ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE
International Journal of Advanced Research in Computer and Communication Engineering

NCDSPICE 2016
National Conference on Digital Signal Processing, Information and Communication Engineering
Govt. Polytechnic College, Kasaragod
Vol. 5, Special Issue 4, November 2016

Cost Effective Real Time Scheduling for Cloud Based on Market Approach

Shihabudheen K M

Head of Department, Computer Engg, Govt. Polytechnic College Meppadi, Wayanad, India

Abstract: Cloud computing is a model for enabling a convenient, on demand network access to a common shared pool of configurable computing resources that can be immediately provisioned and released with least effort . The technologies used in cloud have provided ample opportunities for scalability, cost efficiency, reliability and high resource utilization. Many applications deployed on clouds are real time in nature .Performance of real time cloud applications depends on the generated results and time at which result becomes available .Scheduling has essential role in mapping real time tasks to machines such that deadlines and response time requirements are satisfied. Agent based scheduling technology is flexible to meet different requirements while scheduling. The agent based systems facilitate the interaction between processes by cooperating, coordinating and negotiating with each other. Based on this approach scheduling mechanisms are implemented and corresponding dynamic scheduling algorithms for real time tasks are executed in cloud. In this paper an approach for real time scheduling is implemented for virtualized clouds. This concept is based on agent based scheduling with deadline and cost constraints. Based on these constraints a bidirectional announcement bidding mechanism and probability strategy are developed that will give an improved scheduling of tasks. Investigated the problem of agent-based scheduling for independent real-time tasks in virtualized cloud environments and proposed mechanism for improved scheduling based on deadlines and cost constraints.

Keywords: Real time scheduling, cloud, virtual machines, bidding, cost effective ratio, market approach.

INTRODUCTION

The concept of computing is transforming to a model which consisting of services that are commoditized and delivered in a way similar to traditional utilities such as water, electricity, gas, and telephony. In such a model, users can access services based on their needs without bothering to where the services are hosted or how these are delivered. Several computing techniques have evolved to deliver these utility computing model and these include cluster computing, Grid computing, and Cloud computing. Cloud computing became an efficient model to offer computational resources as services on a "pay-per-use" basis.

In addition virtualization technology is utilized in clouds to provide flexible and scalable services, that make users the illusion of infinite resources. Running tasks on virtual machines (VMs) brought an effective solution. Utilizing the virtualization concept, a single host can simultaneously run multiple virtual machines (VMs).This concept enhances the opportunities for scalability, reliability, high resource utilization and cost-efficiency. The tasks deployed on cloud environment have the real-time in nature where result depends on both computational output generated and the time instant on which these results made available. An agent is a computer system or a program that is capable of performing independent actions, i.e. deciding for itself what needs to be done to perform its design objectives.

In a multi agent system there will have a number of agents that interact with each another. For successful interaction, agents perform operations such as cooperation, coordination, and negotiation with each other. The agent-based technology is evolved from distributed artificial intelligence (DAI) domain.

RELATED WORKS

Cloud computing requires technology for virtualization and allocation of resources. Scheduling concepts plays a vital role in it. A number of technologies and strategies have been proposed for this, which include [3] Task scheduling algorithm based on Greedy strategy in cloud, which makes good use of time in greedy strategy, small tasks and big tasks are put together for scheduling. In view of Min-Min and Max-Min algorithms, the algorithm Min-Max proposed in this paper solves the load imbalance in cloud computing.

This paper is not considering the various factors such as the price of resources and priority. [4] Greedy-Based job scheduling algorithm, which can be applied in cloud environments proposed algorithm has decreased the completion time and increased user satisfaction, the paper has drawback that it is not considering the priority and deadline of tasks which are the very important factors in real time environment. [5] Resource allocation for real

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

NCDSPICE 2016

National Conference on Digital Signal Processing, Information and Communication Engineering

