© 2017 IJNRD | Volume 2, Issue 6 June 2017 | ISSN: 2456-4184

LOAD BALANCING OF TASKS USING HYBRID TECHNIQUE WITH ANALYTICAL METHOD OF ESCE & THROTTLED ALGORITHM

[1]MANDEEP KAUR, [2]NAVTEJ SINGH GHUMMAN
[1]M.TECH STUDENT, [2]ASSISTANT PROFESSOR
DEPARTMENT OF COMPUTER SIENCE
SHAHEED BHAGAT SINGH STATE TECHNICAL CAMPUS, FEROZEPUR.

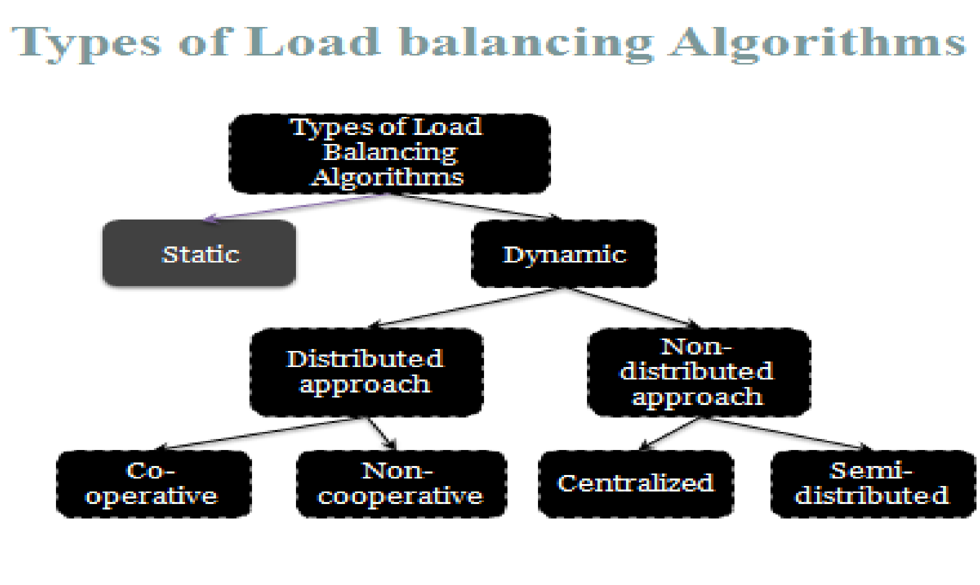
Abstract: Cloud computing is delivery of computing as service rather than as a product. Cloud computing provides a variety of computing resources, from servers and storage to enterprise applications such as email, security, backup, voice, all delivered over the Internet. Cloud computing is a way to increase the capacity or add capability dynamically without investing in new infrastructure, training new personnel, or licensing new software. Cloud computing helps in making internet the ultimate resource of all computing needs. It provide one common platform to manage/monitor/setup/scale. It helps to increase the security. The utilization of hardware with better resource management. Helps in saving money and time by switching to cloud computing. The objective of the algorithm is to improve the response time and processing time in SpaceShared in host level and SpaceShared in VM level. To simplify this process, in this paper we propose CloudSim: an extensible simulation toolkit that enables modelling and simulation of Cloud computing environments. The CloudSim toolkit supports modelling and creation of one or more virtual machines (VMs) on a simulated node of a Data Center, jobs, and their mapping to suitable VMs. It also allows simulation of multiple Data Centers to enable a study on federation and associated policies for migration of VMs for reliability and automatic scaling of applications.

