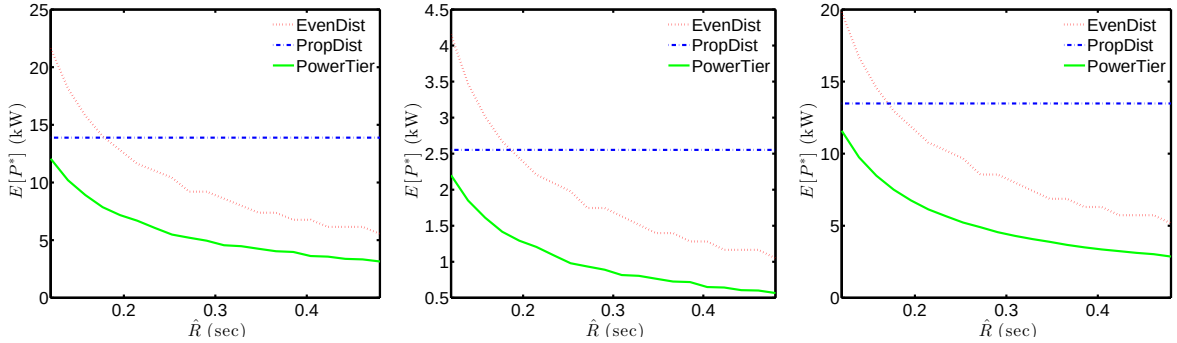
Raghavendra, R., Ranganathan, P., Talwar, V., Wang, Z., and Zhu, X. (2008). No "power" struggles: coordinated multi-level power management for the data center. In *International conference on Architectural support for programming languages and operating systems*.

Reiser, M. and Lavenberg, S. S. (1980). Mean-value anal-



(a) Type-A Servers (b) Type-B Servers (c) Type-C Servers

Figure 7: Comparison in the open-queueing model for $\lambda_1 = 200$ per second.



(a) Type-A Servers (b) Type-B Servers (c) Type-C Servers

Figure 8: Comparison in the closed-queueing model for $N_1 = 50$.

ysis of closed multichain queuing networks. *J. ACM*, 27(2):313–322.

Rolia, J. and Sevcik, K. (1995). The method of layers. *IEEE Transactions on Software Engineering*, 21(8):689–700.

Rusu, C., Ferreira, A., Scordino, C., and Watson, A. (2006). Energy-efficient real-time heterogeneous server clusters. In *IEEE Real-Time and Embedded Technology and Applications Symposium*.

Sharma, V., Thomas, A., Abdelzaher, T. F., Skadron, K., and Lu, Z. (2003). Power-aware QoS management in web servers. In *IEEE Real-Time Systems Symposium*.

Urgaonkar, B., Pacifici, G., Shenoy, P., Spreitzer, M., and Tantawi, A. (2005). An analytical model for multi-tier internet services and its applications. *ACM SIGMETRICS Performance Evaluation Review*, 33(1):302.

U.S. Environmental Protection Agency (EPA) (2007). Report to congress on server and data center energy efficiency, public law 109-431.

Wang, L. and Lu, Y. (2008). Efficient power management of heterogeneous soft real-time clusters. In *IEEE Real-Time Systems Symposium*.

Wang, P., Qi, Y., Liu, X., Chen, Y., and Zhong, X. (2010). Power management in heterogeneous multi-tier web clusters. In *International Conference on*. IEEE.

Wierman, A., Andrew, L. L. H., and Tang, A. (2009). Power-aware speed scaling in processor sharing systems. In *IEEE INFOCOM*.