Understanding Concept of DVFS for Real-Time VM on Cloud Computing

Manthan Shah¹  Kruti Shah²  Richa Sinha³

¹,² ¹Masters in Information Technology ³Asst. Professor
³Kalol Institute of Technology & Research Center, Gujarat, India

Abstract--- Reducing power consumption has been an essential requirement for Cloud resource providers not only to decrease operating costs, but also to improve the system reliability. A cloud system uses virtualization technology to provide cloud resources (e.g., CPU, memory) to users in form of Virtual Machines (VM). We have proposed Power-Conscious provisioning of virtual machines policy for user services. In our approach user is asking for the virtual platform to deploy his application on the Cloud system. After receiving request from the user, Resource Broker will compose the request to Data Centre. We have propose scheme to provision the virtual machine which consumes less energy so, it will increase the profit of Data center and at the same time user will have to pay minimum price if two hosts are consuming same power. It uses Dynamic Voltage Frequency Scaling (DVFS) scheme.

Keywords: Power-awareness, Cloud computing, Real-time

I. INTRODUCTION

Cloud Computing has been pointed as a promising approach to improve resources utilization. This is mainly supported by the use of virtualization that allows providers to run multiple workloads from different customers on the same computing infrastructure. According to the National Institute of Standards and Technology (NIST) [2], Cloud Computing is “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction”. However, in order to offer all these characteristics Cloud providers rely on large and power-consuming data centers. In this context, Real-Time Applications (RTAs) require large amount of computing resources to scale user utilization patterns and fulfill time deadlines at the same time. Therefore, energy-efficient Cloud environments need to deal not only with energy consumption but also with increasing demand, and high QoS expectations. Achieving this balance is fundamental for Real-Time Cloud Computing. This will allow providers to carry out applications’ time constrains offering at the same time sustainable services. Initial energy-efficient efforts have been focused on workload scheduling mechanisms to reduce the number of active servers. However, they are mainly focused on host and data center level improvements neglecting fine-grained concerns. These include inefficiencies at VM resource provisioning due to customer overestimations. According to [3], Cloud Computing customers tend to overestimate the amount of required resources to ensure acceptable performance. This causes overall system underutilization and reduces the data center capacity.

Energy efficiency is becoming a very important concern for Cloud Computing environments. These are normally composed of large and power consuming data centers to provide the required elasticity and scalability to their customers. In this context, many efforts have been developed to balance the loads at host level. However, determining how to maximize the resources utilization at Virtual Machine (VM) level still remains as a big challenge. This is mainly driven by very dynamic workload behaviors and a wide variety of customers’ resource utilization patterns. Its impact on the trade-off between energy efficiency and SLA fulfillment is analyzed [30]. The main idea is to exploit the resource utilization patterns of each customer to decrease the waste produced by resource request overestimations. This creates the opportunity to allocate additional VMs in the same host incrementing its energy efficiency. Nevertheless, this also increases the risk of QoS afeetations. The problem is even worse for large-scale compute infrastructures, such as clusters and data centers. It was estimated that in 2006 IT infrastructures in the US consumed about 61 billion kWh for the total electricity cost about 4.5 billion dollars [30]. The estimated energy consumption is more than double from what was consumed by IT in 2000. Moreover, under current efficiency trends the energy consumption tends to double again by 2011, resulting in 7.4 billion dollars annually.

According to [6], Real-Time Systems (RTSs) are those that their correctness does not depend only on the logical result but also on the time in which such results are produced. As mentioned in [7], the use of RTSs is very relevant in daily life processes. Applications have a wide range including gaming applications, financial processes, scientific experimentation, and medical and flight-control systems. In this type of applications messages, completion, or response are always constrained by time. Failing in accomplish this requirement could result in serious implications. Depending on the flexibility of such constraints or deadlines, Real-Time Applications are generally classified in hard, firm and, soft [6]. Hard Real-Time Applications are those where the non-fulfillment of the time constraints leads to system failures. Similarly, Firm Real-Time Applications have hard constraints, but they allow a certain level of tolerance. Finally, in the case of Soft Real-Time Applications the non-fulfillment of deadlines degrades the system’s performance but not destroy it by failure or crash. In order to accomplish these time constraints, RTSs normally demand large amount of computing resources to scale user demands. In this context, Cloud Computing is a model that can offer this scalability. As it is mentioned in [8], the use of virtualization and the resulting decoupling of infrastructure and application management offered by Cloud Computing...
makes possible to rapidly scale on demand infrastructure to meet the resource requirements of real-time services. However, the advent of other critical Cloud Computing targets such as the improvement of energy-efficiency is creating a challenging atmosphere. Cloud providers require not only accomplishing customer expectations, but also improving the resources usage within data centers. The objective is to increase their profits and diminish the environmental impact while QoS is maintained. Achieve this balance is fundamental for Real-Time Cloud Computing. This will allow providers to carry out applications’ time constrains offering at the same time sustainable Cloud Computing services. In Recent year DVFS Schema is used to reduce Power Consumption.

