Cloud Environment – Task Schedule and Adaptive Cost

Mohammed A. S. Mosleh
School of IT & Science,
Dr. GR Damodaran College of Science
Coimbatore, India

Syed Hamid Hasan
Department of Information Systems
King Abdulaziz University, Jeddah,
Saudi Arabia

ABSTRACT

Execution of tasks in cloud environment warrants use of data stored on remote data servers. Even though the memory requirement of the user computer are reduced yet it is critical to maintain deadlines. This paper proposes Task scheduling based on Adaptive Cost for providing access to data for VMs (Virtual Machines) in such a way that the cost is not impacted. CBAS(Cost Based Adaptive Scheduling) utilizes the time taken for data access to decide on the data access path that is most cost effective. Time taken is computed through variance & mean time of network service and I/O request arrival rate. Task Priority is then assigned for removing the data access time for tasks. In the end, analysis of data access path and allocation is done on the basis of task priority. Task with low priority are allocated to low cost path while tasks that have high priority are assigned more expensive path to meet deadlines. Hence, CBAS achieves task scheduling efficiently. The results of the experiments related to CPU Utilization, bandwidth, communication costs, computational cost, and execution time prove CBAS perform better than current contemporary methodologies.

Keywords

Cloud computing

1. INTRODUCTION

Customers use virtual platforms remotely through "pay-per-use" methodology in the cloud computing environment. NITS[1] define cloud computing as : on demand, convenient and ubiquitous access to resources over a network, that can be provided & provisioned through minimal interaction. Services like platform, software and infrastructure are provided to users for a price. Application and data is maintained through remote servers and internet. Businesses and consumers can access their data remotely through internet and use and an application without installing it locally. Processing and data storage capabilities are enhanced with this approach. Deployment of Cloud is done through various methods: distributed cloud, community cloud, hybrid cloud, private cloud and public clouds etc.

Cloud computing treats every facility available as a service and has its bases on architecture that is service oriented[2]. VMs (Virtual Machines) and PMs (Physical Machines) are provided under IaaS(Infrastructure as a service) for use in Backup, security, scaling, data partitioning, and processing. Development tools, web servers and databases are provided under PaaS(Platform as a Service) for application development. Gaming applications, communication processes, virtual desktops and email services are provided under SaaS(Software as a Service). Each of these facilities is a paid with cost determined by usage levels. Cloud service cost are minimal in comparison to installed services.

On the basis of the task requirement activities are performed on VMs or PMs. Multiple remote data centers provide storage services for data required for these tasks, at a specific cost[3]. During the execution of the tasks on the PMs or VMs requests for data are sent to these Data centers. Time taken for the data to reach the processing machines from the data servers is called access completion times. This approach has a limitation of storage and computation cost pertinent to the access paths. Thus we may have one of the following situations: Cost surge to ensure timely access of data or data access delays to maintain low cost of computing. Which enumerates into scheduling performance reduction on the whole.

Keeping in consideration this issue of data access we have proposed the CBAS system that ensures timely data access through access paths that are cost effective. Completion time of all data access requests[13] for each task is estimated by this approach. The total cost all possible paths are computed by considering the storage costs[14], communication cost and computation cost. Task priority is determined by weighing the data access times. Then high priority task are assigned high cost paths, while low cost paths are attributed to tasks with low priority. This way adaptive selection of paths is done for overall cost reduction and effective data delivery in time.

The paper is segregated into section where Section 2 talks about other works done in this field with a discussion on schemes of scheduling. We discuss the methodologies used in this work in Section 3. Discussion and results of the experiments are presented in Section 4. In Section 5 we conclude the paper.

2. RELATED WORKS

A resource manager that is distributed and cloud enabled is called cloud scheduler. It provides a job execution environment by managing the cloud virtual machines. Typical example of such schedulers are Yahoo – Capacity scheduler, Facebook – fair scheduler and Hadoop - FIFO scheduler, however, the Quality of Service(QoS) constraints are not satisfied by any of these equitable and efficient resource managers. Hence suitability of these schedulers for application and services in the real-time cannot be vouched for, even though we see the hybrid environment on clouds are moving to these facilities. In this section we do not aim at finding alternatives and solution to the issues faced in scheduling of task, but to identify the drawbacks in current scheduling methodologies by thorough analysis.

