ISSN:0975-9646

A.Kousalya et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (3) , 2015, 2687-2690

A Comparative Study of Parallel Job Scheduling Algorithms in Cloud Computing

A.Kousalya[1], R.Radhakrishnan[2]

[1]*Assistant Professor,*

Computer Science and Engineering, Vidhya Mandhir Institute of Technology, Tamilnadu, India

[2]*Principal,*

Computer Science and Engineering, Vidhya Mandhir Institute of Technology, Tamilnadu, India

Abstract--- **The cloud computing has become a popular platform for scientific applications. Cloud computing is a term used to describe a new class of network based computing that takes place over the internet. It offers dynamic flexible resource allocation for reliable and guaranteed services in pay-as-you-use manner to cloud service users. Scheduling the job is most important task in cloud computing environment. The goal of parallel scheduling is to achieve high system throughput, resource utilization and response time. This paper surveyed different types of parallel scheduling algorithms. Existing parallel scheduling algorithms does not consider the node utilization and response time of a parallel workload in the cloud.**

Keywords--- **Cloud computing, parallel job scheduling, Backfilling, Gang scheduling, Task scheduling**

I. INTRODUCTION

Cloud computing is the use of computing resources hardware and software, that are delivered as a service over a network typically the internet. Cloud computing is an environment in which the user use the computing resources in distant data center rather than the local computing systems. The cloud environment offers different virtualized platforms that help user to accomplish their jobs with minimum completion time and minimum costs. End users can access their data from the database in the cloud. Companies are able to rent resources from cloud for storage and other computational purpose so that their infrastructure cost can be reduced significantly.

Scheduling jobs is an important issue in the cloud. There are various scheduling algorithms exist, the main goal of scheduling algorithm is to distribute the load among processors and maximize their utilization while minimize the total execution time. The job scheduler is responsible for assigning the resources to a particular job so that overall computing resources are utilized effectively. Cloud application use virtualization technique to improve its performance, more complex applications are used in data centers, complex application required parallel processing. Parallel computing is a platform that computes the job in parallel manner in different machines independent of location. As parallel processes increases the CPU utilization become low.

II. SCHEDULING

Jobs and resources have to be allocated and scheduled in such a way that cloud users can complete their jobs with minimal time and cost and maximize the throughput of cloud resource provider.

There are two types of parallel scheduling:

- Application scheduling
- Job scheduling

The application scheduling maps a single application tasks to resources to reduce the total response time. The job scheduling chooses an order of jobs for execution to minimize total turn around time. Two types of scheduling strategies are space sharing and time sharing. Time sharing techniques virtualizes the physical machine by slicing the time axis into multiple virtual machines. Space sharing techniques runs the jobs side by side on different nodes of the machine at the same time.

Scheduling process in the cloud run in three stages:

1. **Resource Discovery:**
   Datacenter broker know the status of all the resources that are available in the cloud and also the remaining resources that may available. The resources are generally virtual machines. It frequently collects the status of each resources attached to the cloud.
2. **Resource Selection:**
   Based on information regarding the current queued jobs and information on the status of cloud resources, the cloud scheduler makes decisions regarding the creation or deletion of specific cloud nodes (VMs) in order to best suit the set of jobs waiting to run.
3. **Task Submission:**
   Finally the job is submitted to best available resource selected.

III. CRITERIA FOR SCHEDULING IN CLOUD COMPUTING

- Resource utilization: Resource should be utilized in a way that increases throughput. Resources should not remain idle for long time.
- Throughput: The number of processes completing in a unit of time.
- Turnaround time: The length of time it takes to run a process from initialization to termination, including all the waiting time.
- Waiting time: The total amount of time that a process is in the ready queue.
- Response time: The time between when a process is ready to run and its next I/O request.
- Fault tolerance: The algorithm should continue to work properly even though if any node fails.

IV. LITERATURE SURVEY

A. First Come First Serve (FCFS):

It is the simplest Scheduling algorithm. Processes are dispatched according to their arrival time on the ready queue. FCFS is a non preemptive discipline. The shortest job at the back of the queue have to wait for the large task to finish. Resource utilization is low and response time is more [3].

