Issue 4, Vol.3 (May 2014)
ISSN 2249-6149

Optimized Scheduling Algorithm tolerant towards Fault in Cloud Computing

Ms.P.Jamuna[1], **R.Anand Kumar**[2], **I.Sudha**[3]
[1] Department of Computer science and Engineering,
Achariya College of Engineering Technology,
[2] Department of Computer science and Engineering,
Achariya College of Engineering Technology,
[3] Department of Computer science and Engineering,
Achariya College of Engineering Technology,

ABSTRACT—Cloud computing is a marketing term for Information Technologies that provide computation software data access and storage services that does not require end-user knowledge. Cloud computing also allows for a lot of flexibility, depending on the demand, suitable resources will be allotted for a specific task execution. Various strategies of resource scheduling in cloud computing used either to improve system operating efficiency, or to improve user satisfaction. The cloud scheduling process is responsible for selection of best suitable resources for task execution, by taking some static and dynamic parameters and restrictions of tasks into consideration. Cloud Service providers ensures that how resources effectively used and to their best capacity so that resource potential is not left unused. In this paper, we provide a scheduling method which focuses on parameters like task completion time or task execution cost etc and also addresses these major challenges of task scheduling in cloud. The incoming tasks are grouped on the basis of task requirement like minimum execution time or minimum cost and prioritized. The main objective of this algorithm is to reduce the scheduling length and also tolerant towards fault.

Keywords-scheduling length,Round Robin,FCFS,Resource scheduling

1. Introduction

Cloud computing is known as a provider of dynamic services using very large scalable and virtualized resources over the Internet. Various definitions and interpretations of “clouds” and / or “cloud computing” exist. With particular respect to the various usage scopes the term is employed to, we will try to give a representative (as opposed to complete) set of definitions as recommendation towards future usage in the cloud computing related research space. We try to capture an abstract term in a way that best represents the technological aspects and issues related to it. In its broadest form, we can define a 'cloud' is an elastic execution environment of resources involving multiple stakeholders and providing a metered service at multiple granularities for a specified level of quality of service. To be more specific, a cloud is a platform or infrastructure that enables execution of code (services, applications etc.), in a managed and elastic fashion, whereas “managed” means that reliability according to predefined quality parameters is automatically ensured and “elastic” implies that the resources are put to use according to actual current requirements observing overarching requirement definitions – implicitly, elasticity includes both up- and downward scalability of resources and data, but also load-balancing of data throughput. Job scheduling is one of the major activities performed in all the computing environments. Cloud computing is one the upcoming latest technology which is developing drastically To efficiently increase the working of cloud computing environments, job scheduling is one the tasks performed in order to gain maximum profit. The goal of scheduling algorithms in distributed systems is spreading the load on processors and maximizing their utilization while minimizing the total task execution time Job scheduling, 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. There are main two categories of scheduling algorithm.1) Static scheduling algorithm and 2) Dynamic scheduling algorithm. Both have their own advantage and limitation. Dynamic scheduling algorithms have higher performance than static algorithm but have a lot of overhead compare to it.