Govt. Polytechnic College, Kasaragod

Vol. 5, Special Issue 4, November 2016

time tasks using cloud computing uses the concept of allocating resources for real time tasks using the "Infrastructure as a Service". In this paper formulated the problem as a constrained optimization problem and propose a polynomial-time solution to allocate resources efficiently. The solution proposed in this work can be described as a greedy strategy based on EDF (earliest deadline first). The tasks are considered in the order of task deadlines. The strategy first tries to allocate a task to VMs available from the allocations for previous tasks. If these VMs are insufficient to complete the task before its deadline, the strategy selects the cheapest set of VMs that can complete the task before the deadline. To find the cheapest set of VMs for each task, it constructs a lookup table based on the speeds and costs. The lookup table consists of a range of possible computing speeds constructed from the different types of VMs. The entries in the lookup table are sorted in order of their speeds. For a given task's workload, the greedy strategy searches the lookup table to find the lowest speed that can finish the workload before the deadline. This guarantees that the greedy strategy can find a VM allocation for all the tasks. [6] Agent based computing deals with the design and development of software agents for bolstering cloud service discovery, service negotiation, and service composition. The significance of this work is introducing an agent-based technique for constructing software tools for cloud resource management.[7] Hierarchical queue based task scheduling presents a new scheduling algorithm for scheduling tasks by considering several parameters, including the machine capacity and task priority. Inspired by the concept of multi-queue, a queue-based task scheduling algorithm for achieving a minimum completion time of task being scheduled. [8] On-line scheduling of real time services for cloud computing introduced a novel utility accrual scheduling algorithm for real-time cloud computing services. Here tasks are scheduled with the objective to maximize the total utility. The important characteristic of this approach is that it uses two different time utility functions (TUF) a profit TUF and a penalty TUF associated with each task at the same time, to model the real-time applications for cloud computing that need not only to reward the early completions but also to penalize the abortions or deadline misses of real-time tasks.

PROPOSED SYSTEM

This work devises an agent-based scheduling mechanism for cloud environment that allocates real-time tasks cost effectively and dynamically provision resources. It employs a bidirectional announcement-bidding mechanism and the mechanism consists of three different phases, called basic matching phase, forward announcement bidding phase and backward announcement bidding phase.

The mechanism also considers the elasticity of cloud by dynamically adding virtual machines to improve schedulability. Furthermore, it design calculation rules of the bidding values in both forward and backward announcement bidding phases and two strategies for selecting contractors.

The major contributions of this work are designed a bidirectional announcement bidding mechanism based on an improved contract net protocol, developed an agent-based scheduling algorithm in virtualized clouds for independent real-time tasks by considering the priority, deadline and effective cost.

A. System Model

The target system is considered as a virtualized cloud that is characterized by an infinite set of physical computing hosts which provides hardware infrastructure for creating virtualized resources for users. It can be represented as

$$H=\{h_1,h_2,h_3,.....h_n\}$$

The active host set is modeled by Hawith nelements, where H_a is subset of H. For each host $h_k$is subset of H_a, it contains a set of virtual machines.

$$Vk=\{v_{1k},v_{2k},v_{3k},.....v_{jk}\}$$

All the VMs in the cloud will constitute a virtual machine set represented as

$$V=\{V_1,V_2,V_3,.....V_j\}$$

The cloud environment consists of a set of tasks which can be represented as these tasks are independent, non-preemptive, aperiodic and importantly with deadlines.

$$T= \{t_1,t_2,t_3,..... t_n\}$$

A task ti submitted by a user can be represented by a collection of parameters, i.e.

$$t_i= \{a_i; l_i; d_i; p_i\}$$

where ai, li, di and piare the arrival time, task size, deadline, and priority of task t_i , respectively.
Let s_{ijk}be the start time of task $t_i$on VM v_{jk}. Similarly f_{ijk}represents the finish time of task $t_i$on v_{jk}. Let e_{ijk}be the execution time of task $t_i$on VM v_{jk}.

B. Scheduling Objectives

The scheduling objectives of this work are based on task guarantee ratio (TGR), priority guarantee ratio (PGR) and cost effective ratio (ECR). The scheduling algorithm tries to finish as many tasks as possible before their deadlines.