1. INTRODUCTION

Cloud computing can be defined as “a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers”. Load balancing is the process of improving the performance of the system by shifting of workload among the processors.Load balancing is done so that every virtual machine in the cloud system does the same amount of work. One of the critical issue of cloud computing is to divide the workload dynamically.Load balancing is to equalize the workload among the nodes by:

- Minimizing response time
- Minimizing communication delays
- Minimizing resource utilization
- Maximizing throughput

Following figure defines the types of Load-Balancing Algorithms:



2.1 Key terms & Technologies

Service Request Examiner and Admission Control: When a service request is first submitted, the Service Request Examiner and Admission Control mechanism interprets the submitted request for QoS requirements before determining whether to accept or reject the request. Thus, it ensures that there is no overloading of resources whereby many service requests cannot be fulfilled successfully due to limited resources available. It also needs the latest status information regarding resource availability (from VM Monitor mechanism) and workload processing (from Service Request Monitor mechanism) in order to make resource allocation decisions effectively. Then, it assigns requests to VMs and determines resource entitlements for allocated VMs.

Pricing: The Pricing mechanism decides how service requests are charged. For instance, requests can be charged based on submission time (peak/off-peak), pricing rates(fixed/changing) or availability of resources (supply/demand). Pricing serves as a basis for managing the supply and demand of computing resources within the Data Center and facilitates in prioritizing resource allocations effectively.

Accounting: The Accounting mechanism maintains the actual usage of resources by requests so that the final cost can be computed and charged to the users. In addition, the maintained historical usage information can be utilized by the Service Request Examiner and Admission Control mechanism to improve resource allocation decisions.

VM Monitor: The VM Monitor mechanism keeps track of the availability of VMs and their resource entitlements.

Dispatcher: The Dispatcher mechanism starts the execution of accepted service requests on allocated VMs.

Service Request Monitor: The Service Request Monitor mechanism keeps track of the execution progress of service requests.

VMs: Multiple VMs can be started and stopped dynamically on a single physical machine to meet accepted service requests, hence providing maximum flexibility to configure various partitions of resources on the same physical machine to different specific requirements of service requests. In addition, multiple VMs can concurrently run applications based on different operating system environments on a single physical machine since every VM is completely isolated from one another on the same physical machine.

Physical Machines: The Data Center comprises multiple computing servers that provide resources to meet service demands. In the case of a Cloud as a commercial offering to enable crucial business operations of companies, there are critical QoS parameters to consider in a service request, such as time, cost, reliability and trust/security. In particular, QoS requirements cannot be static and need to be dynamically updated over time due to continuing changes in business operations and operating environments. In short, there should be greater importance on customers since they pay for accessing services in Clouds. In addition, the state-ofthe-art in Cloud computing has no or limited support for dynamic negotiation of SLAs between participants and mechanisms for automatic allocation of resources to multiple competing requests. Recently, we have developed negotiation mechanisms based on alternate offers protocol for establishing SLAs . These have high potential for their adoption in Cloud computing systems built using VMs. Commercial offerings of market-oriented Clouds must be able to:

- support customer-driven service management based on customer profiles and requested service requirements,
- define computational risk management tactics to identify, assess, and manage risks involved in the execution of applications with regards to service requirements and customer needs,
- derive appropriate market-based resource management strategies that encompass both customer-driven service management and computational risk management to sustain SLA-oriented resource allocation,
- incorporate autonomic resource managementmodels that effectively self-manage changes in service requirements to satisfy both new service demands and existing service obligations, and
- leverage VM technology to dynamically assign resource shares according to service requirements.

2.2 Performance criteria:

2.2.1 CPU Utilization We want to keep the CPU as busy as possible that means CPU is not free during the execution of processes. Conceptually the CPU utilization can range from 0 to 100 percent.

2.2.2 Throughput: If the CPU is executing processes, then work is being completed. One measure work is the number of processes that are completed per time unit that means the number of tasks per second which the scheduler manages to complete the tasks.

2.2.3 Response Time: In an interactive system, turnaround time may not be best measure. Often, a process can produce some output fairly early and can continue computing new results while previous results are being output to the user. Thus, response time is the time from the submission of a request until the first response is produced that means when the task is submitted until the first response is received. So the response time should be low for best scheduling.

2.2.4 Turnaround Time: Turnaround time refers to the total time which is spend to complete the process and is how long it takes the time to execute that process. The time interval from the time of submission of a process to the time of completion is the turnaround time. Total turnaround time is calculation is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU, and doing I/O.

