© 2017 IJSRST | Volume 3 | Issue 6 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

Energy-Efficient Provisioning of Virtual Machines for Real-Time Applications of Cloud Computing

Manju Lata
Shri Venkateshwara University, NH-24, Rajabpur Gajraula, Distt:- J.P. Nagar, Uttar Pradesh, India

ABSTRACT

Cloud Computing has emerged as a most prominent computing paradigm for a large number of computing applications. With the growing number of Cloud data centers, the energy efficiency of the environment becomes a prime concern. This is particularly true, when the computation is required in the real time. The cloud computing environments has been presented in the present work for the real time computing applications. The energy efficient provisioning of the cloud virtual machines have been discussed considering the different required parameters. Both the cases of the Soft and Hard real time provisioning have been presented with the parameters and algorithms.
Keywords: Cloud Computing, Real time Services, Energy Efficiency

I. INTRODUCTION

With the growth of high speed networks over the last decades, there is an alarming rise in its usage comprised of thousand of concurrent e-commerce transactions and millions of Web queries a day. This ever increasing demand is handled through large scale data centers, which consolidate hundreds and thousands of servers with other infrastructure such as cooling, storage and network systems. Many Internet companies such as Google, Amazon, eBay, and Yahoo are operating such huge datacenters around the word. The commercialization of these developments is defined currently as cloud computing [1] where computing is delivered as utility on a pay-as-you-go manner. Traditionally, business organizations used to invest huge amount of capital and time in acquisition and maintenance of computational resources. The emergence of cloud computing is rapidly changing this ownership-based approach to subscription-oriented approach by providing access to scalable infrastructure and services on-demand. As modern real time services become available through cloud computing. Cloud computing is an evolving paradigm which is enabling outsourcing of all IT needs such as storage, computation and software such as office and ERP, Internet based access. The shift toward such service-oriented computing is driven primarily by ease of management and administration process involving software upgrades and bug fixes. It also allows fast application development and testing for small IT companies that cannot afford large investments on infrastructure. Most important advantage offered by clouds is in term of economies of scale that is when thousands of users share same facility, cost per user and the server utilization. To enable such facilities, cloud computing encompasses many technologies and concepts such as virtualization, utility computing, pay as you go, no capital investment, elasticity, scalability, provisioning on demand, and IT outsourcing. The literary meaning of cloud computing can be computing achieved using collection of networked resources, which are offered on subscription. Some of the definitions given by many well known scientists and organizations include:

Buyya et al. [1], defined the cloud computing in terms of its utility to end user: "A cloud is a market oriented distributed computing system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers."

National Institute of Standards and Technology (NIST) [2] defined, "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., network,

servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models." The characteristics of clouds include on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. The available service models are classified as SaaS (Software-as-a-service), PaaS (Platform-as-a-Service), and IaaS (Infrastructure-as-a-Service). The deployment model is categorized into public, private, community, and hybrid clouds.

Cloud computing is a highly scalable and cost-effective infrastructure for running High Performance Computing (HPC), enterprise and Web applications. However, the growing demand of cloud infrastructure has drastically increased the energy consumption of data centers, which has become a critical issue. Traditionally, enterprises purchased their own computing infrastructures to run real time application, such as financial analysis, distributed data processing, real-time database, etc. This approach provides performance guarantees, as the hardware is not shared and its capacity is provisioned to meet the needs of specific applications. However, owning an infrastructure incurs high operating costs and leads to the necessity to deal with the maintenance and upgrade of software and hardware. Moreover, this approach does not allow the infrastructure capacity to scale in and out depending on the current resource demand created by the applications. Cloud computing addresses these problems by providing dynamic resource provisioning on a pay-as-you-go basis. And provide easy provisioning configuration in the device into which provisioning is to be done and supports multiple modes like real-time, scheduled and on-demand for processing the provisioning requests. Cloud computing provides a web-based interface for operational and management functions such as real-time monitoring and scheduling also deals with error prevention & detection, and all time availability. Any real-time system will comprise different types of activities those that can be scheduled those that cannot be scheduled, such as operating-system facilities, interrupt handlers and non real-time activities. If non-schedulable activities can execute in preference to schedulable activities, they will affect the ability of the system to handle time constraints. A non real-time is one that cannot guarantee a response time in any situation, even if a fast response is the usual result.

