ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2 Issue Special 1 Jan-March 2014
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

Improve the Utilization and Responsiveness for Parallel Workloads Based on Priority Consolidation in the Cloud

[I]Sutharsiya J, [II]Sundar Rajan R
[I,II]Dept. of IT, Kalasalingam university, Krishnankoil, India

Abstract
Effectively to run the parallel jobs using virtualization technologies. A parallel job often requires a certain number of nodes to run, to schedule parallel job in datacenters. A set of nodes is likely to be fragmented by parallel jobs with different node number requirement. An important consideration in this environment is to improve response time, throughput and utilization. In Existing approach, each nodes computing capacity is partition into two tier using priority based consolidation. The foreground VM tier runs higher priority jobs and Background VM tier runs low priority jobs. This leads to low system utilization and large waiting time. To overcome this, the proposed approach is to effectively partition the datacenter node into 3-tiers in which the first 2-tiers having same priority when compare to background VM tier by using this technique improved the node utilization and responsiveness.

Keywords
Parallel Job Scheduling, Backfilling, Online Scheduling, Virtual machine, Computing capacity

I. Introduction

The cloud makes it possible for you to access your information from anywhere at any time. Cloud computing is internet-based computing whereby shared resources, software and information are provided to computers and other devices on demand, although it has a few Important Parameters such as delivery over the web Scalability, elasticity and multitenancy.

Cloud computing providers offer their services according to several fundamental models: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). SaaS is software that is owned, delivered and managed remotely by one or more providers and that is offered in a pay-per-Use manner. PaaS supplies all the resources required to build applications and services completely from the Internet, without having to download or install software. Infrastructure-as-a-Service (Iaas): Infrastructure as a service delivers a platform virtualization outsourced service.

Virtualization is the ability to run multiple operating systems on a single physical system and share the underlying hardware resources.

A. Scheduling

An essential requirement in cloud computing environment is scheduling the current jobs to be executed with the certain constraints.

The purpose of scheduling is to increase the utilization of resources. Scheduling is used to allocate particular resources for a certain tasks in particular time. Job scheduling in cloud computing has attracted great attention.

Most research in job scheduling adopt a paradigm in which a job in cloud computing system is characterized by its workload, dead-line and the corresponding utility obtained by its completion before deadline, which are factors considered in devising an effective scheduling algorithm. Job scheduling problem is a core and challenging issue in cloud computing.

B. Scheduling Jobs Based on Parallelizability

In this model, jobs have degrees of parallelizability, the scheduler to schedule the jobs to multiple processors in simultaneously.

C. Scheduling Criteria

CPU utilization __ keep the CPU as busy as possible.
Throughput __ number of processes that complete their execution per time unit.
Turnaround time __ amount of time to execute a particular process.
Waiting time __ amount of time a process has been waiting in the ready queue.
Response time __ amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment).

The reminder of this paper is required as follows. Section II Related work, Section III Problem statement after that, Proposed System in Section IV .Finally, the conclusion is given in Section VII.

II. Related Work

The most basic batch scheduling algorithm is First-Come-First-Serve (FCFS). FCFS is that the simplest CPU scheduling algorithm [2] .In this, the processes are unit assign the CPU within the order the request it. The relative importance of the jobs measured solely by the point in time. The average waiting time below the FCFS policy is usually quit long.

A different approach to CPU scheduling is that the shortest-job-first (SJF) scheduling algorithm [3]. Jobs with the shortest interval area unit scheduled initial.

Backfilling [4] is an optimization that tries to balance the goals of utilization and maintain FCFS order. It needs that every job also specifies its most execution time. While the job at the head of the queue is waiting, it is possible for there, smaller jobs, to be scheduled, particularly if they might not delay the beginning of the job on the head of the queue. Processors get to be used that may otherwise stay idle.

Min–Min algorithm [5] chooses little tasks to be executed firstly, which in turn large task delays for long time. Max – Min algorithm [6] chooses massive tasks to be executed firstly that successfully little task delay for long time.

