International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 9, ISSN No. 2455-2143, Pages 103-106
Published Online July – August 2016 in IJEAST (http://www.ijeast.com)

MULTI-TARGET TASKS ASSIGNMENT PLANNING FOR CLOUD-ENVIRONMENT THROUGHPUT OPTIMIZATION

Shubhashree S R
M.tech in Computer Science
Jawaharlal Nehru National College of Engineering
Shivamogga, Karnataka

Abstract— **In cloud-computing data-centers serves to update the resources efficiency. On each data-center many virtual-machines (VMs) are running to use the resources effectively. So, a multi-target assignment algorithm is proposed to enhance the data-center throughput for mapping tasks to a virtual-machine and to reduce the expenses without ignoring SLA (Level of Service Agreement) in cloud-environment. So, this proposed algorithm gives an exact planning method of scheduling. Most of the algorithms will be planning to schedule the tasks that are based on each principle (i.e. running time). Also, the different principles are to be considered such as time of execution, expenses, productivity and so on.**

Keywords— **VMs (Virtual-Machines), QOS (Quality-Of-Services)**

I. INTRODUCTION

Cloud-Computing is the newest and rising example in conveyed registering that energizes programming applications, stage, and equipment foundation as an organization. Cloud administration suppliers offer these organizations in perspective of revamp Administration Level Understandings (SLAs) which describe customer's required Nature of Administration (QOS) parameters. Distributed computing reduces speculation on various resources like hardware, programming and allow advantages for be leased and released. Decreases beginning venture, upkeep cost and working cost. Cloud Organizations are encouraged on administration supplier's own specific base or on untouchable cloud base supplier. Generally, three sorts of administrations are given Platform-as-a-Service (PaaS), Infrastructure-as-as-Service (IaaS) and Software-as-a-Service (SaaS). Cloud customers use these administrations at whatever point they require by using pay-per-use model.

The rest of the paper is organized as follows. Proposed algorithm and non-dominated sorting algorithm are explained in section II. Experimental results are presented in section III. Concluding remarks are given in section IV.

II. PROPOSED ALGORITHM

A. Non-Dominated Sorting algorithm –

This sorting technique is used to solve the multi-tasking issues. Also, multiple-objective functions are to be considered in multi-tasking sorting problem. So, the main aim is to reduce the time of execution of a particular task in the proposed algorithm. Also, the aim is reached by choosing a task with least size of task low value of QOS.

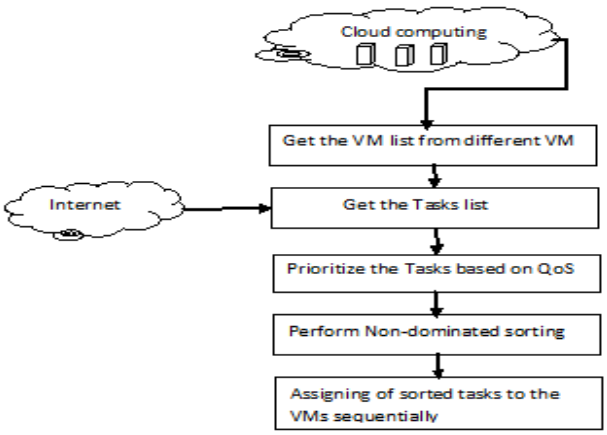

Fig. 1. Multi-Objective task scheduling and dominance relation

A non-dominated sorting in light of the prevalence associations as showed up in Fig 1, game plans in the populaces are appointed to different fronts. The plan of non-ruled courses of action is moreover called as the Pareto-Front. Relationships of transcendence association between the courses of action are the essential movement in non-ruled sorting. In proposed calculation as shown in Fig 2, quality examination is performed on the reason of target programming approach.