II. RELATED WORK

Provisioning of virtual machine for real-time cloud computing applications has been a recent issue in data centers. However, recent research has focused on reducing energy consumption in cluster systems. The objective of energy efficient computing is to improve power management and consumption using power aware ability of system devices, such as processors, disks, and communication links. There are two main reasons for need of energy efficient computing in cluster systems: operational cost and system reliability. One dominating factor in the operational cost of data centers comes from electricity cost consumed by server systems. As the number of managed servers increases, data centers consume from 10 to 100 times more power per square foot than typical office buildings [3]. They can consume as much electricity as a city [4]. Traditionally, energy consumption in data centers comes from computation processing, disk storage, network, and cooling systems. Lowering the power usage of data centers becomes a challenging issue as computing application and data are growing so quickly that increasingly larger servers and disks are needed to process them within the required time [5]. Thus, data center resources need to be managed in an energy-efficient manner to drive Green cloud computing [5].

In cloud computing, application performance depends on the application's ability to simultaneously access multiple types of resources [6]. As CPU, memory and I/O bandwidth are the building blocks of a VM's capacity. Virtual Machine provisioning involves instantiation of one or more Virtual Machines (VMs) that match the specific hardware characteristics and software requirements of an application. Most Cloud providers offer a set of general-purpose VM classes with generic software and resource configurations. For example Amazon EC2 supports 11 types of VMs, each one with different options of processors, memory, and I/O performance. VM Provisioning is to provide applications with sufficient computational power, memory, storage, and I/O performance to meet the level of QoS expected by end-users. The latter is achieved

either by increasing/decreasing capacity of deployed virtual machines or by increasing/decreasing the number of application and VM instances.

Many recent studies have been conducted to provide emerged cloud computing paradigm. To leverages virtualization of computer resources and allows achieving more efficient allocation of workload in terms of higher resource utilization as well as decreased energy consumption [7]. Kusic et al. [8] have targeted on minimizing both energy consumption and SLA violations for online services on virtualized data centers using a limited look-ahead control. The pMapper (Power and Migration Cost Aware Application Placement in Virtualized System) architecture has been also proposed in Verma et al. [9, 10] to solve the same 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. Rusu et al. [11] have developed a QoS-aware power management scheme is presented by combining cluster-wide (On/Off) and local Dynamic Voltage Frequency Scaling (DVFS) power management techniques in the context of heterogeneous cluster. The front-end manager decides which servers are turned on or off for a given system load, while local manager reduces power consumption using DVFS scheme. Wang et al. [12] have proposed a threshold-based method is proposed for efficient power management of heterogeneous soft real time cluster as well as the mathematical analysis of determining the threshold. Kim et al. [13] have investigated power-aware algorithms for scheduling of real-time bag-of-tasks applications with deadline constraints on homogeneous cluster. Substantial amount of work have been done in the area of power-efficient computing. Kim et al. [14] have proposed a framework for provisioning cloud resources for hard real-time services, in order to support general real-time services by considering the soft-real time model. This work investigates the problem of energy efficient provisioning of virtual machine for both hard and soft real-time cloud applications.

III. REAL TIME SYSTEMS

A real-time system is one in which the correctness of the computations not only depends upon the logical correctness of the computation but also upon the time at which the result is produced. If the timing constraints of the system are not met, system failure is said to have occurred. A real-time system is to have a physical effect within a chosen time-frame. Typically, a real-time system consists of a controlling system (computer) and a controlled system (environment). The controlling system interacts with its environment based on information available about the environment. In a real-time system, there may be unexpected or irregular events and these must also receive a response. In all cases, there will be a time bound within which the response should be delivered. The ability of the computer to meet these demands depends on its capacity to perform the necessary computations in the given time.

