



Research Article

ISSN: 2394 - 658X

Deadline - Cost Constraint Task Scheduling for Soft - Real Time Systems

Himani and Kamaljit Kaur

Department of CSE, Guru Nanak Dev University, India
er.himani26@rediffmail.com

ABSTRACT

Cloud computing is a nascent technology which widely spreads among researchers. It provides users with infrastructure, platform and software as amenity which is effortlessly accessible via Internet. In cloud computing, there are numerous tasks needs to be executed by the existing resources to attain best performance, shortest response time, minimal total time for completion, utilization of resources etc. Due to these dissimilar objectives, we need to design, develop a noble scheduling strategy to outperform proper division of tasks on virtual machines. Recently existing scheduling algorithm may work proficiently in some perspective. But they are unable to achieve deadline of task and attain maximum profit. Based on the concept of space-shared scheduling policy, this paper presents cost-deadline based scheduling to schedule tasks with Cloud-sim by taking into account several parameters including task profit, task penalty, and throughput.

Key words: Cloud computing, scheduling, cloud-sim, space-shared, virtual machine.

INTRODUCTION

Cloud computing directed to an advanced way in which IT infrastructure, applications, services are developed, designed and delivered. In this, the utility of IT possessions can be consumed on the basis of pay-per-use model like electricity etc. Many computing service providers like Google, Microsoft, IBM, Yahoo etc provide data centers in various locations around the world to deliver Cloud computing services. A recent Berkeley report [1] stated: "Cloud computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service". Three major services which are contributing in defining cloud computing are Infrastructure-as-a-service (IaaS), Platform-as-a-service (PaaS), Software-as-a-service (SaaS).

So, we firstly present the deadline and cost-oriented scheduling work existing in the background Section II, and we focus on the problems related to the implementation of these algorithms in cloud computing. Further to this, section III describes our proposal methodology and its requirements and assumptions. The rest of the paper is planned as follows, section IV, presents the experiment scenario and we analyze the performances of our work by comparing the ability of our algorithm to meet deadlines to the Space Shared scheduling algorithm implemented in Cloud Sim [2]. We show that the deadline meeting approach of our work allows to minimize the extra-cost implied by the lateness of a task. Finally, section V illustrates the concluding annotations and the further research guidelines of our work.

DEADLINE SCHEDULING LITERATURE

Dr. M. Dakshayini [3] proposed a new scheduling algorithm based on priority and admission control scheme. The priority is assigned to each admitted queue. Admission of each queue is absolute by calculating acceptable delay and service cost. Benefit of proposed policy with the proposed cloud architecture has attained very high (99%) service completion rate with sure QoS. As this policy provides the highest preference for very much paid user service-

requests, overall examining cost for the cloud also increases. Amit Nathani [4] the proposed algorithm finds number of slots in addition to finding one slot while scheduling a deadline sensitive lease. It also applies two concepts; one is swapping and other is backfilling, while rescheduling already accommodated leases to make space for a newly arrived lease. Backfilling has a disadvantage of requiring more preemption, which increases overall overhead of the system. The results show that by applying swapping and multiple slots concepts, the number of accepted leases increases compared to the existing algorithm. It also shows that the proposed algorithm requires rescheduling of fewer leases compared to the existing algorithm of Haizea. R. Santhosh [5] focus on providing a solution for online scheduling problem of real-time tasks using “Infrastructure as a Service” model offered by cloud computing. The real time tasks are scheduled preemptively with the intent of maximizing the total utility and efficiency. It also minimize the response time and to improve the efficiency of the tasks. The tasks are migrated to another virtual machine whenever a task misses its deadline. This improves the overall system performance and maximizes the total utility. The proposed algorithm can significantly outperform the EDF and Non Preemptive scheduling algorithm. Florin Pop, Ciprian Dobre [6] addresses the problem of remote scheduling of a periodic and sporadic tasks with deadline constrains in Cloud computing. Beginning from traditional addressed scheduling techniques and considering asynchronous mechanism to handle tasks, they analyze the possibility of decoupling event listening from task creation and scheduling, activities that can be placed into a peer-peer relation over a network or to client-server in Cloud. They consider multiple independent tasks sources that follow with a specific distribution. They created a simulation experiment in MONARC that highlights the capability of tasks migration in order to respect the deadlines. Nitish Chopra [7] developed a level based scheduling algorithm which executes tasks level wise and it uses the concept of sub-deadline which is helpful in finding best resources on public cloud for cost saving and also completes workflow execution within deadlines. Dr. V. Vaithyanathan [8] the incoming tasks can choice their method on the basis of task requirement like minimum execution time or cost and then it is prioritized. The algorithm is named as TPD Scheduling Algorithm in which T represents Task Selection, P is used for Priority (in terms of cost) and D is for Deadline. The proposed model is experimented and evaluated on cloud-sim toolkit. Outcomes validate the accuracy of the framework and show a momentous improvement over other scheduling methods.

