International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET

Improved Scheduling Algorithm in Cloud Computing

Varinder Saggar[*1], Manoj Kumar Srivastava[2]
[*1] M. Tech (Scholar), CSE Depatment Desh Bhagat University, Mandi Gobindgarh, Punjab, India
[2]CSE Department, Desh Bhagat University, Mandi Gobindgarh, Punjab, India

ABSTRACT

Article Info
Volume 8, Issue 4
Page Number: 156-161

Publication Issue :
July-August-2021

Article History
Accepted : 10 July 2021
Published: 15 July2021

The current era of an emerging technology is cloud computing. It is internet based computing, works as pay-per-use model and process large data. The cloud Service provider goal is to manage resources efficiently, So, in cloud computing the mechanism of Scheduling has an important function. The revised scheduling technique is meant to improve the server performance and decrease the switching time to increase the use of resources. Different sorts of scheduling algorithms have been studied and analysed in this research to deliver efficient cloud services. The improved Scheduling algorithm prioritises the task, which improves computer performance and does my best possible efforts to limit the duration and duration of waiting. A CloudSim tool is used to simulate the suggested approach.

Keywords : Improved Scheduling algorithm, Cloud, Job Scheduling in Parallel, Batch Workloads, Makespan.

I. INTRODUCTION

The new technology is cloud. It has been recently revealed that academics want to use cloud for scientific activities, and also the huge companies are converting to cloud. In order to perform duties successfully, many sophisticated applications want to processes in parallel. The use of CPU resources has decreased as a result of communication and synchronization between the job processed in parallel. The use of nodes while preserving the response level simultaneous processes is important for a data centre.

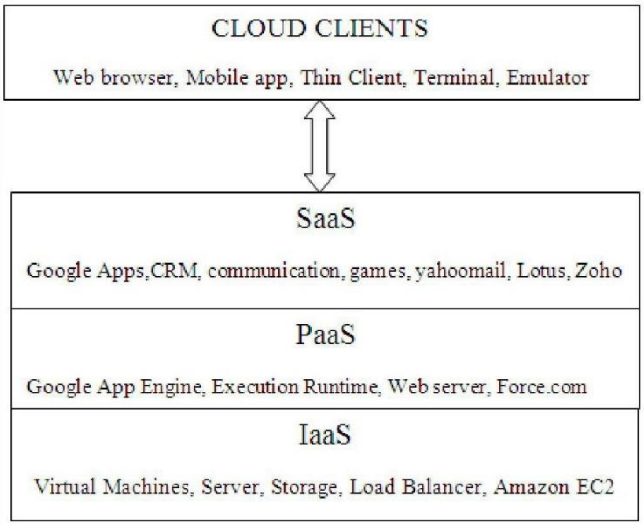


Figure 1. Overview of Cloud Computing

A growing number of apps are being enticed to run in distant data centers thanks to cloud computing. Many sophisticated applications need parallel processing. When there is an increase in parallelism, certain parallel programmes exhibit a drop in CPU resource use. If the tasks are not scheduled effectively, the computer performance suffers.

Regarding the cloud computing scheduling mechanism, several methods and protocols have been developed. However, only a few approaches for detecting the scheduling mechanism in cloud computing have been presented. Most writers include a frequent monitoring zone in their procedures that is not a realistic reality. Because the clouds are dispersed at random, the monitoring region is always irregular. As a result, we present a method for scheduling jobs in cloud computing.

For task processing, the majority of the writers use FCFS scheduling. It reduces resource use and server use in this situation. I thus intend to use Backfilling to minimise the scope of the project to enhance the use of the servers that are allocated to the work, to enhance the use of the resource using Backfilling and to distribute the shortest distance resources to a project to minimise latency. Few authors may perhaps not place a high value on the procedure. In FCFS scheduling, processors process jobs by assigning them the same priority. As a result, the computer's performance suffers. As a result, I arrange the task with priority in mind. Some authors do not take into account the time spent waiting. As a result, the job's time to completion lengthens. As a result, the computer's performance suffers. Some writers propose that the makespan be reduced by reducing the waiting time, however, the timing of the resources shift is ignored. I thus think there is a better way to minimise time and at the same time to lower the duration of the project. The major aim of the improved scheduling algorithm suggested is:

- Improve the correct use of assigned servers .
- To carry out the top priority activity.
- Proper management of the resource.
- Reduces the time to complete (makespan)
- Reduces period of expectation
- Reduce the time of switching

The following is how rest of the work is organized: The relevant work in this topic is addressed in the literature review. The algorithm and proposed model are presented in the next section. Finally, with the conclusion and future scope of the work in mind, we concentrated on outcome analyses.

