ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

A Survey on Task Scheduling For Parallel Workloads in the Cloud Computing System

Abhijeet Tikar [1], S.M.Jaybhaye [2], G.R.Pathak [3]

P.G. Student, Department of IT, Sinhgad College of Engineering, Pune, Maharashtra, India [1]

Assistant Professor, Department of IT, Sinhgad College of Engineering, Pune, Maharashtra, India [2]

Associate Professor, Department of IT, Sinhgad College of Engineering, Pune, Maharashtra, India [3]

ABSTRACT: Cloud computing is a computing paradigm where applications, data, memory, bandwidth and IT services are provided over the Internet. Cloud computing is based on pay per usage model. Cloud service providers provide virtual resources to the cloud users. The ultimate goal of cloud service providers is to gain maximum profit and use resources efficiently. Scheduling refers to a set of policies to control the order of work to be performed by a system. Task scheduling plays vital role in cloud computing system to manage heavy load or traffic. Efficient task scheduling improves resource utilization, response time and also meets user requirements. In this paper, Survey on various task scheduling methods for parallel workloads is made.

KEYWORDS: Cloud Computing, Backfilling, Migration, Gang scheduling, Task Scheduling

I. INTRODUCTION

The Word Cloud Computing is flourishing everywhere among enterprises, organization, independent software vendors, end users etc. Cloud computing is nothing but distributed computing over the internet. End users can access their data from the database in the cloud. Cloud computing varies from one cloud service provider to another, as some cloud service providers provide storage over network with small monthly rentals for end users, whereas some providers offers applications which helps in reducing costs in installations of applications as well as deployment .Cloud Computing has the ability to transform the large part of Information Technology (IT) industry for making software more attractive as a service. Software developers with new ideas for Internet services can directly deploy their code.

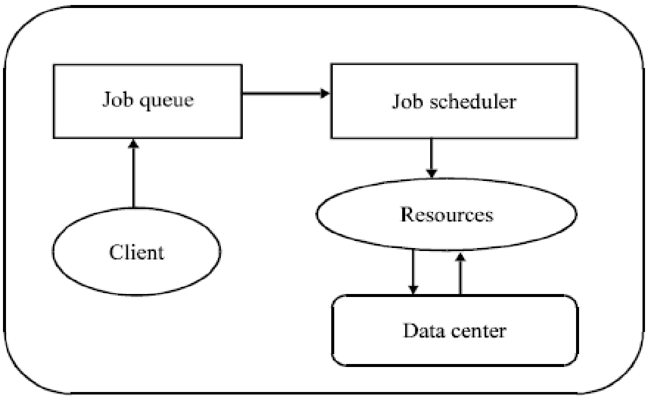

Fig. 1 Job scheduling in cloud computing system

In cloud computing system, scheduling is the method by which jobs are given access to system resources. This is usually done to load balance and share system resources effectively. Figure 1 shows basic structure of job scheduling in cloud computing system. A job queue is a data structure maintained by job scheduler containing jobs to run. Users

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

submit their jobs to the job queue for processing. A job scheduler is responsible for arranging jobs into an appropriate sequence and also responsible for allocating resources to the jobs by using scheduling algorithm.

1.1 Need for task scheduling

Cloud computing paradigm is attracting number of applications to run in data centers. End users are given access to a variety of large amount of data and software's to manage their work. Cloud is pay per usage model. Bill is generated based on amount of usage. User buys virtual resources on rent and pay for only what they use. The need for software and hardware resources has been increased rapidly. Cloud service providers do business by servicing the users. The goal of cloud service providers is to gain maximum profit and use resource efficiently. So, it is important to handle heavy traffic in cloud computing and task scheduling is the way to handle heavy traffic in cloud computing system. A good scheduling algorithm improves the node utilization, response time and throughput. A poor scheduling algorithm may result in bad consequences. For example cloud service providers may lose money and even go out of business.

