# Power-aware Provisioning of Cloud Resources for Real-time Services*

Kyong Hoon Kim
Dept. of Informatics
Gyeongsang National University
900 Gajwadong, Jinju, Korea
khkim@gnu.ac.kr

Anton Beloglazov, Rajkumar Buyya
CLOUDS Lab
Dept. of Computer Science and Software Eng.
The University of Melbourne, Australia
{abe, raj}@csse.unimelb.edu.au

## ABSTRACT

Reducing energy consumption has been an essential technique for Cloud resources or datacenters, not only for operational cost, but also for system reliability. As Cloud computing becomes emergent for *Anything as a Service (XaaS)* paradigm, modern real-time Cloud services are also available throughout Cloud computing. In this work, we investigate power-aware provisioning of virtual machines for real-time services. Our approach is (i) to model a real-time service as a real-time virtual machine request; and (ii) to provision virtual machines of datacenters using DVFS (Dynamic Voltage Frequency Scaling) schemes. We propose several schemes to reduce power consumption and show their performance throughout simulation results.

## Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed applications

## General Terms

Power-aware cloud computing

## Keywords

Power-awareness, Cloud computing, Real-time

## 1. INTRODUCTION

Development in computers and communications technology has led a new computing paradigm called *Cloud computing*, which delivers computing services to users as a utility in a pay-as-you-go manner [3]. The Cloud providers offer various types of services, such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Service providers make good use of IaaS and PaaS for developing their services without consideration of physical hardwares, while users also can access on-demand and pay-per-use services anywhere in Cloud computing.

One of big challenges in datacenters is to manage power in the systems. Datacenters consume 10 to 100 times more energy per square foot than typical office buildings [19]. They can even consume as much electricity as a city [15]. The most part of power consumption in datacenters comes from computation processing, disk storage, network, and cooling systems. In this paper, we study the processing power management throughout the Virtual Machine (VM) provisioning which is an essential technique in Cloud computing.

Pay-as-you-go mechanism in Cloud computing assures *Service Level Agreements (SLAs)* between customers and Cloud providers. SLAs specify the negotiated agreements including *Quality of Service (QoS)*, such as deadline. Thus, datacenters minimize power consumption without violating these SLAs. As many applications require deadline constraints, this paper focuses on power-aware real-time Cloud services, such as financial analysis, distributed image processing, real-time distributed databases, and so on. The main contribution of this paper is (i) to provide real-time Cloud service framework for requesting a virtual platform, and (ii) to investigate various power-aware VM provisioning schemes based on DVFS (Dynamic Voltage Frequency Scaling) schemes.

The remainder of this paper is organized as follows. Section 2 presents related work on power-aware Cloud computing. We propose the real-time Cloud service framework in Section 3. Section 4 describes the problem definition and provides several DVFS schemes. We evaluate them throughout simulations in Section 5, and finally conclude the paper.

## 2. RELATED WORK

Much recent research has focused on reducing power consumption in cluster systems. In work [1, 27], high performance clusters with consideration of energy consumption has been designed and developed. As many recent commodity processors provide DVFS ability, power-aware clusters systems has been built on such processors [10, 11]. Scientific applications developed by MPI library are mostly targeted to reduce energy consumption [12, 10, 21]. Based on the profile of MPI programs, they choose an appropriate voltage scaling to each synchronization point.

General purpose cluster systems also have studied on reducing power consumption. In [22], they deal with online

*This research was supported in part by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-C1090-0904-0001), and by the Korea Research Foundation (KRF) grant funded by the Korea government (MEST) (No. 2009-0077571).


*MGC'09,* 30 November - 1 December 2009, Urbanan-Champaign, Illinois, US.

services executing on heterogeneous clusters. When a new request comes, a heuristic for multidimensional bin packing is used to find a server to allocate the request. If a server cannot be found, a new machine is turned on and all requests are re-allocated. The work in [7] aims at serving web-applications on homogeneous clusters according to utility function. In work of [9], they investigate the problem of minimizing mean response time of web-applications on heterogeneous clusters. In this work the optimal power allocation is determined based on queuing theoretical model.