Another CPU scheduling algorithm is named gang scheduling [7], slices the time in a very round robin manner Gang scheduling permits processes within the same job to run at identical time. This results in higher performance for compute-bound human activity processes.

Effective scheduling [8] ways to enhance response times, throughput, and utilization are an important consideration in massive supercomputing environments. Parallel machines in

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

these environments have historically used space-sharing ways to accommodate multiple jobs at identical time by dedicating the nodes to a one job until it completes. This approach, however, can result in low system utilization and enormous job wait times. This paper discusses 3 techniques which will be used beyond simple space sharing to enhance the performance of huge parallel systems.

III. Problem Statement

A parallel job often desires certain number of nodes to run. Job specifies the number of nodes needed and the scheduler processes jobs according to the order of their arrival.

A remainder of nodes is probably to be disjointed by parallel jobs with completely different node number requirement. We tend to partition the computing capacity of a physical node into two tiers that's foreground and background. The foreground VM tier runs on high processor priority and therefore the background VM tier runs on low processor priority.

• Node utilization might even be low for high efficiency parallel workloads using two tier data centre.

• If the number of available nodes cannot satisfy the necessity requirement of an incoming job.

• A process in a parallel job may frequently wait for the data from alternative processes throughout waiting; the utilization of the node is low.

In Existing system used Conservative Migration supported BackFilling (CMBF) is backfill primarily based. Aggressive Migration supported Backfilling (AMBF) Scheduling Algorithm.

AMBF solely choose backfilling jobs for the work at the head of the queue and allows the head of-queue job to preempt alternative jobs. The rest of jobs within the queue are not allowed to preempt jobs, in another word, they can only be dispatched to idle nodes.

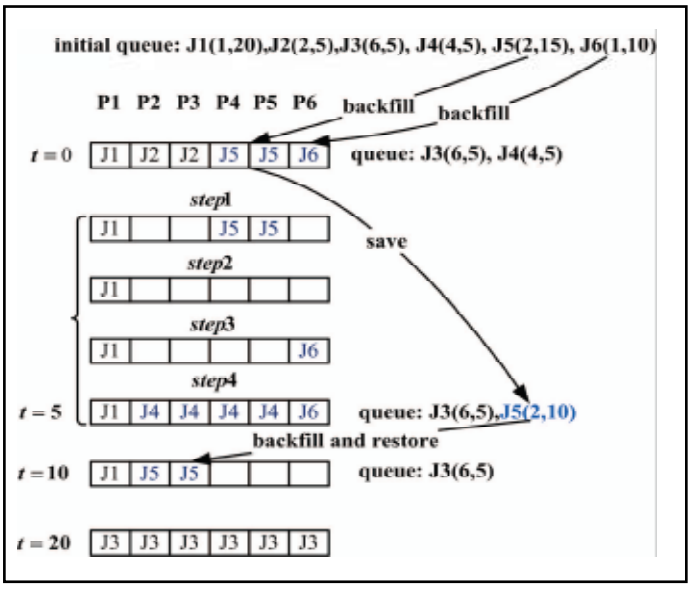

Fig. 1: Job Scheduling in AMBF

In general, the schedulers process jobs in order of priority, which is determined based on job attribute such as job class and time in queue , but also employ backfilling operations, [8] i.e., run jobs out of order, to make better use of the available resources. Its attempt to improve the utilization.