Moreover, if the system cannot finish all tasks due to heavy workload then scheduling algorithm tries to finish tasks with higher priorities. The additional factor considered is minimizing the cost of execution. Consequently the objectives modeled as follows:

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

NCDSPICE 2016

National Conference on Digital Signal Processing, Information and Communication Engineering

Govt. Polytechnic College, Kasaragod

Vol. 5, Special Issue 4, November 2016

1. Task Guarantee Ratio (TGR)

$$\max\left\{\sum_{k=1}^{|H_a|}\sum_{j=1}^{|V_k|}\sum_{i=1}^{|T|}x_{ijk}/|T|\right\}$$

2. Priority Guarantee Ratio (PGR)

$$\max\left\{\sum_{k=1}^{|H_a|}\sum_{j=1}^{|V_k|}\sum_{i=1}^{|T|}x_{ijk}\cdot p_i/\sum_{i=1}^{|T|}p_i\right\}.$$

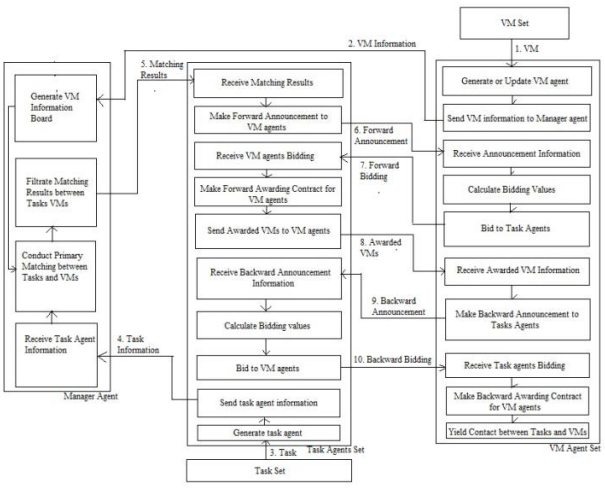
3. Effective Cost Ratio (ECR)

$$Cost(T_iR_j) = NOI(T_i) * CPI(T_iR_j)$$

Where: NOI (T_i): is the number of instructions for task T_i $CPI(T_iR_j)$:is the cost per instruction for T_i on resource R_j.

C. Bidirectional Bidding - Announcement Mechanism.

Bidirectional announcement bidding mechanism have two phases called forward announcement-bidding phase and the backward announcement-bidding phase to make a bidirectional contract between the tasks and VMs. In Forward Announcement-Bidding mechanism the announcement is initiated by the task and VMs bid to tasks and in the case of Backward Announcement-Bidding announcement is initiated by the VMs and tasks bid to Vms.



At the end of forward announcement-bidding there is a chance that multiple tasks have selected the same VM. Thus, the VM has to selects one of the tasks reversely to finish task allocation, which is taken place in the backward announcement-bidding phase. After the bidirectional announcement-bidding, the newly arrived task will be assigned to a VM or to be rejected.

The bidirectional announcement-bidding mechanism basically includes three phases, i.e., basic matching phase, forward announcement-bidding phase, and backward announcement-bidding phase.

1) Basic matching phase:

In this phase, VM scope is shrinked to only VMs that satisfyies the basic requirement of tasks and thus reduce the interactions between task agents and VM agents. The mechanism proceeds as follows

- A new task agent is generated
- The task agent sends basic task requirement information task ID, task type etc to the manager agent
- The manager agent receives the task requirement information and performs matching between each task agent requirements with VM agents resource capacity to choose those VMs that satisfy the basic requirements posted by task agents.
- The manager agent then sends the selected VMs information to corresponding task agents.

At the end of this phase each task agents receives a set VMs that have basic match with task's requirements.

2) Forward bidding phase:

In this phase, the task agents negotiate with basic matching set VM agents so as to select VM that can guarantee the timing constraint. The mechanism proceeds as follows

- The task agent set receives the VMs information from the manager agent
- Each task agent generates forward announcement information such as task arrival time, size of task, deadline, priority etc. Then sends this information to relevant VM agents.
- The VM agent receive the tasks announcement information and calculates the forward bidding values.
- The task agents receive the forward bidding values and make forward awarding contracts with VM Agent based on the basis of strategy used.