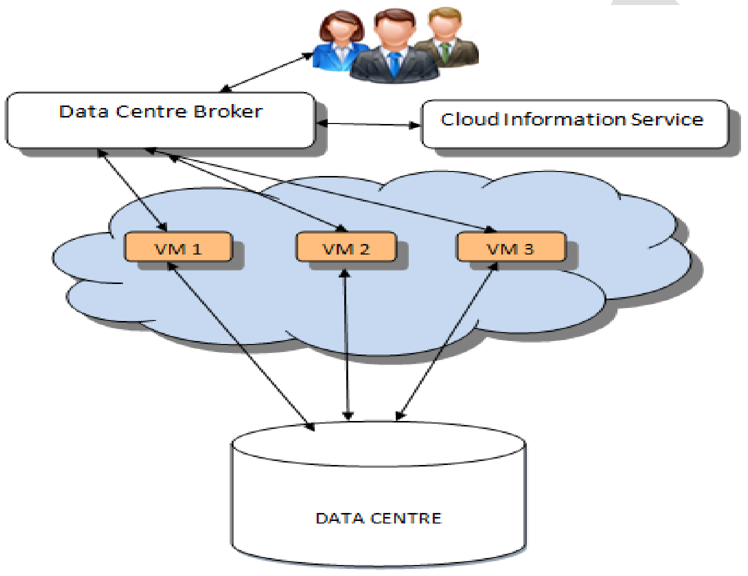

Fig.1 Scheduling in Cloud Environment

2. Scheduling Process

Scheduling process in cloud environment is generalized into three stages are

a. Resource discovering and filtering – Data centre Broker detect the resources present in the network system and collects those information regarding its status.
b. Resource selection – The respective resources can be selected based on certain parameters of task and resource.
c. Task submission – Finally the task is submitted to resource selected.

Two main components are in service request scheduling are Categorization and Scheduling. Categorization which receives the request from the user, process it and classifies as smaller unit. These task units can be scheduled based upon its Priority. Before that each task unit can be assigned with random priority. Scheduling unit which contains several units and having its own priority. Scheduling unit execute the task based on specified algorithm. The task unit having least deadline will be scheduled first.

There has been various types of scheduling algorithm exist in distributed computing system. Most of them can be applied in the cloud environment with suitable verifications. The main advantage of job scheduling algorithm is to achieve a high performance computing and the best system throughput. Traditional job scheduling algorithms are not able to provide scheduling in the cloud environments. According to a simple classification, job scheduling algorithms in cloud computing can be categorized into two main groups; 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. The main examples of BMHA based algorithms are; First Come First Served scheduling algorithm (FCFS), Round Robin scheduling algorithm (RR), Min–Min algorithm and Max–Min algorithm. By On-line mode heuristic scheduling algorithm, Jobs are scheduled when they arrive in the system. Since the cloud environment is a heterogeneous system and the speed of each processor varies quickly, the on-line mode heuristic scheduling algorithms are more appropriate for a cloud environment. Most fit task scheduling algorithm (MFTF) is suitable example of On-line mode heuristic scheduling algorithm.

a. First Come First Serve Algorithm:

Job in the queue which comes first is served. This algorithm is simple and fast.

b. Round Robin algorithm:

In the round robin scheduling, processes are dispatched in a FIFO manner but are given a limited amount of CPU time called a time-slice or a quantum. If a process does not complete before its CPU-time expires, the CPU is preempted and given to the next process waiting in a queue. The preempted process is then placed at the back of the ready list.

c. Min–Min algorithm:

This algorithm chooses small tasks to be executed firstly, which in turn large task delays for long time.

d. Max – Min algorithm:

This algorithm chooses large tasks to be executed firstly, which in turn small task delays for long time.

e. Most fit task scheduling algorithm:

In this algorithm task which fit best in queue are executed first. This algorithm has high failure ratio.

f. Priority scheduling algorithm:

The basic idea is straightforward: each process is assigned a priority, and priority is allowed to run. Equal-Priority processes are scheduled in FCFS order. The shortest-Job-First (SJF) algorithm is a special case of general priority scheduling algorithm. An SJF algorithm is simply a priority algorithm where the priority is the inverse of the (predicted) next CPU burst. That is, the longer the CPU burst, the lower the priority and vice versa. Priority can be defined either internally or externally. Internally defined priorities use some measurable quantities or qualities to compute priority of a process.

A multiprocessor system is regularly deployed for executing computationally concentrated similar applications with various computing necessities. The processor will map the task and it will specify the execution time which is one of the important steps in parallel processing. This idea is called as task scheduling; it will determine the efficiency of the application for parallelization in multiprocessor systems. In multiprocessor systems the Component failures may occur. Consequently, there is a growing need for increasing fault tolerant techniques. In reality, the chance of failure occurrences is increased by the fact that many corresponding applications are getting larger, and might require long period of time for execution.