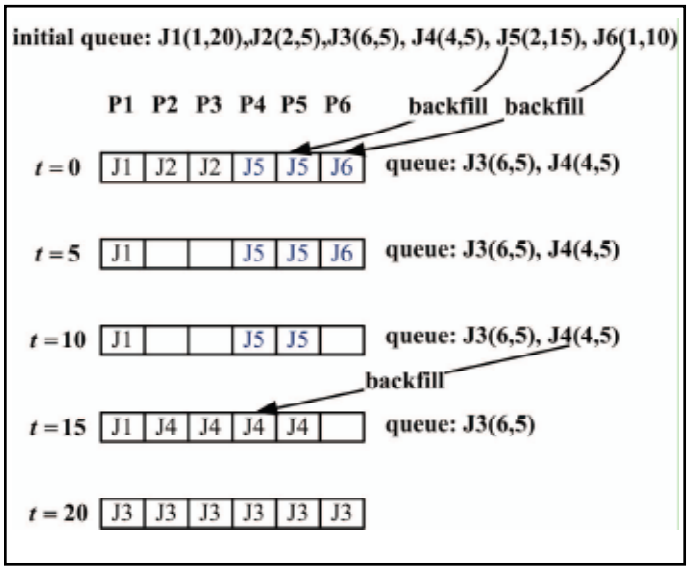

Fig. 2: Job Scheduling in CMBF

CMBF schedules jobs to run according to their arrival time when there is enough number of nodes. When the number of idle nodes is not sufficient for a job, another job with a later arrival time but smaller node number requirement may be scheduled to run Utilization of the node is low for using the AMBF and CMBF algorithm, it is difficult to achieve.

IV. Proposed System

In Existing system, used 2-tier known as foreground and background. The foreground jobs are filled after that jobs are filled in background via backfilling method. We proposed that, the primary two tiers offer a similar priority, the primary two tiers are filled jobs after that jobs are run on the 3-tier based on On-line Scheduling algorithm.

The proposed approach is to effectively partition the datacenter node into 3-tiers in which the first 2-tiers having same priority when compare to background VM tier by using this technique. The first 2-tier using priority based scheduling algorithm, For improved the responsiveness and the utilization additionally add one more tier compare to the existing system. The 3-tier using online scheduling algorithm. Using extensive simulations based on detailed models of parallel workloads, the benefits of combining the various techniques are shown the best performance and highly utilized the node.

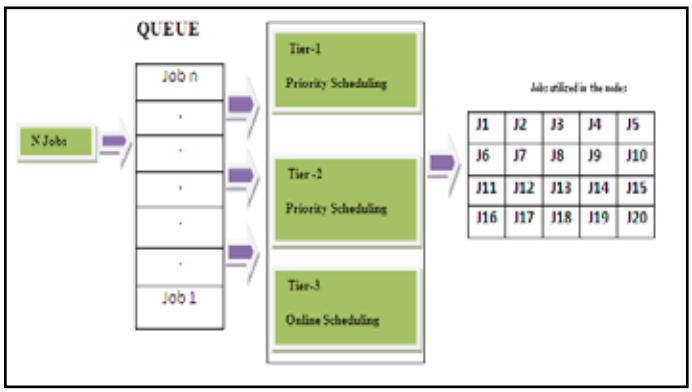

Fig. 3: Proposed Architecture

V. Online Scheduling Algorithm

When a new job comes, it is first inserted at the head of the ready

queue, assuming its expected starting time would be the expected finishing time of the current active task.
Based on this starting time we than can compare its expected utility with the rest of the jobs in the queue.
If it's expected utility is less than that of one following it, we re-insert this job. This procedure continues until the entire queue becomes a listed ordered by their expected utilities. We then recalculated the expected utilities for the jobs behind.
Online scheduling reduces the queuing time for each incoming task by quickly dispatching them to various virtual machines.

Algorithm 1: ON-LINE SCHEDULING

1: Input: Let f J1, J2...Jk be the accepted jobs in the ready queue ordered in non-increasing order of their expected accrued utility, and let ri i = 1… k represent their specific arrival times. Let current time be t and Let T0 be the task currently being executed.
2: Let the expected utility threshold be α
3: if t = the critical time of T_0 then
4: Abort the execution of T_0;
5: Choose J1 from the ready queue and start its execution;
6: Re-calculate the expected accrued utility for each of the tasks in the ready queue;
7: Remove the jobs with expected accrued Utility smaller than α;
8: end if
9: if T0 is completed then
10: Choose T1 from the ready queue and start its execution;
11: Re-calculate the expected accrued utility for each of the tasks in the ready queue;
12: Remove the tasks with expected accrued utility smaller than α;
13: end if
14: if a new job, i.e. J j arrives then
15: insert it into the ready queue;
16: Calculate the expected accrued utility of J_j and also calculate the expected accrued utilities of tasks in the ready queue;
17: Arrange the ready queue according to their expected
accrued utility. Following this sequence, calculate the expected accrued utility for each job in the ready queue;
18: for each job in the ready queue
19: Remove the job with expected accrued utility Smaller than
20: end for
21: end if

We assume that jobs are inserted to the ready queue. Jobs are scheduled to the node according the node requirement of the job and their execution time. As shown in algorithm 1, to allocated the job depends upon the utility.