2.2.5 Waiting Time: The waiting time is not the measurement of time when a process executes or does I/O completion; it affects only the amount of time of submission of a process spends waiting in the ready queue. So the Waiting time is the period of spent waiting in the ready queue to submit the new arriving process for the CPU.

2.2.6 Correctness: For algorithms which permit the user to attach deadlines to a task, this measures the proportion of time or proportion of total scheduling time unit when a task fails to meet its deadline that means all the task should be meet at the given deadline.

2.2.7 Overhead: Overhead refers to the proportion of time wasted due to computation of the schedule, and the system overhead due to context switching the tasks when new task is arriving.

2.2.8 Preventability: Predictability refers to the degree to which a task runs in a conventional manner that is it takes approximately the same time or with the same cost, irrespective of the load on the system.

2.2.9 Efficiency: Efficiency refers to the respective of system when CPU is busy for scheduling of new arriving tasks.

2.2.10 Fairness: In the absence of user or system supplied criteria for selection, the scheduler should allocate a fair amount of the resource to each task.
2.2.11 Load balancing: The scheduler should balance the load across other system resources, such as memory and I/O usage.

3. PROPOSED ALGORITHM

The objective of the algorithm is to improve the response time and processing time in SpaceShared in host level and SpaceShared in VM level.
The expected response time can be determined by the following formula:
Response Time = Finish Time – Arrival Time
Where ,Finish time is calculated by the formula:
Finish Time = No of instructions/(capacity x No of processors)
Where Capacity is the average processing capacity (in MIPS) of a core for job.

Problem Defination

1. In real time environment with number of virtual machines available when a job is requested to DCB , it computes execution time with all the virtual machines
2. Algorithm spends too much time in finding the relevant Virtual machine
3. It suffers from under utilization of resources in the cloud environment.