An cost effective algorithm for dynamic scheduling of task based on deadline constraints was proposed by Sahni & Vidyarthi[4] that was good for scientific workflows. However, pay on the go model of pricing and resource provisioning on demand facilities of cloud computing environment are not aptly served by this algorithm, even though it works well in cluster and grids. Thus, a heuristic, cost effective algorithm for dynamic scheduling of task based on deadline constraints was used for exploiting cloud features that considered delay in instance acquisition and variability in VM performance which determines time schedules.

This approach fails in case of Virtual Machine failure as it leads to affecting the comprehensive time for workflow execution.

Tsai et al[5] proposed the Hyper Heuristic Scheduling Algorithm(HHSA). Improvement detection and diversity detection operator is used by the approach to dynamically ascertain which is the best suited heuristic of low level, for scheduling effectively. The scheduling performance and task scheduling makespan are bettered by HHSA. However, since this approach increases connection overhead it inadvertently reduces scheduling importance thus leading to reduced performance overall.

The ANGEL algorithm, which was dynamic agent based scheduling algorithm, was proposed by Zhu et al[6] to effectively schedule virtual cloud tasks. Here, collaborative process and bidding announcement are performed bidirectional for improving schedule performance. In order to improve scheduling further, VMs are dynamically added for considering elasticity. Bidding process is improved by generating calculation rules that results in reduced delays. However, since dispatching time and communication time is not considered by this approach it reduces overall performance.

EMO (Evolutionary Multi Objective) work flow was suggested by Zhu et al[7] aiming at reducing makespan and cost related to work flow scheduling. Because of the typical nature of the problems in workflow scheduling, the genetic operation and corresponding variation operators like real valued encodings, and binary encoding, in EMO the operators have their basis on them. The drawback is not considering the time overhead and monetary cost of storage and communication.

PRIMS (Phase & Resource Information-aware Scheduler for MapReduce) was proposed by Zhang et al[8] which is a scheduling approach that is fine-grained. In a data-intensive job the running time is effectively reduced by MapReduce, however schedulers based on solutions at a task-level, like MapReduce, Additionally, reduction of execution time of a job is major challenge that these schedulers face. So task were divided into phases by PRISM. Phase level scheduling is performed at each phase which has a usage profile of resources that is constant. Hence, reducing the total time of execution for the job. Yet the point that requires attention is the issue of meeting deadlines since it is a major concern when it comes to scheduling at phase levels.

The EARH approach was presented by Zhu et al[9] which is an EA(Energy Aware) real time scheduling mechanism oriented on RH(Rolling Horizon) optimization. It is best suited for virtualized cloud environment. VMs Cancellation, migration and creation procedures are developed for dynamically adjusting cloud scale for energy reduction and deadline achieving in real-time. However, dynamic update of VMs cycles is not a possibility in EARH.

Scheduling method for jobs with unknown duration was suggested by Srikant and Maguluri[10]. This approach assumes that the job size are unknown both at the arrival and the beginning of services. Thus an algorithm for load balancing and optimal throughput scheduling is introduced in the cloud data centers. The algorithm uses length of the queues as weights in-place of workload, in a maximum -weight schedule setup.

Cloud computing task scheduling is challenging, Su et al [11] suggested a scheduling method that is cost effective for large programs. The problem with this approach is that it does not consider the scheduling cost and completion time hence making it inefficient in terms of overall performance.

Based on the above we can clearly see that high costs of computation and communication are the major drawbacks of the methods described in literature. Another issue is overshooting of deadlines because of inefficient data paths utilized while scheduling tasks. Below we analyze the different scheduling schemes

2.1 Scheduling Schemes Analysis

Normally, scheduling mechanism termed efficient in a grid & cluster environment are really not so efficient when it comes to cloud computing. Pay per use and on Demand resource provision is a major cause of this difference in cloud computing. Thus, use of cloud features for efficient scheduling of tasks without delay in time is essential for the scheduling approach. VM task processing requires obtaining datasets from data centers in multiple and distant locations. Since there is a deadline for the task scheduling approach need to effectively obtain data within stipulated timelines. However, we come across another variable in this complex equation that need to be minimized that is ‘Cost’. Elementary resources of a cloud environment like storage and computational resources formulate the models based on cost.