VI. Experimental Results

A datacenter that hosts HPC parallel applications in addition to normal workload, the utilization of the node is low.

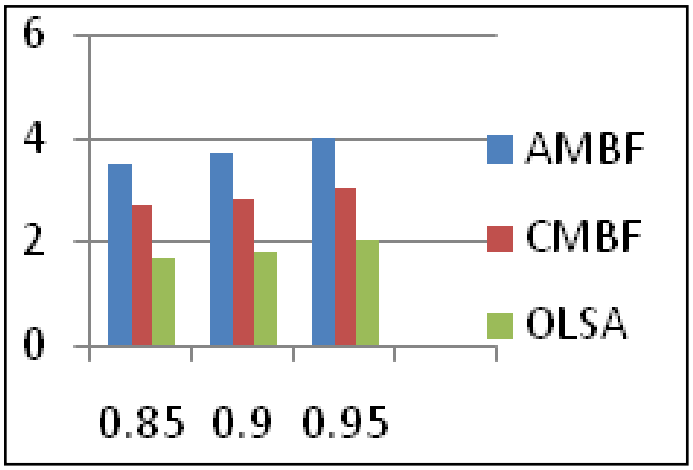

Fig. 4: Node Utilization

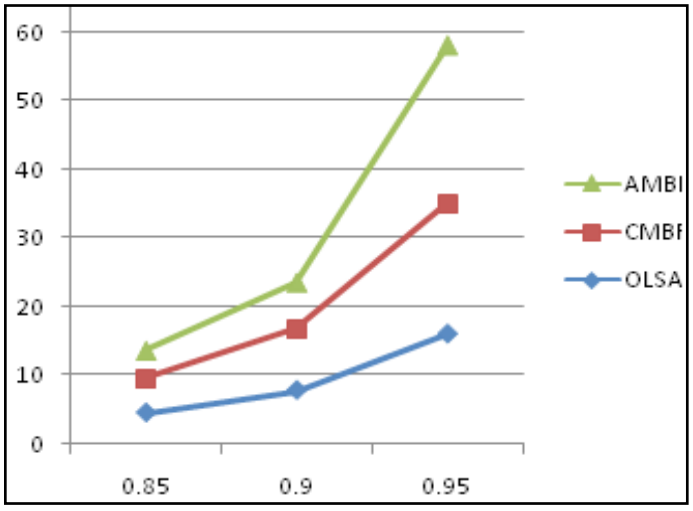

Fig. 5: Average Response Time

Our extensive experiments on well-known workloads show that our algorithm takes the very good utilization. For two common parallel job scheduling objectives, our algorithm produces an up to 41.5% and an average of 24.2% improvement on the average response time .The node utilization is high compare to the existing system. Two consolidation algorithm leads to low utilization .The proposed algorithm show the better results.

VII. Conclusion

An increasing number of high performance computing parallel applications leverages the power of the cloud for parallel processing. How to schedule the parallel applications to improved the node utilization is the key to the successful host of parallel applications in the cloud. The first technique we analyze is AMBF, the second is CMBF, and the third is online scheduling . The main contribution of this paper is an analysis of the effects of combining the above techniques.

References

[1] *Xiaocheng Liu, Chen Wang, "Priority-based Consolidation of Parallel Workloads in the Cloud" Fellow, IEEE vol,24 no 9, september 2013.*
[2] *U. Schwiegelshohn and R. Yahyapour, "Analysis of first-come first- serve parallel job scheduling," in Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2002, pp.*

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

629–638.