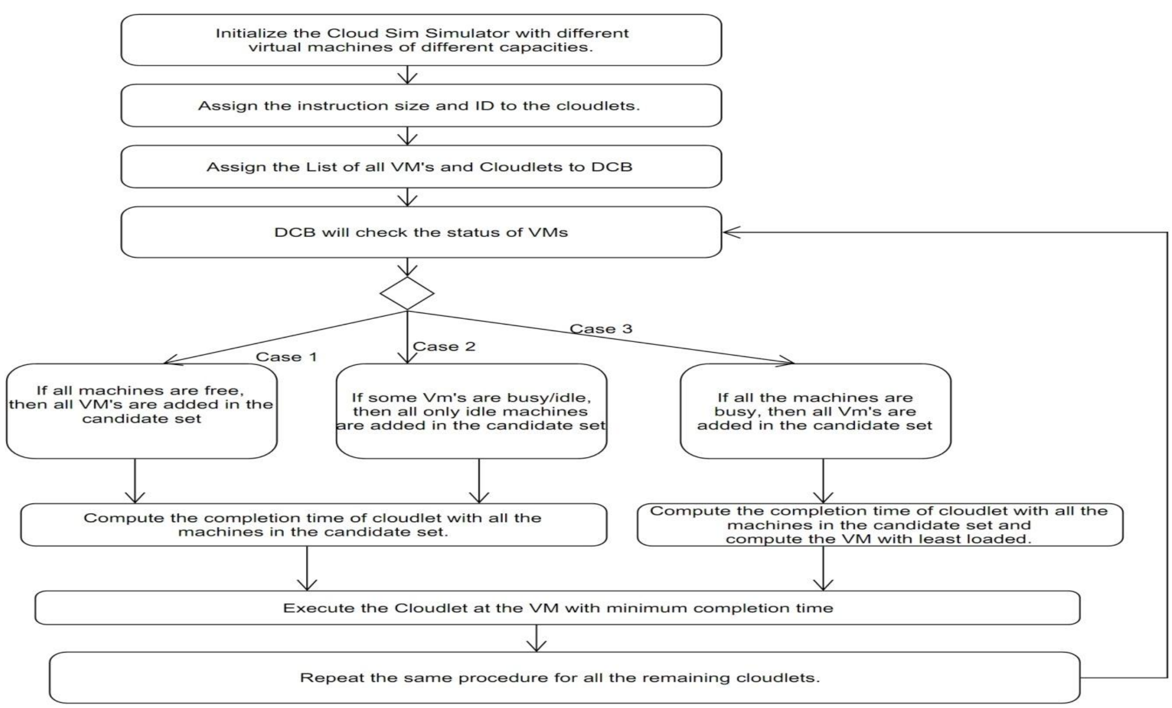
Objectives

- To properly manage the resources of service provider.
- To achieve optimal resource consumption, least average response time and avoid overload.
- To improve response time and processing cost of algorithm
- Compare the results of Proposed Algorithm with existing Algorithm.

Proposed Algorithm

- In proposed algorithm we'll try to solve the above said problem by using 3 cases
- CASE 1: If all the Virtual Machines are Available, then include all of them in the candidate set for calculating the completion time.
- CASE 2: If some Virtual Machines are Busy and some are Available, then include only those machines which are currently available in the candidate set.
- CASE 3: If all Virtual Machines are Busy then compute the completion time with each virtual machine and add them to candidate set.
- Now, When there is request to allocate a VM to DCB, Virtual machine will be selected from candidate set .
- A candidate set contain set of Vm's depending upon different cases.
- If CASE 1 is selected then candidate set include all Vm's and Vm with minimum completion time will be selected.
- If CASE 2 is selected then candidate set will contain only those Vm's which are currently available(Busy Virtual machine's are not considered) and Vm with minimum completion time will be selected.
- If CASE 3 is selected then candidate set will consist of Vm's which are already busy and we have to select the Vm with minimum load and check its completion time.

FLOW CHART (PROPOSED METHODOLOGY)



4. CLOUDSIM SIMULATOR

- CloudSim is a famous simulator for cloud parameters developed in the CLOUDS Laboratory, at the Computer Science and Software Engineering Department of the University of Melbourne.
- CloudSim is a simulation tool that allows cloud developers to test the performance of their provisioning policies in a repeatable and controllable environment, free of cost.
- It helps tune the bottlenecks before real-world deployment.

CloudSim is a toolkit used for modeling, experimentation and simulation. Cloud Analyst separates the simulation experimentation exercise from a programming exercise, as modeler should focus on the simulation complexities rather than to spare much time on the technicalities of programming. The Cloud

Analyst also facilitates the modeler to operate series of simulation experiments with slight changes in respective parameters in easy and quick way. The deployment of large scale applications is quite economical and easy by using clouds . The cloud also generates the new issues for developers. The various users access the internet applications around the world, and due to popularity of applications, it may vary along the world, so experience in the use of application can also vary. In order to analyze various load balancing policies, configuration of the various components of the cloud analyst tool need to be set.

5. RESULTS & DISCUSSIONS

Turn Around Time

- It is defined as total time taken between the submission of a process for execution time and complete output to the user.
- Turn Around Time (RT) = FT – AT

where,

FT = finish time of execution

AT = arrival time of process

Waiting Time

- It is defined as amount of time a process has been waiting in a ready queue..
- Waiting Time = ST - AT

