ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.8

Task Scheduling using Improved Task Migration Consolidation in Cloud Computing

Prof. Navdeep Saluja[1], Prof. Sarvesh Rai[2], Radha Devi Nayak[3]
HOD[1], Assistant Professor[2], PG Scholar[3]
Department of CSE
IMEC, Sagar, India

Abstract:
Cloud computing is an inventive and revolutionary field in distributed system. It allows innovation in real time environment which encourages pay per portrayal according to client necessity. Cloud is a gathering of virtual strategy which incorporates both computational and storage scope. The fundamental target of distributed computing is to supply productive access to inaccessible and geologically conveyed assets. Cloud is creating step by step and faces an immense agreement of difficulties, one of them is planning. Scheduling alludes to an accumulation of approaches to deal with the request of work to be performed by a computer system. Scheduler adjusts its planning scenario of jobs as indicated by the changing circumstance and the sort of undertaking. The exhort strategy Improved Task Migration Consolidation Scheduling calculation for efficient execution of assignment and appraisal with FCFS and Minimum Completion Time Scheduling. This approach use the consolidation with backfilling, in which when task take long time then it remove from queue list and reenter in background machine through backfilling. The methodology implements on CloudSim 3.0.1 toolbox, which is configure in NetBeans 8.1. The outcome demonstrates that it gives improved execution contrasted with supplementary predictable scheduling algorithm. Resource utilization rate is improved by 0.52% and 11.45% as compare than FCFS and MCT respectively.

Keywords: Cloud computing, Distributed Computing, Virtual Machine, FCFS, Minimum Completion Time, Generalized Priority Scheduling, CloudSim.

1. INTRODUCTION

Cloud computing is one of the latest technology that is very popular now a days in IT industries as well as in R&D. This cloud computing technology is a model of development that comes after the introduction of distributed computing. As compare the cloud computing with the distributed computing in this there is a multilevel virtualization.

The whole work that is related to cloud computing works in a virtual environment. To get the advantages of cloud user needs to only connect to the internet and after that user can easily use the powerful computing and capacity of storage. Cloud computing services provided by CSP (cloud service provider) as per user requirements. In order to fulfill the demand of different users, they provide different quality of services.

In order to conclude the term cloud is an executable environment having dynamic behavior of resources as well as users providing multiple services. Scheduling is the one of the most prominent activities that executes in the cloud computing environment.

To increase the efficiency of the work load of cloud computing, scheduling is one of the tasks performed to get maximum profit. The main objective of the scheduling algorithms in cloud environment is to utilize the resources properly while managing the load between the resources so that to get the minimum execution time.