[3] D.Tsafrir, Y.Etsion, D.Feitelson,"Backfilling Using System-Generated Predictions Rather than User Runtime Estimates", IEEE Transactions on Parallel and Distributed Systems, vol.18, n. 6, pp. 789-803, June 2006.

[4] Loukopoulos T., Lampsas P., Sigalas P., "Improved Genetic Algorithms and List Scheduling Techniques for Independent Task Scheduling in Distributed Systems", IEEE, Eightth Internat ional Conference on Parallel and Distributed Computing, Applicat ions and Technologies, Adelaide, SA, pp. 67-74.

[5] Y. Wiseman and D. Feitelson, "Paired gang scheduling," Parallel and Distributed Systems, IEEE Transactions on, vol. 14, no. 6, pp.581 – 592, june 2003.

[6] D. Feitelson and B. Nitzberg, "Job characteristics of a production parallel scientific workload on the nasa ames ipsc/860," in Job Scheduling Strategies for Parallel Processing. Springer, 2005, pp60.

[7] R. Fujimoto, A. Malik, and A. Park, "Parallel and distributed simulation in the cloud," Simulation Magazine, Society for Modeling and Simulation, no. 3, 2010.

[8] Y. Zhang, H. Franke, J. Moreira, and A. Sivasubramaniam, "An integrated approach to parallel scheduling using gang-scheduling, backfilling, and migration" IEEE Transactions on Parallel and Distributed Systems, pp. 236–247, 2003.

[9] G. D'Angelo, "Parallel and distributed simulation from many cores to the public cloud," in Proceedings of the International Conference on High Performance Computing and Simulation (HPCS '11), pp. 14–23, IEEE, Istanbul, Turkey, 2011.

[10] X. Wang, E. Perlman, R. Burns, T. Malik, T. Budavari, C. Meneveau, and A. Szalay. Jaws:" Job-aware workload scheduling for the exploration of turbulence simulations". In SC '10.

[11] A. Do, J. Chen, C. Wang, Y. Lee, A. Zomaya, and B. Zhou, "Profiling applications for virtual machine placement in clouds," in Proceedings of IEEE International Conference on Cloud Computing (CLOUD '11), pp. 660–667, Washington, DC, USA, July 2011.

[12] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," Computer, vol. 40, no. 12, pp. 33–37, 2007.

[13] U. Schwiegelshohn and R. Yahyapour, "Fairness in parallel job scheduling," Journal of Scheduling, vol. 3, no. 5, pp. 297–320, 2000.

[14] Amazon Inc., "High performance computing (HPC) on AWS," http://aws.amazon.com/hpc-applications/, 2011.

[15] A. Mu'alem and D. Feitelson, "Utilization, predictability, workloads,and user runtime estimates in scheduling the ibm sp2 withbackfilling," Parallel and Distributed Systems, IEEE Transactions on, vol. 12, no. 6, pp. 529–543, 2001.

Authors Profile

J.Sutharsiya was born in virudhunagar, Tamil nadu, India on June 12, 1991 .She received the B.Tech degree in Information Technology from kalasalingam university krishnankoil , virudhunagar, Tamil nadu , india in 2012.She is pursuing M.Tech in Information Technology in Kalasalingam University, Krishnan koil, virudhunagar, Tamil nadu, India. Her research interest in Cloud computing.

R.SundarRajan received the B.Sc degree in computer science from Rajapalayam, Rajapalayam Raju's college, Tamil nadu,India and the M.Sc degree in computer science and Information Technology from Rajapalayam, Rajapalayam Raju's college, Tamil nadu,India and the M.E degree in computer science and Engineering from arulmigu kalasalingam college of engineering,Tamil nadu,India. He is pursuing Ph.D in Kalasalingam University, Krishnan koil,Tamil nadu,India.He was published many technical papers in international conferences. Currently,he is an assistant professor, in Information Technology at Kalasalingam University,Krishnankoil,Tamil nadu,India.He is a member of ISTE (Indian Society for Technical Education).He is currently engaged in research in cloud computing