In this Paper Section 2 describe related work for DVFS Schema. Section 3 describes DVFS Schema with energy model and hard-real time algorithm. Section 4 describes Simulation Result for DVFS and Non Power Aware Algorithms. Section 5, 6, 7 describe Conclusion, Future work and References.

II. RELATED WORK FOR DVFS SCHEMA

In recent years there has been a significant amount of work on task scheduling for real time embedded systems using various forms of DVFS enabled techniques. The main idea in most of the existing algorithms is to efficiently use processors’ slack times to satisfy time requirements of all tasks; e.g. deadlines, release times and execution times. Based on provided/estimated information for each task, energy aware task scheduling algorithms in embedded systems can be categorized into two groups: static (offline) and dynamic (online). In static scheduling timing information of all tasks is made available during compile time, scheduling is performed to meet all deadlines while maximizing processor utilization [9], [10], [12], [14], [15]. This type of scheduling is used in most large scale computational problems, such as, bioinformatics [16], chemistry [17] and machine vision applications [18]. In dynamic scheduling, on the other hand, although tasks’ deadlines might be available during compile time, their release and execution times must be estimated during the run time [23], [11], [13], [19]. This class of scheduling is usually used in dynamic large scale approximation and optimization problems such as weather forecasting [20] and search algorithms [21] as well as most power aware devices like laptops, wireless sensors and cell phones. Kappiah et al. in [22] used a just-in-time DVFS technique to fill slack times in MPI programs. A system called Jitter was utilized to reduce working frequency of nodes with more slack times and/or less assigned computation. Jitter ascertains that tasks would arrive just in time without increasing overall execution time. Ge et al. in [23] applied the DVFS technique to processors that do not work at their peak performance during the execution of parallel applications. In this approach, the best processor frequency for each task was selected before its execution based on through analysis of collected computation and communication power profiles. A method to reduce energy consumption was presented in [24] to adaptively activate and deactivate hardware resources (e.g., memory) for intensive HPC applications. Lee and Zomaya in [13] presented a DVFS based algorithm to simultaneously minimize both completion time and energy consumption of precedence constrained (dependent) parallel jobs. Their final result was a trade-off between quality of scheduling and consumption of energy. Ding et al. in [23] formally modeled efficiency/iso efficiency concepts for energy scalability. They also extended their results to produce an analytical model for studying tradeoffs between performance and energy saving in HPC systems. Molson et al. in [24] classified the slack times in real time applications into static, work and shared lack groups for multiple dependent tasks on multiple DVFS enabled processors. Then a dynamic dependency aware task scheduling was proposed to adjust voltage/frequency of the deadlines for tasks assigned to processors. The use of multiple voltages in Dynamic Voltage Scaling enabled processors was used in Ishihara work in [24]. Their work is a simplified version of our work. Kimura et al in [25] proposed an energy reduction algorithm for power scalable high performance cluster supported by DVFS technique. This algorithm selects a suitable set of voltages and frequencies to execute tasks as uniformly as possible using the lowest available frequency with slightly increasing the overall execution time. In our former approach [26], an algorithm was proposed to reclaim slack times of tasks by linear combination of the processor highest and lowest frequencies. To the best of our knowledge, Reference DVFS algorithm (RDVFS) [25], and Maximum Minimum Frequency DVFS (MMF DVFS) [26] are the most efficient algorithms.

