

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

Manthan Shah¹ Kruti Shah² Richa Sinha³

^{1,2}Masters in Information Technology ³Asst. Professor

^{1,2,3} Kalol 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 affectations. 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 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 DVS 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].

$$P = C \cdot f^3, \quad (1)$$

Where C is a proportional coefficient, Let us consider an application with the execution time t running at the CPU with the frequency f_{max} [5]. If the processor runs at the frequency level f ($0 < f \leq f_{max}$), the execution time is

defined by $t / \frac{f}{f_{max}}$. Thus, the dynamic power consumption during the task execution is defined by Equation (2) [4].

$$E = \int_0^{t/\frac{f}{f_{max}}} P = c \times t \times f_{max} \times f^2 = \alpha \times t \times s^2 \quad (2)$$

Where α is a coefficient and S is the relative processor speed for the frequency f ($S = f / f_{max}$) [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_p^{min} to f_p^{max} continuously. The relative processor speed S for each frequency f is defined by f / f_{max} , where $f_p^{min} / f_p^{max} < S \leq 1$ [5].

V. B.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 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 [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 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 $T_k = \{V_j(u_j, m_j, d_j) | j = 1, \dots, n_k\}$ [4]. 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 (3). 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 [5].

$$u_i \times m_i + \sum_{j=1}^{n_k} \frac{W_j}{d_{j-t}} \leq Q_k \quad (3)$$

Algorithm Min-Price RT-VM Provisioning (V_i)

```

1: VM ← null;
2: alloc ← -1;
3:  $e_{min}$  ← MAX VALUE;
4:  $price_{min}$  ← 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:    $e_k$  ← energy_estimae( $PE_k, V_i$ );
8:    $price_k$  ← price for HRT-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:      $price_{min}$  ←  $price_k$ ;
12:      $e_{min}$  ←  $e_k$ ;
13:     alloc ← k;
14:   endif
15: endif
16: endfor
17: if alloc ≠ -1 then
18:   VM ← create_VM (PEalloc,  $V_i$ );
19: endif
20: return VM;

```

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

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 [4]. 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 [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.

Sr.no	Number of Host	Number of VM	Energy Consumption in DVFS(Kwh)	Energy Consumption in NPA(Kwh)
1	10	10	0.10	0.84
2	20	20	0.24	1.72
3	30	30	0.47	2.61
4	40	40	0.63	3.46
5	50	50	0.78	4.31
6	60	60	1.01	5.44
7	70	70	1.16	6.00
8	80	80	1.32	7.06
9	90	90	1.49	8.31
10	100	100	1.64	8.54

