Improved Task Scheduling Algorithm in Cloud Environment

Sumit Arora
M.Tech Student
Lovely Professional University
Phagwara, India

Sami Anand
Assistant Professor
Lovely Professional University
Phagwara, India

ABSTRACT
Cloud computing is one of the hottest word in IT world and it having huge demand in these days. Some big IT companies like Google, IBM, Microsoft, Amazon, Yahoo and others develop cloud computing systems and products related to it for customers. But still customers are having difficulties for adopting the cloud computing, that is only because of the security issues exist in it. Cloud Computing is collection of large number of resources like hardware and software that are provided by the cloud providers to the consumers as a service over the internet. In cloud computing every task requires to be executed by available resource to achieve minimum waiting time, reduce makespan, best performance and maximum utilization of resources. To achieve these requirements we proposed an efficient scheduling algorithm which will work effectively to provide better result as compared with the traditional scheduling approaches. For this CloudSim framework is used to simulate the proposed algorithm under various conditions and presented the better results with reduced the waiting time and processing time with optimum resource utilization and minimum overhead for the same.

Keywords
Cloud Computing, Task Scheduling, Task Grouping, SA Algorithm

1. INTRODUCTION
In the modern era of IT if we talk about the "Cloud Computing" it is one of the hottest topics of the industry. A lot can be done on this topic. If we talk about the term "cloud" appears to have its origins in network diagrams that represented the internet, or various parts of it, as schematic clouds. Cloud computing can be easily defined, there are many definitions, which share the same common denominator: the Internet. Cloud computing is a way to use the Internet in the daily life from your PC and Laptop [1].

1.1 Scheduling

In the simple language if we define scheduling it is assigning number of resources to jobs or tasks to be executed. Now if we talk about the Cloud Computing field Job Scheduling and Load Balancing is one of the main security issues in the today's world.

For this there are number of algorithms are exist in the cloud environment. The main benefit of job scheduling is to achieve a high performance computing and the best system throughput and reduced waiting time. So it is difficult to find an algorithm which effectively utilizes the resources.

1.2 Scheduling Process

Scheduling process basically contains three steps that are the following:-

- Resource discovering and filtering: - In first step Datacentre broker looks for the available resources in the network system and collects the status related to them.
- Resource Selection: - In the second step Target resource is selected based upon some parameters of the task and resource.
- Task Submission: - In the last step task is submitted to the resource selected. [2]

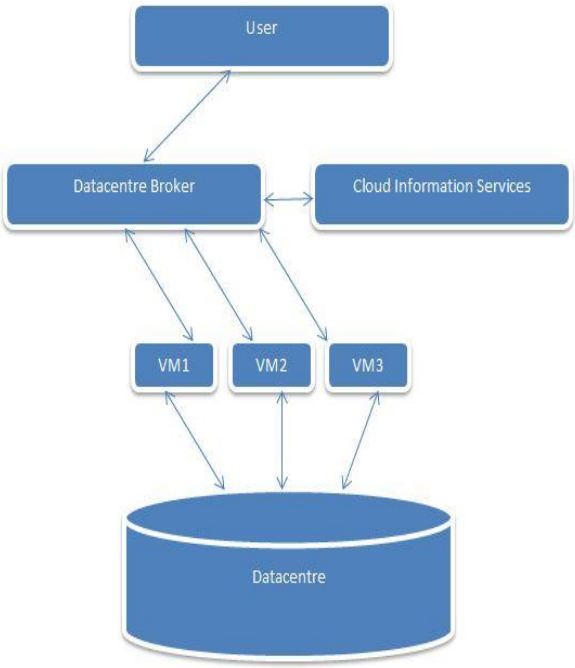

Fig 1: Scheduling Process

This paper mainly focuses on evaluating and improving the deployment policy to get the better resource utilization. CloudSim simulator is used in the experimentation that provides the parameters that helps us to work in it.

The rest of this paper is organized as follows: Section 2 discusses background and related work. Section 3 presents the proposed algorithm and its strategy. Section 4 provides simulations and experiments on the proposed scheduling algorithm using CloudSim and provides results and discussion on the experiments in comparison with existing algorithms. Section 5 concludes the paper and proposes future research directions. Section 6 describes the References.