The diverse types of deadlines and extant three prevalent deadline-conscious scheduling algorithms which are directly related with our work. A task has two basic characteristics: its vilest execution time and a deadline. Then we can categorize deadline based tasks in three types:

- Hard deadline task: in any case missing deadline is not allowed.
- Soft deadline task: the task has a penalty in function of its lateness, i.e. Lateness= task execution time-deadline.
- Firm deadline task: the task gains reward if Task execution time < Deadline.

The reward of the task is function its aheadness= Deadline-task execution time.

1. Earlier Deadline First (EDF)

EDF algorithm gives the highest priority to those tasks which has the earliest deadline.

The algorithm is pre-emptive in nature which means tasks may be sporadic in directive to process other tasks with a higher priority. The EDF algorithm is driven by “dynamic priority in the sense that the priority of a request is assigned as the request arrives” [9]. The norms made for the EDF algorithm, the periodicity of tasks.

2. Least Laxity First (LLF)

The LLF algorithm is one more optimum scheduling algorithm obsessed by lively priorities. It is work on the concept of laxity. The laxity of a task is demarcated as the deadline minus the remaining computation time required to complete the task [11]. Therefore, the laxity is the supreme time that a task can wait before it becomes impossible to encounter its deadline. LLF provides highest priority to the process which has the lowest laxity. LLF is also pre-emptive and the norms are similar to EDF. LLF also have a well provision than EDF of non-periodic tasks [9].

3. Cloud Least Laxity First (CLLF)

CLLF, minimizes the extra-cost implicit from tasks that are executed over a cloud setting by ordering using its laxity and locality. By using CLLF, deadlines are minimized while the total execution time of the job remains in acceptable levels. CLLF is non-optimal distributed scheduler for soft deadline task. It is almost similar to the LLF algorithm but only basic difference is CLLF is non-pre-emptive in nature. It also manages the locality of task [10].

COST-DEADLINE BASED SCHEDULING APPROACH

In this section, we present Cost-Deadline Based Scheduling (CDBS) approach towards scheduling of task which resolves the matters tinted above. We firstly converse the norms made during the scheme of the approach and then discuss the approach to schedule the task on virtual machines (VM).

Assumptions

CDBS approach is used for soft deadline task as defined in above section. The user should have to define the cost and deadline of the task. We are assuming all tasks having same configuration. We consider a cloud having number

of virtual machines (VM). Each virtual machine has number of processing elements (PEs) which all are demarcated by number of millions of instruction that can completed in one second (MIPS). Each virtual machine has its memory (RAM), Bandwidth (BW). We are assuming 20 VMs having millions instruction per second (MIPS). The set of VMs {0, 1, 219} having {500, 1000, 400, 500, 600, 700 , 500, 700}.

Approach

The CDBS approach is discussed in following steps:-

Step 1: Sort the incoming tasks according to the deadline (means give priority earliest deadline first) and maintain a queue.

Step 2: If two tasks devising similar deadline, then sort it according to cost (means highest paying user first) and again maintain queue.