1.2 Parameters for scheduling in cloud computing

1. **Resource utilization:** Using a resource in a way that increases throughput. Resources should not remain idle for long time.
2. **Response time:** The time of submission to the time the first response is produced. It should be as less as possible.
3. **Waiting time:** The sum of the periods spent waiting in the job queue.
4. **Throughput:** It is the total amount of work done in a unit of time.
5. **Turnaround time:** Turnaround time is the total time taken between the submission of a task for execution and the return of the complete output to the client.
6. **Fault tolerance:** The algorithm should continue to work properly despite of failure of nodes.
7. **Energy consumption:** Energy consumption is the amount of energy consumed in a process. Scheduling techniques must lower power consumption.

II. SCHEDULING TECHNIQUES

2.1 Static and Dynamic Scheduling:

In static scheduling, schedule is computed before executing task. For applications that have predictable and fixed demands, it is possible to use static scheduling effectively. With static scheduling, the customer contracts with the service provider for services and the provider prepares the required resources in advance of start of service. The customer is billed on a monthly basis.

In cases where demand by applications may change or vary, dynamic scheduling techniques have been suggested whereby VMs may be migrated on-the-fly to new compute nodes within the cloud. With dynamic scheduling, the service provider allocates more resources as they are needed and removes them when they are not needed. The end user is billed on a pay-per-use basis. In dynamic scheduling user cannot promise a deadline and cloud cannot plan ahead of resource usage.

2.2 Pre-emptive and non pre-emptive scheduling:

Pre-emptive scheduling allows a task to be interrupted in the middle point of its execution, taking the VM away and allocating it to another task. Priorities are considered while assigning tasks. Sometimes it is necessary to run certain task that has a higher priority before another task although it is executing. Therefore, the running task is pre-empted for some time and resumed later when the priority task has finished its execution. This type of scheduling is called as pre-emptive scheduling.

Non pre-emptive scheduling ensures that a task give away control of the VM only when it finishes its complete execution. In this type of scheduling running task is executed till completion.
Example-First come first serve algorithm (FCFS).

DOI: 10.15680/IJIRSET.2014.0310067

III. LITERATURE SURVEY

3.1 First Come First Serve (FCFS): The popular batch scheduling algorithm for parallel jobs is first come first serve (FCFS).In FCFS, each job has to specify number of nodes required .Processing of jobs by scheduler is done on the basis of arrival time. The main disadvantage of FCFS is that it is non pre-emptive. The shortest or small tasks which are at the back of the queue have to wait for the large task at the front to finish. Resource utilization is low and response time is more in FCFS. [4]

3.2 Backfilling algorithm: It is advancement to first come first serve algorithm. Backfilling tries to improve resource utilization while maintaining FCFS order. 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. The aim of backfilling algorithm is to move short jobs ahead to improve response time and node utilization. [5] [6]

3.2.1 Conservative Backfilling: Ahuva et al. [5] proposed conservative backfilling. In this backfilling, Scheduler schedules jobs according to the order of arrival time. If enough nodes are available then the job will be executed. Otherwise the job has to wait and later arriving and small job will be executed if enough nodes for that job are available. It has capability of predicting when each job will execute. Also there is no problem of starvation because reservation is already made for every job when it is submitted for execution. Conservative backfilling maintains two data structures. One is the list of queued jobs and times at which they will start execution. Second is for the profile of expected processor usage at future times.

3.2.2 EASY Backfilling: Ahuva et al. [5] proposed EASY which is aggressive version of backfilling. It allows only job at the head of queue to preempt other jobs. Any job can be backfilled in EASY backfilling if it does not delay the job at the head of queue. Jobs other than head of queue may be delayed. The disadvantage of this algorithm is that prediction cannot be made about how much each job will be delayed in the queue.

3.3 Gang Scheduling: Gang scheduling is an alternative to batch scheduling. It is a scheduling algorithm for parallel systems that schedules related processes or threads to run simultaneously on different processors. Groups of related threads scheduled as a unit, or gang. All members of gang run simultaneously on different processors. All gang members start and end their time slices together. The allocation of time slices is coordinated by operating system support.