2. BACKGROUND AND RELATED WORK

Cloud computing consists of a cluster of computing resources that are delivered over a network, which is accomplished by utilizing virtualization technologies to consolidate and allocate resources suitable for various different software applications. It provides a platform for solutions requiring different configurations, emulating physical hardware combinations in a virtualized cloud environment managed by cloud platform software to deliver enhanced services. The strategies used in the cloud platform software become important, which directly influence the runtime performance of software applications running on its platform. Therefore the effective scheduling policies to maximize the utilization of the virtualized resources are the primary focus of this study. This section provides a summary of related scheduling approaches applied in cloud computing. [3]

2.1 Cloud Computing Scheduling Model

In cloud computing scheduling model is mainly build by Client, Broker, Resources, Resources Supporter and Information Service In the Figure2 below given scheduling model **Broker** is the middle interface between the client and resource provider. It is the main scheduler which provides the scheduling to the jobs and resources. Firstly client submits the task to the broker, then broker searches for the resources in information service and then deploys the tasks to the appropriate resources according to the algorithm provided to the broker. Broker contains Job Control Agent, Schedule Advisor, Explorer, Trade Manager and Deployment Agent.

- Job Control Agent: - It is responsible for monitoring jobs in the software system, such as schedule generation, jobs creation, status of jobs and communicating with clients and schedule advisor.

- Schedule Advisor: - It is used to determine resources, allocate available resources which satisfy the demands of clients such as deadline and cost, as well as to allocate jobs.

- Cloud Explorer: - It is a tool that communicates with cloud information service to find resources and identifies the list of authorized machines as well as records resources status information.

- Trade Manager: - It determines resources access cost and tries to communicate with resources at a low cost under the guidance of schedule advisor.

- Deployment Agent: - It uses scheduler instruction to activate the execution of tasks as well as to update the status of execution sending back to Job Control Agent in regular intervals.[3]

As I early mentioned there are so many scheduling algorithms let us see some of the them that what authors have researched about the scheduling of tasks in cloud computing.

Pinal Sanlot gave the basic introduction about the scheduling and compared various existing scheduling algorithms [2]. He categorized the job scheduling algorithms in two type's i.e. Batch Mode heuristic scheduling algorithms (BMHA) and Online Mode heuristic algorithms. In BMHA jobs are queued and collected into a set when they arrive in the system. The scheduling algorithm will start after a fixed period of time for example First Come First Serve (FCFS), Round Robin Scheduling (RR), Min-Min and Max-Min Algorithm. In Online mode heuristic scheduling algorithm are more not efficient in cloud environment as it schedules the jobs when they arrive in the system example of this is Most Fit Task Scheduling algorithm. MFTSA in this algorithm task which fit the best in the queue are executed first. This algorithm has high failure ratio. He also explained the Resource Aware Scheduling Algorithm, a Priority based Job Scheduling Algorithm in Cloud Computing and many more then he compared all these algorithms with some fixed parameter.

Saeed Parsa et al. proposed a new algorithm i.e. RASA [4]. In this paper they took the two basic algorithms i.e. Max-Min and Min-Min. These two algorithms work on the estimated time for execution and completion of the task. In these execution time of each task is calculated on the resources. In MIN-MIN task with the minimum completion time is selected first and assigned to the resource with the minimum execution time. This procedure is followed with all the tasks. Main disadvantages of this algorithm are that in this longer tasks have to wait for the long time. But MAX-MIN algorithm selects the larger tasks first and then it selects the smaller tasks. Disadvantage of this algorithm is that in this smaller tasks have to wait for the long time.

In the proposed algorithm i.e. Resource Aware Scheduling Algorithm they are relying on the number of resources available for the procession of the tasks. According to their algorithm if the numbers of available resources available are ODD then Min-Min strategy will be applied and if the numbers of available resources are EVEN then the Max-Min strategy will be applied. If the first task is assigned with the Min-Min strategy then the next task will be assigned with the Max-Min strategy.

O.M. Elzeki et al. proposed the new algorithm with the modification of the Max-Min algorithm [5]. This algorithm based upon the expected execution time instead of expected completion time. In this algorithm expected completion time for all the submitted tasks are calculated first then the task with the maximum completion time is selected and then submitted to the resource with the minimum execution time.