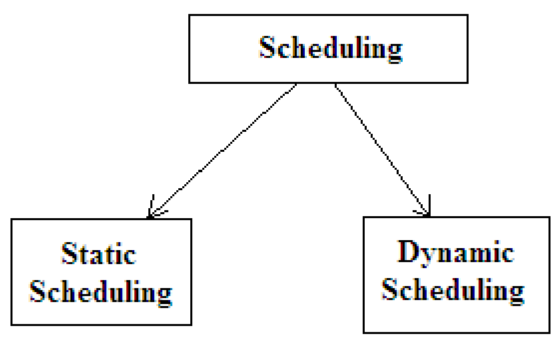

Figure.1. Types of Scheduling

Cloud computing has recently received considerable attention, as a promising approach for delivering Info and Communication Technologies (ICT) services as a utility. In the mechanism of providing these services it is necessary to improve the utilization of datacenter resources which are operating in most dynamic workload environments. Datacenters are the essential parts of cloud computing. In a single datacenter generally hundreds and thousands of virtual servers run at any instance of time, hosting many tasks and at the same time the cloud system keeps receiving the batches of task requests. During this context, one has to notice few target servers out many powered on servers, which can fulfill a batch of incoming tasks. So Task scheduling is a valuable issue which is greatly influences the performance of cloud service provider. Traditional approach that are used in optimization are deterministic, fast, and give perfect answers but often tends to get stuck on local optima. Complexity of the task scheduling problem belongs to Non Polynomial -complete involving extremely large search space with correspondingly large number of potential solutions and takes much longer time to find the optimal answer. There is no readymade and well outlined methodology to solve the problems under such circumstances. However in cloud, it is tolerable to find near best solution, preferably in a short period of time. In this framework IT practioners are focusing on heuristic methods. The cloud computing paradigm promises a cost-effective solution for running business applications through the use of virtualization technologies, highly scalable distributed computing, and data management techniques as well as a pay-as-you-go pricing model. In recent years, it also offers high-performance computing capacity for applications to solve complex problems [1]. Improving resource utilization is essential for achieving cost effectiveness. Low utilization has long been an issue in data centers. Servers in a typical data center are operated at 10 to 50 percent of their maximum utilization level [2]. 10 to 20 percent utilization is common in data centers [3]. For a data center, or a subset of servers in a data center that mainly handles applications with high-performance computing needs and runs parallel jobs most of the time, the problem can be significant.

2. RELATED WORK

Ting et al [1], Cloud Computing refers to constructed data center or "super computer" by virtualization technology and provides computing and storage resources, as well as the application container environment of software running, to software developers in a manner of free or hiring.

Liu et al [2], propose priority-based method to consolidate parallel workloads in the cloud. We leverage virtualization technologies to partition the computing capacity of each node into two tiers, the foreground virtual machine (VM) tier (with high CPU priority) and the background VM tier (with low CPU priority).

Lugun et al [3], analysis the differentiated QoS requirements of Cloud computing resources users' jobs, we build the corresponding non-preemptive priority M/G/1 queuing model for the jobs.

Hsu et al [4], describes the important issue of energy conservation for data centers. We consider the problem of provisioning physical servers to a sequence of jobs, and reducing the total energy consumption.

Kaur et al [5], addresses parallel machine scheduling problems with practical Swarm Optimization (PSO). A PSO approach embedded in a simulation model is proposed to minimize the maximum completion time (make span).

Jung et al [6], In cloud computing, a service provider has to guarantee quality of service to offer stable services. For this, we should use scheduling algorithms.

3. PROPOSED METHODOLOGY

The proposed method is Improved Task Migration Consolidation Scheduling (ITMCS), which is described through following point.

Step 1: Input the list of jobs with their size and allocated execution time and list of resources where jobs has been allocated through proposed scheduling approach. Consider that list of jobs are as follows -

J1(1,10), J2(2,5), J3(2,10), J4(3,10), J5(1,25), J6(1,15), J7(2,10), J8(5,5), J9(4,5), J10(1,15).

Consider to resources R1 and R2 with their node size

Resources	Nodes: P1	P2	P3	P4	P5
R1					
R2					

Step 2: Time interval size is 5 Sec. At time T=0

	P1	P2	P3	P4	P5
R1	J1	J2	J2	J3	J3
R2	N	J4	J4	J4	J5

Queue: J6(1,15), J7(2,10), J8(5,5), J9(4,5), J10(1,15)

Step 3: At time T=5

(a)

	P1	P2	P3	P4	P5
R1	J1			J3	J3
R2	N	J4	J4	J4	J5

Queue: J6(1,15),J7(2,10), J8(5,5), J9(4,5), J10(1,15)

(b)

	P1	P2	P3	P4	P5
R1	J1	J5		J3	J3
R2	N	N			

Queue: J4(3,5), J6(1,15), J7(2,10), J8(5,5), J9(4,5), J10(1,15)

(c)

	P1	P2	P3	P4	P5
R1	J1	J5	J6	J3	J3
R2	N	N	N		

Queue: J4(3,5), J7(2,10), J8(5,5), J9(4,5), J10(1,15)

(d)