Recently emerged Cloud Computing paradigm leverages virtualization of computer resources and allows to achieve more efficient allocation of workload in terms of higher resource utilization as well as decreased power consumption. The work in [14] is targeted on minimizing both power consumption and SLA violations for online services on virtualized datacenters using a limited look-ahead control. The *pMapper* architecture is also proposed in [24, 23] to solve the same problem with consideration of migration cost. In the work [6], they present several techniques for addressing the sharing aware VM allocation problem. Hypervisor distributes resources among VMs according to a sharing based mechanism, when the minimum and maximum amount of resources that can be allocated to a VM is specified.

In addition, many studies have focused on power-aware real-time applications on clusters. A QoS-aware power management scheme is presented by combining cluster-wide (On/Off) and local (DVS) power management techniques in the context of heterogeneous clusters [18]. The front-end manager decides which servers are turned on or off for a given system load, while local PM reduces power consumption using DVS scheme. In [26], a threshold-based method is proposed for efficient power management of heterogeneous soft real-time clusters as well as the offline mathematical analysis of determining the threshold. In addition, power-aware algorithms are investigated for scheduling of real-time bag-of-tasks applications with deadline constraints on homogeneous clusters [13].

Considerable amount of work have been done in the area of power-efficient computing, but few of them deal with power-aware scheduling of real-time applications in Cloud Computing environments. This work investigates the problem of provisioning Cloud resources for real-time services in order to minimize power consumption by modeling real-time virtual machine requirement and using DVFS techniques.

# 3. FRAMEWORK

## 3.1 Real-time Service Model

A usual real-time service such as financial analysis, distributed databases, or image processing, consists of multiple real-time applications or subtasks. As long as a group of applications for a given real-time service meet all their deadlines, the service accomplishes the quality of service agreed with users. A real-time service is defined by $\{\tau_i(r_i, c_i, d_i, p_i, f_i)|i = 1, \ldots, n\}$, where $n$ is the number of subtasks. Each real-time subtask $\tau_i$ is defined by the following parameters.

- $r_i$: release time
- $c_i$: worst-case execution time
- $d_i$: relative deadline
- $p_i$: period
- $f_i$: finish time

A real-time application can be started at time $r_i$ and requires the worst-case execution time $c_i$. In order to accomplish the application's objective, it should be completed by the time $r_i + d_i$ after released. Also, $p_i$ specifies its periodicity so that the task releases a job of $c_i$ computation time at time $(r_i + kp_i)$, and should be finished by $r_i + kp_i + d_i$ $(k = 0, 1, \ldots)$. In case of non-periodic application, $p_i$ is set to zero. We also consider duration or finish time, $f_i$, since a user cannot access a cloud computing resource forever, although a periodic real-time task in embedded system assumes an infinite sequence.

This group of sub-tasks of a real-time service is developed and launched on a specific run-time platform including middlewares, operating systems, and so on. The cloud computing environment is a suitable solution for real-time services by leveraging *virtualization*. When users request their requirements for real-time services to the cloud computing environment, appropriate virtual machines are allocated for executing those services.

## 3.2 Real-time Virtual Machine Model

Cloud resource brokers take a role in finding Cloud resources or virtual machines for real-time services requested by users. In this paper, we define *RT-VM (Real-Time Virtual Machine)* as the requirement of a virtual machine for providing a real-time service. RT-VM $V_i$ of a real-time service includes three parameters $u_i$, $m_i$, and $d_i$.

- $u_i$ : utilization of real-time applications
- $m_i$ : MIPS (Million Instructions Per Second) rate of the based virtual machine
- $d_i$ : lifetime or deadline

The service is developed and launched on a specific platform or infrastructure (e.g. 1GHz-Linux machine). We select the MIPS rate, $m_i$, for the specification of the base machine. For a given set of real-time applications, we can analyze the required CPU utilization $u_i$ on the base machine. Thus, the above requirement implies that the real-time service is guaranteed when the allocated virtual machine keeps providing $u_i \times m_i$ amount of processing capacity by the deadline $d_i$. This real-time service on virtualized cloud resource is achieved by *compositional real-time computing* and *real-time virtual machine* techniques.

The compositional or hierarchical real-time framework [8, 20] enables a group of real-time applications to be a single real-time resource requirement to the upper layer of real-time environments. Thus, we assume that a RT-VM $V_i$ is defined from multiple real-time applications, $\{\tau_k(r_k, w_k, d_k, p_k, f_k)|k = 1, \ldots, n\}$, of the service by using compositional real-time technique. Thus, VM provisioner in Clouds maps virtual machines for the service, not for individual applications. Furthermore, recent work on implementing real-time virtual machines [25, 28] assures real-time services (e.g. real-time CPU allocation, real-time I/O) of a virtual machine. This paper focuses on how to provision virtual machines to a given RT-VM request with consideration of power consumption by leveraging these techniques.