A real-time system has some timing properties. These timing properties should be considered when scheduling tasks on a real-time system. The timing properties of the real-time system have been described by a number of researchers [15-18]. Real-time systems span a broad spectrum of complexity from very simple microcontrollers to highly sophisticated, complex and distributed systems. Some examples of real-time systems include process control systems, flight control systems, flexible manufacturing applications, robotics, intelligent highway systems, and high speed and multimedia communication systems [19, 20, 17, and 15]. A real-time system will usually have to meet many demands within a limited time. The importance of the demands may vary with their nature (e.g. a safety-related demand may be more important than a simple data-logging demand) or with the time available for a response. Thus, the allocation of the system resources needs to be planned so that all demands are met by the time of their respective deadlines. This is usually done using a scheduler which implements a scheduling policy that determines how the resources of the system are allocated to the demands. Scheduling policies can be analyzed mathematically so the precision of the formal specification and program development stages can be complemented by a mathematical timing analysis of the program properties [20, 15, 18].

Real-Time Service: – In real-time systems, DVFS technique is used in order to save energy consumption as

well as to meet the task deadline. All service such as financial analysis, distributed databases, or image processing, consists of multiple real-time applications or subtasks. All services accomplished the quality of service agreed with users. The group of subtasks of a real-time service is developed and launched on a specific run-time platform including middleware's, and operating systems, and so on. 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. A real-time service is defined by $\{\tau_i (r_i, w_i, d_i, p_i, f_i) | i = 1, \ldots, n\}$, where n is the number of subtasks. Each real-time subtask τ_i is defined by the following parameters.

r_i : release time
w_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 w_i. In order to accomplish the application's objective, it should be completed by the time $r_i + d_i$ after release a job of w_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. And consider duration and finish time, f_i, since a user cannot access a cloud computing resource for-ever, although a periodic real-time task in embedded system assumes an infinite sequence. Many real-time applications involve combined scheduling of hard and soft real-time tasks. Hard real-time tasks have critical deadlines that are to be met in all working scenarios to avoid catastrophic consequences. In contrast, soft real-time tasks (e.g., multimedia tasks) are those where deadlines are less critical such that missing the deadlines occasionally has minimal effect on the performance of the system. It is possible to allocate separate resources for each of hard and soft tasks. However, sharing the available resources among both types of tasks would have enormous economical and functional impact.

Therefore, it is necessary to support combined scheduling of hard and soft real-time tasks in such systems, in which multiprocessors are increasingly being used to handle the compute intensive applications. Real-time tasks besides being hard or soft can be periodic or aperiodic, sporadic. Described by a number of researcher [20-24] Periodic real-time tasks are activated (released) regularly at fixed rates (periods). A majority of sensory processing is periodic in nature. For example, radar that tracks flights produces data at a fixed rate. Aperiodic real-time tasks are activated irregularly at some unknown and possibly unbounded rate. Sporadic real-time tasks are activated irregularly with some known bounded rate. The bounded rate is characterized by a minimum inter-arrival period that is a minimum interval of time between two successive activations. The time constraint is usually a deadline defined by [20, 22]. Many researchers have investigated mechanisms to support combined scheduling of hard and soft real-time tasks. A number of scheduling approaches were adopted to handle a mix of hard and soft real-time tasks in uniprocessor systems. Examples are, Constant Bandwidth Server (CBS) [25] and [26] Total Bandwidth Server (TBS) [27] and [25] Constant Utilization Server [28] Deferrable Server (DS) [29] Priority Exchange (PE) [29] Extended Priority Exchange (EPE) [30] Dual priority Scheduling [31] and Dynamic Slack Stealing (DSS) [32].