B. Backfilling Algorithm:

It improve FCFS by increasing the utilization of the system resources and by decreasing the average waiting time of the job in the queue of the scheduler. If the job at the head of the queue is waiting for resources then it is possible for other short jobs to be scheduled and executed. Maximum execution time for each job must be known for backfilling algorithm. It move short jobs to compute first to improve response time and node utilization [4] [7].

C. Conservative Backfilling:

In this backfilling, it allows each job to reserve the resources it needs, when it is inserted into the job queue. If enough nodes are available then the job will be executed. Otherwise the job has to wait and later arriving small job will be executed if enough node for that job are available. It can predict when each job will execute. Starvation problem will not occur because reservation is already made for every job when it is submitted for execution [4] [7].

D. EASY Backfilling:

It is developed for the IBM SP2 super computer, only the first job in the submission queue is allow to reserve the resources it needs. This approach is more aggressive, because it increases resource utilization even if it does not guarantee that a job is not delayed by another one submitted later [5].

E. Flexible Backfilling:

In this approach, jobs are prioritized. Each job waiting in the queue is given a slack factor for the computation of which the waiting time of job in the queue is considered(For example, if the slack factor is equal to three, means the jobs may wait three times the average waiting time). Shortest jobs have smaller slacks than other jobs [5].

F. Multiple Queue Backfilling:

Lawson and Amirin proposed this method, in which each job is given to a queue in accordance with its expected running time and each queue is given to the partition of the parallel system and only jobs of this queue can be run [5].

G. Gang Scheduling:

In this approach, the tasks that compose a job are grouped together and scheduled concurrently in a set of processors. At any given time the highest priority job is executed. When scheduled, a job can be preempted before completion. Preemption occurs only if a higher priority job is released. It is based on ousterhout matrix. In this matrix row represent time slices and column represent processors. The threads of a job are grouped into a row of matrix [6] [7].

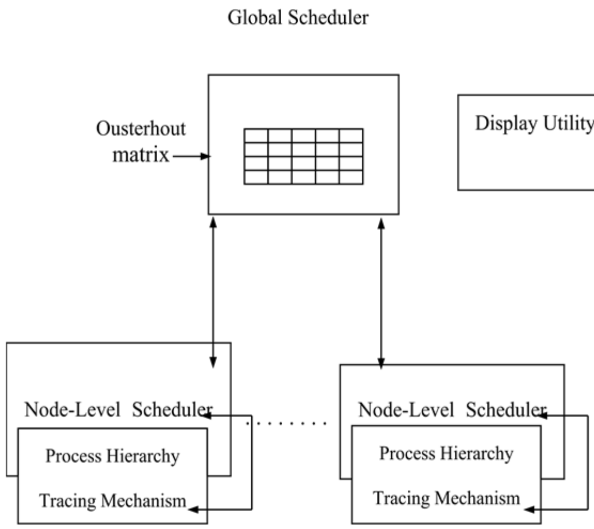

Fig.1 overview of Gang Scheduling system

H. Paired Gang Scheduling:

It improves the resource utilization well without causing interference between the processes of completing jobs. Because a process which occupies the CPU most of the time will be matched with a process that occupies an I/O devices. So they will not interfere with each other's work. The CPU and the I/O devices will not be idle while there are jobs which need to be executed [8].

I. Gang EDF Scheduling:

It applies the EDF (Earliest Deadline First) policy to the gang scheduling scheme. Jobs with earlier deadlines are assigned higher priorities [9].

Algorithm: Gang EDF scheduling policy

Q_{ready}: Set of ready tasks that are stored in order of early deadlines.

Q_{run}: Set of running tasks being scheduled

First_task(Q_{ready}): Function that returns the first task in Q_{ready}

$Q_{run} = Ø$;
While $Q_{ready}= Ø \ \wedge \Sigma r_i \in Q_{run}\ v_i \neq m$ do
r_i = first_task(Q_{ready});
if $v_i + \Sigma r_i \in Q_{run}\ v_i \leq m$ then
$Q_{run} = Q_{run} \cup \{ r_j \}$;
end if
$Q_{ready} = Q_{ready} \setminus \{ r_j \}$;
end while
execute all the tasks in $Q_{run;}$

J. Nephele Data Processing Framework:

In this approach, it includes Job Manager(JM) and Task Manager™. The job manager is responsible for receiving the clients job and scheduling them. It has the capability of communicating with the controller and allocate or deallocate virtual machines for the current job. The task manager™ is responsible for receiving tasks from job manager and execute them and later inform the job manager about their completion or errors occurred. This framework improve the overall utilization and reduce the communication cost [10].