where,

ST = Start time

AT = Arrival Time

Processing Cost

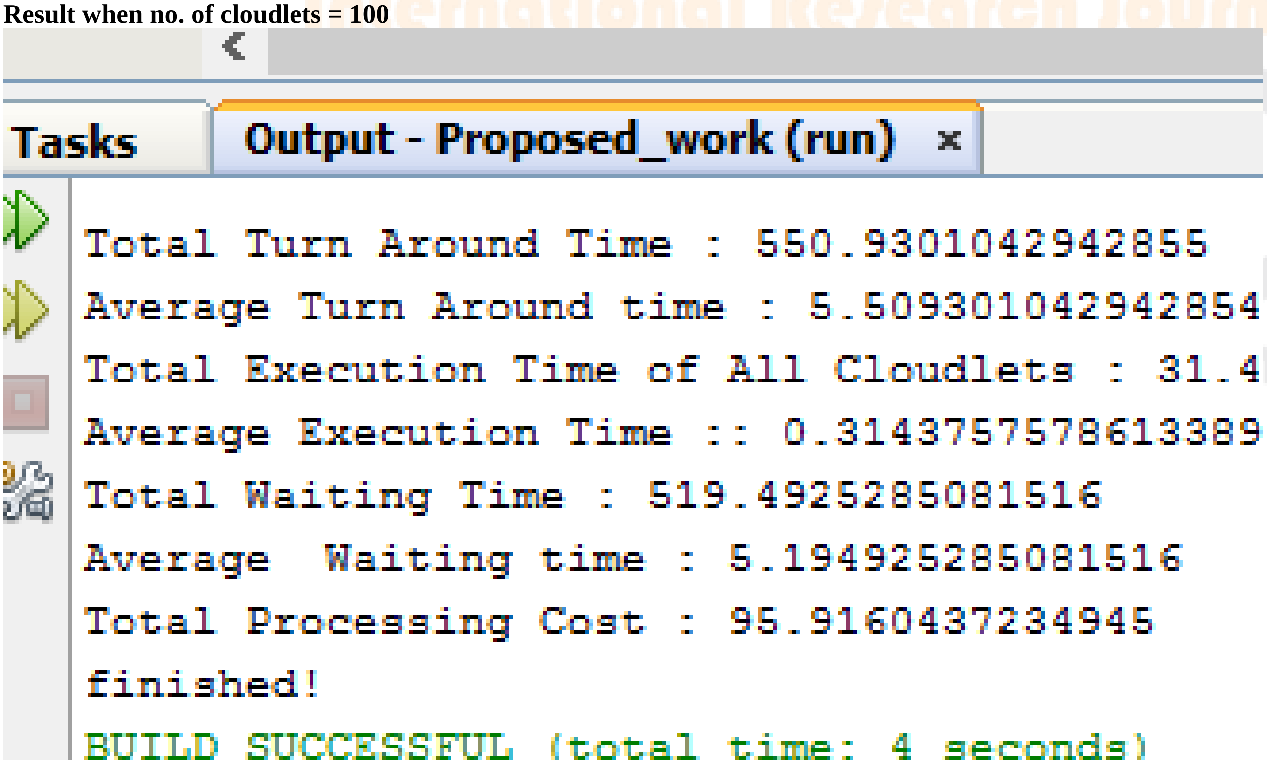
- It is obtained by addition of cost per storage, cost per memory.
- Processing cost = processing time * total cost;

Where, Processing time is the time it takes to complete a prescribed procedure;