Issue of high costs can be tackled by choosing the path with minimum cost from all the data paths available. However, all the task do not have the same time for execution, some tasks may require the data more quickly. While if the minimum cost path is chosen, the task would have to wait for data based on queue or there are chances of data loss because of the queue overflowing. Thus, selection of specific cost path has to be in line with the deadline for the tasks, in other words adaptive selection is required. This research focuses on these two issues majorly.

3. COST-BASED ADAPTIVE SCHEDULING

We will discuss Cost-based Adaptive task scheduling (CBAS) now. The proposed scheduling methodology can be used for effective task scheduling in VMs. We have taken inspiration from Su et al. They have considered the total cost and execution time variables for choosing the suitable scheduling methodology. Even though they considered the cost and execution time both, yet the method was not really efficient, since cost and time are work in collaboration. Time taken for completing a task is execution time, which mean it’s the overall time taken starting from assigning the task to VM and ending when the output is received. However, we cannot minimize the time for all sub-processes of the task and time taken by each sub-process varies. Which means, time for fetching data from data centers is a lot more than the actual execution time of the tasks. Likewise, the cost is a combination of resources like data transfer, storage, communication and computation etc. where communication and computational cost are pretty high in comparison to others. However, only the combined factors were considered by Su et al. for scheduling. Thus here in CBAS, we focus on distinctively considering each individual process as scheduling factor. Communication cost, computation cost and completion time for data access are the major factors considered. <Done>

Remote data centers store the data required for processing in the PM or VM. Cost effective paths need to be used for fetching this data to the VMs. The average arrival rate of the network I/O request has an independent Poisson distribution associated with each VM access. PM(Driver domain) process the data access request on first com first serve basis so the later request have to queue for their turn. Arbitrary distribution represents the data access service time in the PM(Driver domain).

Data path selection is done on the basis of completion time for data access. Physical machine’s network I/O request parameters are used for calculating data access completion time. μ gives the network Input/Output request service time in the PM while σ gives the variance. Network I/O request arrival rate is λ. So the formula for getting the completion time '*t*' is as follows

$$t = \frac{2\mu - \lambda + \lambda\mu^2\sigma^2}{2\mu^2 - 2\lambda\mu}. \tag{1}$$

λ will be calculated through the following

$$\lambda = \sum \lambda^{(e)} \cdot r^{(e)} + \sum \lambda^{(n)} \cdot r^{(n)}, \tag{2}$$

here $r^{(n)}$ and $r^{(e)}$ are CPU time allocation ratio for new and existing VMs. $\lambda^{(n)}$ and $\lambda^{(e)}$ signify Network I/O request arrival rate for the new and existing VMs. <Done>

Data access paths are used by VMs to fetch the data from remote data centers for performing the tasks. Every path has its associated resource for data storage and processing requests. There are certain cost associated for each resource usage. I/O request execution cost combined with re-access cost of the data gives the cost of computation. Datasets regeneration cost is also included. Total resource utilization cost for I/O request processing is the cost of communication. We can express it as the price of network and size of data set multiplied.

Cost for all available data paths is calculated and then the least costly path identified. Estimate of the path cost is done by

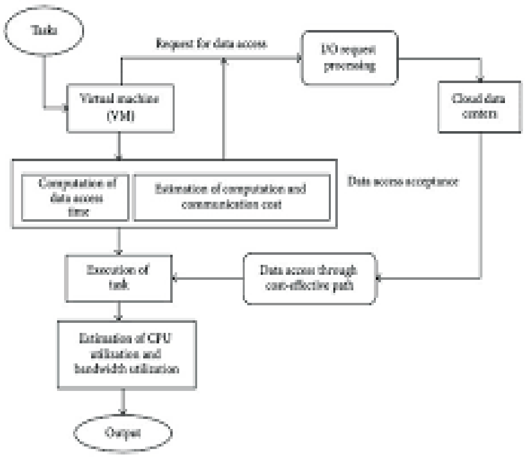
$$\text{Cost} = \text{Computation cost} + \text{Communication cost}. \tag{3}$$

Communication and computation cost are critical to determine the paths that are cost effective since they required for handling the I/O requests. A request for Data access is sent to the data centers by the VM, when executing a task. The I/O request is received by PM and access to data is provided.