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

- [1] Rajkumar Buyya, Rajiv Ranjan and Rodrigo N. Calheiros, "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities.
- [2] D. Amrhein, et al., "Cloud Computing Use Case," Cloud Computing Use Case Discussion Group, White paper, 2010.
- [3] B. Newton and H. VanHook, "Cloud Cover Delivering on the Value of the Cloud in Public Sector IT Organizations," BMC Software, White Paper 129978, 2010.
- [4] Kyong Hoon Kim, R. Buyya, A. Beloglazov, Power-aware Provisioning of Cloud Resources for Real-time Services, ACM 978-1-60558-847-6/09/11.
- [5] Kyong Hoon Kim, R. Buyya, A. Beloglazov, Power-aware Provisioning of Cloud Resources for Real-time Services,, 2011 John Wiley & Sons, Ltd.
- [6] J. A. Stankovic, "Misconceptions about real-time computing: a serious problem for next-generation systems," Computer, vol. 21, pp.10-19, 1988.
- [7] K. G. Shin and P. Ramanathan, "Real-time computing: a new discipline of computer science and engineering," Proceedings of the IEEE, vol. 82, pp. 6-24, 1994.
- [8] V. Sarathy, et al., "Next Generation Cloud Computing Architecture: Enabling Real-Time Dynamism for Shared Distributed Physical Infrastructure," presented at the Proceedings of the 2010 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, 2010.
- [9] J. Zhuo and C. Chakrabarti, "Energy-efficient dynamic task scheduling algorithms for DVS systems," ACM Trans. Embed. Comput. Syst., vol. 7, pp. 1-25, 2008.
- [10] C. M. Krishna and Y.-H. Lee, "Voltage-Clock-Scaling Adaptive Scheduling Techniques for Low Power in Hard Real-Time Systems," IEEE Trans. Comput., vol. 52, pp. 1586-1593, 2003.
- [11] D. C. Snowdon, et al., "Koala: a platform for OS-level power management," presented at the Proceedings of the 4th ACM European conference on Computer systems, Nuremberg, Germany, 2009.
- [12] R. Xiaojun, et al., "An Energy-Efficient Scheduling Algorithm Using Dynamic Voltage Scaling for Parallel Applications on Clusters," in Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on, 2007, pp. 735-740.
- [13] Y. C. Lee and A. Y. Zomaya, "Minimizing Energy Consumption for Precedence-Constrained Applications Using Dynamic Voltage Scaling," presented at the Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid), 2009.
- [14] D. Zhu, et al., "Scheduling with Dynamic Voltage/Speed Adjustment Using Slack Reclamation in Multi-Processor Real-Time Systems," presented at the Proceedings of the 22nd IEEE Real-Time Systems Symposium, 2001.
- [15] P. d. Langen and B. Juurlink, "Trade-Offs Between Voltage Scaling and Processor Shutdown for Low-Energy Embedded Multiprocessors," presented at the Embedded Computer Systems: Architectures, Modeling, and Simulation, 2007.
- [16] T. K. Yap, et al., "Parallel Computation in Biological Sequence Analysis," IEEE Trans. Parallel Distrib. Syst., vol. 9, pp. 283-294, 1998.
- [17] M. F. Guest, et al., "High-performance computing in chemistry: NW Chem," Future Gener. Comput. Syst., vol. 12, pp. 273-289, 1996.
- [18] T. Simunic, et al., "Dynamic voltage scaling and power management for portable systems," presented at the Proceedings of the 38th annual Design Automation Conference, Las Vegas, Nevada, United States, 2001.
- [19] P. Bougeault, "High Performance Computing and the Progress of Weather and Climate Forecasting," presented at the 8th International Conference on High Performance Computing for Computational Science (VECPAR 2008), Toulouse, France, June 24-27, 2008.
- [20] P. Lu, et al., "An Effective Iterative Compilation Search Algorithm for High Performance Computing Applications," presented at the Proceedings of the 2008 10th IEEE International Conference on High Performance Computing and Communications, 2008.
- [21] J.-J. Chen, et al., "Energy-Efficient Real-Time Task Scheduling in Multiprocessor DVS Systems," presented at the Proceedings of the 2007 Asia and South Pacific Design Automation Conference, 2007.
- [22] R. Springer, et al., "Minimizing execution time in MPI programs on an energy-constrained, power-scalable cluster," presented at the Proceedings of the eleventh ACM SIGPLAN symposium on Principles and practice of parallel programming, New York, New York, USA, 2006.
- [23] R. Ge, et al., "Performance-constrained Distributed DVS Scheduling for Scientific Applications on Power-aware Clusters," presented at the Proceedings

- of the 2005 ACM/IEEE conference on Supercomputing, 2005.
- [24] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," presented at the Proceedings of the 1998 international symposium on Low power electronics and design, Monterey, California, United States, 1998.
- [25] H. Kimura, et al., "Emprical study on Reducing Energy of Parallel Programs using Slack Reclamation by DVFS in a Power-scalable High Performance Cluster," in Cluster Computing, 2006 IEEE International Conference on, 2006, pp. 1-10.
- [26] N. B. Rizvandi, et al., "Linear Combinations of DVFS-enabled Processor Frequencies to Modify the Energy-Aware Scheduling Algorithms," presented at the Proceedings of the 2010 10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), Melbourne, Australia, May 17-20, 2010.
- [27] Nikzad Babaii Rizvandi^{1,2}, 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.
- [28] 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.
- [29] 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.
- [30] A. Beloglazov, R. Buyya, Y. Lee, A. Zomaya, A Taxonomy and Survey of Energy Efficient Data Centers and Cloud Computing Systems, Advances in Computers, Volume 82, 47-111pp, M. Zelkowitz (editor), Elsevier, Amsterdam, The Netherlands, March 2011.