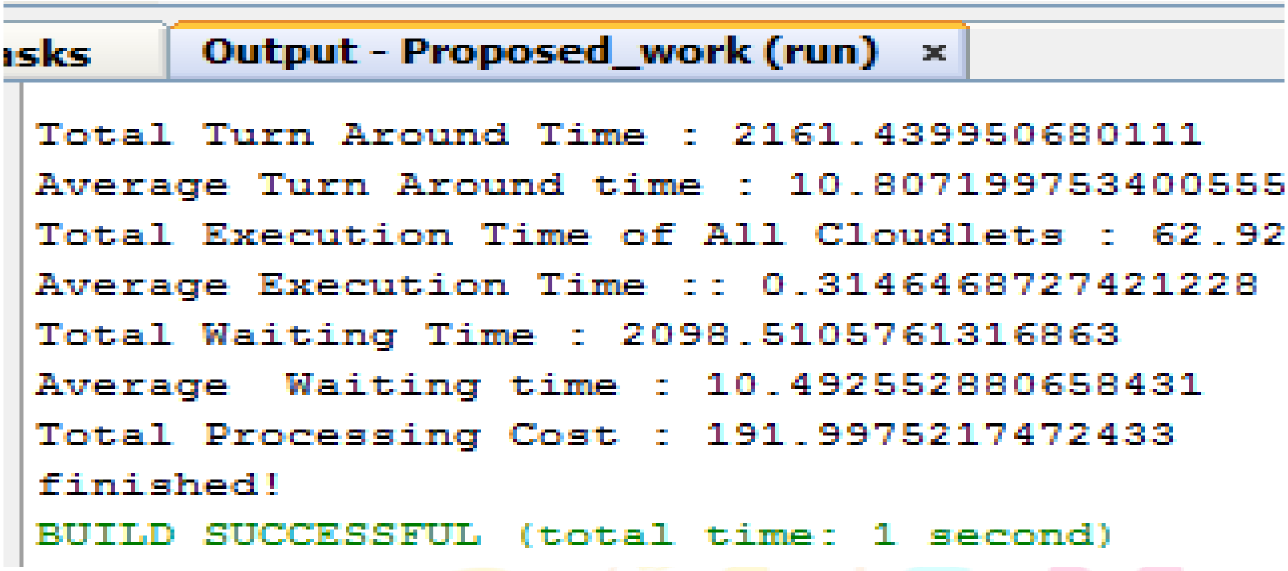
Total cost = cost Per Mem + costPerStorage + costPerBw;

Following are some results based on Our proposed Agorithm:

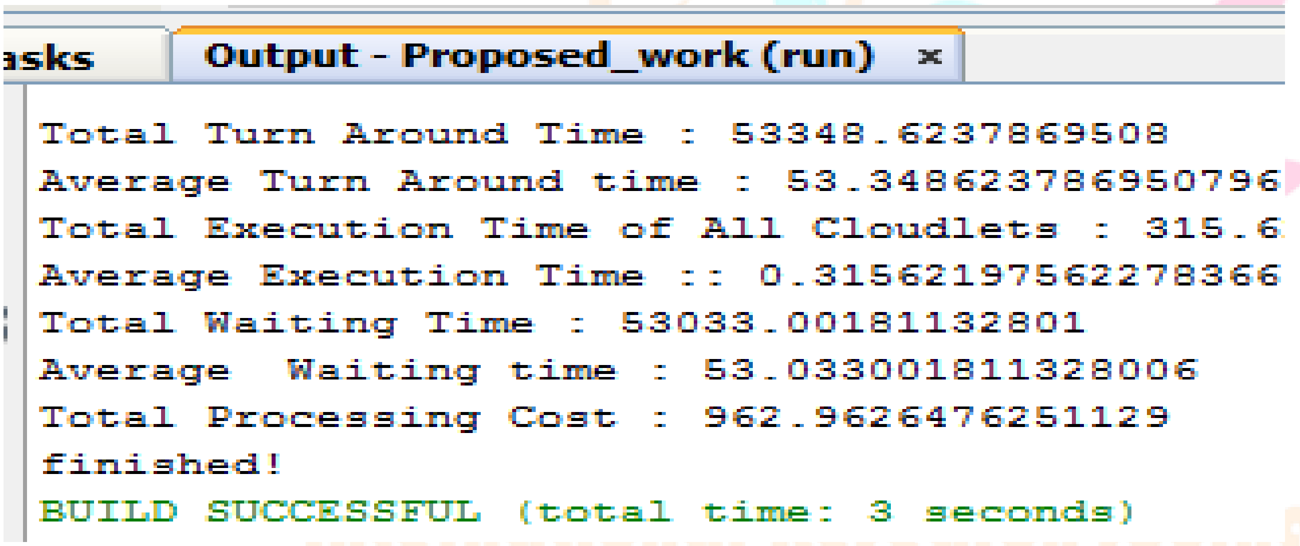
Result when no. of cloudlets = 100



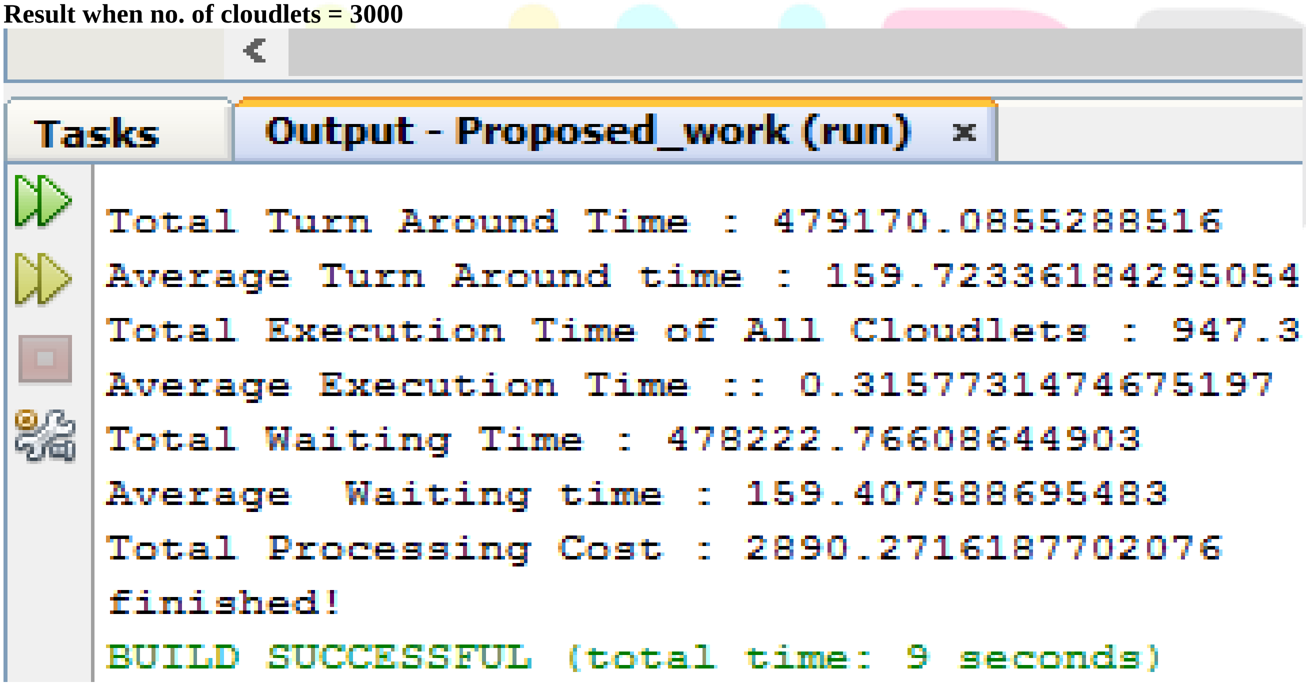
Result when no. of cloudlets = 200



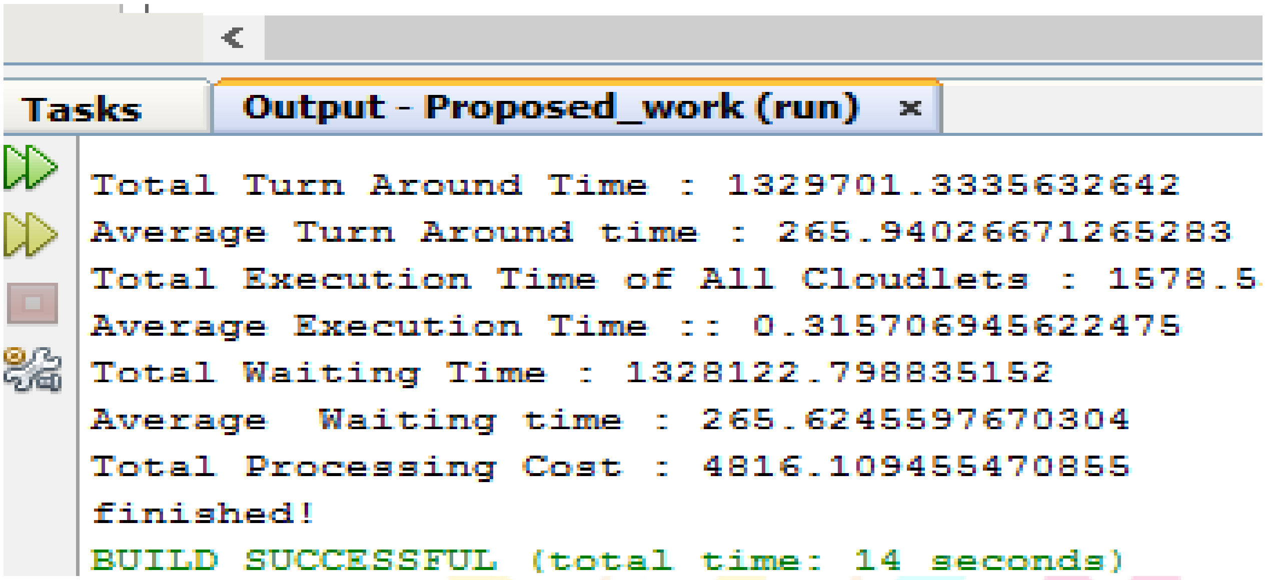
Result when no. of cloudlets = 1000



Result when no. of cloudlets = 3000



Result when no. of cloudlets = 5000



6. CONCLUSION

From the results and bar charts, it is clear that the proposed algorithm is performing better than the existing algorithm. It shows a decrease in average response time (ART) and total processing cost by taking into consideration the MIPS(millions of instructions per seconds) of virtual machine.

Finally, proposed algorithm presents better results with increase in number of cloudlets.

Future Scope

Consider a situation in which a user is allocated a virtual machine and it suddenly fails. There must be a requirement to handle such a situation.

The proposed algorithm can be made more robust by setting a facility of automatic fault tolerance.

This would automatically shift the workload on some suitable machine and maintains the performance of the system.

7. REFERENCES

[1] http://research.ijcaonline.org/ncetct/number1/NCETCT4017.pdfthe-president's-budget-making-cloud-computing-a-priority-for-thefuture as on Sep. 2012.

[2] Rajwinder Kaur, Pawan Luthra " Load Balancing in Cloud System using Max Min and Min Min Algorithm", National Conference on Emerging Trends in Computer Technology (NCETCT-2014)