## 3.3 Real-time Cloud Service Framework

In this subsection, we describe the real-time cloud service framework based on real-time virtual machine model. As shown in Figure 1, the steps for a real-time service are as follows.

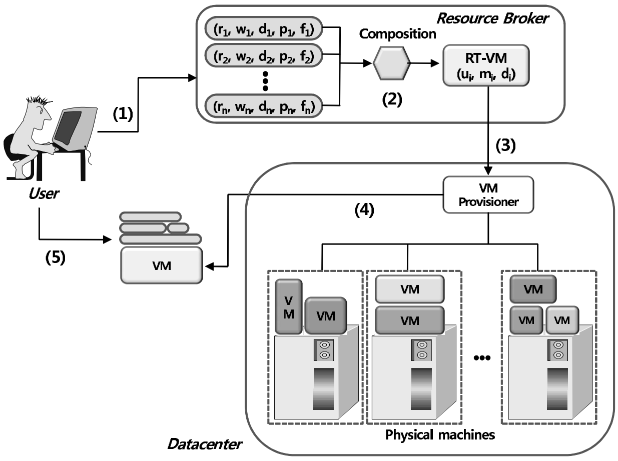



**Figure 1: Framework**

(1) *Requesting a virtual platform*: A user who wants to launch a real-time service submits all the information about real-time applications to the broker.

(2) *Generating the RT-VM from real-time applications*: The resource broker first analyzes the submitted real-time applications and generates one RT-VM request, $V_i = (u_i, m_i, d_i)$.

(3) *Requesting a real-time virtual machine*: The broker requests a virtual machine for RT-VM $V_i$ to the VM provisioner in the cloud computing.

(4) *Mapping the physical processors*: The VM provisioner finds appropriate processing elements which meet the $V_i$ requirement. And then, it provides the VM to the user.

(5) *Executing the real-time applications*: The user finally launches and executes real-time applications on the provided VM.

## 3.4 Energy model

The most part of power consumption in datacenters comes from computation processing, disk storage, network, and cooling systems. This paper focuses on CPU power saving in terms of virtual machine provisioning in the cloud computing.

The main power consumption in CMOS circuits is composed of dynamic and static power. We only consider the dynamic power dissipation because it is more dominating factor in the total power consumption [16]. And, datacenters can increase their profit by reducing dynamic power consumption. The dynamic energy consumption by an application is proportional to $V_{dd}^2$ and $f$, where $V_{dd}$ is the supply voltage and $f$ is the frequency [4]. Since the frequency is usually in proportion to the supply voltage, the dynamic power consumption of processor is given by

$$P = C \cdot f^3$$

where $C$ is a proportional coefficient. Let us consider an application of $t$ execution time at the frequency $f_{max}$ of the processor. If the processor runs at $f$ frequency level ($0 < f \leq f_{max}$), the execution time is defined by $t/\frac{f}{f_{max}}$. Thus, the dynamic energy consumption during the task execution is defined by Equation (1).

$$E = \int_0^{t/\frac{f}{f_{max}}} P = C \cdot t \cdot f_{max} \cdot f^2 \\ = \alpha \cdot t \cdot S^2 \quad (1)$$

where $\alpha$ is a coefficient and $S$ is the associated processor speed related to the frequency $f$ ($S = f/f_{max}$). The DVFS (Dynamic Voltage Frequency Scaling) scheme reduces the dynamic energy consumption by decreasing the supplying voltage and frequency, which results in slowdown of the execution time. We assume that each PE (Processing Element) $p$ in a datacenter can adjust its processor frequency from $f_p^{min}$ to $f_p^{max}$ continuously. The associated processor speed $S$ with each frequency $f$ is defined by $f/f_{max}$, which follows that $f_p^{min}/f_p^{max} < S \leq 1$.