Upendra Bhoi et al. proposed a new algorithm with the modification in the Max-Min algorithm. This algorithm selects the task with average execution time [6]. In Enhanced Max-min Task Scheduling Algorithm in Cloud Computing the task with the average or nearest greater than average task is selected and then assigned to the resource with the minimum completion time.

Jia Ru et al. proposed a new scheduling algorithm in which they integrated task grouping, priority aware and Shortest Job First algorithm. In this algorithm they used Gaussian distribution to generate the tasks [3].

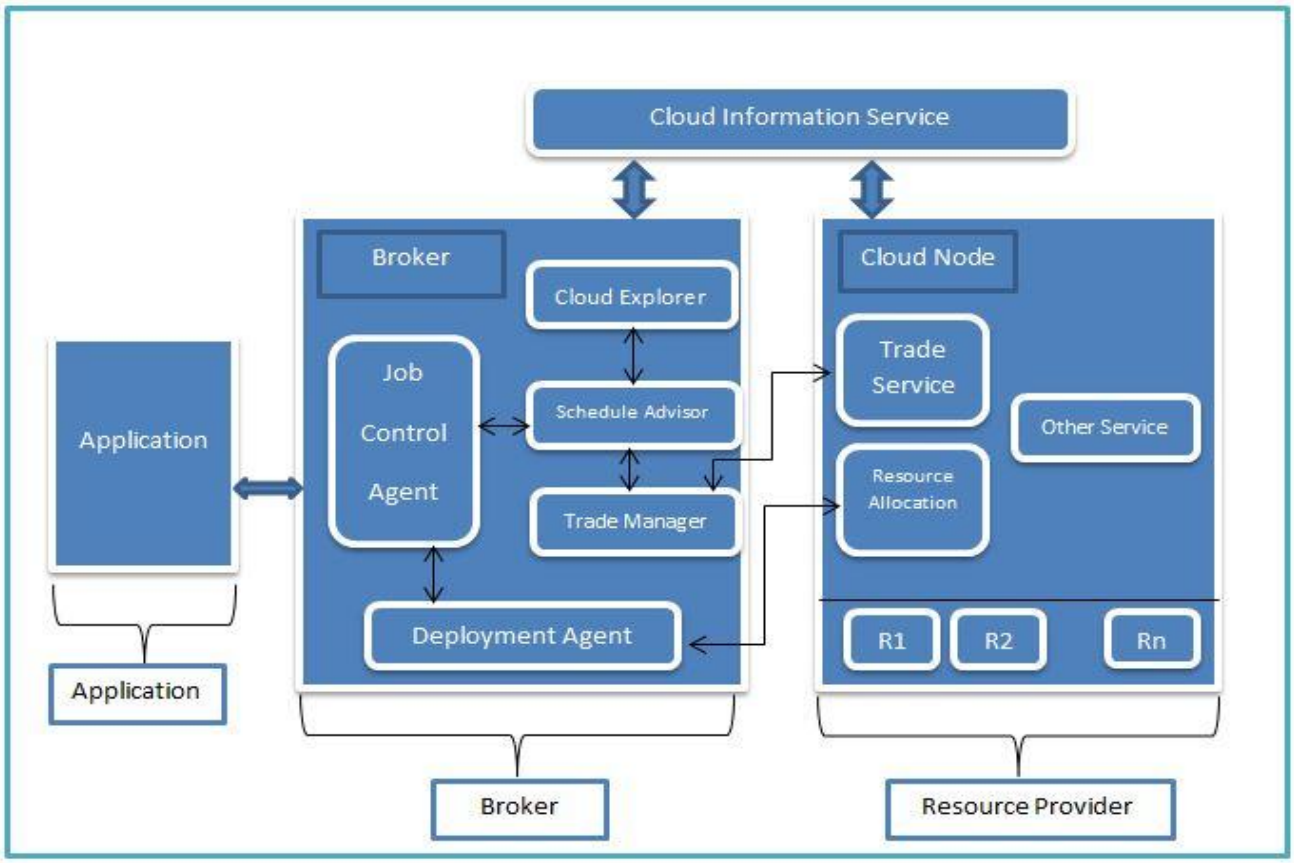

Fig 2: Cloud computing scheduling model

Raksha Sharma et al. discussed on various types of Job and Resource scheduling algorithms in Grid Environment. They motivated the researchers to further continue their research work in the grid computing [7]. They explained the basic elements of the Grid model i.e. User, Resource Broker, Grid Information service and resources.