	P1	P2	P3	P4	P5
R1	J1	J5	J6	J3	J3
R2	N	N	N	J7	J7

Queue: J4(3,5), J8(5,5), J9(4,5), J10(1,15)

Step 4: At time T=10

(a)

	P1	P2	P3	P4	P5
R1		J5	J6		
R2		N	N	J7	J7

Queue: J4(3,5), J8(5,5), J9(4,5), J10(1,15)

(b)

	P1	P2	P3	P4	P5
R1	J4	J5	J6	J4	J4
R2		N	N	J7	J7

Queue: J8(5,5), J9(4,5), J10(1,15)

(c)

	P1	P2	P3	P4	P5
R1	J4	J5	J6	J4	J4
R2	J10	N	N	J7	J7

Queue: J8(5,5), J9(4,5)

Step 5: At time T=15

(a)

	P1	P2	P3	P4	P5
R1		J5	J6		
R2	J10	N	N		

Queue: J8(5,5), J9(4,5)

(b)

	P1	P2	P3	P4	P5
R1	J10	J5	J6		
R2	N	N	N		

Queue: J8(5,5), J9(4,5)

Step 6: At time T=20

	P1	P2	P3	P4	P5
R1	J10	J5			
R2	N				

Queue: J8(5,5), J9(4,5)

Step 7: At time T=25

R1					
R2					

(a)

Queue: J8(5,5), J9(4,5)

(b)

	P1	P2	P3	P4	P5
R1	J8	J8	J8	J8	J8
R2					

Queue: J9(4,5)

(c)

	P1	P2	P3	P4	P5
R1	J8	J8	J8	J8	J8
R2	J9	J9	J9	J9	

Step 8: Total makespan of given queue is 25

4. RESULT AND ANALYSIS

The simulation is initialized by the Main class which creates instances of the scheduler, the job and machine loader, the failure loader and other entities as required by the standard CloudSim 3.0.2.

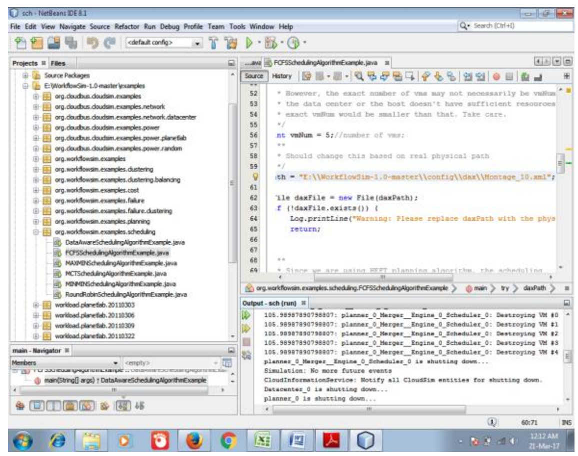

Figure.2. CloudSim 3.0.2 Environment in NetBeans IDE Environment

The makespan(in ms) can be evaluated through FCFS, MCT and ITMCS (Proposed Method) is as follows:

Table.1. Comparison of Makespan among FCFS, MCT and ITMCS

Jobs	MAKESPAN		
	FCFS	MCT	ITMCS (Proposed)
5	119.47	132.54	72.11
10	267.53	210.34	151.69
15	472.35	470.46	379.05
20	527.33	541.69	465.91
25	928.62	925.09	710.66

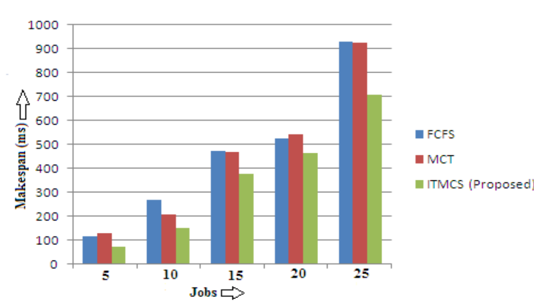

Figure. 3. Makespan among FCFS, MCT and ITMCS

As per above graphical analysis, makespan is less for ITMCS as compare than FCFS and MCT. Therefore, ITMCS is better than FCFS and MCT.