The proposed CBAS takes into account data access completion time and associated costs for effective task scheduling. Priority is assigned by CBAS on the basis of completion time. A fixed time *T* gets ascertained and then priority is assigned based on the comparison of completion time with *T.* Task with longer completion time are given low priority tasks and the lowest cost path is selected for it, to minimize the overall price. This approach is chosen since low priority tasks may be completed with standard time without escalation. Likewise, task with a shorter deadline are assigned higher priority and are assigned paths with quicker access, not the longer minimum cost paths. Which may lead to cost increase, but since the goal is providing data within deadlines, small cost variation may be ignored. Post task execution, bandwidth and CPU utilizations are estimated.

The CBAS proposed solution is shown in Figure 1. In this paper we focus on task scheduling for VMs using paths with least cost for reducing the complexity of accessing data from remote data centers. Allocation of tasks is done to an under-loaded VM, on the basis of standard load condition. These tasks require data on remote data centers. Cost of processing the I/O request is calculated along with estimating the data access completion time. The bandwidth and CPU utilization are then calculated and applied for forthcoming task execution.

E.g. *V* is taken as basic mathematical program task with flexible property like MIPS(mil. Instruction per seconds), RAM and bandwidth. The user can choose these parameters from the cloud task parameters that he wants for the tasks. Besides, CloudSim is used for these simulation which is pretty user friendly. The chosen tasks dependent and non-preemptive.


Tasks
Request for data access
I/O request processing
Virtual machine (VM)
Cloud data centers
Computation of data access time
Estimation of computation and communication cost
Data access acceptance
Execution of task
Data access through cost-effective path
Estimation of CPU utilization and bandwidth utilization
Output

Cloud environment initiates VMs where *E* denotes existing VMs while *N* denoted new VMs. To accommodate introduction of new VMs to handle large loads. Basic mathematical program is executed by the task, the length for the program differs based on the codes initiated. Simple adding *(a + b)* is executed for the task once with four bit and then repeated until the predetermined length is achieved.

Considering *m* resources are available for task *v*. First possibility of execution is checked for the task. Each VM is unrelated with dedicated resources and is running parallely. No resource sharing is happening between VMs. A task that is dependent and non-preemptive is scheduled for the each VM. For every task *v*, *T* (v_i and m_j) and arrival rate λ_j is calculated. Equation [2, 7 and 9] are used to calculate the Ccommunication and Ccomp Costs for every path *d*. Cost of computation in [7] is calculated as total cost of executing task *V* on VM *m* that is provided by *p* [8] whereas cost of communication [9] is the product of prices of incoming network traffic and required data cost. Task priority is assigned on basis of completion time. The paths are then sorted based on cost and *T* (v_i and m_j). The paths are then allocated for every task. VMs which are under-loaded are then loaded with task that accesses data at remote data servers at the time of the deadline. Utilization of CPU [11] & utilization of bandwidth[12] is then calculated to determine system efficiency.

3.1. Description

VM resource *m*, number of VM and task V are initialized. New VM *N* and existing VM *E* are assigned. Completion time for data access *T* (v_i and m_j) is calculated for each task. Also each data path cost is estimated by calculating the communication and computation cost. Scheduling is done on the basis of every data path's communication and computation cost, along with completion time. Priorities are assigned to tasks based on the time of completion. Data paths that have fast C_{dt} access are assigned to the task with less time of completion i.e. time deadlines, since cost is not a priority. The tasks that have high time of completion are assigned access path with minimum cost C_{dmin}. Post execution of tasks, bandwidth and CPU utilizations are calculated.

4. RESULTS

We conducted a number of experiments for evaluating the CBAS and tabulated the results. [13] IT presents task scheduling that is cost efficient (CETS) in order to compare performance. CETS does not consider the completion time and cost of data access while comparing it with CBAS that considers completion time for data access and it's cost. Cloudsim tool [14] is used for the experiments.