As in real-time cloud service cloud resource brokers take the role of finding VMs for real time services requested by users. Thus, a user requests VMs by either HRT-VM (Hard Real-Time Virtual Machine) or SRT-VM (Soft Real-Time Virtual Machine) depending on deadline types. And define 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

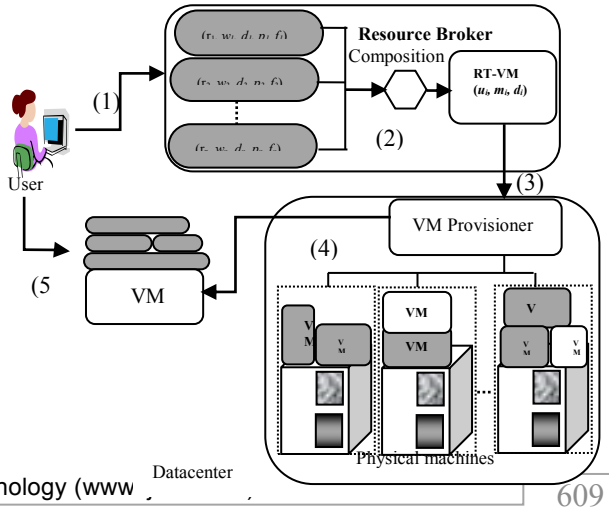

Figure 1: Real Time Cloud Services

The service is developed and launched on a specific platform or infrastructure (e.g. 1GHz-Linux machine). There select the MIPS rate, m*i*, for the specification of the base VM. For the given set of the real-time applications, and 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. Thus, there 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 provisional in clouds map virtual machines for the service, not for individual applications. Furthermore, recent work on implementing real-time virtual machines [33] assures real-time services (e.g. real-time CPU allocation, real-time I/O) of a virtual machine. As shown in Figure 1. There are some steps taken by the user to execute a real-time service.

- ✓ *Requesting a virtual platform:* A user who wants to launch a real-time service submits all the information about the real-time applications to the broker.
- ✓ *Generating a RT-VM from real-time applications*: The resource broker first analyzes the submitted real-time applications and generates one RT-VM request $V_i = (u_i, m_i, d_i)$.
- ✓ *Requesting a real-time VM*: The broker request a VM for RT-VM V_i from the VM provisional of a cloud computing environment.
- ✓ *Mapping physical processors*: The VM provisional finds appropriate processing elements that meet the V_i requirements and provides the VM to the user.
- ✓ *Executing the real-time applications*: The user launches and executes the real-time applications using the provided VM.

Soft Vs Hard Real Time PROVisioning

Hard real-time means strict about adherence to each task deadline. When an event occurs, it should be serviced within the predictable time at all times in a given hard real-time system. A hard real-time is one, which has predictable performance with no deadline miss. Even in case of sporadic tasks (sudden bursts of occurrence of events requiring attention). Automobile engine control system and anti lock brake are the example of hard real time systems. Hard real-time system design provisioning of asynchronous IOs, and provisioning of locks or spin locks. Hard real-time system design to response in all the time slots for the given events in the system, and thus providing the guaranteed task deadlines even in case of sporadic and aperiodic tasks. Sporadic task means tasks executed on the sudden bursts of the corresponding events at high rates. And aperodic tasks mean task having no definite period of event occurrence. Some examples of hard real-time system are video transmission, each picture frame and audio must be transferred at fixed rate. Soft real-time is one in which deadlines are mostly met. Soft real-time means that only the precedence and sequence of the task-operations are defined interrupt latencies and context switching latencies of the tasks and observed time constraints and a few deadline misses are accepted. Mobile phone, digital cameras and orchestra playing robots are examples of soft real-time systems.

A soft real-time service is one that is not the hard real-time special case, and is thus the general case. The sequencing timeliness optimality factor may be anything. Very common examples (outside the traditional real-time computing community) of sequencing timeliness criteria in terms of both factors are minimize the expected number of missed deadlines and minimize the mean total tardiness. And ensures that hard real time deadlines are met and that deadline tardiness is bounded for soft real-time tasks. In the sequencing timeliness criterion 2-dimensional space of optimality and predictability of optimality, soft real-time is the entire space except for the hard real-time corner point. Some soft real-time services are non-stochastically non-deterministic. Hard real-time must have at least some actions with hard deadlines. But the converse is not true soft real-time service may have actions with hard deadlines. Those services are soft real-

time because they do not employ the hard real-time sequencing timeliness criterion. They employ soft real-time ones such as minimize the expected number of missed deadlines regardless of whether the deadlines are hard or soft. Hard real-time and soft real-time apply only to a system because these definitions are based on sequencing. In this sense, a system is any resource management facility that includes sequencing of thread anywhere from the hardware up through the OS and middleware to the application.