The Minimum Scheduling Execution Time (in sec) can be evaluated through FCFS, MCT and ITMCS (Proposed Method) is as follows:

Table.2. Comparison of Minimum Scheduling Execution Time (sec) among FCFS, MCT and ITMCS

Jobs	MSET		
	FCFS	MCT	ITMCS (Proposed)
5	0.16	0.13	0.12
10	0.18	0.2	0.13
15	0.68	0.81	0.21
20	0.74	0.88	0.24
25	0.89	1.02	0.55

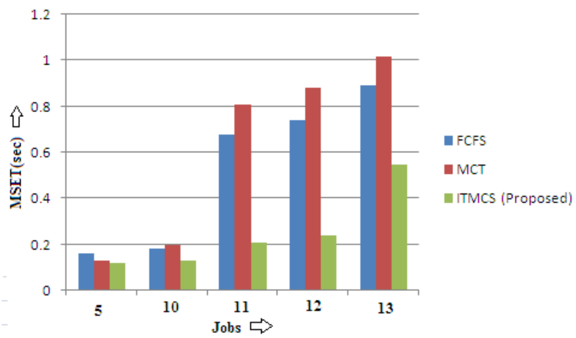

Figure. 4. MSET (sec) among FCFS, MCT and ITMCS

As per above graphical analysis, Minimum Scheduling Execution Time is less for ITMCS as compare than FCFS and MCT. Therefore, ITMCS is better than FCFS and MCT.

The Resource Utilization Rate (in per) can be evaluated through FCFS, MCT and ITMCS (Proposed Method) is as follows:

Table.3. Comparison of Resource Utilization Rate (in per) among FCFS, MCT and ITMCS

Jobs	RUR		
	FCFS	MCT	ITMCS (Proposed)
5	88.12	97.7	98.21
10	78.33	74.34	83.06
15	61.07	59.18	78.13
20	42.41	39.72	59.08
25	28.97	25.08	31.96

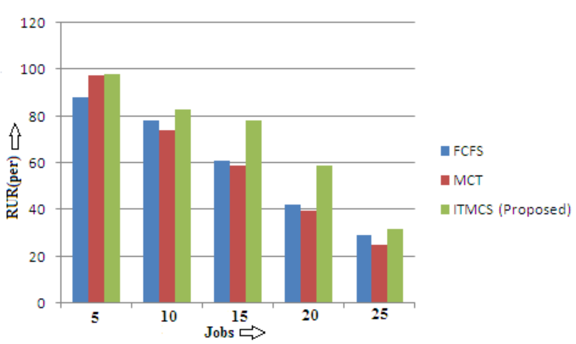

Figure.5. Resource Utilization Rate (sec) among FCFS, MCT and ITMCS

As per above graphical analysis, Resource Utilization Rate (in per) is more for ITMCS as compare than FCFS and MCT. Therefore, ITMCS is better than FCFS and MCT. The Skewness of Makespan (SM) and Makespan Standard Deviation (MSD) for Montage and Cybershake dataset can be evaluated through FCFS, MCT and ITMCS (Proposed Method) is as follows:

Table.4. Comparison of SM and MSD among FCFS, MCT and ITMCS

Scheduling Policy	Montage		Cybershake	
	SM	MSD	SM	MSD
FCFS	1.31	28.12	16.36	148.82
MCT	2.63	41.22	13.8	122.34
ITMCS	1.06	27.73	11.32	119.17

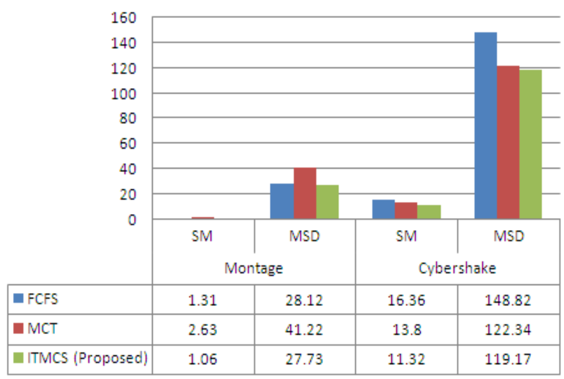

Figure .6. SM and MSD among FCFS, MCT and ITMCS

As per above graphical analysis, SM and MSD is less for ITMCS as compare than FCFS and MCT. Therefore, ITMCS is better than FCFS and MCT.