# 4. POWER-AWARE RT CLOUD SERVICE

## 4.1 Problem Description

Let us consider a physical machine with one PE of 2400 MIPS and a set of RT-VMs, $\{V_1(0.2, 1000, 10)$, $V_2(0.8, 500, 15)$, $V_3(0.5, 1200, 20)\}$, as an example. $V_1$ requires the utilization 20% on 1000-MIPS machine by the deadline 10 sec. Similarly, $V_2$ and $V_3$ require 80% and 50% of 500-MIPS and 1200-MIPS machines by 15 and 20 seconds, respectively. Figure 2(a) shows the proportional sharing scheduling result of three VMs under the maximum processor capacity. The proportional share of $V_i$ is defined by $m_i \times u_i / \sum_{j=1}^{3}(m_j \times u_j)$. Three RT-VMs share the processor capacity in proportion to their required MIPS rates, $m_i \times u_i$, and finish before the deadlines. The total energy consumption is 8.34 ($= 1 \times 8.34 \times 1.0^2$) by Equation (1) under the assumption of $\alpha = 1$.

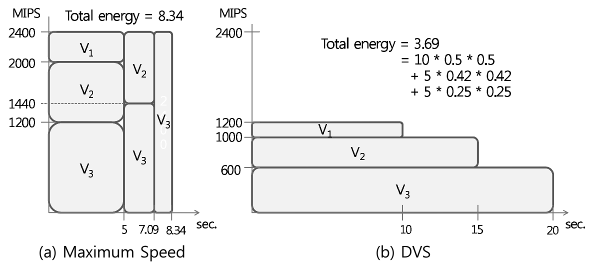



**Figure 2: Proportional sharing of VM provisioning and energy consumption ($\alpha = 1$)**

The energy consumption can be reduced by combining DVS and proportional sharing scheduling. As shown in Figure 2(b), the minimum required processor capacity is allocated to each virtual machine, so that the processor dynamically adjust its speed to $\sum(m_j \times u_j)/2400$. The total energy consumption of DVS scheme is 3.69 ($= 10 \times 0.5^2 + 5 \times 0.42^2 + 5 \times 0.25^2$). Thus, DVS scheme can reduce much energy compared to the maximum-speed static scheme.

However, there are tradeoffs in dynamic scaling of processor speed in on-line real-time cloud computing. Operation in higher speed processor can accept more RT-VMs with more energy consumption. On the contrary, scaling down to lower processor speed consumes less energy with lower acceptance. For example, let us assume that a new RT-VM $V_4$ (0.8, 2000, 10) is requested at time 10. Figure 2(a) accepts $V_4$ since the processor is idle at time 10, while DVS scheme of Figure 2(b) cannot provision it due to lack of processor capacity.

Datacenters can increase their profit by provisioning more virtual machines to users. In addition, reducing energy consumption also increases profit by reducing the cost for clouding service. Thus, this paper provides several schemes on power-aware provisioning of real-time virtual machines for the purpose of maximizing profits of cloud computing datacenters.

We use proportional sharing scheduling for scheduling multiple virtual machines on a processor. The proportional sharing scheduling is simple but guarantees the real-time services of RT-VMs if the total required MIPS rate is less than or equal to the processor capacity. Furthermore, it can be easily implemented. For example, the default scheduling in Xen Hypervisor [17] is *Credit scheduler* which is based on credit value set by weight of each VM. The VMM (Virtual Machine Monitor) can dynamically adjust credit values of VMs according their required MIPS rates in order to support the proportional sharing scheme.

Before explaining the VM provisioning, we define the remaining service time, $w_i$, of $V_i$. The initial value of $w_i$ is defined by $u_i \times m_i \times (d_i - t_s)$, at its submission time $t_s$. If $V_i$ is provided with $q_i$ MIPS rate for the period $t_p$, $w_i$ is decreased by $q_i \times t_p$. For instance, Table 1 shows the remaining service times of three RT-VMs at the time of proportional share change of Figure 2(a). $V_i$ finishes its service when $w_i$ becomes zero.

**Table 1: Remaining service times of Figure 2(a)**

| | $t=0$ | $t=5$ | | $t=7.09$ | | $t=8.34$ | |
|---|---|---|---|---|---|---|---|
| | $w_i$ | $ST_i$ [0, 5] | $w_i$ | $ST_i$ [5, 7.09] | $w_i$ | $ST_i$ [7.09, 8.34] | $w_i$ |
| $V_1$ | 2000 | 2000 | 0 | - | - | - | - |
| $V_2$ | 6000 | 4000 | 2000 | 2000 | 0 | - | - |
| $V_3$ | 12000 | 6000 | 6000 | 3010 | 2990 | 2990 | 0 |

($ST_i[t_1, t_2]$ : The service time of $V_i$ from $t_1$ to $t_2$)

## 4.2 DVS-enabled RT-VM Provisioning

When a datacenter receives a RT-VM request from a resource broker, it returns the price of providing the RT-VM service if it can provide real-time virtual machines for that request. The broker selects the minimum-price virtual machine among available datacenters. Thus, the provisioning policy in this paper is to select the processing element with the minimum price for the sake of users. Figure 3 shows the pseudo-algorithm of provisioning the virtual machine for a given RT-VM request .