II. LITERATURE REVIEW

Ke Liu [1] It created a one-of-a-kind compromised-time-cost scheduling technique that takes into account cloud computing characteristics in order to handle instance-intensive cost-constrained processes by sacrificing execution time and cost with on-the-fly user input. The simulation showed that the CTC approach (compromised time cost) may produce reduced costs while still satisfying user needs.

Swin De WC is the tool for simulation (Swinburne Decentralized Workflow for Cloud).

Linlin Wu [2] Introduced a PSO technique for cloud planning of applications that includes both calculation and data transfer expenses It is used by changing the cost of communication and computation in workflow applications. The cost reduction obtained by using PSO is compared with the traditional approach of 'Best Resource Selection.' According to the findings, PSO save three time as much money then BRS while also providing superior work allocation.

Cui Lin, Shiyong Lu [3] It suggested a SHEFT workflow scheduling technique for elastically scheduling a process on a Cloud computing environment. SHEFT not only outperforms numerous sample workflow scheduling algorithms in terms of reducing workflow execution time, but it also allows resources to scale elastically during runtime, according to the results.

Salim Bitam [4] used to optimise the distribution of computing jobs among processing data in the cloud datacenters. This is an NP-Complete issue that aims to distribute workloads among processing resources as efficiently as feasible in order to reduce total work load and hence improve the overall efficacy of cloud services. Work Scheduling seeks to distribute jobs to cloud datacenters in order to reduce the execution time (makes pan) of total job tasks.

Abirami S.P and Shalini Ramanathan [5] Focus on allocating resources among requestors in such a way that the given QoS criteria are maximized. As the cost function, the QoS parameter was chosen. The scheduling concept encompasses both the jobs and the virtual machines that are available. The scheduling strategy is predicated on the notion that the initial answer to a request is provided only after collecting the resource for a specified period of time but not allocating the resource as it occurs. In response to a statement for additional resources, the scheduler may execute dynamically. This is performed by re-evaluating the threshold value on a regular basis. This scheduling mechanism, as well as the dynamic threshold value computation in the scheduler, takes both the job and the resource into account. Regardless of hunger or deadlock situations, this enhances system throughput and resource usage..

WeiWang [6] The Trust Model is based on the Bayesian trust evaluation model in trustworthy scheduler, and the Schedule Advisor is based on Cloud-DLS. Sending queries to cloud nodes determines the trust value. Every node maintains two trust tables, one for implied trust and the other for direct trust. The reliable values are generated from the queries sent between the nodes. The trust of a node represents its disposition/suitability to engage in cloud peer-to-peer activities. Then, nodes with greater trust levels are chosen for specific scheduling. Its mission is to increase, security, dependability and efficiency.

R. Santhosh [7] proposes scheduling technique which is novel that focuses on giving a solution for the online problem of scheduling real-time jobs utilising the cloud computing "Infrastructure as a Service" concept. The real-time tasks are planned ahead of time in order to maximise overall utility and efficiency. The goal is to reduce response time and increase work efficiency. When a job misses its deadline, it is transferred to another virtual computer. This boosts overall system performance while also increasing total utility. The suggested approach outperforms the EDF and Non Preemptive scheduling algorithms by a large margin.

El-Sayed T. El-kenawy[8] The effects of the RASA algorithm is used to develop a new algorithm in this study. The improved Max-min algorithm selects based on the predicted execution time rather than the total duration. Petri nets are used to represent distributed systems' concurrent behaviour. Instead of RASA and original Max-Min, Max-min displays attaining schedules with comparable reduced makespan.

Xiaomin Zhua [9] The RTC and the AVC work together to determine whether or not a coming job in the global queue can be allowed. The scheduler will assign a voltage level to the job after it has been approved. Every cluster has a queue of accepted jobs waiting for the node. The local control system in every node attempts to save energy consumption by lowering voltage levels for permitted tasks..