For using the new algorithm, Cloudsim simulator classes were overridden (extended). CloudSim provides the option of hypothesis evaluation, before beginning development of software through an environment capable of reproducing test. A controlled and repeatable environment is provided to customers for testing their service by the Cloud simulation since access to live cloud infrastructure means incurring of access fees even for test purposes. It Additionally provides opportunity to fine-tune any performance bottleneck prior to real cloud deployment. Bandwidth, CPU utilization, execution time, communication and computation cost are compared to assess efficiency of both approaches.

VMs considered and number of task are flexible as per the user, i.e. bandwidth, mips and memory values are provided by the user for utilization in the VM. We recommend highly that VM and task characteristics are appropriately determined to obtain evaluation results of the performance at desired levels. We used the VM with the following characteristics: bandwidth (850, 950, 1000, and 1100) bps; ram (568, 624, 1024, and 1168) bytes; mips (400, 440, and 470). Similarly task that were *I/O intensive* were: lengths (7, 9, 13, 6, 4, 8, and 12); memory (384, 440, 506, 408, 564, 681, and 503) bytes. The task reads the I/O data and then stores it in files. Users provides the values and lets say we have 10 VMs then the bandwidth, mips and ram combination is chosen randomly. E.g. for 10 VMs the value set could be 384, 440, 840, 984, 384, 440, 840, 984, 384, and 440, respectively.

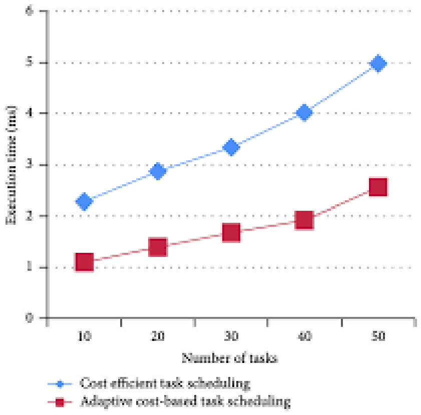


Figure 2: Computation cost comparison.

4.1. Cost of Computation

Cost required for utilization of computation resource while computing for the data access I/O request is the Computation cost. It is obtained by using equation (7).

In Figure 2 we compare the existing CETS (excluding completion cost & time) with CBAS (including completion cost and time) with respect to computation cost. *X*-axis represents number of task, while *Y*-axis measures computation cost. With 50 tasks CETS has Computation cost of 2975 while CBAS gives us a cost of 2540. Hence, CBAS provided better scheduling with lesser cost of computation.

4.2. Cost of Communication

Cost required for utilization of communication resource while communicating for the data access I/O request is the Communication cost. It is obtained by using equation (9).

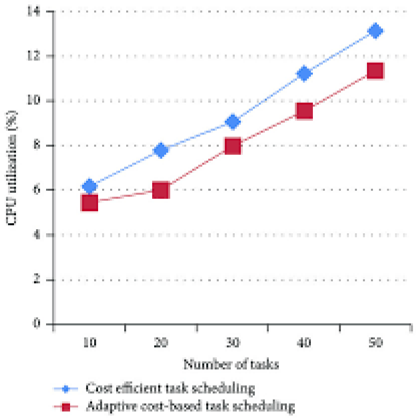


Figure 3: Communication cost comparison

In Figure 3 we compare the existing CETS (excluding completion cost & time) with CBAS (including completion cost and time) with respect to communication cost. *X*-axis represents number of task, while *Y*-axis measures communication cost. With 50 tasks CETS has communication cost of 1130 while CBAS gives us a cost of 950. Hence, CBAS provided better scheduling with lesser cost of communication.

4.3. Time of Execution

Time required for executing a task by the VM is the Execution time. It is obtained by multiplying number of instruction with execution cycles per instruction and time per cycle.

In Figure 4 we compare the existing CETS (excluding completion cost & time) with CBAS (including completion cost and time) with respect to Execution time. *X*-axis represents number of task, while *Y*-axis measures execution time (ms). With 50 tasks CETS has execution time of 4.875ms while CBAS gives us a time of 2.52ms. Hence, CBAS provided better scheduling with lesser time of execution.