Scheduling in Real-Time Service: Most real-time services are designed as concurrent processing systems rather than as monolithic control systems. The concurrency allows for the system to react to events more easily. The scheduling of concurrent activities (tasks or threads) is critical to achieving real-time constraints. Different scheduling approaches are available for different types of real-time services hard versus soft real-time, periodic, aperiodic or sporadic tasks. Most scheduling algorithms aim to meet deadlines associated with tasks while optimizing the use of resources. Thus, the goal of a hard real-time service is to ensure that all deadlines are met, but for soft real-time services the goal becomes meeting a certain subset of deadlines in order to optimize some application specific criteria. The particular criteria optimized depends on the application, but some typical examples include maximizing the number of deadlines met, minimizing the lateness of tasks and maximizing the number of high priority tasks meeting these deadlines.

In addition many studies have focused on the penalty function with deadline. Regarding the deadline model, real-time services can be categorized into two types: hard and soft. In the hard deadline model, a service provider receives some penalty if a service does not meet the deadline. On the other hand, a service with a soft deadline provides a diminished value or utility even when the execution time exceeds the deadline. Kyong Hoon Kim et al. [34] defined a penalty function such as linear decreasing function, is used in the soft deadline model to define the value decrease. The requirement for a VM providing a soft real-time service includes an additional parameter, called the penalty function.

The penalty function indicates the diminished value of a service by executing a VM that has missed the deadline. Figure 2, shows the examples of soft real-time service models with various penalty functions. If a real-time service meets its deadline, it provides the pre-determined value. However, when it misses the deadline, the value or utility of the service decreases according to its penalty function.

$$value = \begin{cases} 1 & if\ delay \le 0 \\ \varphi_i(delay) & if\ delay > 0 \end{cases} \quad (1)$$

SRT-VM V_i for a soft real-time service is define by (u_i, m_i, d_i, φ_i), where φ_i denotes the penalty function of the soft real-time service model. If VM V_i is provide with $u_i \times m_i$ of processing capacity by the deadline d_i, the service quality is guaranteed. However, if the processing capacity of $u_i \times m_i$ requires the time beyond the deadline, the service quality is defined by the penalty function φ_i. And assume that V_i is provided with the capacity past delay time units. Then, as shown in Figure 3, the service quality or value is given by Equation (1).

When a soft real-time service misses its deadline, it gives a diminished value of the service to the user. In this case, the user does not need to pay the whole price for the cloud resources because the resource provider has not met the QoS requirements. And there assumption that a refund due to a service delay is in proportion to the diminished value of the service. The profit of the cloud resource provider for providing a soft real-time service i with consideration of a refund is defined by Equation (2).

$$\begin{aligned} Profit_i &= price_i \times value_i - cost_i \\ &= price_i \times (1 - \varphi_i(delay)) - cost_i \end{aligned} \quad (2)$$

Where $value_i$ is the service quality of the soft real-time service depending on the finish time, as shown in Figure 3, and $cost_i$ is the total cost including the power consumption. Since the soft real-time service gives the value beyond the deadline, and the provisioning of a soft real-time may produce more profit in case of delayed service execution. And Yuichi Asahiro et al. [35] consider the problem of finding an orientation that minimizes $\sum_{v \in V} c_v$

Where, c_v is a penalty incurred for v's violating the degree constraint.

As penalty functions sever classes of penalty functions can be considered, e.g., linear functions, convex functions, and concave functions. The nature of Minimum Penalty Degree Constrained Orientation (MPDCO) depends on the penalty functions. Penalty functions are linearly separable. It can be written as

$$p(c(\Delta)) = \sum_{u \varepsilon v} g(c_i(\Delta))$$

Where, g is non-negative variable function with $g(0) = 0$, such as the example of following type of functions of g: convex (include linear) and concave (including step) functions.