For a given RT-VM $V_i(u_i, m_i, d_i)$, the datacenter checks the schedulability of $V_i$ on the processing element $PE_k$ of $Q_k$ MIPS rate. Suppose that the current running RT-VMs on the processing element $PE_k$ at time $t$ is known as $T_k = \{V_j(u_j, m_j, d_j) | j = 1, \cdots, n_k\}$. And the remaining service time of $V_j$ at time $t$ is denoted as $w_j$. Then, the schedulability is guaranteed if it satisfies Equation (2). Since $w_j/(d_j - t)$ is the minimum MIPS rate for $V_j$ by its deadline $d_j$, Equation (2) means that total summation of all the required MIPS rates including the new RT-VM $V_i$ is less than the processor capacity $Q_k$.

**Algorithm Min-Price RT-VM Provisioning** ($V_i$)

1: $VM \leftarrow null$;
2: $alloc \leftarrow -1$;
3: $e_{min} \leftarrow$ MAX_VALUE;
4: $price_{min} \leftarrow$ MAX_VALUE;
5: **for** $k$ **from** 1 **to** $N$ **do**
6:     **if** ( $u_i \times m_i + \sum_{j=1}^{n_k} \frac{w_j}{d_j - t} \leq Q_k$ ) **then**
7:         $e_k \leftarrow$ *energy_estimate* $(PE_k, V_i)$;
8:         $price_k \leftarrow$ *price for the RT-VM* $V_i$ *in* $PE_k$;
9:         **if** $price_k < price_{min}$ **or**
10:           ($price_k = price_{min}$ **and** $e_k < e_{min}$) **then**
11:             $price_{min} \leftarrow price_k$;
12:             $e_{min} \leftarrow e_k$;
13:             $alloc \leftarrow k$;
14:         **endif**
15:     **endif**
16: **endfor**
17: **if** $alloc \neq -1$ **then**
18:     $VM \leftarrow$ *create_VM* ($PE_{alloc}$, $V_i$);
19: **endif**
20: **return** *VM*;

**Figure 3: Min-Price RT-VM Provisioning**

$$u_i \times m_i + \sum_{j=1}^{n_k} \frac{w_j}{d_j - t} \leq Q_k \tag{2}$$

If $PE_k$ is able to schedule $V_i$, it estimates energy and price of provisioning (line 7, 8). Since the provisioning policy is to provide lower price to users, the algorithm finds the minimum-price processor. For the same price, less energy is preferable because it produces higher profit (line 9-14). Finally, a virtual machine is mapped on $PE_{alloc}$ if RT-VM $V_i$ is schedulable on the datacenter.

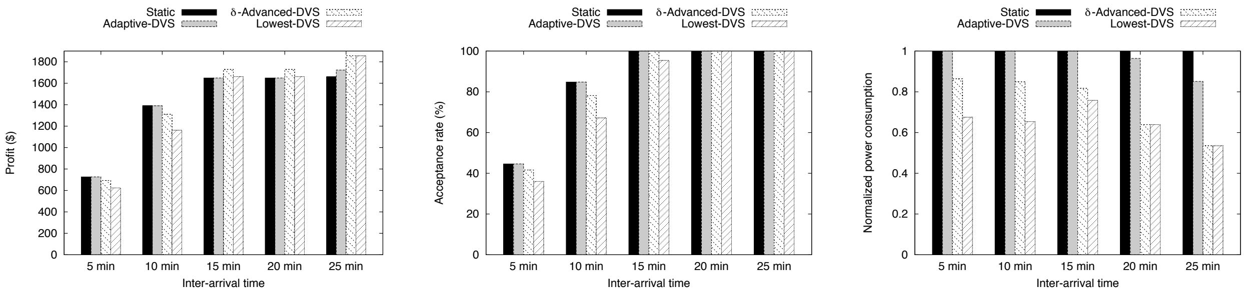
When a user launches the service on the VM, the resource provider provision the VM using DVS schemes to reduce the power consumption. We propose three power-aware VM provisioning schemes: *Lowest-DVS*, *δ-Advanced-DVS*, and *Adaptive-DVS*. The following subsections describe them.