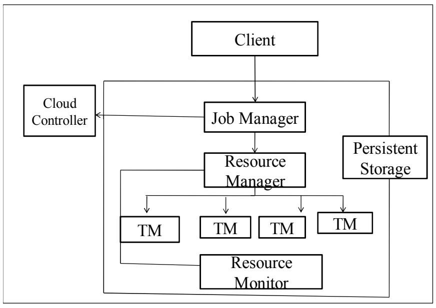

Fig.2 Nephele Data Framework

K. Conservative Migration Supported Backfilling (CMBF):
It is a backfilling based algorithm. CMBF schedules the job according to their arrival time when there is enough number of nodes. When the number of idle nodes is not sufficient for a job to run then the job with the smaller node number requirement may be executed via backfilling. The starvation problem is overcome in this algorithm. Virtual machines are parallelized into foreground and background. When foreground VM cannot utilize all the available resources the background virtual machine are allocated to utilize the available resources [11].

L. Aggressive Migration Supported Backfilling(AMBF):
It keeps track of backfilling jobs at the head of the queue and allows the head of queue job to preempt other jobs. It overcome the problem of CMBF where it keeps track of backfilling jobs for every job as it results high cost [11].

M. Conservative Migration and Consolidation Supported Backfilling (CMCBF):
It is based on the priority based consolidation method where the computing capacity of a node is divided into two tiers namely foreground and background. Foreground machine will have the high priority and background machine will have the low priority. In CMCBF, the threshold value is assigned for every virtual machine , it allows the job run in background virtual machine only when foreground virtual machine has lower utilization then the given threshold, when compared to CMBF, CMCBF provide better result in terms of resource utilization and response time [11].

N. Aggressive Migration and Consolidation Supported Backfilling (AMCBF):
In this approach, it uses the CPU usage information of parallel processes to make scheduling decisions. In AMCBF, other than the first job in the queue is delayed. It provides better result in terms of resource utilization [11].

O. Buddy Allocation Mechanism:
In this approach, the computing capacity of a data center node is partitioned into k-tiers (ie, both the foreground and background virtual machine). The background job does not affect the foreground job. Both the background virtual machine and the foreground virtual machine execute in parallel to improve the resource utilization and minimize the communication cost [12] [13].

V. COMPARISON OF EXISTING SCHEDULING ALGORITHMS

Following scheduling algorithms are used and implemented in parallel job scheduling systems are summarized in table.1

Table 1. Comparison of existing parallel scheduling algorithms

S.No	Scheduling Method	Scheduling Factors	Findings	Tools
1	FCFS	Submission time, start time, completion time	Resource utilization is low and response time is more	CTC SP2
2	Backfilling Algorithm	Jobs run time, prioritized the jobs in the queue	Shortest jobs are move ahead to improve response time and node utilization	Event Based Simulator
3	Conservative Backfilling Algorithm	Arrival time of job, reservation for all jobs, estimated time of execution for each job	It improves node utilization	Event Based Simulator
4	EASY Backfilling	Run time of a job	Requires each job to specify its maximum execution time	Trace driven simulation
5	Flexible Backfilling	Jobs estimated execution time, Jobs deadline time, number of resources	Shortest job will be executed first in the queue. Resource utilization is improved	Event driven adhoc simulator
6	Multiple Queue Backfilling	Job priorities, job reservation, job submission time, estimated execution time	It reduces cost of reservation and on demand	CTC, KTH, SDSC-PAR(1996) and SDSC-SP2
7	Gang Scheduling	Average slow down, average wait time, average loss of capacity	Improves job wait time and maximized system utilization	Discrete event simulation