	P1	P2	P3	P4	P5
Time Slice 0	J1	J1	J1	J1	J1
Time Slice 1	J2	J2	J2	J2	J2
Time Slice 2	J3	J3	J4	J4	J4

Fig. 2 Ousterhout matrix of gang scheduling

Gang scheduling is based on ousterhout matrix which is shown in figure 2. In ousterhout matrix, row represents time slices and column represents processors. The threads of a job are grouped into a row of matrix. Gang scheduling is useful for applications where performance of application degrades when any part of the application is not running. [7] [8]

3.4 Backfilling and Migration Gang Scheduling: Moreiraz et al. [10] proposed Backfilling and Migration Gang Scheduling. Backfilling gang scheduling is a combination of gang scheduling and backfilling. Each virtual machine created by gang scheduling is treated as a target for backfilling which gives better results than individual backfilling and gang scheduling methods. In migration gang scheduling, the process of migration includes moving a job to any time slice in which there are enough free processors to execute that job.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

3.5 Paired gang scheduling: Y. Wiseman et al. [9] proposed paired gang scheduling which overcomes the drawbacks of gang scheduling. Paired gang scheduling matches compute bound jobs with I/O bound jobs and schedule them together. The main purpose of matching these gangs is that these two gangs will not interfere in each other's work because they use two different devices.

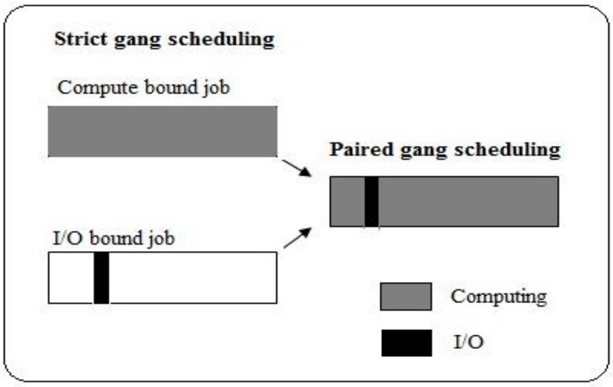

Fig. 3 Paired gang scheduling

Figure 3 shows the example of paired gang scheduling in which compute bound job and I/O bound job is paired together. Because of these pairing efficiency as well as resource utilization is improved.

3.6 Nephele Data Processing Framework: Warneke et al. [11] proposed Nephele data processing framework to exploit dynamic resource allocation for task scheduling as well as execution. Nephele architecture has the ability to assign specific virtual machine types to specific tasks of a processing job. Also it has ability to automatically allocate/de-allocate virtual machines in the course of a job execution. Nephele framework improves the overall resource utilization and reduces the processing cost.

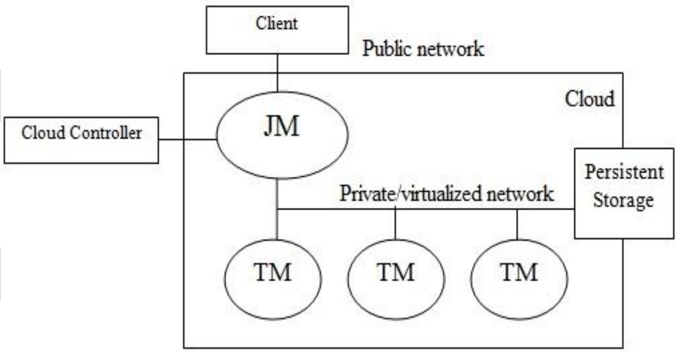

Fig. 4 Structural overview of nephele

Figure 4 shows structural overview of nephele framework. It includes Job Manager(JM) and Task Manager(TM).JM is responsible for receiving client requests and scheduling them.JM can allocate as well as de-allocate virtual machines by means of cloud controller.TM is responsible for receiving tasks for JM ,executing them and again informing JM whether task is properly executed or for possible errors.