### 4.2.1 Lowest-DVS for VM Provisioning

This scheme adjusts the processor speed to the lowest level at which RT-VMs meet their deadlines. That is, each RT-VM $V_i$ executes its service at the required MIPS rate, as shown in Figure 2(b). It consumes the lowest energy in the case that the RT-VM arrival rate is low enough to accept all the requests.

### 4.2.2 δ-Advanced-DVS for VM Provisioning

In order to overcome the low service acceptance rate of Lowest-DVS scheme, this scheme over-scales more up to $\delta\%$ of the required MIPS rate for current RT-VMs. Thus, it operates the processor speed $\delta\%$ faster in order to increase the possibility of accepting coming RT-VM requests. The processor scale $s$ is adjusted as in Equation (3) at time $t$ for a given RT-VM set $T_k$. The proportional share of each VM




(a) Total profit (b) Acceptance rate (c) Normalized power consumption

**Figure 4: Simulation results**

is in proportional to $w_i/(d_i - t)$.

$$s = \min\left\{1,\ (1+\frac{\delta}{100}) \times \frac{1}{Q_k}\sum_{V i \in T_k}\frac{w_i}{d_i - t}\right\} \tag{3}$$

The value of $\delta$% is predefined in the systems according to the system load. Throughout the simulation results in Section 5, we analyze the impact of $\delta$.

### 4.2.3 *Adaptive-DVS for VM Provisioning*

When the RT-VM arrival rate and its service time are known in advance, we can analyze an optimal scale. Let us consider the $M/M/1$ queuing model with arrival rate $\lambda$ and service rate $\mu$. If the processor speed scale is set to $s$, then the average response time, $RT$, is given by $RT = 1/(s\mu - \lambda)$ by $M/M/1$ queuing model. And, the response time should be less than or equal to the average deadline, $d$, in order to to meet their real-time services ($1/(s\mu - \lambda) \leq d$ ). Thus, an optimal scale, $s^*$, to reduce the power-consumption is given by Equation (4).

$$s^* = \frac{1}{\mu}\left(\lambda + \frac{1}{d}\right) \tag{4}$$

Adaptive-DVS scheme manages the average arrival rate $\hat{\lambda}$, the average service rate $\hat{\mu}$, and the average deadline $\hat{d}$ for the last $h$ service requests (e.g. $h = 10$). And, it adjusts the processor scale $s$ as in Equation (5) at time $t$ for a given RT-VM set $T_k$.

$$s = \max\left\{\min\left\{1,\ \frac{1}{\hat{\mu}}(\hat{\lambda} + \frac{1}{\hat{d}})\right\},\ \frac{1}{Q_k}\sum_{V i \in T_k}\frac{w_i}{d_i - t}\right\} \tag{5}$$

In Equation (5), the optimal scale is calculated by Equation (4) not greater than one. Since it should be greater than the minimum required utilization of the current RT-VMs on the processor, we select the maximum between two values. So, the processor adjusts the processor speed by Equation (5) when a new RT-VM is provided or an existing one finishes its service.

## 5. SIMULATION RESULTS

We evaluate simulations of power-aware real-time services using the CloudSim toolkit [5] with additional development of power-awareness capability. We create a datacenter with four machines with 16 DVS-enabled processors of which characteristics are shown in Table 2.

**Table 2: Characteristics of datacenter**

| | # of PEs | MIPS of PE | DVS level | $\alpha$ ($10^{-3}$) |
|---|---|---|---|---|
| Machine 0 | 4 | 1,800 | [0, 1.0] | 2.92 |
| Machine 1 | 4 | 2,400 | [0, 1.0] | 4.08 |
| Machine 2 | 4 | 3,000 | [0, 1.0] | 5.37 |
| Machine 3 | 4 | 3,400 | [0, 1.0] | 6.21 |

The price model in the simulations follows the Amazon EC2 Standard small (default) instance type [2], so that the unit price per hour is given by $0.10. We use the cost function as the power consumption of each machine in Table 2.