III. PROPOSED ALGORITHM

The algorithm's pseudo code is as follows:

Step 1: Consider switching Time Else if there is no dependence between the jobs and resources.

Sort and check for dependencies.

Step 2: Return Independent work is assigned to propagation to optimize resource usage and higher priority resources.

Step 3: Assign jobs with longer completion times to higher processors, and so forth.
Step 4: To process the job in parallel, a multilevel queue is built.

IV. METHODOLOGY

In cloud computing, the efficiency of the scheduling mechanism is determined by how efficiently it manages processes and improves server and resource performance. As we previously noted, the old scheduling technique has a number of flaws that must be addressed in any way feasible in order to improve efficiency.
In this part, we present a work scheduling technique that optimises resource use by scheduling jobs efficiently. There are four steps in the complete process or technique.
Step-1
If there is no interdependence between occupations and resources, we consider switching time since it is more flexible and dependable. As a result, the jobs might be handled in this way.
If not, evaluate the dependencies and put them in a queue before moving on step two.
As a result, we take changeover time into account.
So, in order to schedule the jobs, we store them in the following manner.
J2 ⟶ J1⟶J3 // here J is Job

If there are any interdependencies between the jobs or resources, go straight on step 2. There is reliance among the resources and employment, for example, in the diagram below. As a result, there is a risk of stalemate and critical portions. To circumvent this, we'll go for step 2.

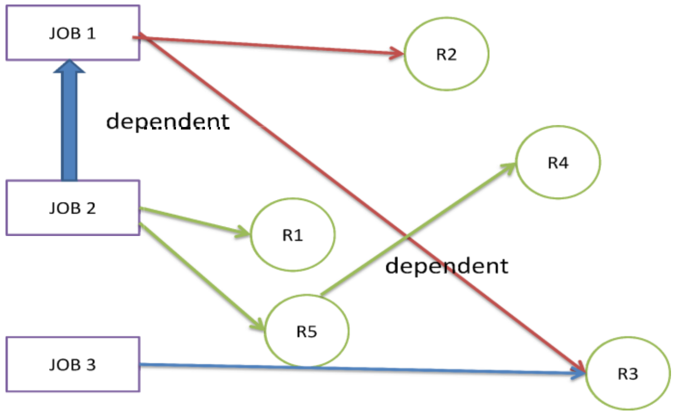


Figure 2. Dependencies in the jobs and resources

Step-2

The server does not assign any priority to the tasks, as we can see in the previous situation. To circumvent this difficulty, we use priority as an additional option to determine which work should be executed first. Let's say we have a maximum priority of 1 and a minimum priority of 5, and we want to allocate jobs in a priority order to enhance server performance and resource usage. To increase resource consumption, we use the backfilling approach.

J i (j,k,l) //where J is job.

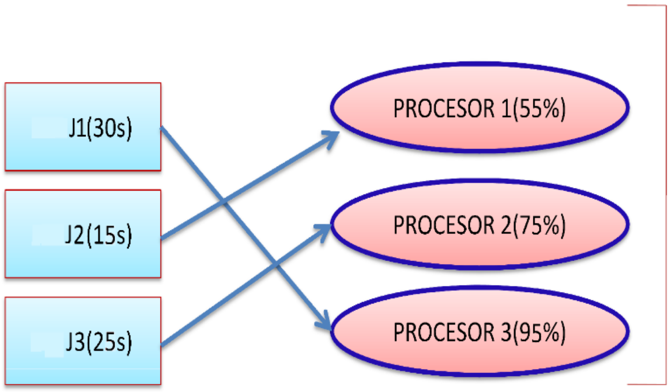
What is the location of

i= Job number j = necessary resource k = time necessary to finish the job l = process priority

If there is reliance between resources or jobs, the independent task should always take precedence over the dependent task. As a result, the delay will be reduced..

Step-3
After the jobs have been prioritised, the question of which job should be assigned to which processor arises. So, to figure out which job should be assigned to which processor, I average the execution times of all the jobs. Then look at the processor's processing

speed. The jobs are then assigned to the server or processor . Let me give you an example.:



Parallel Processing

Figure 3. The job allocation to the processor

There are three occupations depicted in the diagram above: J1, J2, and J3. As a result, the average time interval is 23.33 seconds. As a result, all of the tasks must be completed. Following the prioritization of jobs, the question of which tasks should be given to which processor emerges. As a result, I average the execution times of all the jobs to determine which work should be given to which processor. Then have a peek at the processor's speed. The processor, CPU, or server is subsequently allocated the tasks. Consider the following scenario: k in 25 seconds. As a result, jobs with longer completion times are assigned to higher processors. The next work with a shorter completion time will be assigned to the processor after that. All of the positions are assigned in this manner. All workloads are assigned to the processor in this manner. As a result, all of the processors begin processing in parallel.

Step-4

Then a multi-level queue is built for each processor to store and process the parallel jobs.

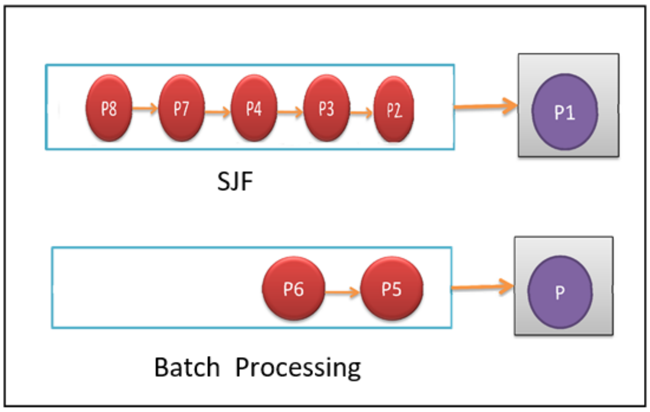


Figure 4. The parallel Processing

In the diagram above, one processor produces three queues, each with its own scheduling method for processing workloads. The processor then places the jobs in the proper queue and processes them in parallel when they come to the processor.

Comparison Between Scheduling Algorithms

Scheduling Algorithm	**Scheduling Type**	**Scheduling Mode**	**Scheduling Parameters**	**Scheduling Factors**	**Findings**	**Tools**
A compromised -Time-Cost Scheduling Algorithm	Dynamic	Batch mode	Cost and Time	An array of workflow instances	1. It's utilised to save money and time..	SwinDeW-C

A Particle Swarm Optimization based Heuristic for Scheduling	Dynamic	Mode of Dependency	Resource utilization, time	Group of tasks	1.It can be utilised three times as much as BRS for cost savings 2.It's used to ensure hat workload is listributed evenly mong resources.	Amazon EC2
Bees life algorithm for scheduling in cloud computing	Dynamic	Batch Mode	Cost, time	Set of tasks	1. It seeks to optimally redistribute the burden between processing resources to decrease the total runtime. 2. It also enhances the efficiency of all cloud - based services.	Experiment al Tests
Linear scheduling for task and resource	Dynamic	Batch Mode	Priority threshold value	Resource package	1. It tries to increase the use of resources, system performance. 2. Improved cloud resource performance.	Nimbus and Cumulus Services.

V. RESULT

This section describes the result obtained by experimentation of the algorithm using cloudSim

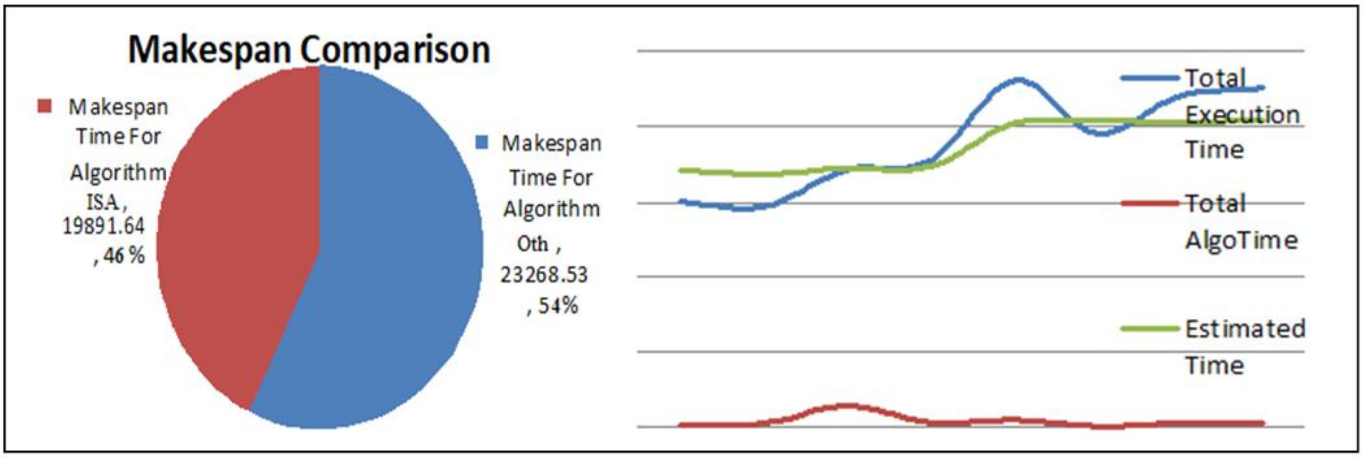


Figure 5. Makespan Time Comparision of ISA and other algorithm. 2.Line graph of ISA Algo

VI.CONCLUSION AND FUTURE SCOPE

In the case of cloud computing, the scheduling method is crucial. A scheduling method is essential for improving server and resource use, as well as increasing computer performance. As a result, a improved scheduling mechanism or method for scheduling jobs in the cloud is proposed in this paper. This method is a very simple and novel method for scheduling jobs that is also very efficient. This Improved scheduling method is superior to other proposed algorithms, or methods because it helps to schedule the task and jobs in very efficient manner as priority assign to the jobs with minimising the makespan. And also it increases resource utilisation and server utilization.

The proposed method is straightforward and simple to grasp. This study is the result of a tenacious effort on my part to investigate various facets of the scheduling process as well as detection. As it is depicted from the line graph.

VII. REFERENCES

[1]. K. Liu; Y. Yang; J. Chen, X. Liu; D. Yuan; H. Jin, “A Compromised-Time- Cost Scheduling Algorithm in SwinDeW-C for Instance-intensive Cost- Constrained Workflows on Cloud Computing Plat- form”, International Journal of High Performance Computing Applications, vol.24, May,2010, Page no.4 445 456.

[2]. Suraj Pandey1; LinlinWu1; Siddeswara Mayura Guru; Rajkumar Buyya, “A Particle Swarm Opti- mization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments” 24th IEEE International Conference onAdvanced Information Networking and Applications, 2010.

[3]. Cui Lin, Shiyong Lu, “Scheduling ScientificWorkflows Elasticallyfor Cloud Comput- ing” inIEEE 4th International Conferenceon Cloud Computing, 2011.

[4]. Salim Bitam, “Bees LifeAlgorithm for Job Sched- uling in Cloud Computing,” in second international conferenceoncommunicationandinformationtech- nology, Feb 2012.

[5]. Abirami S.P and Shalini Ramanathan, “ Linear Scheduling Strategy for Resource Allocation in Cloud Environment”, InternationalJournalon Cloud Computing: Services and Architecture(IJCCSA),Vol.2, No.1,February 2012.

[6]. Wei Wang, Guosun Zeng, Daizhong Tang, Jing Yao, “Cloud-DLS: Dynamictrusted scheduling for Cloud computing”, SciVerse ScienceDirect , Ex- pert Systems withApplications 39, 2012.

[7]. R. Santhosh, T. Ravichandran, “Pre-emptive Scheduling ofOn-line RealTime ServiceswithTask Migration for Cloud Computing”, International Conference on Pattern Recognition, IEEE, Feburary 2013.

[8]. El-Sayed T. El-kenawy, Ali Ibraheem El-Desoky, Mohamed F. Al-rahamawy “Extended Max-Min Scheduling Using Petri Net and Load Balancing” International Journal of Soft Computing and Engi- neering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-4, September 2012.

[9]. Xiaomin Zhua, Chuan Hea, Kenli Li, Xiao Qin, “Adaptive energy-efficient scheduling for realtime tasks on DVS-enabled heterogeneous clusters”, J.Parallel Distrib. Comput, SciVerse ScienceDirect, 2012, Elsevier Inc.

[10]. Anish Das Sarma, Christopher Olston,Xiaodan Wang, Randal Burns : CoScan: Cooperative Scan Sharing in the Cloud.

Cite this article as :

Varinder Saggar, Manoj Kumar Srivastava, " Improved Scheduling Algorithm in Cloud Computing", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 8 Issue 4, pp. 156-161, July-August 2021.
Journal URL : https://ijsrset.com/IJSRSET21827