Step 3: Schedule virtual machines according to Time-shared policy.

Step 4: Schedule task on virtual machines according to Space-shared policy.

Step 5: Introduce four parameters on behalf of which this approach is compared with Simple Space-shared policy.

The parameters used in this approach are Task profit, Task Penalty, Throughput and Net Gain.

On the bases of these parameters our approach outperform than the space-shared policy and time-shared policy.

SIMULATION RESULTS

In this section, we examine the practical abilities of CDBS approach. We firstly extant the experimental situation and then compare the performances of CDBS with Space-shared and Time-shared scheduling algorithm of cloud-sim.

Scenario

The performance measurement has been gathered by using the CloudSim [2] framework. The goal of this study is to show the ability of CDBS approach to meet deadlines in conditions where the other algorithms (Time Shared and Space Shared) could not able to meet deadlines. To accomplish this, we recommend the following situation. We label identical data centre composed of hosts. Each Processing element has a speed of 1000 MIPS, an available bandwidth of 10 Gbit/s, a storage capacity of 1 TB and a Random Access Memory (RAM) of 2048 MB.

We process 5000 cloudlets on the earlier demarcated data centre using three policies - CDBS, Space-Shared and Time- Shared approach. The lengths of cloudlets are equal and each cloudlet is given a randomly generated deadline where minimum value is half the number of cloudlets and maximum value is equal to number of cloudlets. Cost of each cloudlet is also depending upon the number of cloudlets and it is also randomly generated.

Task Profit

The number of tasks which have been completed successfully before they meet the deadline. CDBS approach is more efficient to finish tasks before the deadline arrives than Time-shared and Space-shared scheduling algorithm. We compare number of tasks completed by each approach for the given set of cloudlets. In figure 1, one can see the results of experiment.

On the x-axis it shows the number of cloudlets and y-axis indicates number of tasks completed successfully before deadline reaches. By examining the figure we can say that Time-shared algorithm inefficient to meet deadline before it comes. Table 1shows results of the extracted data from experiment that uses 3000 cloudlets.

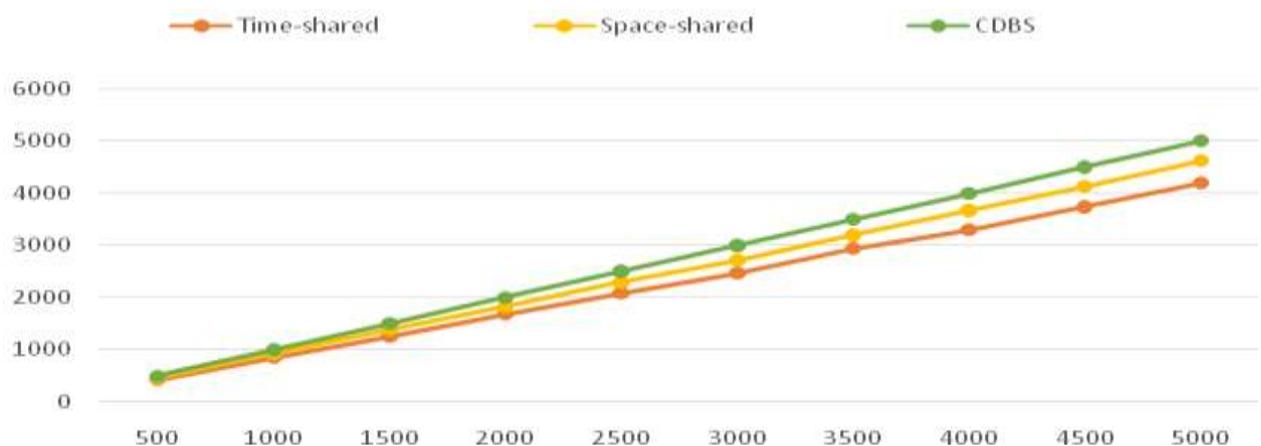


Fig. 1 Comparison of task profit

We can notice that, in this case all approaches seems to be quiet indistinguishable but CDBS completes almost all tasks before it meets the deadline. By increasing the number of cloudlets and virtual machines we can observe the more difference and conclude that our approach is better than Space-shared and Time-shared approach.