In Figure 4, HRT-VM (Hard Real-Time Virtual Machine) is a requirement for a VM providing a hard real time service. HRT-VM V_i for a hard real-time service is also described by all these three parameters: u_i, m_i, and d_i. When a data center receives a HRT-VM request from a resource broker, it returns the price of providing the HRT-VM service if it can provide real-time VMs for the request. The broker selects the minimally priced VM among available data centers. Thus, the provisioning policy is to select the processing element with the minimum price for the sake of users, and provisioning a VM for a HRT-VM request. For a given HRT-VM V_i (u_i, m_i, d_i), the data center check the schedulability of V_i on the processing element PE_k of Q_k MIPS rate. Suppose that the current running HRT-VMs on the processing element PE_k at time t. And the remaining service time V_j at the time t is denoted as w_j. Then, the schedulability is guaranteed. Since $w_j / (d_j - t)$ is the minimum MIPS rate for V_j by its deadline d_j, Its means that the required MIPS rate including the new HRT-VM V_i is less the processor capacity Q_k. If PE_k is able to schedule V_i, it estimate the energy and price of provisioning. Science the provisioning policy to lower price to users, then finds the minimally priced processor. For the same price, less energy is preferable because it produces higher profit. Finally, a VM is mapped on PE_{alloc} if HRT-VM V_i is schedulable in the data center. When the user launches the service on the VM, the resource provider provisions the VM using DVFS schemes to reduce energy consumption.

When an SRT-VM request is received, a data center provides for the resource broker an appropriate VM. Similarly to the HRT-VM provisioning in Figure 4, a data center finds for the user the minimally price resource. The difference from the HRT-VM provisioning is the acceptance test of a VM. In Figure 4, a new VM request is accepted if all VMs on a PE including the new one meet their deadlines. However, the SRT-VM provisioning acceptance results in higher profit. For example, there are two VM requests V_1 and V_2 accepted at the time 0 on a PE. When a new VM V_3 requiring 800 MIPS rate during 3 time units arrives at the PE, the HRT-VM provisioning algorithm rejects the request since it cannot meet the deadline. On the contrary, SRT-VM can accept V_3 if the acceptance produces higher profit. Therefore, the provisioning of SRT-VM should consider the profit in the acceptance test. Figure 5, shows for the SRT-VM provisioning. The left side of Figure 5, describes the minimally priced VM allocation, in which it calculates the profit by calling the function *Calculate_Profit* (). The function *Calculate_Profit* () is shown in the right side of Figure 5, for a given set of VM-request at the time t. If the number of VMs in T is n_k, there may be n_k different finish times. The function calculates the profit for each finish time by adding the VM profit minus the cost. The function *Calculate_Profit*() implements this by calling the function recursively without deleting the earliest finishing VM, and adding the return profit. The time complexity of the profit calculation is $O(n_k^2)$, where n_k is the number of VMs on a PE.

IV. CONCLUSION

This paper proposed the real-time cloud applications, where each real-time service request is modeled as RT-VM in resource brokers. The soft and hard real-time cloud services, for hard real-time services are provided several schemes. HRT-VM provisioning produced higher profit with lower power consumption regardless of the system load. For soft real-time services, analyzed profitable VM provisioning and proposed the provisioning algorithm. This work includes further analysis and improvement of the proposed adaptive schemes, and the soft real-time VM provisioning with the consideration of various penalty functions.

V. REFERENCES

[1] Buyya, R., Yeo, C.S. and Venugopal, S. 2008. "Market-oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities." In Proceedings of the 10th IEEE International conference on High Performance Computing and Communications, Los Alamitios, CA, USA, PP. 1-9.

[2] Mell, P. and Grance, T. 2009. "The NIST Definition of Cloud Computing", National In'stitute of Standards and Technology, Version15, 10-7-09.

[3] Scheihing P. "Creating Energy Efficient data center." In Data Center Facilities and Engineering Conference. Washington, DC, USA, May 2007.