S.No	Scheduling Method	Scheduling Factors	Findings	Tools
8	Paired Gang Scheduling	Time slice, number of processors used by job	Improves resource utilization well without causing interference between the processes of completing jobs	ParPar cluster
9	Gang EDF scheduling	Jobs with earlier deadlines assign high priority	The earliest deadline jobs can execute all the threads simultane-ously at any time instants	Global EDF schedulability test
10	Nephele data processing framework	Job manager and Task manager	Improves the overall utilization and reduces communication cost	OMNET++ version 3.2 simulator
11	CMBF	Foreground VM and Background VM	Requires keeping track of backfilling jobs for all jobs in the queue	Trace Driven Simulation
12	AMBF	Foreground VM and Background VM	Keeps track of backfilling jobs for the job at the head of the queue	Trace Driven Simulation
13	CMCBF	Foreground VM and Background VM	It consider threshold value for each virtual machine. Improves node utilization	Trace Driven Simulation
14	AMCBF	Foreground VM and Background VM	Allows job at the head of the queue to preempt other jobs and consider threshold VM. Does not consider communication cost.	Trace Driven Simulation
15	Buddy Allocation Mechanism	Partitioning of Foreground and background VM	Improves the overall resource utilization and reduces the communication cost	Cloudsim

VI. CONCLUSIONS

The Buddy Allocation Mechanism produces best performance among the existing scheduling methods. The AMCBF and CMCBF does not consider the communication cost, because the processes of a job may need to be allocated to nodes that are close to each other. FCFS produces high response time compared to EASY backfilling method. Nephele framework assign specific virtual machine to a specific task. Even though the priority based consolidation methods improves the resources utilization compared to other algorithms, the drawback in CMCBF and AMCBF is overcome by Buddy Allocation Mechanism.

REFERENCE

[1]. Michael Armbrust, Armando Fox, Rean Griffith, Anthony D.Joseph, Randy Katz, Andy Knowinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharai.2010. "A View of cloud computing" Ana. Commun. ACM 53, 4(April 2010),50-58.

[2]. Rao.S, Rao.N& Kumari, K2009, "Cloud Computing: An Overview", Journal of Theoretical and Applied Information Technology, Vol.9, No.1, pp.71-76.

[3]. U.Schwiegelshohn abd R.Yahyapour," lysis of First- come-First-serve parallel job scheduling", Proc.Ninthe Ann.ACM-SIAM Symp Discrete Algorithms, pp.629-638, 1998.

[4]. Alexander M.Lindsay, Maxwell Galloway-Carson, Christopher R. Johnson, David P. Bunde, and Vitus J. Leung," Backfilling with Guarantees Granted upon Job Submission" 17th European conference on Parallel and Distributed Computing (Europar'11) *Part 1,* volume 6852 of LNCS, pages 142-153, 2011.

[5]. Jackson David, Quinn Snell, and Mark Clement." Core Algorithms of the Maui Scheduler." In job scheduling strategies for parallel processing, pp.87-102.Springer Berlin Heidelberg 2001.

[6]. D.Feitelson and M.Jettee, " Improved Utilization and Responsiveness with Gang Scheduling" , Proc. Workshop Job Scheduling Strategies for Parallel Processing, pp. 238-261, 1997.

[7]. Zhang Yanyong, Hubertus Franke, Jose Moreira, and Anand Sivasubramaniam," An Integrated Approach to parallel scheduling using gang scheduling, backfilling, and migration," IEEE Transaction on parallel and Distributed Systems, vol.14,no.3(2003),pp.236-247.

[8]. Y.Wiseman and D.Feitelson," Paired Gang Scheduling," IEEE Trans. Parallel and Distributed Systems, vol.14, no.6, pp. 581-592, june 2003.

[9]. S.Kato and Ishikawa," Gang EDF Scheduling of Parallel Task Systems",Real Time System Sym. 2009, 30th IEEE.

[10]. Abhijit and Adhikari," Efficient Dynamic Resource Allocation in Parallel Data Processing for Cloud using PACT", International Journal of Advanced Research in Computer and Software Engineering, vol.3, Issue.11, 2013.

[11]. Xiaocheng Liu, Chen Wang, Bing Bing Zhou, Junliang Chen, Ting Yang, Zomaya. AY,"Priority- Based Consolidation of parallel workloads in the cloud," IEEE Transaction on parallel and Distributed Systems, vol.24, no.9, pp.1874-1883,sept 2013.

[12]. E.Von puttkamer," A simple Hardware Buddy System Memory Allocator," IEEE Trans. Computers, vol.C-100, no.10, pp.953-957,oct 1975.

[13]. K.Knowlton, "A Fast Storage Allocator", Comm. ACM, vol.8, no.10, pp.623-624,1965.