Here the forward bidding values reflects the capabilities of VM agents. The forward bidding values fb_{ijk} calculated as follows

$$fb_{ijk} = d_i - e_{ijk} - fp_{jk}$$

where fp_{jk}represents the finish time of task $t_i$preceding task $t_p$on the same VM v_{jk}. If $fb_{ijk}>0$ means that the VM v_{jk}has the ability to finish the task $t_i$before its deadline without affecting the executions of other tasks allocated to v_{jk}. Bigger value of fb_{ijk}means that $t_i$has more flexible time to run on v_{jk} before its deadline. If $fb_{ijk}<0$ indicates that VM v_{jk}cannot finish t_i before its deadline.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE
International Journal of Advanced Research in Computer and Communication Engineering

NCDSPICE 2016
National Conference on Digital Signal Processing, Information and Communication Engineering
Govt. Polytechnic College, Kasaragod
Vol. 5, Special Issue 4, November 2016

3) Backward Bidding Phase:
The backward announcement-bidding is needed to realize one-to-one match between task and VM. Here the VM agents only make backward announcement to those task agents that have forward awarding contracts with them. If a VM agent is selected only by one task agent, then the task agent and VM agent confirm a contract directly. Otherwise the task agents have to perform backward bidding and based on the strategy VM agent will select a specific task for its execution. The mechanism is described as follows

- The VM agents send backward announcement to task agents that have forward contracts with them.
- The task agents receive the VM agents announcement information and calculate the corresponding bidding values based on the strategy used.
- The VM agents receive the task agent's bidding values and make backward awarding contracts.
- If a bidirectional contract is set up between a task agent and a VM agent, the task is allocated to the VM.

The backward bidding value is represented as bb_{ijk} represents the bidding value of task $t_i$to v_{jk}, and is calculated as

$$bb_{ijk} = \frac{p_i^{\theta} \cdot e_{ijk}}{(d_i - f_{ij}) \cdot \sum_{k=k_1}^{K} \sum_{j=j_1}^{J} fb_{ijk}}$$

where the parameter Θ represents the weight of priority, J denotes the count of VM agents that have bid to task agent t^A_i. K denotes the count of hosts in which there exist VM agents that have bid to task agent t^A_iin the forward announcement-bidding phase. The backward bidding value is based on the following considerations. Firstly, the higher priority, then stronger requirement to allocate the task. Secondly, the tighter a task's finish time approaches its deadline, then higher likelihood this task cannot be allocated successfully, thus it should be allocated preferentiably. Thirdly, the smaller number of VMs to execute a task, the smaller feasibility to finish this task, hence the task should be allocated with higher preference.

D. Section Strategies.
In both the announcement-bidding phases, an announcer is responsible for selecting a bidder to award a contract if the bidder has the ability to execute this task. In this work two different selection strategies are used, one is caleedMAX strategy and other is called asP strategy.
In the case of MAX strategy when more than one bidder simultaneously bid to one announcer, the announcer will choose the bidder with maximal bidding value. While P strategy means probability selection strategy, in this case the announcer will select a bidder according to probability policy. As there are two selections in the bidirectional announcement bidding mechanism, four kinds of scheduling algorithms can be constructed
1) MAX-MAX
2) MAX-P
3) P-MAX
4) P-P

E. Scheduling algorithms.
The algorithms consider the Priority, Deadline and Cost efficiency of tasks. There are four different algorithms for Manager agent, Task agent, VM agent and resource scaling up function.

Algorithm 1 - Algorithm for Manager agent:
1) Assign allthe tasks arrive at the same time instant to list T_{wait};
2) foreach t^A_i in T_{wait} do
3) Assign the VM agents that satisfy the basic requirements of $t^A_i$to list V^A_i;
4) Send the list V^A_i to task agent t^A_i;
5) while $T_{wait}$is not empty do
6) The task agents start the forward announcement bidding phase using Algorithm 2;
7) The VM agents start the backward announcement bidding phase using Algorithm 3;

Algorithm 2 - The Algorithm for Task Agent:
1) valueList ←Φ;
2) foreach v^A_{jk} in V_i do
3) Task agent t^A_i sends the announcement information to V^A_{jk};
4) fbijk ← agent v^A_{jk} calculates the forward bidding value;
5) fbijk ≥ 0 then
6) valueList:add(fbijk);
7) if valueList ≠Φ then
8) Task agent t_i selects a bidder v select based on the values in valueList using the MAX/P Strategy;
9) else
10) v_{new}← scaleUpResources();
11) if v_{new}≠NULL then
12) The new VM v_{new} generates a new VM agent v^A new and sends the information to the manager agent;
13) Allocate t_i to v_{new};
14) $T_{waiting}$. remove(t^A_i);
15) else
16) Reject t_i;
17) $T_{waiting}$. remove(t^A_i);