International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 9, ISSN No. 2455-2143, Pages 103-106
Published Online July – August 2016 in IJEAST (http://www.ijeast.com)

Step 1: Non-dominated sorting (task list)
i=0
Step 2: Create empty non-dominated list
Step 3: Initially put task (i) into non-dominated list
For all i=1 to size of task list
For all j=0 to size of non-dominated list
Step 4: If task (j) dominates task (i) then put task (j) into non-dominated set
Else
Step 5: If task (i) dominates task (j) then put task (i) into non-dominated set
Else
Step 6: Put task (i) and task (j) into non-dominated set
End if
End for
End for

Fig 2. Non-Dominated Sorting algorithm

B. Proposed algorithm – Multi-Target Tasks Scheduling algorithm

With the progressive consumption and value heightening of conventional energy, working data-center in energy proficient way is a developing critical issue. Nevertheless, most existing researches of task assignment in data-center did not take into full thought how to diminish energy utilization.

This algorithm creator proposed multi-task scheduling algorithm that enhances the data center execution without damaging SLA. The proposed algorithm is as appeared in Fig 3, that utilizations non-dominating sorting algorithm for comprehending the multi-objective (task size, QOS value). After a fixed time interval the list will be updated dynamically. This algorithm will give the optimized throughput when compared with the existing algorithm.

Advantages:

- It diminishes the execution time.
- It enhances throughput of the data-center.

According to the below steps the process will be initiated. Firstly, cloud provider has to give the list of VMs and list of tasks to the cloud-broker. Then broker creates the list of received VMs and tasks. After this step, the sorting will be done by taking list of tasks as variable. Initially, the task will be initialized to zero. Then create an empty non-commanded sorted list. So, one by one task will be putting into the sorting list. Again sort the list of tasks according to the set of non-commanded task list. Finally, sort the list of VMs in decreasing order and bind the tasks from the tasks list to the list of virtual-machines.

1. Submit both VMs list of successfully created VMs in datacenter and t Broker.
2. Create a received list of tasks.
3. Create a received list of VMs.
4. Non-dominated sorting (list of task)
$i \leftarrow 0$
Create empty non-dominated list
dominated list ←list of task
Initially put $task_i$ in the non-dominated list
for all $i \leftarrow 1$ to size of task's list **do**
for all $j \leftarrow 0$ to size of non-dominated list **do**
if $task_j$ dominates $task_i$ **then**
put $task_j$ into non dominated set
else
if $task_i$ dominates $task_j$ **then**
put $task_i$ into non dominated set
end if
else
put $task_i$ and $task_j$ into non dominated set
end if
end for
end for
5. Sort the list of task according to the non-dominated task set.
6. Sort the VM received list in descending order .
7. $j \leftarrow 0$ **7.**
for all $i \leftarrow 0$ to the size of task's list **do**
if $j \geq 0$ **then**
Bind $task_i$ to the VM_j j++
if j== number of VMs **then**
j=0
end if
end if
end for

Fig 3. Multi-Target Tasks Scheduling algorithm

III. Experiment and Result

The test set for this evaluation of tasks is randomly selected from the internet. Java and Oracle software are used to perform this experiment.

The proposed scheme is tested using ordinarily text file uploading, downloading and creation processing. From the simulation of the experiment results, we can draw to the conclusion that this method is execution time is minimum to many kinds of existing tasks scheduling algorithms.

Fig 4, provides the key difference between two algorithms i.e. RSA (Rivest-Shamir-Adlemon) algorithm holds 1024 bits maximum of 80% as well as proposed algorithm i.e. ECC(Elliptic Curve Cryptography)holds 256 bits minimum of 20%.
Consider time difference of the selected file between two algorithms i.e. RSA and ECC by clicking on time difference button. In this stage, RSA holds 2.887 seconds of execution time and ECC holds 0.203 seconds of execution time for which the file had been selected as shown in Fig 5.

