Azerbaijan Journal of High Performance Computing, Vol 5, Issue 1, 2022, pp. 131-142
https://doi.org/10.32010/26166127.2022.5.1.131.142

Task Scheduling Algorithms in Cloud Computing

Aliaa Kadhim Gabbar Alwaeli and Karrar Ezzulddin Kareem Al-Hamami

Islamic Azad University, Tehran, Iran, aliaa85aliaa@gmail.com, karrare.alhamami@student.uokufa.edu.iq

*Correspondence: Aliaa Kadhim Gabbar Alwaeli, Islamic Aazd University, Iran, Tehran, aliaa85aliaa@gmail.com

Abstract

Utilizing virtualization technology, a cloud computing service provides on-demand access to computer resources & services through the internet. There are new ways to control the functioning of cloud resources since they are constructed in diverse places. More than one algorithm Possibly included in these plans. One of the most important components of a high-performing cloud computing system is the scheduling mechanism. Fault tolerance & load balancing methods are also included in the scheduling system, not only task scheduling strategies. Fault handling Possibly accomplished via using scheduling systems. We analyze & contrast a few different scheduling algorithms to see what they have going for them in terms of benefits & drawbacks.

Keyword: Cloud Computing, Scheduling Algorithm, Task Scheduling, Scheduling Approaches.

1. Introduction

Clouds Possibly employed for a variety of reasons & in a variety of industries. Virtualized resources (like hardware, development platforms, &/or services) may then be dynamically changed to adapt to a varied load, allowing for optimal resource efficiency. Software as a service development might be influenced via the cloud. Today, software engineering techniques are not optimal, & the end product frequently suffers

Fig. 1. Cloude computing

because of this. There is still much manual work involved in the software development process. When different software engineering teams (e.g., development, system test, user-acceptance test/staging) collaborate on the same project, scheduling hardware & software resources is an example of such a method Service-Oriented Architecture (SOA)-based infrastructures exacerbate the situation. Staged deployment of SOA systems is difficult because of the system's scalability & dependency on other services. As a result, virtualization & application simulation may eliminate the need for back-end systems & other external services (Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M., 2008; Nizamic, F., Groenboom, R., & Lazovik, A., 2011). Utilizing cloud infrastructures, these simulators Possibly used to provide a flexible platform to serve the software development & testing teams. Even with virtualization & virtualized services, it is necessary to carefully manage system dependencies to guarantee that all services are in place & that resources that are not needed are not launched (or shut down) to make the best utilization of resources. In the absence of automated scheduling, this activity is very difficult & error-prone, resulting in a lack of potential savings. As seen in Fig. 1 (Alam, T., 2020),

1. There are several interesting characteristics of cloud infrastructure architectures that render them appealing for future IT apps & service (Soni, A. N., 2020)

1.1. Self-service on-demand

CUP time, storage, network connection, worker time, web applications, & so on Possibly outsourced to consumers without any human interaction as needed.

1.2. Cost-efficiency

If not completely free, the services offered via cloud service providers are very cost-effective. No legitimate argument exists for purchasing a base, & therefore reducing maintenance expenses, as a result of the pricing model.

1.3. Pooling Capital

Tools for both physical & virtual processing are included in the cloud. These services

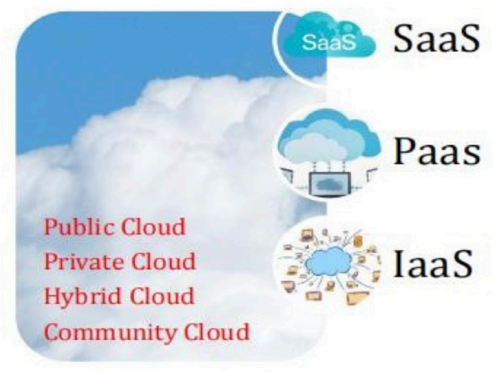

Fig. 2. Cloude-based services and deployment model

are not covered via the field since the customer has no say or access to information about their location.

1.4. Scalability

The cornerstone of cloud computing is entirely scalable. Vendors of cloud services will be able to connect more hubs & servers to the cloud by making minor changes to the foundation & code.

1.5. Dependability

In cloud computing, this is accomplished via using a large number of repeating locations. High resiliency makes the Cloud ideal for disaster relief & business-critical organizations.

1.6. Virtualization

With cloud infrastructure, users may access services anywhere, on any device. The services it requires come from the cloud rather than a well-known company. You can get whatever you need to be done online, whether you are on a PC or a mobile phone. In a simple & safe manner, customers may complete or swap it wherever they are. A task that cannot be completed on a single computer should be assigned to clients.

2. Cloud Providers can Provide Three Services in the Cloud:

As a Service Platform (PaaS) Infrastructure as a Service (IaaS) is shown in Figure 2 (Alam, T., 2020).

2.1. Public Cloud

Most services are provided in a public setting, where users can access a resource pool administered by a host firm (Alam, T., 2020). Due to its presence, this setting will raise significant security concerns (Alam, T., 2020).

A third-party provider provides services that differentiate it from public access (Alam, T., 2020). As a result, it is superior to the prior development paradigm in preventing unwanted access.

2.2. Community Cloud

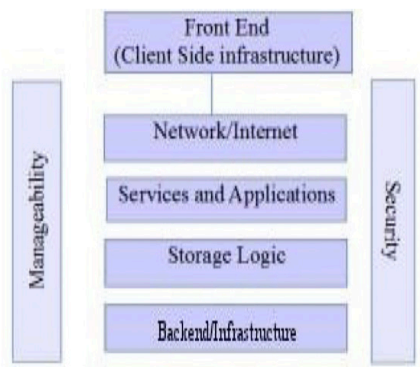

Fig 3. Cloud architecture

Cloud services are made available to a particular group, with each member having equal access to the same services (Alam, T., 2020).

2.3. Hybrid Cloud

Cloud services are delivered through a combination of public, private, & community clouds . It may simply inherit any vulnerability or risk among the parties indicated above.

3. Cloud Architecture

As seen in Figure 3, the cloud computing architecture is composed of several features & components. Each piece is just weakly related. The Front End components offer the apps & interfaces for clients & mobile devices to utilize the cloud platform.

4. Task Scheduling (Amandeep Kaur, R. M., 2018)

The Data Center of a Cloud Service Provider houses a huge number of servers & other computer gear. Thousands of Virtual Machines are used to maximize resource utilization inside a data center. By allocating specific tasks to defined resources, task scheduling Possibly utilized to improve resource utilization. It improves the service's quality & performance automatically. There are two task scheduling algorithms; the 1st is Bulk Mode. A random resource is chosen if many resources give the same amount of execution time. Heuristic Algorithms assess which task requires the least amount of time to complete & then assign the necessary resources to the job at hand.

The FCFS Algorithm, the Round-Robin Algorithm, the Min-Min Algorithm, & the Max-Min Algorithm are all examples of BMHAs that Possibly used.

Heuristic algorithms prioritize work in an online fashion.

They are scheduled in the sequence in which they enter the system.

Several Significant Terminologies Used in These Scheduling Algorithms

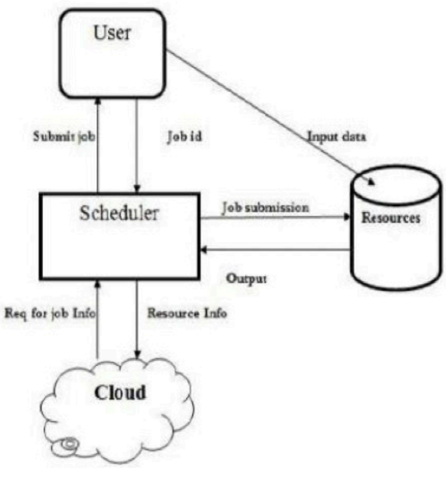

Fig. 4. Cloud computing scheduling

AT stands for Arrival Time.

BT is an abbreviation for Burst Time.

CT is an abbreviation for Completion Time.

T.T = Turnaround Time = Time Required for Completion - Time Required for Arrival = C.T - A.T

W.T. = Waiting Time = Turnaround Time minus Burst Time = T.T. minus B.T.

4.1. Task Scheduling in Cloud Computing Environment

Task scheduling is the process through which incoming tasks are assigned to available resources. The primary objective of work scheduling algorithms is to optimize resource consumption while maintaining the cloud's service parameters. Figure.5

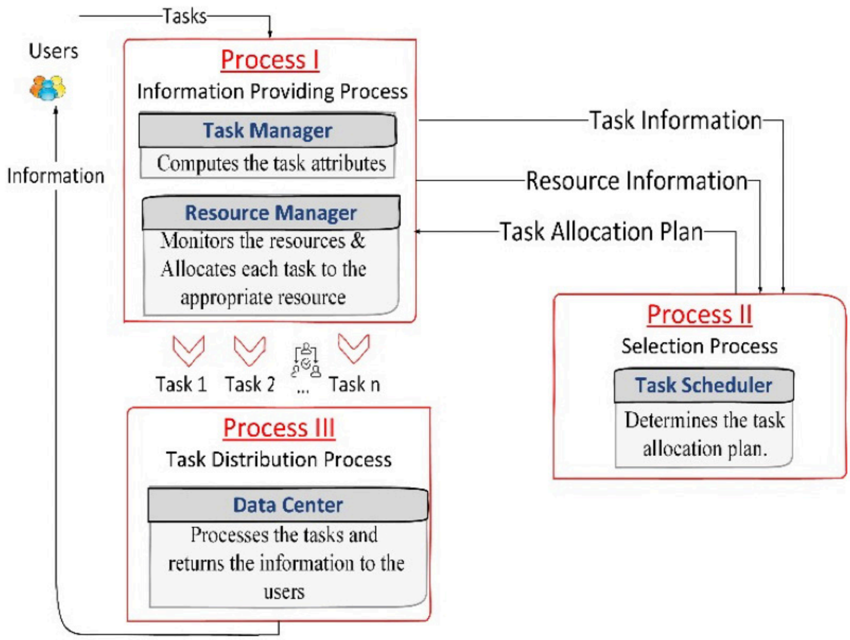

Fig. 5. shows that task scheduling is divided into three processes.

illustrates the fundamental scheduling mechanism that occurs in a cloud system. As seen in the illustration, the work schedule is separated into three steps. The 1st phase is the information gathering process, during which the task scheduler gathers task & resource information from the task manager & resource manager, respectively. The second step is a selection one, in which the target resource is chosen based on the resource's & task's unique criteria. These factors include the size of the job, its priority, its dependability factor, its activity-based cost, & the task's dynamic slotted duration. The task scheduler then delivers the resource management and the job allocation plan. The last step is task distribution. The task manager then assigns each task to the proper resources.

Scheduling is a problem with a broad solution area in cloud computing. As a result, finding an ideal solution takes a lengthy time. Deterministic scheduling algorithms are much simpler & quicker to construct than probabilistic scheduling algorithms since they are all based on a single or a few rules for managing & arranging jobs. However, these algorithms cannot find the ideal solution in a fair amount of time, particularly when the issue grows complicated or the number of jobs becomes excessive.

4.2. Tasks Scheduling Algorithms Overview (Nasr, A. A., & Elbooz, S. A., 2018)

Algorithms for task scheduling are described as the method for allocating resources to perform tasks in order to minimize waiting & execution time.

4.3. Tasks Scheduling Algorithms Definition & Advantages (Aladwani, T., 2020)

Algorithms for task scheduling are described as a collection of rules & policies that are used to allocate jobs to the most appropriate resources (CPU, memory, & bandwidth) in order to maximize performance & resource usage.

• Manage the performance & quality of service (QOS) of cloud computing.

• Take control of the RAM & CPU.

• Appropriate scheduling algorithms maximize resources while keeping overall job execution time to a minimum.

• Improving task fairness across the board.

• Increasing the number of tasks done satisfactorily.

• Scheduling jobs in real-time using a real-time system.

• Increasing the system's throughput.

• Improving load distribution.

5. Different Task Scheduling Approaches Lakshmi, R. D., & Srinivasu, N. (2015)

5.1. 1st-come-1st-serve Scheduling Algorithm

• In this instance, the jobs that arrived initially are serviced 1st. • There are two locations where work Possibly queued. When a task is received, it is placed last. Each process is removed sequentially from the top of the queue. This method is simple & fast as a result of the following: • There is no way to organize the items. This implies that each procedure must be completed before a new one can be introduced.

By & large, this sort of algorithm does not perform well with traffic that is very concerned with delays due to the amount of time & delay involved.

• Because context changes occur only when a process completes, no process queues are required, & there is very no scheduling cost.

5.2. Algorithm for Scheduling the Shortest Job 1st

For the SJF Algorithm, determining the shortest possible waiting time is an important part of preemptive algorithmic planning. It is then transmitted to the processor in the shortest burst time possible. The SJF Algorithm is the name given to this approach.

Characteristics

For SJF to work, it has to figure out what the next processor request will be.

It shortens the average waiting time because it runs small processes 1st, then big, to speed things up.

• When many small processes run simultaneously, they will starve.

5.3. Round Robin Scheduling Algorithm

Processes are executed in the same manner as in FIFO, except that they are time-limited via the CPU, a concept called timeslice. The processor proceeds to the following waiting process in the queue if the process is not finished within the processor's time limit. Next, the preemption or new process is put to the bottom of a queue while other tasks continue to fill it.

Characteristics

The CPU efficiency will be reduced if we utilize a shorter time slice or quantum.

• If we utilize a lengthy timeslice or quantum, the reaction time will be slow.

• Due to the length of the wait, there is very little probability that deadlines will be reached.

5.4. Priority Scheduling Algorithm

Each process is allocated a priority in this method, & processes are run according to their priority. Priorities with the same priority level utilize FCFS.

Characteristics

• If there are a large number of processes with the same priority results in a long wait time.

• Prioritized processes result in reduced waiting time & less delay.

Processes with a low priority risk starving.

5.5. Min–Min Algorithm:

This method prioritizes little activities, delaying big jobs over an extended period of time.

5.6. Algorithm Max – Min:

This algorithm prioritizes big activities, causing little jobs to be delayed for an extended period of time.

Table 1. Process with its id & burst time

Process ID	Burst Time (ms)
P1	10
P2	2
P3	8
P4	6

Table 2. With its identification, burst time, & priority

Process ID	Burst Time (ms)	Priority
P1	10	3
P2	2	1

P3	8	4
P4	6	2

Table 3. Time spent waiting for individual processes & the average time spent waiting for each scheduling waiting time (ms)

Process ID	FCFS	SJF	Round Robin	Priority Scheduling
P1	0	16	12	8
P2	10	0	5	0
P3	12	8	17	18
P4	20	2	20	2
Avg Waiting Time	10.5	6.5	13.5	7

Table 4. Time required for each individual process & the average time allowed for each scheduling time for turnaround

Process ID	FCFS	SJF	Round Robin	Priority Scheduling
P1	10	26	22	18
P2	12	2	7	2
P3	20	16	25	26
P4	26	8	26	8
Avg Turnaround Time	17	13	20	13.5

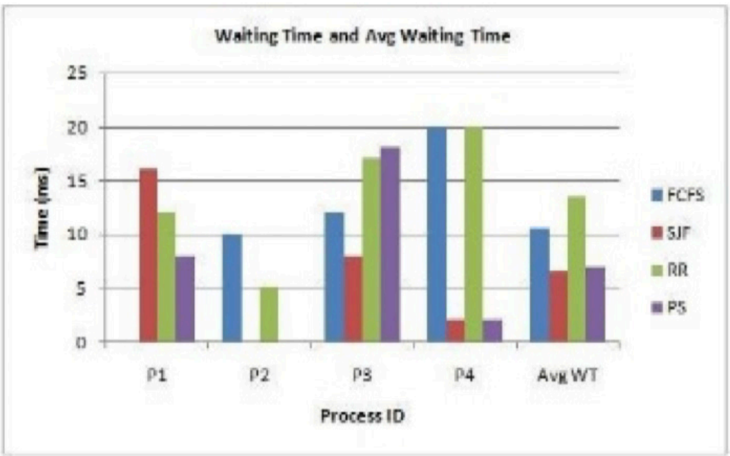

Fig. 6. Algorithm-waiting time comparison of a fundamental scheduling algorithm

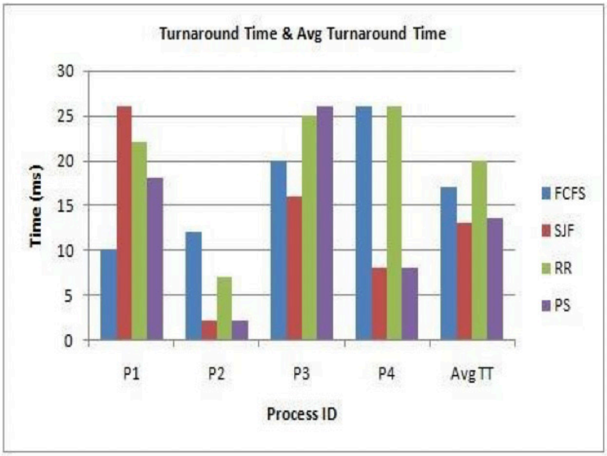

Fig. 7. Fundamental Scheduling Algorithm-Turnaround Time Comparison

5.7. Most fit task scheduling algorithm:

The tasks that fit the best in the queue are done 1st in this algorithm. This algorithm has a high rate of failure.

Consider the following collection of processes, the duration of which is specified in milliseconds in Table1.

A Gantt chart for SJF scheduling Possibly used to calculate a process's turnaround time, which is the time from submission to completion.

The average turnaround time for processes P1, P2, P3, & P4 is 8.

The waiting time for a process is calculated as the time it takes for the process to wait in the ready queue, as shown on the SJF scheduling Gantt chart. One ms is the average of the waiting periods for each of the four processes, which are 16, 0, 8, & 2.

Tables 3 & 4 show the turnaround & waiting times for all other algorithms.

1st Come, 1st Serve (FCFS) & Shortest Job 1st (SJF) is often suited for batch operating systems. However, due to the previous reasoning, Round Robin (RR) & Priority Scheduling (PS) are appropriate for time-sharing systems.

This method, SJF, is the best one for all scheduling algorithms. Consequently, it is an algorithm that can be used in every situation since it has the best possible requirements. Compared to all other critical algorithms, the SJF scheduling algorithm is superior in terms of turnaround time, waiting time, & response time when compared to the Gantt charts in Figure 13 to Figure 16. Additional benefits include excellent throughput & CPU utilization.

This investigation & discussion has led us to believe that the FCFS is simple to understand & is best suited for batch systems with long wait times. Waiting times are kept to a minimum because of SJF scheduling. In terms of turnaround time, this is within the norm. Priority is the foundation for distributing priorities, & the highest priority task Possibly completed via this approach.

1st things 1st, even if it means starvation, must be the rule. Utilizing the round-robin policy, the preemptive round-robin scheduling approach is preemptive. Useful in both interactive & time-sharing environments, it is one of the most common scheduling techniques. Only after thoroughly evaluating & including a scheduling approach in the operating system can the algorithm's performance in real-time systems be assessed. In terms of turnaround time, this is within the norm.

Consequently, even though hunger arises from prioritizing tasks according to their importance, the highest-priority tasks Possibly completed 1st while the lowest-priority

Table 5. Experiment Result table

S. No	Algorithms	Evaluation Criteria	Advantages	Disadvantage
1	1st Come 1st Serve	There is much waiting involved. high & low The Waiting Period	Easy implementation	No other criteria for scheduling
2	Round Robin	There is a moderate delay in response time; nevertheless, the turnaround time is fast.	Less complex & Appropriately load is balanced	Preemption is required
3	Shortest 1st Scheduling Job	Less waiting time, Short execution time, maximum throughput	CPU is allocated to the process that has the minimum amount of burst time.	Difficult to understand
4	Priority based	High turnaround time & high response time	It is designed based on multiple criteria decision-making model.	Difficult to understand & code

tasks must wait. The preemptive round robin scheduling approach is based on the round robin policy & is preemptive. Interactive systems & time sharing systems utilize this scheduling technique, which can also be found in other scheduling algorithms. The algorithm's performance in real-time systems can only be assessed after assessing the scheduling method's implementation & included in the operating system.

6. Results & Discussion

SJF has a quicker turnaround time and a shorter waiting time than any other algorithm, according to the results of this study. Additionally, SJF seems to optimize throughput and CPU use. FCFS has the most extended wait and turnaround times owing to its frequent yet short procedures. When dealing with time-sensitive traffic, it is best to avoid it because of the long waits required in even simple activities. Even if each activity is given equal time in RR, certain situations may need a reduction in the average waiting time. If you do not choose your time slice or quantum wisely, you risk slowing down your processing in some way. Due to the similarity between Priority Queuing and FCFS, Priority Queuing may become troublesome when there are many processes with equivalent priority.

7. Conclusions

The SJF scheduling algorithm aims to provide the best scheduling criteria for all work kinds. SJF scheduling's consideration of the shortest procedures often increases waiting time for lengthy processes. Moreover, the lengthy procedure will never be completed, even though it results in the shortest average wait time & fastest average turnaround time.

The shortest job first scheduling method takes a different approach; the primary advantage of this technique is that it provides the smallest average waiting time.

It is suggested that any simulation performed on any CPU scheduling strategy is imprecise.

References

Aladwani, T. (2020). Types of task scheduling algorithms in cloud computing environment. *Scheduling Problems-New Applications and Trends.*

Alam, T. (2020). Cloud Computing and its role in the Information Technology. *IAIC Transactions on Sustainable Digital Innovation (ITSDI), 1(*2), 108-115.

Alrehaili, A., Mir, A., & Junaid, M. (2020). A Retrospect of Prominent Cloud Security Algorithms. *Int. J. Innov. Technol. Exploring Eng.(IJITEE), 9,* 2278-3075.

Amandeep Kaur, R. M. (2018). Different task scheduling algorithms in cloud computing. *International Journal of Latest Trends in Engineering and Technology, 9*(3), 217-223.

Black, D. L. (1990). Scheduling support for concurrency and parallelism in the Mach operating system. *Computer, 23*(5), 35-43.

Djebbar, E. I., & Belalem, G. (2016, June). Tasks scheduling and resource allocation for high data management in scientific cloud computing environment. In *International Conference on Mobile, Secure, and Programmable Networking (pp. 16-27). Springer, Cham.*

Lakshmi, R. D., & Srinivasu, N. (2015). A review and analysis of task scheduling algorithms in different Cloud computing environments. *International Journal of Computer Science and Mobile Computing, 4*(12), 235-241.

Lakshmi, R. D., & Srinivasu, N. (2015). A review and analysis of task scheduling algorithms in different Cloud computing environments. *International Journal of Computer Science and Mobile Computing, 4(*12), 235-241.

Lepakshi, V. A., & Prashanth, C. S. R. (2013). A study on task scheduling algorithms in cloud computing. *International Journal of Engineering and Innovative Technology (IJEIT), 2*(11), 119-125.

Nasr, A. A., & Elbooz, S. A. (2018). Scheduling strategies in cloud computing: Methods and implementations. *American Journal of Engineering and Applied Sciences, 11*(2), 426-432.

Neeta Patil & Deepak Aeloor, "A REVIEW - DIFFERENT SCHEDULING ALGORITHMS IN CLOUD COMPUTING ENVIRONMENT", *11th International Conference on Intelligent Systems & Control (ISCO) ,* pp: 182-185, IEEE, 2017.

Nizamic, F., Groenboom, R., & Lazovik, A. (2011). Testing for highly distributed service-oriented systems using virtual environments. *Proceedings of 17th Dutch Testing Day.*

Patil, N., & Aeloor, D. (2017, January). A review-different scheduling algorithms in cloud computing environment. In *2017 11th International Conference on Intelligent Systems and Control (ISCO) (pp. 182-185).* IEEE.

Saleh, H., Nashaat, H., Saber, W., & Harb, H. M. (2018). IPSO task scheduling algorithm for large scale data in cloud computing environment. *IEEE Access, 7,* 5412-5420.

Sindhu, M., Rajkamal, R., & Vigneshwaran, P. (2010, June). An optimum multilevel CPU scheduling algorithm. In *2010 International Conference on Advances in Computer Engineering* (pp. 90-94). IEEE.

Singh, P., Singh, V., & Pandey, A. (2014). Analysis and comparison of CPU scheduling algorithms. *International Journal of Emerging Technology and Advanced Engineering, 4(1)*, 91-95.

Soni, A. N. (2020). Diabetes Mellitus Prediction Using Ensemble Machine Learning Techniques. *Available at SSRN 3642877.*

Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2008). A break in the clouds: towards a cloud definition. *ACM sigcomm computer communication review, 39*(1), 50-55.

Submitted: 18.01.2022
Accepted: 23.05.2022