They explained the various Job Scheduling algorithms like Efficient Utilization of Computing Resources Using Highest Response Next Scheduling in Grid, Node Allocation in Grid Computing Using Optimal Resource Constraint Scheduling, and Hierarchical Job Scheduling for Clusters of Workstations, Resource CoAllocation for Scheduling Tasks with Dependencies in grid, Scheduling Framework for Bandwidth-Aware Job, Grouping-Based scheduling in grid computing, Grouping-based Fine-grained Job Scheduling in grid computing, A Job Schedule Model Based on grid environment. They also explained the various Resource Scheduling algorithms like Research on Novel Dynamic Resource Management and job scheduling in grid computing, Agent Based Resource Management with Alternate Solution, New Resource Mechanism with Negotiate Solution based on agent in grid environments, Improved Resource discovery approach using P2P model for condor Virtual Computing Grid using Resource Pooling.

Haozheng Ren et al. proposed a new scheduling algorithm which uses the trigger strategy based on the fractal methods. It determines the timing of the virtual machine migration which can avoid the problem of peak instantaneous load trigger [8]. They effectively mad e a better algorithm which provides better load balancing; they presented the dynamic migration of the virtual machines which effectively contains the modules like load balancing, overload detection and the load scheduling.

Vishnu Kant Soni et al. proposed a new scheduling algorithm in the grid environment which have reduced the processing time, processing cost and enhance the resource utilization when compared with the other algorithms [9]. They proposed a new scheduling model where resources are arranged in hierarchical based where Heap Sort Tree (HST) is used to obtain the highest computational power resource or root node, so as to make balanced and effective job scheduling.

Tarun Kumar Ghosh et al. proposed a new scheduling algorithm in which they enhanced the traditional algorithm Max-Min Scheduling algorithm [10]. They have divided the work actually in two phases. In the first phase they are performing the Max-Min after getting result from this they are again re-scheduling tasks to get better performance.

3. PROPSED ALGORITHM

3.1 Proposed SA Scheduling Algorithm

1. All the tasks are submitted in the cloud.

2. After the tasks are submitted they are sorted in ascending order of the length.

3. After sorting is done groups (let's say combining 3 jobs) are created of the tasks according to the available resources.

4. Calculation of Standard deviation for all the tasks is done according to their respective lengths.

5. Calculation of Mid value for total number of available lengths is done.

6. We can perform allocation of resource to the task in the following manner

If SD is greater than the value at mid of the total numbers of tasks then tasks in groups with the longer length will be allocated first to the resources in descending MIPS.

Else the tasks in groups with the minimum length will be allocated first to the resources with the resource in ascending MIPS.

7. All the resources will be deleted at end.

3.2 Description of Algorithm

- In the proposed scheduling algorithm we are dealing with the Shortest Job First algorithm we are just combining two algorithms by which tasks will be executed according to their length. First of all sort all the tasks according to their length.
- Then make their groups according to the requirement. Making group means we are combining the number of cloudlets and sending it to the single resource for the execution.
- Now calculate the Standard deviation of the cloudlets by using the formula in the Equation 1.

$$S=\sqrt{\Sigma\,(X-\bar{X})^2}\,/N-1$$

Equation 1: Standard Deviation

Where S is Standard Deviation

Is the "sum of"

X is the each value

$\bar{X}$ is the mean of all values

N is the number of values

- Now if SD is greater than the value at mid of the total numbers of tasks, then the tasks in groups with the longer length is allocated to the resources for execution.

Table 1: Maximum Case

Index(s)	1	2	3	4	5	6
Length	10	20	50	60	500	1000

Now in this case we have 6 numbers of tasks and Standard deviation is 402.0779 in this case we have SD>s/2 i.e. 402.0779 > 50 so we will Run the maximum case i.e. the tasks in groups with the longer length is allocated to the resources for execution.