International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 9, ISSN No. 2455-2143, Pages 103-106
Published Online July – August 2016 in IJEAST (http://www.ijeast.com)

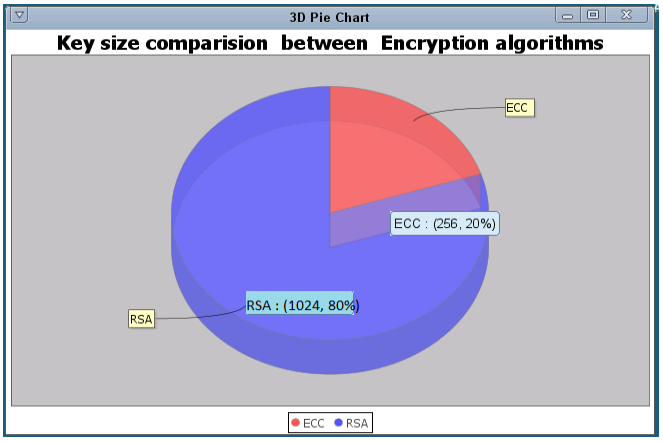

Fig 4.Key Size Comparison between two algorithms

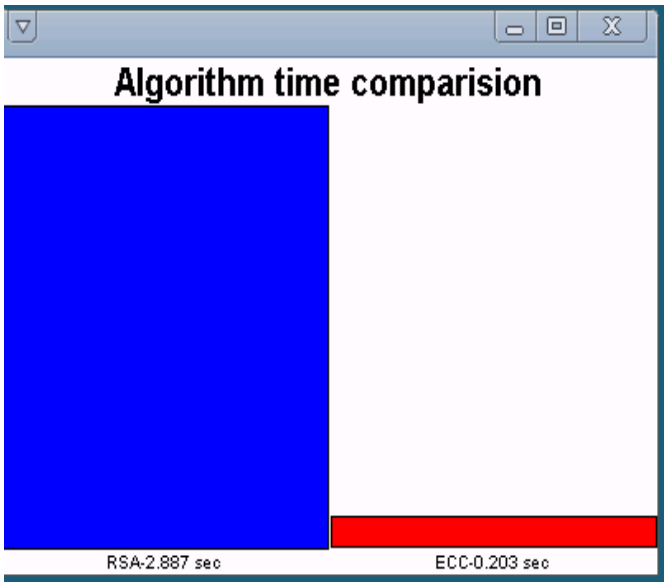

Fig 5. Execution Time Comparison between two algorithms

Consider graph of analysis by clicking on simulation button. This will provide the simulated output by comparing FCFS and Priority-based task-scheduling algorithm with proposed algorithm based on workload in x-axis and turn-around time in y-axis as shown in Fig 6. The examination of turnaround time is appeared and three tasks scheduling calculations with six distinct workloads as organized in Table 1. From the correlation of these three undertakings assignment calculations are to be considered that the proposed calculation performs superior to anything compared to the other two calculations with least execution-time and expanded throughput in the cloud-environment framework.

Table 1 Workload

Workload	Number of VMs	Number of Task
Workload 1 (10,000)	**1**	**10,000**
Workload 2 (20,000)	**1**	**20,000**
Workload 3 (30,000)	**2**	**30,000**
Workload 4 (40,000)	**2**	**40,000**
Workload 5 (50,000)	**3**	**50,000**