In the simulations, we generate 500 RT-VMs. The total service amount ($w_i$) of a RT-VM is randomly selected from 2,400 GIs ($10^3$ MIs) to 3,600 GIs. The deadline is selected from 10 to 30 minutes more than the execution time based on 1000-MIPS machine. The interarrival time between two RT-VMs follows a Poisson distribution. We simulate various interarrival times.

Figure 4(a) shows the total profits of each scheme according to interarrival time. **Static** does not use DVS so that it runs virtual machines at the maximum processor speed. In **$\delta$-Advanced-DVS** we fix $\delta$ as 15%. In higher arrival rates, **Static** produces more profits since it accepts more RT-VMs. **Adaptive-DVS** gives no less profit than **Static**, while other DVS schemes show more profit in lower arrival rates due to lower energy consumption.

Figure 4 (b) and (c) show the RT-VM acceptance rate and the normalized power consumption compare to **Static**, respectively. The acceptance rate of **Adaptive-DVS** is close to **Static** but reduces much energy in case of low arrival rate. **$\delta$-Advanced-DVS** shows more acceptance rate with similar energy consumption compared to **Lowest-DVS**. Generally, **$\delta$-Advanced-DVS** shows the best performance in terms of profit per consumed power since the amount of scaling up is controlled automatically according to the system load. In case of **Adaptive-DVS**, its performance is limited by the simplified queueing model.

Next, we also vary the value of $\delta$ in order to analyze the impact of $\delta$. As shown in Figure 5, higher $\delta$ shows better performance in higher arrival rate since it may accept more VMs. On the contrary, lower $\delta$ produces more profit in case of lower arrival rate. In the simulations, the system utiliza-

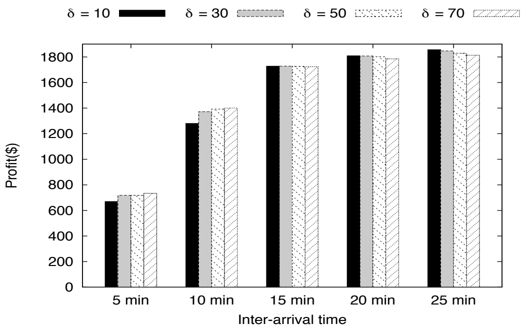



**Figure 5: Impact of $\delta$ in $\delta$-Advanced-DVS**

tion is generally high regardless of arrival rates, so that $\delta$ has little impact on the profit.

# 6. CONCLUSIONS

In this paper, we have proposed a real-time Cloud service framework where each real-time service request is modeled as RT-VM in resource brokers. And then, we have investigated power-aware provisioning of virtual machines for real-time Cloud services. Simulation results show that datacenters can reduce power consumption and increase their profit using DVS schemes. The proposed adaptive schemes, Adaptive-DVS and $\delta$-Advanced-DVS, show more profit with less power consumption regardless of system load.

Our immediate future work includes more analysis and improvement of the proposed adaptive schemes. For example, we will compare them with other approaches, such as bin packing or linear programming, and analyze the impact in the cooling systems. We also plan to implement the proposed framework in Cloud broker and experiment it.

# 7. REFERENCES


[1] N. D. Adiga, et al. An overview of the BlueGene/L supercomputer. In *Proc. of ACM/IEEE Conf. on Supercomputing*. Baltimore, USA, November 2002.

[2] Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2.

[3] M. Armbrust, et al. Above the Clouds: A Berkeley view of cloud computing. *Tech. Report No. UCB/EECS-2009-28*, University of California at Berkeley, USA, February 2009.

[4] T. D. Burd and R. W. Brodersen. Energy efficient cmos microprocessor design. In *Proc. of Annual Hawaii Intl. Conf. on System Sciences*, pages 288–297, January 1995.

[5] R. Buyya, R. Ranjan, and R. N. Calheiros. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In *Proc. of the 7th High Performance Computing and Simulation (HPCS 2009)*. Leipzig, Germany, June 2009.

[6] M. Cardosa, M. R. Korupolu, and A. Singh. Shares and utilities based power consolidation in virtualized server environments. In *Proc. of IFIP/IEEE Intl. Symp. on Integrated Network Management*. USA, June 2009.

[7] J. S. Chase, et al. Managing energy and server resources in hosting centers. In *Proc. of 8th ACM Symp. on Operating Systems Principles*. Banff, Canada, October 2001.

[8] X. A. Feng and A. K. Mok. A model of hierarchical real-time virtual resources. In *Proc. of 23rd IEEE Real-Time Systems Symposium*. Austin, USA, Dec. 2002.