Algorithm 3 - The Algorithm for VM Agent :
1) $T_{candidate}$← the task agents that send the forward contract with the VM agent v^A_{jk};
2) if $T_{candidate}$ ≠ NULL then
3) valueList ←Φ;

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

NCDSPICE 2016

National Conference on Digital Signal Processing, Information and Communication Engineering

Govt. Polytechnic College, Kasaragod

Vol. 5, Special Issue 4, November 2016

4) foreach t^A_i in $T_{candidat}$e do
5) $bb_{ijk} \leftarrow t^A_i$'s backward bidding value;
6) valueList:add(bb_{ijk});
7) VM agent v^A_{jk} selects a bidder t^A select based on the values in valueList using the MAX/P Strategy;
8) Build a bidirectional contract between the v^A_{jk} and t^A_{select}
9) $T_{waiting}$.remove(t^A_i);

Algorithm 4 - Function scaleUpResources() :
1. Create newVM with processing power P_{new};
2. find ←FALSE; v_{new}←NULL;
3. foreach h_k in H_a do
4. if h_k can accommodate newVM then
5. Allocate newVM to h_k;
6. v_{new} ← newVM ; find ← TRUE;
7. break;
8. if find == FALSE then
9. sourceHost ← Migrate VMs among the hosts to make room for newVm;
10. if sourceHost ≠ NULL then
11. Allocate newVm to sourceHost;
12. Allocate newVm to sourceHost;
13. v_{new}← newVM; find ← TRUE;
14. if find == FALSE then
15. Turn on a host hnew in H- H_a;
16. if the capability of hnew satisfies P_{new} then
17. Allocate newVM to h_{new};
18. v_{new}← newVM;

IMPLEMENTATION

CloudSim is used for the simulation of cloud computing environment. The core hardware infrastructure services related to the Clouds are modeled in the simulator by a Data center component for handling service requests.

These requests are for the VMs, which need to be allocated a share of processing power on Datacenter's host components. A Datacenter is composed by a set of hosts, which is responsible for managing VMs during their life cycles. Host is a component that represents a physical computing node in a Cloud, it is assigned a pre-configured processing, memory, storage, and a scheduling policy for allocating processing cores to virtual machines [2].
Implementation of agent based scheduling for real time tasks in virtualized clouds Following steps are performed

1. Create cloud information service (CIS).
2. Create manager agent.
3. Create task agents based on received set of tasks.
4. Create datacenter, Each datacenter have some host and each host have a set of virtual machines.
5. Create broker that will submit tasks to the datacenter, broker initially talks to CIS and retrieve the information which are registered with CIS.
6. Create cloudlets and submitted to broker.
7. If set of task agents are greater than the set of VMs then create additional Virtual Machines
8. Perform Forward announcement bidding, execute Effective cost ratio function and task agents make forward awarding contracts for VM agents, go to 10
9. If task agent unable to make forward awarding contract with any of the VM agents then
4. execute ScaleUpResource function to add new VM from set of Hosts, go to 11
10. Perform Backward announcement bidding and the task agent make backward awarding contracts for VM Agent.
11. Execute the task agent on VM agent to which the backward contract is made or newly created VM by using ScaleUpResource function.

The MAX strategy and P strategy can be employed in either the forward announcement bidding phase or the backward announcement-bidding phase.

Thus, four kinds of scheduling algorithms can be generated by employing different selection strategies.

We can use M-M, M-P, P-M, and P-P to denote the four scheduling algorithms based on MAX-MAX, MAX-P, P-MAX and P-P strategies, respectively. The performance metrics by which that evaluate the system performance include:

1. Task Guarantee Ratio (TGR) : TGR=Total count of tasks guaranteed to meet their deadlines/Total count of tasks;
2. Priority Guarantee Ratio (PGR): PGR=Sum of priorities of tasks that are finished before their deadlines/Sum of priorities of all tasks.
3. Effective Cost Ratio (ECR): ECR= Ratio of cost of execution of task on selected virtual machine to cost of execution of task on highest cost virtual machine.

RESULTS AND DISCUSSIONS

The works implement and execute four different algorithms called M-M, M-P, P-M and P-P and a benchmark algorithm called I-Greedy is also executed. The I-Greedy is based on a single directional bidding algorithm. Fig. 1 shows the performances of the five algorithms in terms of task guarantee ratio (TGR).