5. CONCLUSIONS

Cloud computing has been generally recognized as an critical computing pattern to execute compute and data intensive business process workflow (e.g., media processing, analytics pipelines, orchestration of services, coordinating resources, people, information, and systems) and scientific workflow applications for processing of large sets of scientific data, as witnessed by the recent work on Amazon SWF (Simple Workflow Service). We have been introduced an improved workflow scheduling system. A novel method is Improved Task Migration Consolidation Scheduling (ITMCS) scheduling policy was proposed for scheduling workflow applications in a cloud computing environment. An analysis of different performance metrics was carried out. A wide-ranging simulation was performed to estimate the concert of the proposed scheduling procedure. The performance of the ITMCS was then compared with different scheduling policies to highlight the performance and robustness of the proposed solution. The obtained results show that our ITMCS outperforms other scheduling policies. Importantly, ITMCS was shown to utilize computational resources properly by reducing idle time of cloud resource nodes. Further, we conducted proof-of-concept experiments by employing real-world scientific workflow applications. The proof-of concept experiment indicates that the proposed ITMCS scheduling policy offers significant improvements for larger workflow applications. Importantly, a key lesson learned from this study is that multi-tenancy helps improve the utilization of resources. Although we have demonstrated the advantages of multi-tenant cloud environments for scheduling workflow applications, there are several potential directions for future work, including the development of a complex model of scheduling policies by considering resource failures and complex reservation scenarios for multi-tier application scaling, where scaling may affect different applications.

6. SCOPE OF FUTURE WORK

For the future, we intend to further investigate the optimization of the ITMCS scheduling and apply it in the context of mobile cloud computing. Optimization can be performs through Ant Colony Optimization (ACO), Genetic Algorithm (GA) and Particle of Swarm Optimization (PSO) etc.

7. REFERENCES

[1]. HUANG Qi-yi and HUANG Ting-lei, "An Optimistic Job Scheduling Strategy based on QoS for Cloud Computing", IEEE Conf on Cloud Computing, Oct 2010.

[2]. Xiaocheng Liu, Chen Wang, Bing Bing Zhou, Junliang Chen, Ting Yang and Albert Y. Zomaya, "Priority-Based Consolidation of Parallel Workloads in the Cloud", IEEE Trans on Parallel and Distributed Systems, Vol. 24, No. 9, Sep 2013.

[3]. Luqun Li, "An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Providers", Third International Conference on Multimedia and Ubiquitous Engineering, 2009.

[4]. Yung-Ching Hsu, Pangfeng Liu and Jan-Jan Wu, "Job Sequence Scheduling for Cloud Computing", International Conference on Cloud and Service Computing, 2011.

[5]. Mandeep Kaur, Sugandha Sharma and Rajinder Kaur, "Optimization of Job Scheduling in Cloud Computing Environment", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 7, July 2014.

[6]. Sung-Min Jung, Nam-Uk Kim and Tai-Myoung Chung, "Applying Scheduling Algorithms with QoS in the Cloud Computing" IEEE Conf on Cloud Computing jan 2013.

[7]. Dr. Amit Agarwal and Saloni Jain, "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment", International Journal of Computer Trends and Technology (IJCTT) – volume 9 number 7– Mar 2014

[8]. Seema Vahora and Ritesh Patel, "CloudSim-A Survey on VM Management Techniques", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 1, January 2015.