[3] Eddy Caron , Luis Rodero-Merino "Auto-Scaling , Load Balancing And Monitoring In Commercial And Open-Source Clouds " Research Report, January2012

[4] Bhavisha Kanani, Bhumi Maniyar," Review on Max-Min Task scheduling Algorithm for Cloud Computing JETIR March 2015, Volume 2, Issue 3.

[5] Anthony T. Velte ,Toby J. Velte, Robert Elsenpeter, " Cloud Computing: A Practical Approach ", The McGraw-Hill Companies(2010), [Book]

[6] Saroj Hiranwal , Dr. K.C. Roy, "Adaptive Round Robin Scheduling Using Shortest Burst Approach Based On Smart Time Slice" International Journal Of Computer Science And Communication July-December 2011 ,Vol. 2, No. 2 , Pp. 319-323

[7] Jaspreet Kaur "Comparison of load balancing algorithm in cloud", june International Journal of Engineering Research and Applications 2012.

[8] Bhathiya Wickremasinghe ,Roderigo N. Calherios "Cloud Analyst: A Cloud-Sim-Based Visual Modeler For Analyzing Cloud Computing Environments And Applications". Proc Of IEEE International Conference On Advance Information Networking And Applications ,2010.

[9] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling And Simulation Of Scalable Cloud Computing Environments And The Cloudsim Toolkit: Challenges And Opportunities," Proc. Of The 7th High Performance Computing And Simulation Conference (HPCS 09), IEEE Computer Society, June 2009.

[10] www.cloudbus.org/cloudsim.