4.4. Utilization of CPU

Usage processing resource or work handled amount by a CPU is the CPU utilization for that task. It varies depending upon the type and amount of computing task. It is obtained by using equation (11)

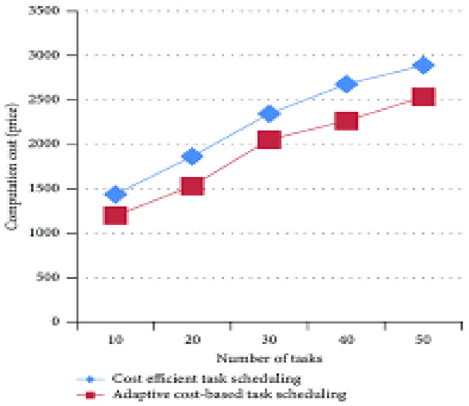


Figure 4 Execution time Comparison

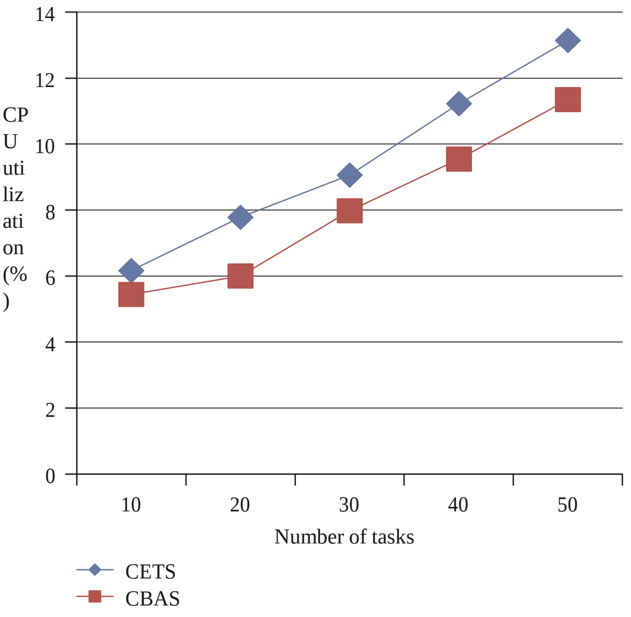


Figure 5: CPU utilization Comparison.

In Figure 5 we compare the existing CETS (excluding completion cost & time) with CBAS (including completion cost and time) with respect to CPU utilization. *X*-axis represents number of task, while *Y*-axis measures % CPU utilization. With 50 tasks CETS has CPU utilization of 13.24% while CBAS gives us a CPU utilization of 11.245%. Hence, CBAS provided better scheduling with lesser utilization of CPU.

4.5. Utilization of Bandwidth

Total amount of data that can be transferred in a specified amount of time is bandwidth and it's measured in bps(bits per second). We obtain it using equation 12.

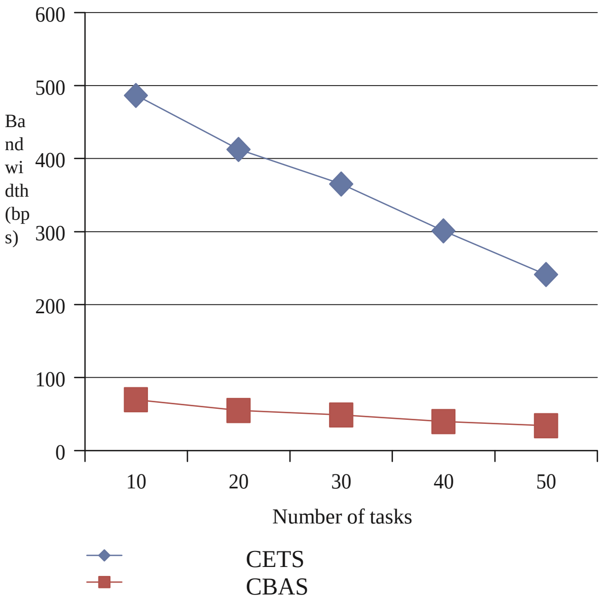


Figure 6: Bandwidth utilization Comparison.

In Figure 6 we compare the existing CETS (excluding completion cost & time) with CBAS (including completion cost and time) with respect to bandwidth utilization. *X*-axis represents number of task, while *Y*-axis measures bandwidth utilization. With 50 tasks CETS has bandwidth utilization of 240.95 bps while CBAS gives us a bandwidth utilization of 34.11bps Hence, CBAS provided better scheduling with lesser utilization of bandwidth.

