Execution of real time task on cloud environment

Sampa Sahoo, Syed Nawaz, Sambit Kumar Mishra, Bibhudatta Sahoo
Department of Computer Science and Engineering
National Institute of Technology, Rourkela, India

***Abstract*—Cloud computing is an internet-based computing where resources, soft wares and information are shared on demand basis i.e. user can access documents anytime anywhere. Execution of real time task on cloud computing environment is an emerging research area. Real-time task needs to meet their deadlines regardless of system load or makespan. This paper discusses about scheduling of real time task on cloud environment considering Basic Earliest deadline first (BEDF), FFE (first fit EDF), BFE (best fit EDF), WFE (Worst fit EDF) algorithms. Different performance parameters such as guarantee ratio (GR), utilization of VMs (UV), throughput (TP) are used to measure the effectiveness of the algorithms.**

***Keywords*—** EDF, Real-time, periodic task, aperiodic task

I. Introduction

Cloud computing is used to provide ubiquitous, on demand network access such that user can access computing resources anywhere and at any time. It is built on the base of distributed computing, grid computing and virtualization which delivers the on demand computing resources over the internet on a pay-for-use basis. Users can access the resources without considering the installation details. For example, Yahoo mail, Gmail or Hotmail etc. can be accessed through internet connections whereas server and email management software is in cloud and are managed by respective service providers like Yahoo, Google.

Cloud can be categorized as private, public or hybrid depending on the resource ownership. In private cloud resources are owned by an organization. So, applications that need high control over data, security and QoS are suitable for this cloud. Public clouds are owned and managed by third party which makes data less secure. Hybrid cloud is a combination of public and private cloud i.e. it takes the benefits of both the approaches.

According to service models used for cloud it can be classified as SaaS, PaaS, and IaaS. In Software as a service (SaaS) consumer can use providers applications on a cloud infrastructure. (E.g. web browser). Platform as a service (PaaS) provides consumer to build and deliver web-based applications. Here users doesnt need to consider the cost and complexity of buying and managing underlying hardware, software etc. Infrastructure as a service (IaaS) allows consumer to access resources on pay per use basis.

Users perspective of cloud computing is that it is a dynamic utility where resources are available at any time, consumed at any amount and paid according to the amount consumed. The availability and reliability of resources are contractual. Here user doesnt consider the technical details of how service providers give the service and providers do not bother which device is used to consume the service.

Real-time systems need to generate correct results not only in their values but also in the times at which the results are produced called as timeliness. Here a quantitative expression of time is used to describe the behavior of the system. Common characteristics used to differentiate real-time system from non-real time systems are time constraints i.e. deadlines associated with tasks and safety-criticality.Real-time system uses time to measure the performance of the system. Real-time tasks must be assigned and completed before deadlines. If it misses deadlines there may be system failure.

Depending on the penalties of a task missing its deadline a real-time task can be classified as hard, soft or firm real-time task.

Hard real-time task : a hard real-time task must produce its results within deadline otherwise system is considered to be failed. These tasks are safety-critical i.e. deadline miss can cause severe consequences. E.g. robot system or anti-missile system, where system needs to respond within deadline to avoid critical consequences.

Soft real-time task : Soft real-time tasks have deadlines but they are not critical. If system misses its deadline it is not considered to be failed but it results degradation of performance of the system. E.g. web-browsing, railway reservation application. In web-browsing if a web-page takes several minutes to open or in railway reservation system if it takes several minutes to give required response after seat reservation we consider that system is not working properly rather than system failure.

Firm real-time task : a firm real-time task has deadline and it needs to produce the results within deadlines. If it misses its deadline then system is not failed but late results are discarded. E.g. video conferencing, satellite-based tracking of enemy movements involves firm real-time task.

Depending on the way real-time tasks recur over a period of time following types of real-time task are there

Periodic task : A periodic task recurs after certain fixed time interval known as period of task. E.g. temperature, pressure and chemical concentration monitoring in chemical plant.

Sporadic task : A sporadic task repeats at random instants. These tasks can vary from highly critical to moderately critical tasks. E.g. I/O device interrupt, DMA interrupt, task handling the reporting of the conditions.

Aperiodic task : There is similarity between aperiodic task and sporadic task except that here the minimum separation between two consecutive instances can be zero. These tasks are generally soft real-time tasks. E.g. logging task in a distributed system, keyboard presses, mouse movements.

II. Related Works

The authors [1] have proposed a rolling-horizon scheduling architecture for scheduling of real-time task in virtualized clouds and analyzed task-oriented energy consumption model. They have designed energy-aware scheduling algorithm named EARH with resource scaling up and scaling down strategies for real-time, aperiodic, independent tasks based on their architecture. The researchers [5] have described a fault-tolerant framework for optimization of resource utilization assuring availability and responsiveness. They have also designed pluggable framework for plugging VM replica deployment algorithms and a real-time publish/subscribe framework for real-time applications. This paper [6] compared EDF (Earliest deadline first) and RM (rate monotonic) algorithms. Their study shows that EDF allows full processor utilization, efficient use of computational resources and better responsiveness for aperiodic activities. This property of EDF makes it suitable for embedded system where computational resources are limited and multimedia system where quality of service is controlled by resource reservation mechanisms. In comparison to EDF, RM is simpler to implement and applicable to only high priority task. The writers [7] have identified the technical challenges for supporting soft real time applications such as online video streaming, cloud-based gaming, and telecommunication management in cloud environment. Their survey displays the recent advancement in the field of real-time virtualization and cloud computing technology for implementation of cloud-based real-time applications. This paper [8] showed Clairvoyant EDF (CEDF) algorithm to insert idle-times for non-preemptive task. Their simulation results show that CEDF is better than EDF. The authors of [9] used EDF scheduler and ant colony optimization (ACO) scheduler for preemptive periodic task. Simulation results shows that during overloaded case ACO based schedulers execution time is more than EDF scheduler and for under loaded case EDF schedulers execution time is less. The reviewers [10] have used profit time utility function an d penalty time utility function for modeling of real-time activities in cloud computing. There is a reward for early completion and a penalty for deadline misses of real time tasks. Several experiments conducted display that the proposed algorithm is better than the traditional EDF approach, the traditional single time utility function approach, and an earlier heuristic approach for homogeneous profit/penalty task model. This paper [11] uses global scheduling to explain processing of task migration with EDF-RM algorithm for real time tasks in distributed system. The presenters [12] proposed a preemptive online scheduling with task migration algorithm to execute real time applications. Their approach is subject to minimum response time and improve efficiency of a task. Simulation results shows that proposed algorithm surmount the existing EDF and non-pre-emptive algorithms. Their approach is subject to minimum response time and improve efficiency of a task. In this paper [13] the authors have addressed the problem of scheduling and resource provisioning for scientific workflow on IaaS clouds. The main aim of their work is to maximize the completion of user-prioritized workflows with budget and deadline constraints. In [14] an energy efficient scheduling algorithm EEVS is proposed for virtual machines to reduce the total energy consumption of cloud. It also supports DVFS (dynamic voltage and frequency scaling). The researchers [15] proposed a scheduling algorithm for the improvement of deadline guarantee and resource utilization by modifying the conservative backfilling algorithm by utilizing the earliest dead-line first (EDF) algorithm and the largest weight first (LWF) algorithm in the cloud data center. The writers of [16] proposed an online-scheduling strategy with goal to meet the deadline constraint at a lower price for scientific workflows on hybrid clouds. In [17] most-efficient-server-first task-scheduling scheme is proposed to minimize energy consumption of servers in a data center while keeping the datacenter response time within a maximum constraint. This paper focuses on EDF algorithm to schedule real time task on cloud environment. Three new algorithms FFE, BFE, WFE are proposed which are variation of basic EDF algorithms for both periodic and aperiodic tasks.

III. Scheduling Model

Target system is characterized by m number of physical hosts $H = h_1, h_2 h_m$. Each hosth_k , contains a finite set of virtual machines (VMs) $V_k = v_{1k}, v_{2k}, , v_{|k||k|}$. Each VM has ready queue to hold executing task, new task and waiting task to be executed. The tasks submitted by different users are stored in ready queue. The Central scheduler consists of ready queue, real-time controller, VM controller. The real-time controller checks whether task will completes its execution before deadline. If not then it informs VM controller, which adds VMs so that task can be completed within timing constraints otherwise task will be rejected. Fig. 1 illustrates the scheduling architecture of the system.

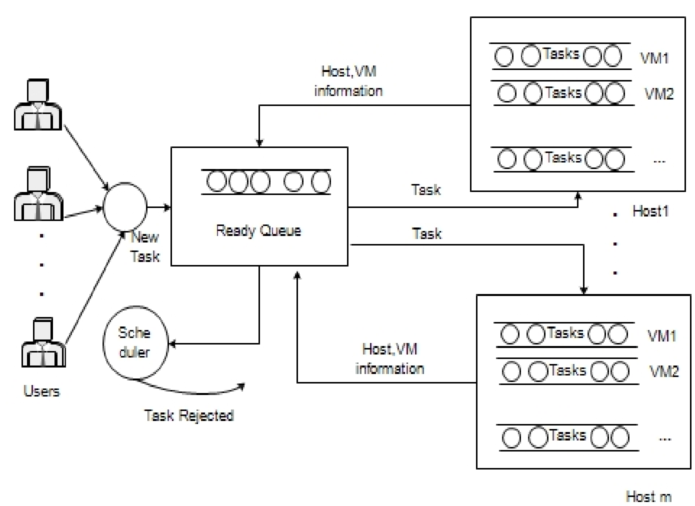

Fig. 1. Scheduling Model

Following steps are taken for scheduling a task stored in ready queue

Step 1: Tasks are sorted according to their deadlines.

Step 2: scheduler checks the status of hosts and VMs such as running tasks remaining execution time, the information of tasks in waiting pool including their deadlines, currently allocated VMs, start time, etc.

Step 3: When a task in ready queue is ready to execute, it is assigned to VM of selected host.

Step 4: After the completion of the task VM becomes free to accept new task.

Step 5: If a task misses its deadline then it is rejected by the scheduler.

Fig. 2 shows the states of task during its execution

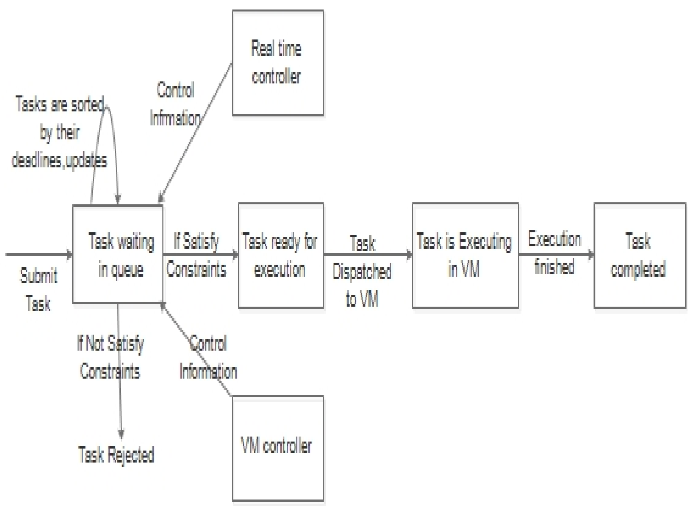

Fig. 2. Task Execution states

There are several metrics to measure the performance of a real time system. Following parameters are considered for measuring the performance of the system. Guarantee Ratio (GR): It is defined as the ratio between the numbers of tasks meets their respective deadline to total number of tasks, that is Guarantee Ratio= (number of tasks meets their respective deadline)/ (total number of tasks) Utilization of VMs (UV): It is defined as the amount of useful work done by VMs in its life time i.e. task executed on VM must meets its deadline, that is Utilization of VMs= (size of tasks meets deadline, task Task set)/ (computation power of VMs in each Host, Host host set and VM VM set) Through Put (TP): It is defined as the number of tasks meets their deadlines successfully.

IV. SCHEDULING ALGORITHMS

Real-time task scheduling algorithm can be categorized as clock-driven, event-driven and hybrid algorithms. Clock-driven algorithms take decision based on system clock interrupts whereas event-driven algorithm take the help of events such as arrival and completion of tasks. Hybrid algorithm is defined by both i.e. clock interrupts and event occurrences. Event driven scheduling is further classified as earliest deadline first (EDF) and rate monotonic analysis (RMA). This work concentrates on EDF and its variations. Assumption: tasks are sorted in ascending order according to their deadline. Basic Earliest Deadline First (BEDF): Search for appropriate host and VM which can execute the task with shorter deadline. Then assign the task to respective VM without considering whether task will complete within deadline or not. The VM becomes free after completion of assigned task. First Fit EDF (FFE): Here scheduler first check whether a task meet the deadline or not among all the free VMs. If the constraint is satisfied by the available VMs it search for the first VM which can execute the task and assign the task to that VM. This process will continue until all tasks are not assigned. Best Fit EDF (BFE): Here scheduler first finds the VMs which can run the task within the deadline and then selects the VM that has minimum finishing time for that task. Worst Fit EDF (WFE): VMs that can satisfy the constraint of executing task within deadline are selected by the scheduler. Then the tasks are assigned to VM that will take maximum finishing time.

V. SIMULATION RESULTS

Simulation of above system is done in MATLAB R2009a. Before doing simulation following assumptions are made Assumptions: Task 1. Number of Tasks in sequence 2, 4, 6 50. 2. Arrival Time of Tasks in Range (CC) [1, 10]. 3. Size of Tasks in Range (NIs) (0, 10000]. 4. Dead line of Tasks in Range (CC) (1, 20]. Host and VMs 1. Number of Hosts are 3. 2. Number of VMs in each Host is in Range [2, 20]. 3. Computation Power of VMs in Range (NIPC) (0, 5000]. Simulation is done for both aperiodic and periodic task taking above assumptions into account.

Fig. 3 to Fig. 5 shows the result for an aperiodic task. It is observed from Fig. 3 that proposed algorithms (FFE, BFE, and WFE) performs better than basic EDF (BEDF) for guarantee ratio. From Fig. 4, we can see that proposed algorithms are better than the BEDF algorithm and when number of task increases the utilization of VMs also increases. Fig. 5 shows that throughput of the system increases with increase in number of tasks for the system with proposed algorithms.

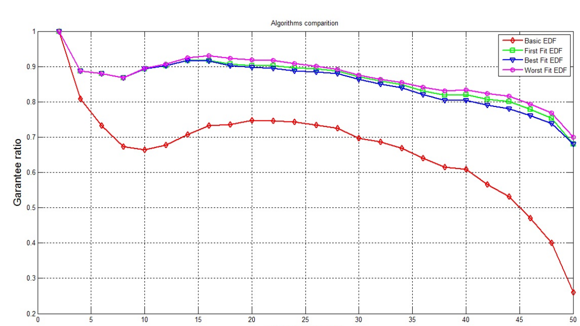

Fig. 3.

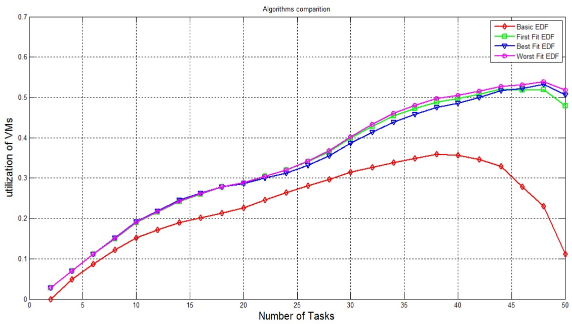

Fig. 4.

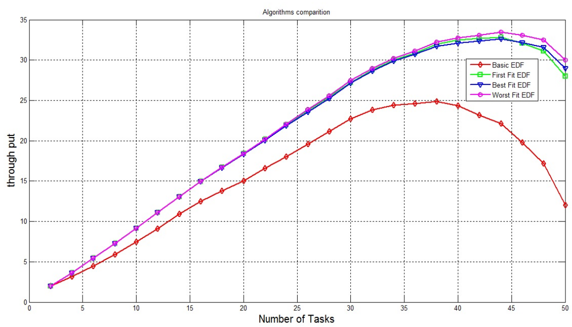

Fig. 5.

Fig. 6 to 8 shows the result for a periodic task. It is observed from Fig. 6 that proposed algorithms (FFE, BFE, and WFE) performs better than basic EDF (BEDF) for guarantee ratio. From Fig. 7, we can see that proposed algorithms are better than the BEDF algorithm and when number of task increases the utilization of VMs also increases. Here Best fit EDF shows the better result. Fig. 8 shows that throughput of the system increases with increase in number of tasks for the system with proposed algorithms.

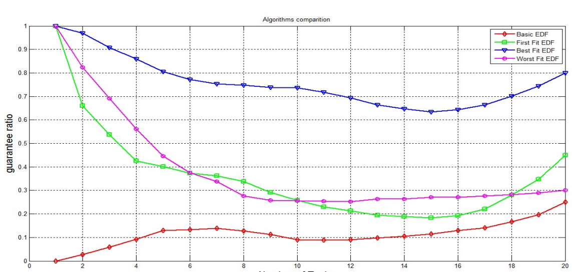

Fig. 6.

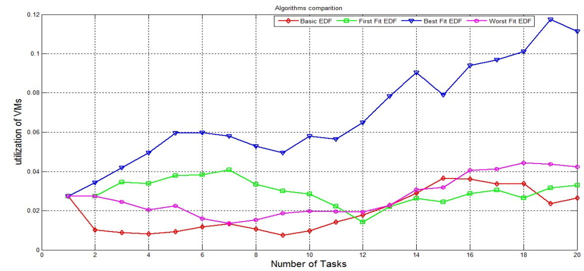

Fig. 7.