[4] Markoff J. Lohr S. "Intel's huge bet turns iffy". New York Times Technology Section, September 29, 2002.http://searchstorage.teachtarget.com.au/articles/28102-Predictions-2-9-Symantec-s-Craig-Scroggie

[5] Buyya R., Beloglazov A, Abawajy J. "Energy-Efficient Management of Data Center Resources for Cloud Computing: a vision, architectural elements, and open challenges." In proceedings of the 2010 International Conference of Parallel and Distributed Processing Techniques and Applications (PDPTA 2010). Las Vegas, USA, July 12-15, 2010, 12 Page; ar Xiv: 1006.0308.

[6] P. Padala, K.-Y. Hou, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, and A. Merchant, " Automated control of multiple virtualized resources." In EuroSys, 2009, PP. 13-26, ISBN: 978-1-60558-482-9.

[7] Beloglazov A, Buyya R., Lee YC, Zomaya A. "A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems." Advances in Computers, M. Zelkowitz (editor), Vol. 82, PP. 47-111, ISBN 13: 978-0-12-012141-0, Elsevier, 2011.

[8] Kusic D, Kephart JO, Hanson JE, Kandasamy N, Jiang G. "Power and Performance Management of Virtualized Computing Environments via Lookahead Control." In Proceedings of the 5th IEEE International Conference on Autonomic Computing (ICAC 2008), Chicago, USA, Cluster Computing 12(1), PP. 1-15.

[9] Verma A, Ahuja P, Neogi A. "Power-Aware Dynamic Placement of HPC Applications", In Proceedings of the 22nd ACM International Conference on Supercomputing (ICS 2008). Agean Sea, Greece, ICS'08, June 7–12, 2008,PP. 175-184, Island of Kos, Aegean Sea, Greece. Copyright 2008 ACM 978-1-60558-158-3/08/06.

[10] Verma A, Ahuja P, Neogi A. "pMapper: Power and Migration Cost Aware Application Placement in Virtualized System." In Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware. Leuven, Belgium, December 2008, Vol. 5346, 2008, PP. 243-264, ISBN: 3-540-89855-7.

[11] Rusu C, Ferreira A, Scordino C, Waston A, Melhem R, Mosse D. "Energy-Efficient Real-Time Heterogeneous Server Clusters. " In Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium. San Jose, USA, April 2006, PP. 418-428.

[12] Wang L, Lu Y. "Efficient Power Management of Heterogeneous Soft Real-Time Clusters." In Proceedings of the 29th IEEE Real-Time Systems Symposium. Barcelona, Spain, December 2008.

[13] Kim KH, Buyya R, Kim J. "Power-Aware Scheduling of Bag-of-Tasks applications with Deadline Constraints on DVS-enabled Clusters." In Proceedings of the 7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid' 07). Rio de Janeiro, Brazil, May 2007, PP. 1-8.

[14] Kim KH, Beloglazov A, Buyya R. "Power-Aware Provisioning of Cloud Resources for Real-Time Services." In Proceedings of the 7th International Workshop on Middleware for Grids, Clouds and e-Science (MGC 2009). Urbana Champaign, USA, December 2009, PP. 1-6, MGC'09, 30 November - 1 December 2009, Urbanan-Champaign, Illinois, US. Copyright 2009 ACM 978-1-60558-847-6/09/11.

[15] M. Joseph, "Real-time Systems: Specification, Verification and Analysis," Prentice Hall, 1996, PP. 1-278, ISBN 0-13-455297-0.

[16] P. A. Laplante, "Real-time Systems Design and Analysis, An Engineer Handbook," IEEE Computer Society, IEEE Press, 1993, Science, Page 339.

[17] W. Fornaciari and P. di Milano, "Real Time Operating Systems Scheduling Lecturer," www.elet elet.polimi polimi.it/ fornacia it/ fornacia.

[18] J. A. Stankovic and K. Ramamritham, "Tutorial on Hard Real-Time Systems," IEEE Computer Society Press, 1988, Vol. 819, Computers, Page 618, ISBN: 0818608196, 80818608193

[19] J. W. de Bakker, C. Huizing, W. P. de Roever, and G. Rozenberg, "Real-Time: Thory in Practice," Preceedings of REX Workshop, Mook, The

Netherlands, Springer-Verlag company, June 3-7, 1991, ISBN-10:3540555641, 732 Pages.