3.7 A Novel Approach for Task Scheduling in Cloud: Vijayalakshmi et al. [13] proposed a novel approach for task scheduling in cloud. In this approach user tasks are prioritized according to priority. The task with highest priority is assigned to a Virtual Machine (VM) with highest processing power (MIPS).The key factor for task is priority and for virtual machine is processing power. Consider 5 tasks where priorities of tasks are given as 1,2,3,4 and 5. Also consider 5 Virtual Machines represented by their Id's and MIPS as

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

V = {{1, 250}, {2, 100}, {3, 2500}, {4, 500}, {5, 350}}.
Here 1st task will be assigned to VM-3 because of its highest processing power. Rest of the tasks 2,3,4,5 will be assigned to VM-4, VM-5, VM-1, and VM-2 respectively.

3.8 Conservative Migration supported BackFilling (CMBF): Xiaocheng Liu et al. [12] proposed Conservative Migration supported BackFilling (CMBF), which is backfill based algorithm. The CMBF algorithm assumes that the state of a job can be saved and can be later restored. So, the scheduler is able to suspend a job and resume it on other nodes in a later time. If there is enough number of nodes available then CMBF schedules jobs according to arrival time .When the number of idle nodes is not sufficient for a job then smaller node number requirement may be scheduled to execute via backfilling. CMBF also overcomes the drawback of starvation which normally occurs in backfilling algorithms. The drawback of CMBF algorithm is it cannot handle virtual machine resources effectively and also requires keeping track of backfilling jobs for every job in a queue.

3.9 Aggressive Migration Supported BackFilling (AMBF): Xiaocheng Liu et al. [12] proposed Aggressive Migration Supported BackFilling (AMBF).AMBF is an alternative to CMBF.CMBF requires keeping track of backfilling jobs for every job when making preemption decision which results in high cost. AMBF overcomes this drawback by only keeping track of job which is at head of queue. It allows only head of queue job to preempt other jobs. Meaning that rest of jobs is not allowed to preempt jobs but they are allowed to dispatch to idle nodes.

3.10 Priority based consolidation method
Priority based consolidation method divides the computing capacity of node into two tiers, foreground and background. Foreground having high priority and background having low priority. This method assumes that physical node can run at most two virtual machines, one in foreground and second in background. Following are two priority-based methods to consolidate Parallel workloads in the cloud computing system.

3.10.1 Conservative Migration and Consolidation supported BackFilling (CMCBF): Xiaocheng Liu et al. [12] proposed CMCBF, which overcomes the drawback of CMBF algorithm. It ensures that background workload does not affect foreground job. There is a threshold for every virtual machine, CMCBF allows job to run in background only when foreground virtual machine has utilization lower than given threshold. The utilization of foreground virtual machine can be obtained from runtime monitoring data or from profile of jobs. This method is better than CMBF in terms of resource utilization and response time.

3.10.2 Aggressive Migration and Consolidation supported BackFilling (AMCBF): CMBF and CMCBF has similar problem of keeping track of backfilling jobs for each job in a queue. To reduce cost new modified algorithm is Aggressive Migration and Consolidation supported BackFilling (AMCBF).In this method only job at the head of queue is allowed to preempt other jobs. [12]

IV. MERITS AND DEMERITS OF SURVEYED SCHEDULING METHODS

Merits and Demerits of surveyed scheduling methods are given in Table 1.FCFS has higher response time and waiting time compared to other scheduling methods. Both conservative and EASY backfilling requires maximum execution time of each job to be already specified. Paired gang scheduling is an advancement to gang scheduling. A novel approach for task scheduling in cloud [13] considers processing power but does not support migration. AMBF & AMCBF are advancements to CMBF and CMCBF respectively.

Sr/No	Scheduling Method	Merits	Demerits
1	FCFS [4]	Simple to understand & implement.	Response time is more, Low resource utilization.
2	Conservative backfilling [5]	Capability of predicting when each job will execute.	Requires each job to specify its maximum execution time.
3	EASY backfilling [5]	Allows only job at head of queue to preempt other jobs.	