III. DVFS SCHEMA

Dynamic voltage frequency scaling (DVFS), already incorporated into many recent processors, is perhaps the most appealing method for reducing energy consumption. DVFS reduces energy consumption of processors based on the fact that such energy consumption in CMOS circuits has a direct relationship with (1) working frequency and (2) the square of the supplied voltage. Thus, DVFS saves energy by switching between processor’s voltages/frequencies to execute tasks during slack times. Although DVFS was originally designed for task scheduling on single processors [27], however, it has recently been extended and used in parallel and distributed computing systems as well [27]. To deploy DVFS, it must be properly integrated with a task scheduler by using one of the following two approaches: (1) during the scheduling process or (2) slack reclamation after scheduling. In the first approach, tasks graph are scheduled on DVFS enabled processors by minimizing both energy and make span at the same time [27]. In the second approach, an independent scheduler is first used to distribute tasks among processors without considering energy consumption. This procedure is then followed by an independent DVFS technique to minimize energy consumption of tasks by filling the generated tasks’ slack times. The existing methods based on DVFS techniques, however, have two major limitations: (1) most of them still focus on the scheduler and rarely explore other opportunities for slack reclamation, and (2) they only use one frequency (among a discrete set of frequencies) to perform each task the use of one frequency usually results in underutilized slack times leading to energy wastage by processors and other devices.
IV. ENERGY MODEL

The main part of power consumption in data centers comes from computation processing, disk storage, network, and cooling systems. This paper focuses on reduction of CPU power consumption using energy-aware VM provisioning in Cloud computing environments.

The most of power consumption in CMOS circuits is composed of dynamic and static power. We only consider the dynamic power consumption, as it is the dominating factor in the total power consumption [28]. Data centers can increase their profit by reducing the dynamic power consumption. The dynamic power consumption by an application is proportional to \( V_{dd}^2 \) and \( f \), where \( V_{dd} \) is the supply voltage and, \( f \) is the frequency [29]. Since the frequency is usually in proportion to the supply voltage, the dynamic power consumption of a processor is defined in Equation (1) [4].

\[
P = C \cdot f^2 \tag{1}
\]

Where \( C \) is a proportional coefficient, Let us consider an application with the execution time \( t \) running at the CPU with the frequency \( f_{\text{max}} \) [5]. If the processor runs at the frequency level \( f \) (\( 0 < f \leq f_{\text{max}} \)), the execution time is defined by \( t / f \leq \frac{t}{f_{\text{max}}} \). Thus, the dynamic power consumption during the task execution is defined by Equation (2) [4].

\[
E = \int_{0}^{t} P \cdot c \cdot f_{\text{max}} \cdot f^2 = \alpha \cdot t \cdot s^2 \tag{2}
\]

Where \( \alpha \) is a coefficient and \( S \) is the relative processor speed for the frequency \( f \) (\( S = f / f_{\text{max}} \)) [4]. The DVFS scheme reduces the dynamic power consumption by decreasing the supplying voltage and frequency, which results in a slowdown of the CPU and increased execution time [5]. We assume that each PE (Processing Element) \( p \) in a datacenter can adjust its processor frequency from \( f_{\text{min}} \) to \( f_{\text{max}} \) continuously. The relative processor speed \( S \) for each frequency \( f \) is defined by \( f / f_{\text{max}} \), where \( f_{\text{min}} / f_{\text{max}} < S \leq 1 \) [5].

V. DVFS-ENABLE 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 [4]. The broker selects the minimum-price virtual ma-chine 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 [5]. Figure 1 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 \( P_{E_k} \) of \( Q_k \) MIPS rate. Suppose that the current running RT-VMs on the processing element \( P_{E_k} \) at time \( t \) is known as \( T_k = \{ V_j(u_j,m_j, d_j), j = 1, \cdots, k \} \) [4]. And the remaining service time of \( V_i \) at time \( t \) is denoted as \( w_i \). Then, the schedulability is guaranteed if it satisfies Equation (3). Since \( w_i/(d_i - t) \) is the minimum MIPS rate for \( V_i \) by its deadline \( d_i \), 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 \( Q_k \) [5].

\[
u_i \cdot m_i + \sum_{j=1}^{n} \frac{w_j}{d_j-t} \leq Q_k \tag{3}
\]

Algorithm Min-Price RT-VM Provisioning (\( V_d \))