The proposed algorithm aims to tolerate multiple processor failures and achieving a least possible schedule length. The goal of this algorithm is to reduce the communications duplication between the task replicas, while keeping the fault tolerance of the necessary number of processor failures.

3. OPTIMIZED SCHEDULING ALGORITHM

The main objective of this algorithm is to reduce the scheduling length and also tolerant towards Byzantine fault [4].Mapping the tasks to the processors with diverse capabilities like different speed a processing capability. In this proposed algorithm Mapping the task to directed cyclic graph to processor in distributed computing system, which aim to minimize the scheduling length and tolerate the number of processor failure (NPF).In order to achieve this, replication scheme was used replicating the processor as (npf+1) copies to allocate the each task to different processor and also each node was scheduled to multiple the processors in order to achieve fault tolerant. In improved algorithm, free nodes are arranged by the priority value which equals to tlevel + blevel of node. Where level refers the dynamic top priority level and it is calculated using

$$tlevel\ (n_i) = \max_{n_j \in pred\ (n_i)} \{FT(n_j, Proc\ (n_j)) + c(n_i, n_j)\},$$

Where, FT (n_j, Proc (n_j)) – finish time of node n_j.
Proc (n_j) – Previously scheduled processor.
C (n_i , n_j) – communication cost between nodes n_i and n_j.
blevel refers the static bottom priority level and its calculated using

$$blevel\ (n_i) = \max_{n_j \in succ\ (n_i)} \{\overline{w(n_i)} + c(n_i, n_j) + blevel\ (n_j)\},$$

Where, $w(n_i)$ is the average execution time for the node n_i.. The finish time of free node which has highest priority is calculated as

$$FT(n, P_i) = C(P_i) \times w(n) + \max\{ \max_{n_j \in pred(n)} [\min_{1 \le k \le npf+1} (FT(n_j^k, Proc(n_j^k))) + c(n_j, n)], r(P_i)\},$$

Where $r(P_i)$ – ready time for the processor
The nodes are already scheduled onto NPF+1, replication processor which deliver the minimum finish time of node.

Where n_j^k refer to the Kth replication of node n_j.
The scheduling length can be calculated as

$$SL = \max_{n \in V} \{ \min_{1 \le k \le npf+1} [FT(n^k, Proc(n^k))]\}$$

4. CONCLUSION

Time and Cost are the main challenge of every IT engineer to build up products that can improve the business performance in the cloud based IT sectors. The Current strategy lack in efficient scheduling and resource allocation techniques which leads to increased operational cost and time. The proposed algorithm aims to tolerate multiple processor failures and achieving a least possible schedule length. The goal of this algorithm is to reduce the communications duplication between the task replicas, while keeping the fault tolerance of the necessary number of processor failures. Thus time and cost can be efficiently managed.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] Tabbaa N, Entezari-Maleki R, Movaghar A. A Fault Tolerant Scheduling Algorithm for DAG Applications in Cluster Environments. In: International Conference on Digital Information Processing and Communication, Springer-CCIS, Ostrava, Czech Republic, 2011; 188:189-199

[5] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[6] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[7] Provisioning: Service Discovery and Load Balancing “ Cloud Computer Communication and Networks, 2010, pp 195-217.

[8] International Conference on Cluster, Cloud and Grid Computing, pp 15-24. Zhipiao Liu, Shangguang Wang,Qibo Sun.

[9] Report, 2010 Linlin Wu, Saurabh Kumar Garg and Rajkumar Buyya, "SLA-based Resource Allocation for Software as a Service Provider (SaaS) in Cloud Computing Environments", IEEE/ACM International Symposium on Cluster, Cloud and Grid.

[10] Colony Optimization Algorithm", International Journal of Computer Science Issues (IJCSI), 2012, pp 54-58.

[11] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose and R. Buyya "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, vol. 41, no. 1, pp. 23-50, 2011.