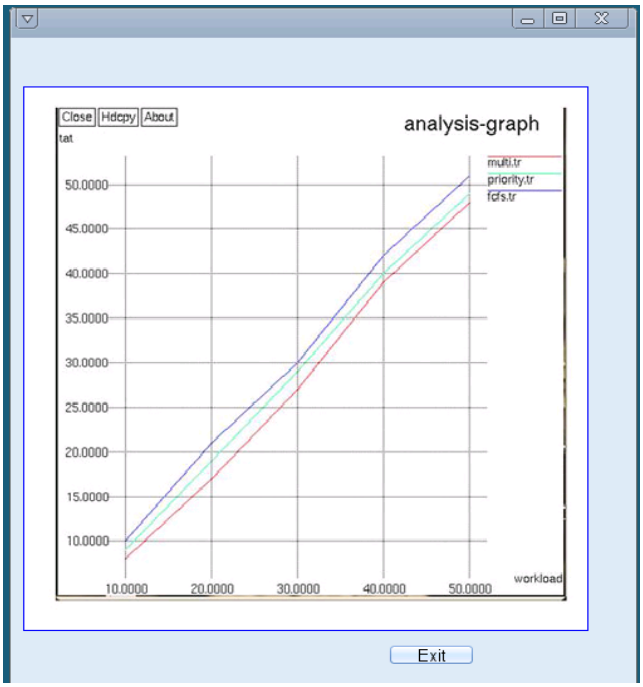

Fig 6. Analysis Graph of Workloads v/s Turn-around Time

IV. CONCLUSION

In cloud-environment, the proposed task-scheduling algorithm is one of the ideal methods for assignment of tasks in which it provides the general time of execution. Cloud-computing works effective for scheduling process. Also, the calculation

of this current algorithm can be enhanced with some parameters of QOS. Also, the multi-objective target-assignment calculation provides QOS with the better enhancement as discussed in early chapters. The proposed algorithm is guaranteed that the execution time, key size and workload of the particular task is efficient than compared to other existing algorithm calculations. As seen in the above results chapter, finally got the minimum time of execution result as expected. According to these results, execution time is the Quality-Of-Service parameter in proposed algorithm which got the minimum execution time as QOS result. Hence, this multi-objective task-scheduling algorithm is an optimal scheduling algorithm and it is more enhanced to provide minimum execution time in the cloud environment.

V. REFERENCE

[1] Shamsollah Ghanbaria, Mohamed Othmana, "Job Scheduling Algorithm based on priority in Cloud-Computing", ICASCE 2012.

[2] Ravi Iyer, Ramesh Illikkal, "Monitoring of Shared Resources and Throughput Streamlining in Distributed computing Data-Centers", 2011 IEEE International.

[3] Shamsollah Ghanbaria, Mohamed Othmana, "Enhanced Min-Max Algorithm Using Petri-Net and Load Modifying", ICASCE 2012.

[4] [4] Das, A.K, Adhikary, T. Razzaque, "An insightful framework for virtual-machine and provisioning of QOS in Cloud-Environment", March 2013.

[5] [5] Shalmali Ambike, Dipti Bhansali, "An Optimistic Isolated Job-Scheduling Framework for Distributed-Computing", www.ijera.com Mar- Apr 2012.

[6] [6] Quyet Thang NGUYEN, Nguyen QUANG-HUNG, Nguyen HUYNH TUONG, Van Hoai TRAN, Nam THOAI, "Allocation of virtual-machine in Cloud-Environment for Reducing Complete Time of execution on each machine", 2013.

[7] [7] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T Meyarivan, "A Non-Dominated Sorting Genetic-Algorithm for Multi-Target Enhancement", March 2014.

[8] [8] Hilda Lawrance, Dr. Salaja Silas, "Productive QOS based Resource Scheduling for Distributed Computing utilizing PARPARIKA strategy", March 2013.

[9] [9] Yogita Chawla, Mansi Bonsle, "Cost-based task-scheduling for Dynamic-Optimization in Cloud-Computing", June 2013.

[10] [10] Rajeev Kaur, Supriya Kinger, "Job Scheduling Algorithm Analysis in Cloud-Computing", March 2014.

[11] [11] Atul Vikas Lakra, Dharmendra Kumar Yadav, "Multi-Objective Task Scheduling Algorithm for Cloud Computing Throughput Optimization", ICCC 2015.

[12] http://www.explainthatstuff.com/cloud-computing-introduction.html

[13] http://cloudcomputing.ieee.org/education-careers

[14] http://searchcloudcomputing.techtarget.com/definition/cloud-computing

[15] https://cloud.google.com

[16] http://cloudthat.in/course/level-1-fundamentals-of-cloud-computing