- If SD is less than the value at mid of the total number of tasks, then the tasks in groups with the shorter length is allocated to the resources for execution.

Table 2: Minimum Case

Index(s)	1	2	3	4	5	6
Length	10	30	70	80	100	120

Now in this case we have 6 numbers of tasks and Standard Deviation is 41.6733 so here we have SD<s/2 i.e. 41.6733<70 so we will run the minimum case i.e. the tasks in groups with the shorter length is send/allocated to the resources for execution.

- After completing the execution destroy the resources.

4. EXPERIMENTAL SETUP AND RESULTS

4.1 Illustrative Example

For evaluation purpose they have used the CloudSim Framework for the practically implement the proposed algorithm. In our example we have taken ten tasks .i.e. T1, T2, T3….T10 and they have five resources for the same i.e. R1, R2….R5. They have taken two different-different scenarios for the same number of tasks with different lengths but same resources.

These are the five resources they are having for the tasks. They have evaluated the results for these scenarios on existing algorithm and our proposed algorithm.

Table 3: Resources for Execution

Resources	MIPS
R1	250
R2	50
R3	20
R4	10
R5	30

They have the following problem set in which two different problem set are mentioned with different-2 task lengths.

Table 4: Problem set with Different Length of Tasks

Problem Set	Number of Tasks	Length of Tasks
P1	T1	3
	T2	80
	T3	90
	T4	4
	T5	50
	T6	343
	T7	300
	T8	9930
	T9	3000
	T10	10
P2	T1	4500
	T2	3000
	T3	7500
	T4	12000
	T5	10000
	T6	14000
	T7	30000
	T8	18000
	T9	20000
	T10	2000

These are the two problem sets with different-2 task lengths. In both the problem sets we have done the grouping of two-two tasks i.e. each resource will process only two tasks. In the P1 i.e. in the first problem set our proposed algorithm will run the Maximum case i.e. tasks with the maximum length will be getting executed first. In the P2 our proposed algorithm will run the Minimum case i.e. tasks with the minimum length will be getting executed first. When evaluated on the CloudSim framework we get the following output when compared with the existing SJF algorithm.

In the below Figure 3 we can see that our proposed algorithm is taking less processing time when compared with the existing algorithm.

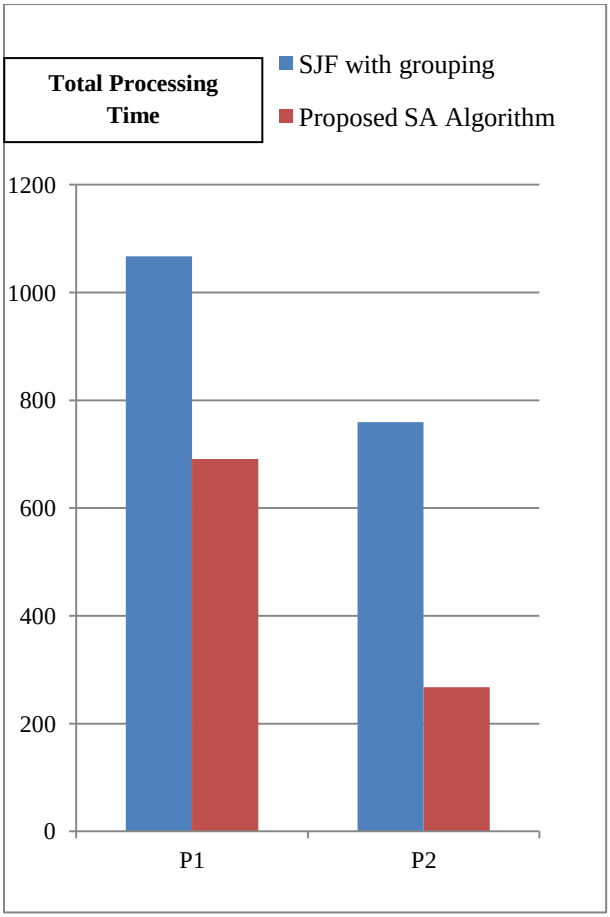

Fig 3: Existing Vs. Proposed Algorithm

4.2 Example with Randomly Generated Values

We have also evaluated the results with the randomly generated values of the length of the tasks. We have made the scenarios of various randomly generated values like i.e. of 10,20,30,40, 50, 60,70,80,90 and 100 tasks and compared it with the existing algorithm. In all the cases we got better results in our proposed algorithm. Here I am just putting results of three cases with i.e. 10, 50 and 100 tasks.