Hence, from the above described results of the experiment it is clearly visible that Cost-based adaptive scheduling (CBAS) that consider the communication & computation and the completion time is more efficient in comparison to the existing CETS.

5. CONCLUSION

Task scheduling in a cloud environment with effective cost management and reduced delays is quite challenging. Thus through this research, cost-based adaptive scheduling (CBAS) is proposed which considers cost of accessing data and time for accessing data. When we consider the above mentioned factors, we can effectively fetch data from the remote data center and improve performance of scheduling. In this approach we focus on maintaining cost while providing access to data while each task is executed. Results of the Experiment also indicate the proposed cost-based adaptive scheduling provide much better performance for CPU Utilization, bandwidth utilization, communication cost, computation cost and execution time in comparison with existing CETS approach.

We have performed and proposed scheduling of task for predetermined tasks demands, while we acknowledge that it is challenging to propose scheduling methodology undetermined task demands. It may be achieved by utilization of techniques of resource provisioning efficiently in future. Dataset regeneration cost is not calculated in CBSA and it may not be fully efficient for exceptional scenarios. We may conduct further research on the same.

We would also be working on problems in Load-balancing to provide efficient services in cloud computing, since we presume it to be the next roadblock and future research scope.

6. REFERENCES

[1] "The NIST definition of cloud computing," National Institute of Standards and Technology, vol. 53, no. 6, p. 50, 2009. P. Mell and T. Grance

[2] "Cloud computing: stateof-the-art and research challenges," Journal of Internet Services and Applications, vol. 1, no. 1, pp. 7–18, 2010. Q. Zhang, L. Cheng, and R. Boutaba

[3] "A model for cost-benefit analysis of cloud computing," Journal of International Technology and Information Management, vol. 22, no. 3, article 6, 2013. K. Nanath and R. Pillai

[4] "A cost-effective deadline-constrained dynamic scheduling algorithm for scientific workflows in a cloud environment," IEEE Transactions on Cloud Computing, 2015. J.Sahniand D.Vidyarthi

[5] "A hyper-heuristic scheduling algorithm for cloud," IEEE Transactions on Cloud Computing, vol. 2, no. 2, pp. 236–250, 2014. C. W. Tsai, W. C. Huang, M. H. Chiang, M. C. Chiang, and C. S. Yang

[6] "ANGEL: agentbased scheduling for real-time tasks in virtualized clouds," IEEE Transactions on Computers, vol. 64, no. 12, pp. 3389–3403, 2015. X. Zhu, C. Chen, L. T. Yang, and Y. Xiang

[7] "Evolutionary multiobjective workflow scheduling in cloud," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 5, pp. 1344–1357, 2016. Z. Zhu, G. Zhang, M. Li, and X. Liu

[8] “PRISM: fine-grained resource-aware scheduling for MapReduce,” IEEE Transactions on Cloud Computing, vol. 3, no. 2, pp. 182–194, 2015. Q. Zhang, M. F. Zhani, Y. Yang, R. Boutaba, and B. Wong

[9] “Realtime tasks oriented energy-aware scheduling in virtualized clouds,” IEEE Transactions on Cloud Computing, vol. 2, no. 2, pp. 168–180, 2014. X. Zhu, L. T. Yang, H. Chen, J. Wang, S. Yin, and X. Liu

[10] “Scheduling jobs with unknown durationinclouds,”IEEE/ACMTransactionsonNetworking,v ol. 22, no. 6, pp. 1938–1951, 2014. S. T. Maguluri and R. Srikant

[11] “Costefficient task scheduling for executing large programs in the cloud,” Parallel Computing, vol. 39, no. 4-5, pp. 177–188, 2013. S. Su, J. Li, Q. Huang, X. Huang, K. Shuang, and J. Wang

[12] “Integrating QoS awareness with virtualization in cloud computing systems for delaysensitive applications,” Future Generation Computer Systems, vol. 37, pp. 478–487, 2014. J.-W. Lin, C.-H. Chen, and C.-Y. Lin

[13] “A highly practical approach toward achieving minimum data sets storage cost in the cloud,” IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, pp. 1234–1244, 2013. D. Yuan, Y. Yang, X. Liu et al.

[14] “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” *Software—Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya.