Asian Journal of Information Technology 13 (8): 442-445, 2014
ISSN: 1682-3915
© Medwell Journals, 2014

A Survey on Scheduling of Resources in Cloud

B. Parkavi and G. Malathy
Department of Computer Science and Engineering,
KSR Institute for Engineering and Technology, Tiruchengode, India

Abstract: Cloud computing is an internet based computing where the computation is moved from single desktop to remote systems. Cloud computing provides a variety of computing resources from servers and storage to enterprise applications such as email, security, backup, etc., delivered over the internet. One of the major problems in cloud is the scheduling of resources dynamically based on user's request by optimizing quality of service, cost and time consumption. Since, resources are shared over a distributed environment, scheduling of those resources to the users is an important task in the cloud computing environment because billing is made as on-demand usage. This study surveyed various scheduling algorithms and compared with different parameters.

Key words: Cloud computing, scheduling, QoS, algorithm, share

INTRODUCTION

Cloud computing is the delivery of computing as a service rather than a product where by sharing of resources, software and information are provided to computers and other devices as a metered service over the network. The cloud delivers a hosting environment that is immediate, flexible, scalable, secure and available while saving corporation's money, time and resources. The National Institute of Standards and Technology (NIST) provides the following definition for cloud computing: "cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" (Mell and Grance, 2011). The services provided by cloud are listed as follows:

- Virtual servers
- Database services
- E-mail applications
- Storage

In a cloud environment, client can access those resources pooled in the cloud by requesting cloud service providers based on client's request, resource is provisioned to the client by pay per usage demand.

During resource provisioning, there may occur delay in response from cloud service provider since, cloud is a way of distributed computing some other clients may request the same resource (or) the server may busy with its resource allocation. Hence, there is a need for scheduling based on client's request and availability of resources in the datacenter. Resource scheduling problem is similar to Banker's algorithm which prevents deadlock by denying or postponing the request if it determines that accepting the request could put the system in an unsafe state. When a new process enters a system, it must declare the maximum number of instances of each resource type that may not exceed the total number of resources in the system. Figure 1 shows the architecture of cloud where the bottom layer belongs to deployment models, the middle layer is service models and the top layer describes its features.

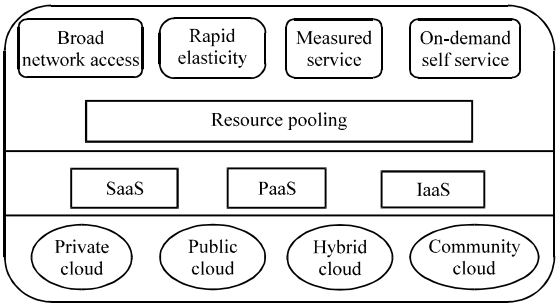

Fig. 1: Cloud architecture

Corresponding Author: B. Parkavi, Department of Computer Science and Engineering,
KSR Institute for Engineering and Technology, Tiruchengode, India

KEY CONCEPTS

Cloud computing: A cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and represented as one or more unified computing resources based on service level agreements established through negotiation between the service providers and consumers. Based on the type of services rendering, there are three cloud service models such as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The IaaS Model include processing, storage and network in which user can deploy and run the chosen software including operating systems and applications (Buyya *et al*., 2009). Examples of commercial implementations of IaaS include IBM SmartCloud™ Enterprise and Enterprise+, IBM Smart Cloud managed backup, Amazon Elastic Compute Cloud (EC2). The PaaS Model includes services that build on IaaS services. They add value to the IaaS services by providing a platform in which the cloud users can provision their own applications or conduct application development activities. Examples of commercial implementations of PaaS environments include IBM SmartCloud Application Services, Amazon Relational Database Service and Microsoft Windows Azure (Buyya *et al*., 2009). The SaaS Model provides software services that are complete applications that are ready to use. The cloud user simply connects to the application which is running at a remote location the user might not know where the process running. Examples of commercial implementations of SaaS environments include IBM Payment Systems, IBM Smart Cloud for Social Business, People Soft HR and Google Apps for Business.

Job scheduling: Scheduling in distributed systems is spreading the load on processors and maximizing their utilization while minimizing the total task execution time. Job scheduling is one of the most famous optimization problems, plays a key role to improve flexible and reliable systems. The main purpose is to schedule jobs to the adaptable resources in accordance with adaptable time which involves finding out a proper sequence in which jobs can be executed under transaction logic constraints. Figure 2 shows an outline of job scheduling in which the resources are allocated from data center to the client.

Task scheduling: The task scheduling is the key role in cloud computing in which a job is divided into number of tasks such that each one runs on different processors in parallel processing environment. Task scheduling problems are primary which relate to the efficiency of the whole cloud computing facilities. Because of different QoS parameters such as CPU speed, CPU utilization, turnaround time, throughput, waiting time, etc., task scheduling in cloud computing is different from conventional distributed computing environment.

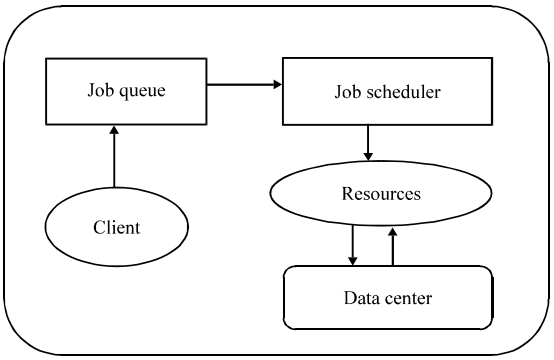

Fig. 2: Job scheduling

Categories of scheduling

Static scheduling: A schedule computed before executing the job. The user can be promised a deadline and the resources can be planned. It is computing optimal schedule of NP-hard.

Dynamic scheduling: Task characteristics are not available and do not allow prior knowledge of when a job finishes. A user cannot be promised a deadline and cloud cannot plan ahead on future resource usage.

EXISTING RESOURCE SCHEDULING ALGORITHM

The following resource scheduling algorithms are currently prevalent in cloud and these algorithms are summarized:

Analysis of First Come First Serve (FCFS) Parallel Job Scheduling: Schwiegelshohn and Yahyapour (1998) has analysed the working of First Come First Serve (FCFS) algorithm in which each job specifies the number of nodes required and the scheduler will processes those jobs in the order of their arrival. When there is a sufficient number of nodes to process the job then the scheduler dispatches the job to run on these nodes else it waits for the currently running job to finish. So, it causes fragmentation of nodes and delay in getting resources.

Improved Utilization and Responsiveness with Gang Scheduling: Feitelson and Jettee (1997) have proposed that gang scheduling improves node utilization and responsiveness over parallel jobs. It allows sharing of

Table 1: Comparison of Existing Scheduling algorithms

Scheduling algorithm	Scheduling parameter	Scheduling factor	Findings	Environment	Tools
First Come First Serve	Make span	An array of job queue	Reduce make span in a DAG (Directed Acyclic Graph)	Grid environment	GridSim
Gang Scheduling	Processing time	Grouped task	Improves node utilization	Grid environment	GridSim
Paired Gang Scheduling	Quality of service, finish time	Grouped task	No interference between parallel processes	Cloud environment	Amazon EC2
Gang Scheduling, Backfilling and Migration	Make span	Workflow with large number of job	Minimize the execution time, make span is minimized	Cloud environment	Amazon EC2
Conservative Migration Supported Backfilling	Resource utilization time, cost	Multiple workflows	Schedule the workflow dynamically	Cloud environment	CloudSim
Conservative Migration and Consolidation Supported Backfilling	Performance,CPU time	Multiple workflows	Resource starvation is reduced	Cloud environment	CloudSim
Double Level Priority Based Task Scheduling	Priority to each queue	An array of job queue	Less finish time	Cloud environment	CloudSim

resources among multiple parallel jobs in which the computing capacity of a node is divided into time slices. The allocation of time slices of different nodes to parallel processes is coordinated by OS support. It manages to make all the processes of a job progress together so that one process will not be in sleep state when another process needs to communicate with it. So, it stretches the execution time of individual jobs.

Paired Gang Scheduling: Wiseman and Feitelson (2003) has proposed that paired gang scheduling tries to overcome the drawbacks of gang scheduling in which it utilizes the system resources well without causing interference between the processes of competing jobs. The processes will not have to wait much because a process which occupies the CPU most of the time will be matched with a process that occupies an I/O device most of the time, so they will not interfere with each other's work. On the other hand, the CPU and the I/O devices will not be idle while there are jobs which can be executed.

An Integrated Approach to Parallel Scheduling Using Gang-scheduling, Backfilling and Migration: Zhang *et al*. (2003) have proposed an effective scheduling strategy to improve response time, throughput and utilization of resources in cloud. Gang-scheduling and backfilling are two optimization techniques that operate on orthogonal axes, space for backfilling and time for gang scheduling and the proposed technique is made by treating each of the virtual machines created by gang-scheduling as a target for backfilling. The difficulty arises in estimating the execution time for parallel jobs so migration is taken into account which improves the performance of gang-scheduling without the need for job execution time estimates.

Conservative Migration Supported Back Filling (CMBF): Liu *et al*. (2013) has proposed CMBF algorithm which schedules jobs according to their arrival time when there is enough number of nodes. If the number of idle nodes is not sufficient for a job then another job with a later arrival time but smaller node number requirement may be scheduled to run via backfilling. So, it avoids starvation of a pre-empted job but it cannot handle VM (Virtual Machine) resources effectively.

Conservative Migration and Consolidation Supported Backfilling (CMCBF): Liu *et al*. (2013) has proposed CMCBF algorithm which overcomes the drawbacks of CMBF algorithm. It ensures a job to run in foreground VMs whenever the number of foreground VMs that are either idle or occupied by jobs arriving later than it satisfies its node requirement. It also allows jobs to run in background VMs simultaneously with those foreground VMs to improve node utilization.

Double Level Priority Based Task Scheduling with Energy Awareness in Cloud Computing: Parikh and Sinha (2011) has proposed a double level priority based task scheduling in which three different waiting-queues are considered such as low-priority queue, medium-priority queue and high-priority queue and the local scheduler maintains these queues. The scheduler needs to effectively schedule tasks in terms of both performance and energy consumption. For this, power-threshold of processor is monitored. When a processor reaches its power threshold, the task is assigned into another processor. Table 1 shows the comparison of Existing Scheduling algorithms.

CONCLUSION

Resource scheduling is one of the major issues in cloud computing environment. The resources are pooled in multiple cloud data centers where the user's request can be satisfied with delay in processing. In this study, researchers have analyse various scheduling algorithm

and tabulated various parameter. Existing Scheduling algorithm gives high throughput and cost effective but they do not consider reliability and availability. So, researchers need algorithm that improves availability and reliability of resources in cloud computing environment.

REFERENCES

Buyya, R., C.S. Yeo, S. Venugopal, J. Broberg and I. Brandic, 2009. Cloud computing and emerging IT platforms: Vision, hype and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst., 25: 599-616.

Feitelson, D.G. and M.A. Jettee, 1997. Improved Utilization and Responsiveness with Gang Scheduling. In: Job Scheduling Strategies for Parallel Processing, Feitelson, D.G. and L. Rudolph (Eds.). Springer, Berlin, Heidelberg, ISBN: 978-3-540-63574-1, pp: 238-261.

Liu, X., C. Wang, B.B. Zhou, J. Chen, T. Yang and A.Y. Zomaya, 2013. Priority-based consolidation of parallel workloads in the cloud. IEEE Transa. Parallel Distri. Syst., 24: 1874-1884.

Mell, P. and T. Grance, 2011. The NIST definition of cloud computing: Recommendations of the National Institute of Standards and Technology. Special Publication 800-145. National Institute of Standard and Technology, U.S. Department of Commerce, September 2011, Gaithersburg, MD., USA. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

Parikh, S.V. and R. Sinha, 2011. Double level priority based task scheduling with energy awareness in cloud computing. Int. J. Eng. Technol., 2: 142-147.

Schwiegelshohn, U. and R. Yahyapour, 1998. Analysis of first-come-first-serve parallel job scheduling. Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, January 25-27, 1998, San Francisco, California, USA., pp: 629-638.

Wiseman, Y. and D.G. Feitelson, 2003. Paired gang scheduling. IEEE Trans. Parallel Distri. Syst., 14: 581-592.

Zhang, Y., H. Franke, J.E. Moreira and A. Sivasubramaniam, 2003. An integrated approach to parallel scheduling using gang-scheduling, backfilling and migration. IEEE Trans. Parallel Distri. Syst., 14: 236-247.