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

4	Gang Scheduling [7] [8]	Schedules related processes to run simultaneously on different nodes.	Processors remain idle when processes perform I/O operations.
5	Backfilling & Migration gang scheduling [10]	Backfilling gang scheduling treats each VM as target for backfilling. Migration gang scheduling has the ability to move the job to another time slice.	Waiting time of backfilling and migration gang scheduling is poor.
6	Paired gang scheduling [9]	Pairs compute bound jobs with I/O bound jobs.	Prediction is difficult which is required while pairing jobs.
7	Nephele data processing framework [11]	Ability to assign specific VM to specific task.	Does not handle resource overload or under utilization.
8	A novel approach for task scheduling in cloud [13]	Prioritize tasks based on their priorities and VM based on their processing power.	Higher waiting time and does not support migration.
9	CMBF [12]	Capability to suspend a job and later resume it on another node.	Requires keeping track of backfilling jobs for every job in a queue.
10	AMBF [12]	It only keeps track of backfilling jobs for the job at head of queue.	Sometimes background work affects foreground work.
11	CMCBF [12]	Considers threshold for each virtual machine.	Does not consider communication cost, Which is important in case of large data centers where job to be allocated to the node will not close to each other.
12	AMCBF [12]	Allows job at head of queue to preempt other jobs and also considers threshold for VM.	

Table 1.Merits & demerits of surveyed scheduling methods

V. CONCLUSIONS

AMCBF produces best performance among surveyed scheduling methods. EASY backfilling overcomes the drawback of conservative backfilling and FCFS. Nephele framework has the ability to assign specific virtual machine to specific task. CMBF and AMBF outperform FCFS but produces high response time compared to EASY. Priority based consolidation method partitions the computing capacity of a node into two, foreground and background. Also priority based consolidation methods lead to better resource utilization compared to other surveyed methods. The drawback of CMBF and CMCBF is also overcome by AMCBF method.

REFERENCES

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

[2] Bento, Al, 2011, "Cloud Computing: A New Phase in Information Technology Management", Journal of Information Technology Management, Vol. 22, No. 1, pp. 39-46.

[3] Rao, S, Rao, N &Kumari, K 2009, "Cloud Computing: An Overview", Journal OF Theoretical And Applied Information Technology, Vol 9, No. 1, pp. 71-76.

[4] U. Schwiegelshohn and R. Yahyapour, "Analysis of First-Come-First-Serve Parallel Job Scheduling," Proc. Ninth Ann. ACM-SIAM Symp. Discrete Algorithms, pp. 629-638, 1998.

[5] Mu'alem, Ahuva W., and Dror G. Feitelson. "Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling." Parallel and Distributed Systems, IEEE Transactions on 12.6 (2001): 529-543

[6] 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.

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

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

[8] Jonathan Weinberg, "job scheduling on parallel systems", University of California, San Diego, 2001.

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

[10] Zhang, Yanyong, Hubertus Franke, Jose Moreira, and Anand Sivasubramaniam. "An integrated approach to parallel scheduling using gang-scheduling, backfilling, and migration." Parallel and Distributed Systems, IEEE Transactions on 14, no. 3 (2003): 236-247.

[11] Warneke,D.; Odej Kao, "Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud," Parallel and Distributed Systems, IEEE Transactions on , vol.22, no.6, pp.985,997, June 2011

[12] Xiaocheng Liu; Chen Wang; Bing Bing Zhou; Junliang Chen; Ting Yang; Zomaya, AY., "Priority-Based Consolidation of Parallel Workloads in the Cloud," Parallel and Distributed Systems, IEEE Transactions on , vol.24, no.9, pp.1874,1883, Sept. 2013

[13] Vijayalakshmi, R.; Prathibha, S., "A novel approach for task scheduling in cloud," Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference on , vol., no., pp.1,5, 4-6 July 2013