Table -1 Result for 3000 Cloudlets

Approach	Completed task before deadline	Ratio
CDBS	2997	99.9%
Space-Shared	2710	90.3%
Time-Shared	2464	82.1%

Task Penalty

We can analyze result of another experiment from Figure 2. In this parameter, we annoyed to find task penalty of cloudlets i.e. number of cloudlets misses their deadline. The x-axis represents the number of cloudlets and y-axis defines missed deadline. Once again, it is notice that CDBS outperforms than other approaches. CDBS approach missed negligible number of deadline as compared to Space and Time-shared polices. Table 2 shows the mined results for 4500 cloudlets. From this table, we can analyze that Time-shared approach misses deadlines nearly double than space-shared and 16% more than CDBS approach. In this, proposed work is far better than default approaches for soft deadline cloudlets.

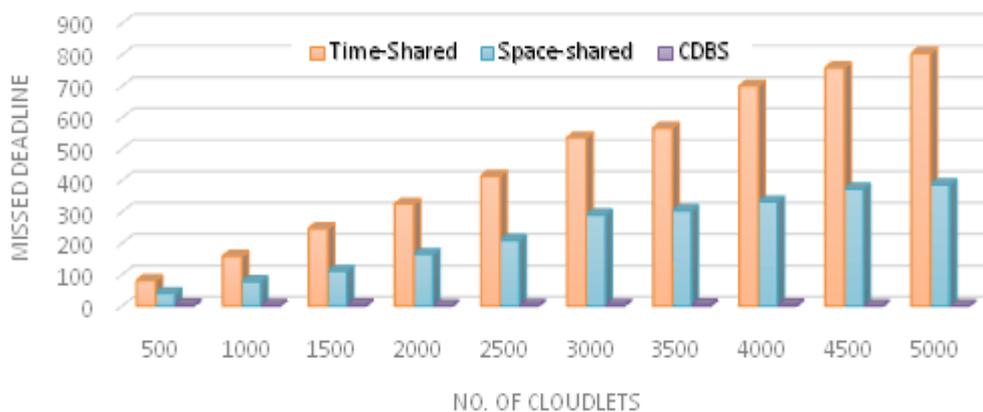


Fig.2 Comparison of task penalty

Table - 2 Missed Deadline for 4500 Cloudlets

Approach	Missed Deadline	Ratio
CDBS	1	0.1%
Space-shared	373	8.3%
Time-shared	759	16.9%