[9] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy. Optimal power allocation in server farms. In *Proc. of Intl. Joint Conf. on Measurement and Modeling of Computer Systems*, pages 157–168. Seattle, USA, June 2009.

[10] R. Ge, X. Feng, and K. W. Cameron. Performance-constrained distributed DVS scheduling for scientific applications on power-aware clusters. In *Proc. of ACM/IEEE Conf. on Supercomputing*. Seattle, USA, November 2005.

[11] C. Hsu and W. Feng. A power-aware run-time system for high-performance computing. In *Proc. of ACM/IEEE Conf. on Supercomputing*. Seattle, USA, November 2005.

[12] N. Kappiah, V. W. Freeh, and D. K. Lowenthal. Just in time dynamic voltage scaling: Exploiting inter-node slack to save energy in MPI programs. In *Proc. of ACM/IEEE Conf. on Supercomputing*. Seattle, USA, November 2005.

[13] K. H. Kim, R. Buyya, and J. Kim. Power aware scheduling of bag-of-tasks applications with deadline constraints on DVS-enabled clusters. In *Proc. of 7th IEEE Intl. Symp. on Cluster Computing and the Grid (CCGrid'07)*, pages 541–548. Rio de Janeiro, Brazil, May 2007.

[14] D. Kusic, et al. Power and performance management of virtualized computing environments via lookahead control. In *Proc. of 5th IEEE Intl. Conf. on Autonomic Computing (ICAC 2008)*, pages 3–12. Chicago, USA, June 2008.

[15] J. Markoff and S. Lohr. Intel's huge bet turns iffy. *New York Times Technology Section*, September 2002.

[16] L. Niu and G. Quan. Reducing both dynamic and leakage energy consumption for hard real-time systems. In *Proc. of CASES'04*. Washington, DC, USA, Sept. 2004.

[17] D. Ongaro, A. Cox, and S. Rixner. Scheduling i/o in virtual machine monitors. In *Proc. of ACM SIGPLAN/SIGOPS Intl. Conf. on Virtual Execution Environments*, pages 1–10. Seattle, USA, March 2008.

[18] C. Rusu, A. Ferreira, C. Scordino, A. Watson, R. Melhem, and D. Mosse. Energy-efficient real-time heterogeneous server clusters. In *Proc. of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 418–428. San Jose, USA, April 2006.

[19] P. Scheihing. Creating energy efficient data centers. In *Data Center Facilities and Engineering Conference*. Washington, DC, USA, May 2007.

[20] I. Shin and I. Lee. Compositional real-time scheduling framework with periodic model. *ACM Transactions on Embedded Computing Systems*, 7(3), April 2008.

[21] S. W. Son, et al. Integrated link/cpu voltage scaling for reducing energy consumption of parallel sparse matrix applications. In *Proc. of 20th IEEE Intl. Parallel and Distributed Processing Symposium*. Greece, April 2006.

[22] S. Srikantaiah, A. Kansal, and F. Zhao. Energy aware consolidation for cloud computing. In *Workshop on Power Aware Computing and Systems (HotPower '08)*. San Diego, USA, December 2008.

[23] A. Verma, P. Ahuja, and A. Neogi. pMapper: Power and migration cost aware application placement in virtualized systems. In *Proc. of 9th ACM/IFIP/USENIX Intl. Conf. on Middleware*. Leuven, Belgium, December 2008.

[24] A. Verma, P. Ahuja, and A. Neogi. Power-aware dynamic placement of HPC applications. In *Proc. of ICS'08*, pages 175–184. Agean Sea, Greece, June 2008.

[25] VirtualLogicx Real-Time Virutulalization and VLX. VirtualLogix, http://www.osware.com.

[26] L. Wang and Y. Lu. Efficient power management of heterogeneous soft real-time clusters. In *Proc. of IEEE Real-Time Systems Sym.* Barcelona, Spain, Dec. 2008.

[27] W. Warren, E. Weigle, and W. Feng. High-density computing: A 240-node Beowulf in one cubic meter. In *Proc. of ACM/IEEE Conf. on Supercomputing*. Baltimore, USA, November 2002.

[28] S. Yoo, M. Park, and C. Yoo. A step to support real-time in a virtual machine monitor. In *Proc. of 6th IEEE Consumer Communications and Networking Conference*. Las Vegas, USA, January 2009.