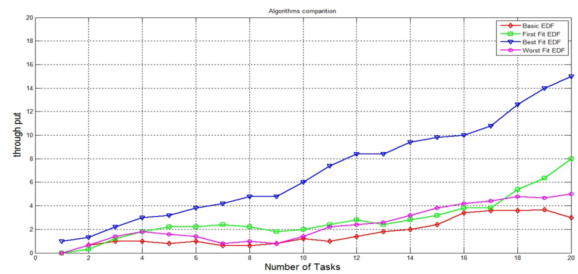

Fig. 8.

VI. Conclusion

The timing requirements of real-time task makes it difficult to implement in cloud computing. It requires real-time virtualization technology for adoption of real-time applications. This paper presented three scheduling algorithms called first fit EDF (FFE), best fit EDF (BFE), worst fit EDF (WFE). These algorithms performances are measured against basic EDF algorithm. Different performance metrics used are guarantee ratio, utilization of VMs, throughput. Simulation results shows that in all the cases proposed algorithm gives better result. In future work scheduling of sporadic task will be done. Sporadic task is real time task that is activated irregularly with some known bounded rate ,which is characterized by a minimum inter arrival period, that is, a minimum interval of time between two successive activations.

References

[1] Xiaomin Zhu,Laurence T. Yang, Huangke Chen ,Ji Wang, Shu Yin, Xiaocheng Liu Real time tasks oriented energy aware scheduling in virtualized clouds IEEE transactions on cloud computing ,vol 2 ,no 2,2014.

[2] Aser Avinash Ekka Fault Tolerant Real Time Dynamic Scheduling Algorithm for Heterogeneous Distributed System, M.Tech research thesis, NIT, Rourkela.

[3] Rajib mall Real-time Systems theory and practice, Pearson education, ISBN: 81-317-0069-0, 2007.

[4] Jane W. S. Liu Real-time systems, Pearson education, ISBN: 81-7808-463-5, 2000.

[5] Kyoungho An, Shashank Shekhar, Faruk Caglar, Aniruddha Gokhale, Shivakumar Sastry A Cloud Middleware for Assuring Performance and High Availability of Soft Real-time Applications in Journal of Systems Architecture, Volume 60, Issue 9, October 2014, Pages 757769.

[6] Giorgio C. Buttazzo Rate Monotonic vs. EDF: Judgment Day in Real-Time Systems, 29, 526, 2005.

[7] Marisol Garca-Valls, Tommaso Cucinotta, Chenyang Lu Challenges in real-time virtualization and predictable cloud computing in Journal of Systems Architecture 60 (2014), 726740.

[8] Cecilia Ekelin Clairvoyant Non-Preemptive EDF Scheduling in Proceedings of the 18th Euromicro Conference on Real-Time Systems (ECRTS06), 2006.

[9] M.Kaladevi , Dr.S.Sathiyabama A Comparative Study of Scheduling Algorithms for Real Time Task in International Journal of Advances in Science and Technology, Vol. 1, No. 4, 2010.

[10] Shuo Liu, Gang Quan, Shangping Ren On-line Scheduling of Real-time Services for Cloud Computing in IEEE 6th World Congress on Services, 2010.

[11] Rashmi Sharma, Nitin Task Migration with EDF-RM Scheduling Algorithms in Distributed System in International Conference on Advances in Computing and Communications, 2012.

[12] R. Santhosh, T. Ravichandran Pre-emptive Scheduling of On-line Real Time Services With Task Migration for Cloud Computing in Proceedings of International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013.

[13] Maciej Malawskia,, Gideon Juve , Ewa Deelman, Jarek Nabrzyski Algorithms for cost- and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds in Fu-ture Generation Computer Systems 48(2015).

[14] Youwei Ding, Xiaolin Qin, Liang Liu, Taochun Wang Energy efficient scheduling of vir-tual machines in cloud with deadline constraint in Future Generation Computer Systems 50 (2015).

[15] SaeMi Shin, Yena Kim and SuKyoung Lee Deadline-Guaranteed Scheduling Algorithm with Improved Resource Utilization for Cloud Computing in 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), 2015.

[16] Bing Lin, Wenzhong Guo and Xiuyan Lin Online optimization scheduling for scientific workflows with deadline constraint on hybrid clouds in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.3582,2015.

[17] Ziqian Dong, Ning Liu and Roberto Rojas-Cessa Greedy scheduling of tasks with time constraints for energy-efficient cloud-computing data centers in journal of cloud compu-ting, 2015.