1. \( \text{VM} \leftarrow \text{null}; \)
2. \( \text{alloc} \leftarrow -1; \)
3. \( \text{price}_{\text{min}} \leftarrow \text{MAX VALUE}; \)
4. \( \text{price}_{\text{max}} \leftarrow \text{MAX VALUE}; \)
5. for \( k \) from 1 to \( N \) do
6. \( \text{if}(u_i \cdot m_i + \sum_{j=1}^{n} \frac{w_j}{d_j-t} \leq Q_k) \) then
7. \( \text{w}_{\text{estimated}} \leftarrow \text{energy estimate}(P_{E_k}, V_i); \)
8. \( \text{price} \leftarrow \text{price for RT-VM V}_i \text{in P}_{E_k}; \)
9. \( \text{if} \text{price} < \text{price}_{\text{min}} \text{or} \)
10. \( \text{price}_{\text{min}} \leftarrow \text{price}; \)
11. \( \text{price}_{\text{max}} \leftarrow \text{price}; \)
12. \( \text{price} \leftarrow \text{price}_{\text{min}} \text{and} \text{price} \left\langle \text{price}_{\text{max}} \right\rangle \)
13. \( \text{alloc} \leftarrow k; \)
14. \( \text{endif}; \)
15. \( \text{endif}; \)
16. \( \text{endfor}; \)
17. if \( \text{alloc} \neq -1 \) then
18. \( \text{VM} \leftarrow \text{create VM (P}_{E\text{alloc}}, V_i); \)
19. \( \text{endif}; \)
20. return VM.

Fig. 1: Min-Price RT-VM Provisioning [4]

If \( P_{Ek} \) 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 [4]. For the same price, less energy is preferable because it produces higher profit (line 9-14). Finally, a virtual machine is mapped on \( P_{E\text{alloc}} \) if RT-VM \( V_i \) is schedulable on the datacenter [5].

When a user launches the service on the VM, the resource provider provision the VM using DVS schemes to reduce the power consumption [4].

VI. SIMULATION RESULT

The Simulation Result Shows Comparison between Non Power Aware and DVFS Schema, The Results are taken in the Cloudsim Toolkit with different inputs.

<table>
<thead>
<tr>
<th>Sr.no</th>
<th>Number of Host</th>
<th>Number of VM</th>
<th>Energy Consumption in DVS(Kwh)</th>
<th>Energy Consumption in NPA(Kwh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>10</td>
<td>0.10</td>
<td>0.54</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>20</td>
<td>0.24</td>
<td>1.72</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>30</td>
<td>0.47</td>
<td>2.61</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>40</td>
<td>0.63</td>
<td>3.86</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>50</td>
<td>0.78</td>
<td>4.31</td>
</tr>
<tr>
<td>6</td>
<td>60</td>
<td>60</td>
<td>1.01</td>
<td>5.44</td>
</tr>
<tr>
<td>7</td>
<td>70</td>
<td>70</td>
<td>1.16</td>
<td>6.00</td>
</tr>
<tr>
<td>8</td>
<td>80</td>
<td>80</td>
<td>1.32</td>
<td>7.06</td>
</tr>
<tr>
<td>9</td>
<td>90</td>
<td>90</td>
<td>1.49</td>
<td>8.31</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>100</td>
<td>1.64</td>
<td>8.54</td>
</tr>
</tbody>
</table>

Table 1: Simulation Result
VII. CONCLUSION
In this paper, we have Study Clouds are essentially Data Centers hosting application services offered on a subscription basis. They consume high energy to maintain their operations. So, it has high operational cost and adverse environment impact. For the solution of higher energy consumption we have proposed Virtual Machine provisioning scheme based on DVFS. After receiving the VM request, system will always select a node with minimum-price of providing the VM. For same price, the node with less energy consumption will be provided. Proposed energy conscious VM provisioning scheme will significantly reduce energy consumption in light load, while providing low level of SLA violation. It profits the data centers, hence service provider and at the same time user has to pay minimum in case of same energy consumption on service. The simulation results have shown that datacenter can reduce power consumption and increase their profit using DVFS schemes.

REFERENCES


[27] Nikzad Babaii Rizvandi1,2, Javid Taheri1, and Albert Y. Zomaya1, "Some Observations on Optimal Frequency Selection in DVFS–based Energy Consumption Minimization", National ICT Australia (NICTA), Australian Technology Park Sydney, NSW 1430, Australia.