[9]. Yogita Chawla and Mansi Bhonsle, "A Study on Scheduling Methods in Cloud Computing", International Journal of Emerging Trends & Technology in Computer Science, Volume 1, Issue 3, September – October 2012.

[10]. Gabriele D'Angelo, "Parallel and Distributed Simulation from Many Cores to the Public Cloud", IEEE International Conference on High Performance Computing and Simulation (HPCS), July 2011.

[11]. Kushang Parikh, Nagesh Hawanna, Haleema.P.K, Jayasubalakshmi.R and N.Ch.S.N.Iyengar, "Virtual Machine Allocation Policy in Cloud Computing Using CloudSim in Java", International Journal of Grid Distribution Computing, Vol.8, No.1, pp.145-158, 2015.

[12]. Bhupendra Panchal and Prof. R. K. Kapoor, "International Journal of Advanced Research in Computer Science and Software Engineering", Volume 3, Issue 9, September 2013.

[13]. Mohammed Rashid Chowdhury, Mohammad Raihan Mahmud and Rashedur M. Rahman, "Implementation and performance analysis of various VM placement strategies in CloudSim", International Journal of Cloud Computing: Advances, Systems and Applications 2015.

[14]. Anton Beloglazov and Rajkumar Buyya, "Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers", International Journal of Concurrency and Computation: Practice and Experience 2012.

[15]. Manan D. Shah and Harshad B. Prajapati, "Reallocation and Allocation of Virtual Machines in Cloud Computing", International Journal of Internet Services and Applications, 2010.

[16]. A. Mu'alem and D. Feitelson, "Utilization, Predictability, Workloads, and User Runtime Estimates in Scheduling the IBM sp2 with Backfilling," IEEE Trans. Parallel and Distributed Systems, vol. 12, no. 6, pp. 529-543, June 2001.

[17]. D. Jackson, Q. Snell, and M. Clement, "Core Algorithms of the Maui Scheduler," Proc. Workshop Job Scheduling Strategies for Parallel Processing, pp. 87-102, 2001.

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

[19] .A. Do, J. Chen, C. Wang, Y. Lee, A. Zomaya, and B. Zhou, "Profiling Applications for Virtual Machine Placement in Clouds," Proc. IEEE Int'l Conf. Cloud Computing (CLOUD), pp. 660-667, 2011.

[20] Y. Zhang, H. Franke, J. Moreira, and A. Sivasubramaniam, "An Integrated Approach to Parallel Scheduling Using Gang-Scheduling, Backfilling, and Migration," IEEE Trans. Parallel and Distributed Systems, vol. 14, no. 3, pp. 236-247, Mar. 2003.

[21]. D. Tsafrir, Y. Etsion, and D. Feitelson, "Backfilling Using System-Generated Predictions Rather than User Runtime Estimates," IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 6, pp. 789-803, June 2007.

[22]. D. Feitelson, "Packing Schemes for Gang Scheduling," Proc. Workshop Job Scheduling Strategies for Parallel Processing, pp. 89-110, 1996.

[23]. J. Jann, P. Pattnaik, H. Franke, F. Wang, J. Skovira, and J. Riordan, “Modeling of Workload in Mpps,” Proc. Workshop Job Scheduling Strategies for Parallel Processing, pp. 95-116, 1997.

[24]. D. Feitelson, L. Rudolph, U. Schwiegelshohn, K. Sevcik, and P. Wong, “Theory and Practice in Parallel Job Scheduling,” Proc. Workshop Job Scheduling Strategies for Parallel Processing, pp. 1-34, 1997.

[25]. E. Von Puttkamer, “A Simple Hardware Buddy System Memory Allocator,” IEEE Trans. Computers, vol. C-100, no. 10, pp. 953-957, Oct. 1975.

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

[27]. http://www.netbeans.com

[28]. http://www.gwa.ewi.tudelft.nl

[29]. http://www.Plagiarismdetector.com