It can be observed from Fig. 1 that the TGRs of MM, M-P, P-M and P-P are higher than that of I-Greedy because the work employ the bidirectional announcement bidding mechanism which considers both the conditions of tasks and VMs. As a result, they are able to reach better global solutions.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

NCDSPICE 2016

National Conference on Digital Signal Processing, Information and Communication Engineering

Govt. Polytechnic College, Kasaragod

Vol. 5, Special Issue 4, November 2016

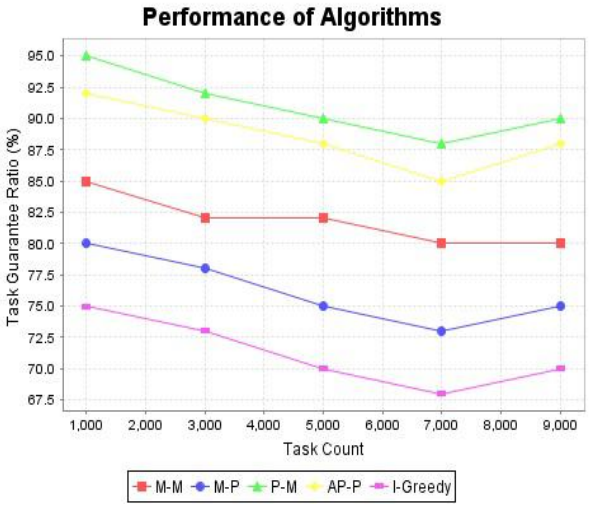


Fig. Performance graph for Task Guarantee Ratio

From Fig. 2, it can observe that M-M, M-P, P-M and P-P exhibit a better performance than I-Greedy in terms of Priority Guarantee Ratio because I-Greedy dose not take the task priority into consideration during the scheduling.

However, when M-M, M-P, P-M and P-P calculate the bidding value in the backward announcement-bidding phase, the priority is taken into account. Therefore, a task with higher priority is more likely to be allocated successfully even under the random sequence, which makes work achieve higher PGR.

It can be concluded from Fig.1 and 2 that M-M, M-P, P-M and P-P are superior to I-Greedy with regards to both PGR and TGR. The proposed algorithms can efficiently solve the scheduling problem for real-time tasks and fully utilize the main advantage of clouds and dynamically vary according to the current state of the system.

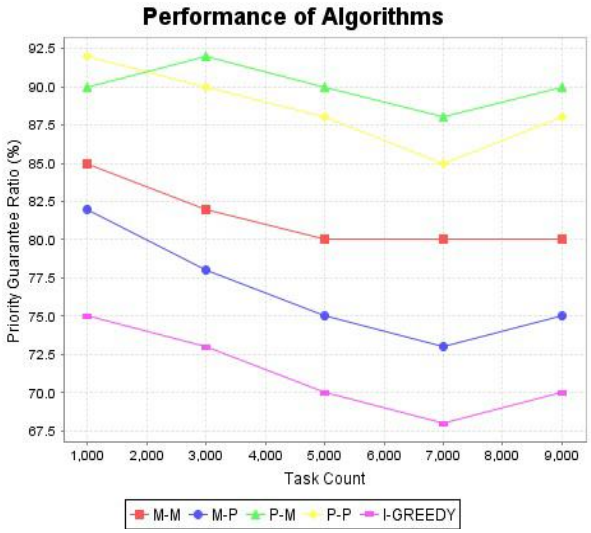


Fig. 2 Performance graph for Priority Guarantee Ratio

Fig.3 shows that the ECR of the proposed system has decreased compared to the existing work. The graph shows the the ECR of the proposed work are lowered as per the objectives of the work this reductionin ECR values are due to effective scheduling of tasks based on the cost of VMs to which which they are allocated and tasks priority. Thus all the three objectives of the work ie. increase the Task Guarantee Ratio and Priority Guarantee Ratio at the same time decrease Effective Cost Ratio of tasks are met effectively as per the resultant graphs generated by running the simulation.