In the first problem set i.e. in P1 with ten tasks, we have made the groups of two-two tasks.

We have compared the existing algorithm with the proposed algorithm and got the following results.

We can see in the Figure 4 that our proposed algorithm in the above result is also performing efficiently.

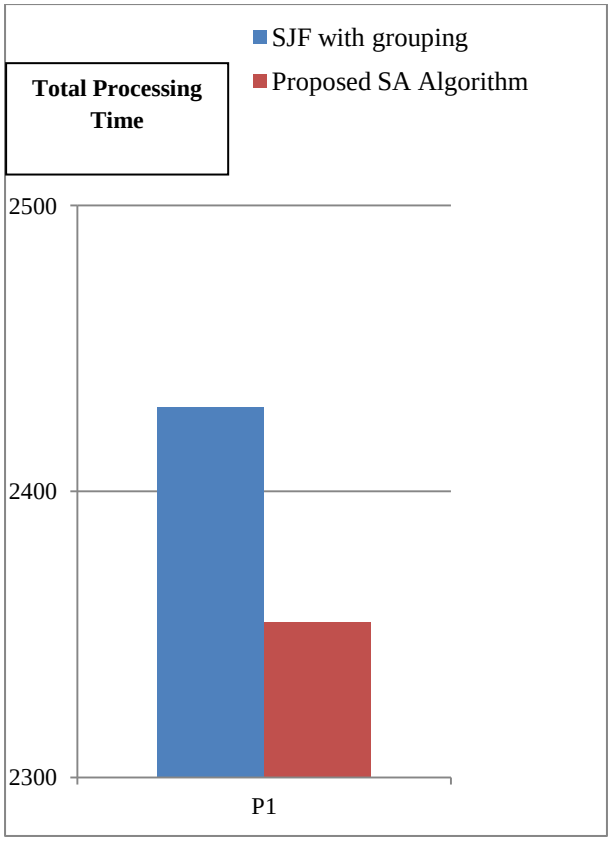

Fig 4: Existing Vs. Proposed with Tasks 10

Now we are taking the P2 i.e. where we have taken the 50 tasks and made the groups of the ten tasks i.e. ten tasks will be executed by the one resource.

We have compared the existing algorithm with the proposed algorithm and got the following results.

We can see in the Figure 5 that our proposed algorithm in the above result is also performing efficiently.

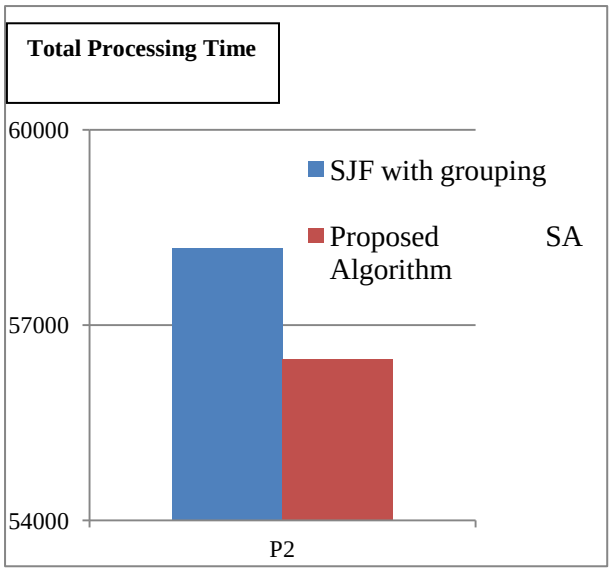

Fig 5: Existing Vs. Proposed with Tasks 50

Now we are taking the P3 i.e. where we have taken the 100 tasks and made the groups of the twenty tasks i.e. twenty tasks will be executed by the one resource.

We have compared the existing algorithm with the proposed algorithm and got the following results.