[20] G. C. Buttazzo, "Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications," Springer, 13-15 September, 2006, PP. 457-466.

[21] H. Kaneko, J.A. Stankovic, S. Sen, and K. Ramamritham. "Integrated Scheduling of Multimedia and Hard Real-Time Tasks." Proceedings of Real-Time Systems Symposium, PP. 206-217, 1996.

[22] W. A. Halang and A. D. Stoyenko, "Real Time Computing," NATO ASI Series, Series F: Computer and Systems Sciences, Springer-Verlag, Vol. 127, 1994, PP. 283-307.

[23] D. Isovic and G. Fohler, "Efficient Scheduling of Sporadic, Aperiodic and Periodic Tasks with Complex Constraints," Proceedings of the 21st IEEE Real-Time Systems Symposium, PP. 207-216, Florida, USA, November 27-30, 2000.

[24] C. M. Krishna and K. G. Shin, "Real-Time Systems," MIT Press and McGraw-Hill Company, 1997, ISBN: 0-07-057043.

[25] M. Spuri and G.C. Buttazzo. "Scheduling Aperiodic Tasks in Dynamic Priority Systems." Journal of Real-Time Systems, Vol. 10, No. 2, 1996, PP. 31-32.

[26] T.M. Ghazalie and T. Baker. "Aperiodic Servers in a Deadline Scheduling Environment." Real-Time Systems, Vol. 9, July 1995, Issue 1, PP. 31-67, © 1995 Kluwer Academic Publisher, Boston, Manufactured in the Netherlands.

[27] M. Spuri and G.C. Buttazzo. "Efficient Aperiodic Service under Earliest Deadline Scheduling." Proceedings of 15th IEEE Real-Time Systems Symposium ,San Juan, Puerto Rico, December 1994.

[28] Z. Deng, J.W. S. Liu, and J. Sun. "A Scheme for Scheduling Hard Real-Time Applications in Open System Environment." In Proceedings of the 9th Euromico Workshop on Real-Time Systems, June 1997, PP. 191-199, Toledo, Spain, IEEE.

[29] J.P. Lehoczky, L. Sha, and J.k. Strosnider. "Enhanced Aperiodic Responsiveness in Hard Real-Time Environments." Proceedings of Real-Time Systems Symposium, PP. 261-270, 1987.

[30] B.Sprunt, J.Lehoczky, and L. Sha. "Exploiting Unused Periodic Time for Aperiodic Service Using the Extended Priority Exchange Algorithm." Proceedings of the 9th IEEE Real-Time Systems Symposium, 6-8 December 1988, PP. 251-258.

[31] R.Davis and A. Wellings. Dual Priority Scheduling. Proceedings of Real-Time Systems Symposium, PP. 100 - 109, 1995.

[32] R. Davis, K.W. Tindell, and A. Burns. "Scheduling Slack Time in Fixed Priorty Pre-emptive Systems." Proceedings of the 14th Real Time System Symposium, PP. 222-231, North Carolina, USA, December 1993.

[33] S. Yoo, M. Park, and C. Yoo. "A Step to Support Real-Time in a Virtual machine Monitor." In Proc. Of the 6th IEEE Consumer Communications and Networking Conference. Las Vegas, USA, January 2009.

[34] Kyong Hoon Kim, Anton Beloglazov, and Rajkumar Buyya, "Power-Aware provisioning of virtual machines for Real-time cloud services", Concurrency Computat., Pract. Exper. 2011; 00: 1-7, Prepared using cpeauth.cls, version: 2002/09/19 v2.02.

[35] Yuichi Asahiro, Jesper Jansson, Eiji Miyano, Hirotaka Ono, "Upper and Lower Degree Bounded Graph Orientation with Minimum penalty", Funded by KAKENHI Grant Number 21680001, Page 9, cripit.com/abstract/CRPITV128 Asahiro.html