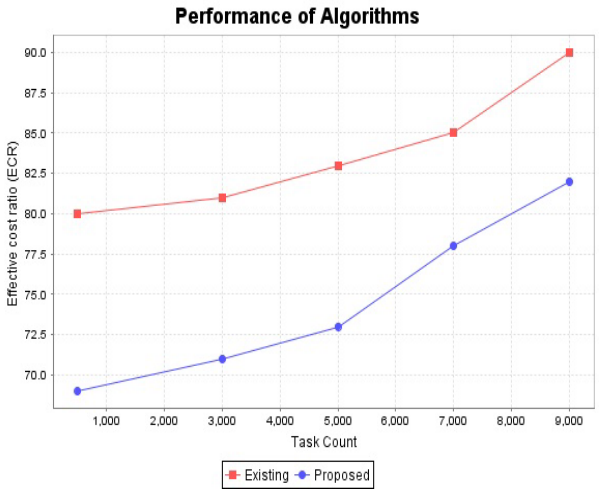


Fig. 3 Performance graph for Cost Effective Ratio

CONCLUSION

In this work, it has investigated the problem of agent-based scheduling for independent real-time tasks in virtualized cloud environment and proposed dynamic scheduling algorithm. The work employs a new bidirectional announcement bidding mechanism, in which the contributions include designing the basic matching policy, forward announcement bidding, backward announcement bidding mechanisms and cost effective approach for scheduling the tasks. Two selection strategies called Max strategy and P strategy were put forward to determine the contractors.

Again, it is sufficiently considered the elasticity of clouds and proposed a scalingup policy to dynamically add VMs so as to enhance the system schedulability. The work comprehensively addresses the issue of schedulability, priority, scalability, realtime in virtualized cloud environment. As future work the extension can be done to address a new scheduling mechanism for the real time cloud environment in which the task's communication time during scheduling operation are taken into account. The other area of extension is to add the concept of energy conservation by performing shutting down of idle VMs till next batch of tasks arrive.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

NCDSPICE 2016

National Conference on Digital Signal Processing, Information and Communication Engineering

Govt. Polytechnic College, Kasaragod

Vol. 5, Special Issue 4, November 2016

REFERENCES

[1] Xiaomin Zhu,Chao Chen, Laurence T. Yang and Yang Xiang, "ANGEL: Agent-Based Scheduling for Real-Time Tasks in Virtualized Clouds", IEEE TRANSACTIONS ON COMPUTERS, VOL. , NO. , 2015 (This article has been accepted for publication in a future issue of this journal. Citation information: DOI 10.1109/TC.2015.2409864, IEEE Transactions on Computers)

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," Future Generation Comput. Syst., vol. 57, no. 3, pp. 599-616, 2009.

[3] Zhou Zhou and Hu Zhigang "Task Scheduling Algorithm based on Greedy Strategy in Cloud Computing" The Open Cybernetics & Systemics Journal, 2014, 8, 111-114.

[4] Ji Li, Longhua Feng, Shenglong Fang "An Greedy-Based Job Scheduling Algorithm in Cloud Computing" Journal of software, VOL. 9, NO. 4, APRIL 2014, 921-924.

[5] Karthik Kumar, Jing Feng, Yamini, and Yung-Hsiang Lu "Resource Allocation for Real-Time Tasks using Cloud Computing" IEEE Trans. Network and Service Management, vol. 10, no. 1, pp. 43-55, 2013.

[6] Kwang Mong Sim, Senior Member, IEEE "Agent-Based Cloud Computing" IEEE Transaction on services computing, VOL. 5, NO. 4, October-December 2012 1939-1374

[7] Wanqing You, Kai Qian, and Ying Qian "Hierarchical Queue-Based Task Scheduling" Journal of Advances in Computer Networks, Vol. 2, No. 2, June 2014 DOI: 10.7763/JACN.2014.V2.98

[8] X. Qin and H. Jiang, "A Novel Fault-Tolerant Scheduling Algorithm for Precedence Constrained Tasks in Real-Time Heterogeneous Systems," Parallel Comput., vol. 32, no. 5, pp. 331-356, 2006.

[9] K. Plankensteiner, R. Prodan, T. Fahringer, A. Kertesz, and P. Kacsuk, "Fault-Tolerant Behavior in State-of-the-Art Grid Workflow Management Systems," Technical Report TR-0091, Inst. On Grid Information, Resource and Workflow Monitoring Services, CoreGRID-Network of Excellence,2007.

[10] H. M. Fard, R. Prodan, and T. Fahringer, "A Truthful Dynamic Workflow Scheduling Mechanism for Commercial Multicloud Environments," IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 6, pp. 1203-1212,2013.