We can see in the Figure 6 that our proposed algorithm in the above result is also performing efficiently.

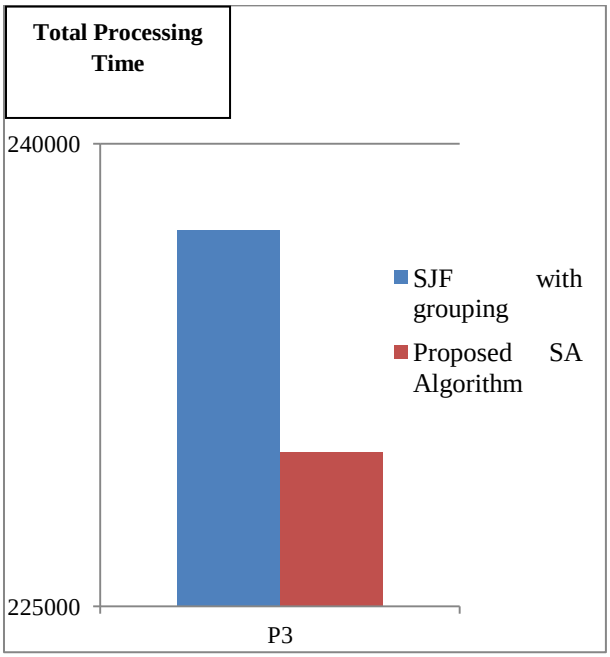

Fig 6: Existing Vs. Proposed with Tasks 100

5. CONCLUSION AND FUTURE WORK

In this paper we have proposed an improved scheduling algorithm which is working efficiently and when evaluated using the CloudSim framework it is performing best as compared with the existing shortest job first algorithm with task grouping. Our proposed algorithm is continuously taking less processing time i.e. time taken by the resources to complete the execution when compared with the existing algorithm.

In future work can be done on this to reduce the average waiting time and processing time of the execution.

6. REFRENCES

[1] Arora Sumit "A Review on Cloud Security Issues and Solutions" International Journal for Advance Research in Engineering and Technology, Vol. 2, Issue III, March 2014, pp.15-18.

[2] Salot Pinal "A Survey of Various Scheduling Algorithm in Cloud Computing Environment" 2013 IJRET Volume: 2 Issue: 2, pp.131-135.

[3] Ru Jia and Keung Jacky "An Empirical Investigation on the Simulation of Priority and Shortest-Job-First Scheduling for Cloud-based Software Systems" 2013 22nd Australian Conference on Software Engineering, pp.78-87.

[4] Parsa Saeed and Maleki Reza Entezari "RASA: A New Task Scheduling Algorithm in Grid Environment" 2009

World Applied Sciences Journal 7 (Special Issue of Computer & IT), pp. 152-160.

[5] Elzeki O. M., Reshad M. Z., Elsoud M.A "Improved Max-Min Algorithm in Cloud Computing" International Journal of Computer Applications (0975 – 8887) Volume 50 – No.12, July 2012, pp.22-27.

[6] Bhoi Upendra, Ramanuj Purvi N. "Enhanced Max-min Task Scheduling Algorithm in Cloud Computing" International Journal of Application or Innovation in Engineering & Management, Volume 2, Issue 4, April 2013, pp.259-264.

[7] Sharma Raksha, Soni Vishnu Kant, Mishra Manoj Kumar, Bhuyan Prachet "A Survey of Job Scheduling and Resource Management in Grid Computing" 2010 World Academy of Science, Engineering and Technology, pp.419-424.

[8] Ren Haozheng, Lan Yihua, Yin Chao "The Load Balancing Algorithm in Cloud Computing Environment" 2012 2nd International Conference on Computer Science and Network Technology, pp. 925-928.

[9] Soni Vishnu Kant, Sharma Raksha, Mishra Manoj Kumar, Das Sarita "Constraint-Based Job and Resource scheduling in Grid Computing" 2010 IEEE, pp.334-337.

[10] Ghosh Tarun Kumar, Goswami Rajmohan, Bera Sumit, Barman Subhabrata "Load Balanced Static Grid Scheduling Using Max-Min Heuristic" 2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing,pp.419-423.