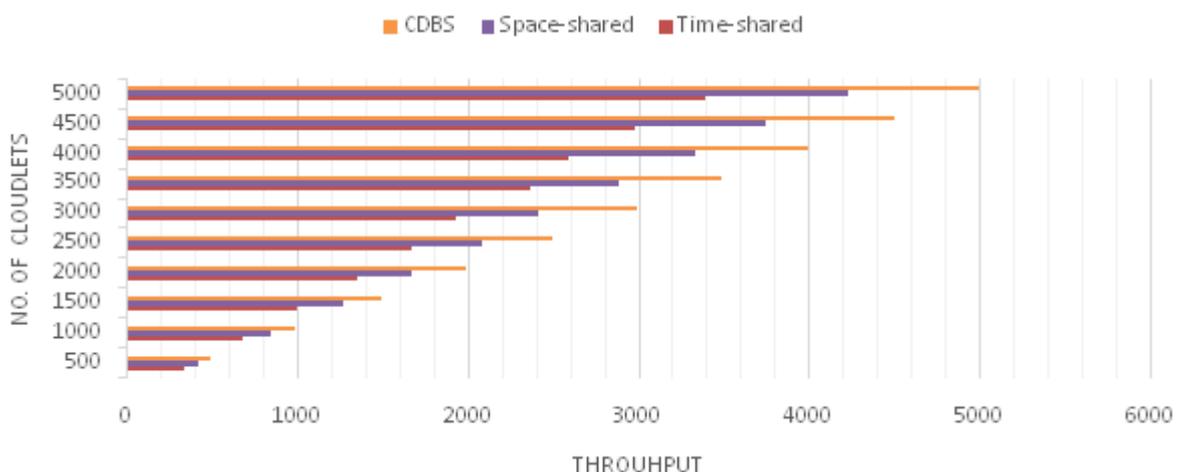


Fig.3 Comparison of throughput

Throughput

Another parameter which shows CDBS approach is best among the other two policies i.e. Space-shared and Time-shared policies. Throughput of tasks is the difference between earlier defined parameters. Figure 3 shows experimental results of all approaches in which x-axis indicates the throughput and y-axis represents the total number of cloudlets.

CONCLUSION

This work represents a soft real-time scheduling approach with deadline and cost constraints. It has been planned to take the concrete problems related to cloud computing. We determine that earlier approaches are not able to meet the deadline efficiently. However, we illustrate that a deadline-meeting methodology to schedule tasks over a cloud allows reducing the number of missed deadline. We compare the CDBS with Space-shared and Time-shared policies and shows that the given approach is more effective in defined parameters as Task Profit, Task Penalty, Throughput and Net Gain. The future step of our research includes for cost optimization we can develop a prediction model for estimating the initial value of MaxUserPay (Initial Cost of Task) based on the length of the task, number of processing elements required, output file size, etc. This prediction model can be valuable for enhancing the profit and saving the money for users as well.

REFERENCES

- [1] M Armbrust, A Fox, R Griffith, AD Joseph, R Katz, A Konwinski, G Lee, D Patterson, A Rabkin, I Stoica and M Zaharia, A View of Cloud Computing, *Communications of the ACM*, **2009**, 53 (4), 50-58.
- [2] R N Calheiros, R Ranjan, C A F de Rose, and R Buyya, CloudSim: a Tool Kit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithm, *Software: Practice and Experience*, **2011**, vol 41, no 1, pp 23-50.
- [3] M Dakshayini and H S Guruprasad, An Optimal Model for Priority Based Service Scheduling Policy for Cloud Computing Environment, *International Journal of Computer Applications*, **2011**, Vol.32, No. 9.
- [4] Amit Nathani, Sanjay Chaudhary and Gaurav Somani, Policy Based Resource Allocation in IaaS cloud, *Future Generation Computer Systems*, Elsevier, **2011**, 28, 94 -103.
- [5] R Santhosh and T Ravichandran, Pre-emptive Scheduling of On-line Real Time Services with Task Migration for Cloud Computing, *IEEE International Conference on Pattern Recognition*, **2013**.
- [6] Florin Pop, Ciprian Dobre, Valentin Cristea and Nik Bessis, Scheduling of Sporadic Tasks with Deadline Constrains in Cloud Environments, *27th IEEE International Conference on Advanced Information Networking and Applications*, **2013**.
- [7] Nitish Chopra and Sarbjeet Singh, Deadline and Cost based Workflow Scheduling in Hybrid Cloud, *IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, **2013**.
- [8] V Vaithyanathan, R Arvinth Kumar, S Vignesh, B Thamocharan and B Karthikeyan, An Efficient TPD Scheduling Algorithm for Cloud Environment, *International Journal of Engineering and Technology (IJET)*, **2013**, Vol. 5 No. 3.
- [9] Apache, Capacity Scheduler Guide, **2010**, Available: <http://hadoop.apache.org/>
- [10] Quentin Perret, Gabriel Charlemagne, Stelios Sotiriadis and Nik Bessis, A Deadline Scheduler for Jobs in Distributed Systems, *27th IEEE International Conference on Advanced Information Networking and Applications Workshops*, **2013**, pp 757-764.