Performance and Analysis of various Fault-Tolerant Algorithms for Cloud Computing under CloudSim

Virendra Singh Kushwah[1] and Sandip Kumar Goyal[2]

[1,2] Department of Computer Science & Engineering,
Maharishi Markandeshwar University, Mullana, Ambala, India
{kushwah.virendra248, skgmmec}@gmail.com

***Abstract*—A fault is a procedure to make an obstacle for any sort of computing. Fault in such a distributed computing assumes a vital part to prematurely end the working usefulness of the computing. Thus, resistance of a fault is expected to make the figuring in straightforward and simple working. There are various counts for accuse tolerant in the midst of modifying burdens in the appropriated registering. This paper presents correlation investigation of predefined fault tolerant calculations analyzes total failed clouds, add up to cost and execution time. It has been likewise demonstrated what calculation is superior to alternate calculations regarding characterized and utilized parameters. It has likewise been resolved that MIN-MIN calculation has the most minimal Average number of failed cloudlets and less fault tolerant is required in such calculation. Examinations of different fault tolerant calculations on the characterized parameters have been done to utilize appropriate asset usage with the most reduced an incentive from aggregate cost and execution time. Finally, necessities of load adjusting is required for overseeing flaws while balance heaps of assets. Need of adaptation to non-critical failure in distributed computing is an imperative perspective.**

Keywords— Fault Tolerant, Cloud Computing, CloudSim, WorkflowSim

I. INTRODUCTION

Distributed computing infers that as opposed to all the PC gear and writing computer programs, you are using sitting on your desktop, or some place inside your association's framework, it has given to you as an organization by another association and got to over the Internet, typically in a steady way. Adaptation to internal failure alludes to a right and ceaseless operation even within the sight of broken segments. In most of the genuine time based cloud applications, preparing of processing hubs are done remotely. Adaptation to noncritical failure is sent by blunder registering which has two constituent stages. The stages are Effective Error Computing which focused at taking the viable blunder back to a lethargic state, i.e. prior to the event of blunder and Latent Error Computing went for guaranteeing that the mistake does not get to be distinctly compelling once more [1].

II. RELATED WORK

The idea of spot instances on Clouds has been given [1]. In this, the work is composed by two scheduling heuristics that map workflow tasks onto the spot and the on-demand instance. They proposed a booking calculation that timetables assignments on Cloud assets utilizing two distinctive valuing models (spot and on-request occasions) to diminish the cost of execution while meeting the work process due date. The proposed calculation is blame tolerant against the untimely end of spot cases and furthermore vigorous against execution varieties of Cloud assets. They minimize the execution cost

A Fault Injector Module, named by FIM-SIM [2], have been sketched out and realized, on cloud proliferation with the essential target to give an obliging mechanical assembly to endorsement and testing of various adjustment to non-basic disappointment models or any new course of action that can be imperfect.

Mishra et al. [3] added to a fruitful weight conforming computation using creepy crawly settlement streamlining strategy to help or limit particular execution parameters like CPU weight, memory restrict, concede or framework stack for the surges of various sizes.

Song et al. [4] have been proposed another supporting center data adjustment to inside disappointment appropriated processing framework, named IDF Support structure to satisfactorily deal with the testing cloud midway data adjustment to non-basic disappointment issue.

Anju Bala et al. [5] proposed an arrangement that is adequately effective to in a perfect world use the benefits. The results examination exhibited that the execution time of the proposed scheduler is not precisely the present techniques. However there are yet various restrictions that are ought to have been be overcome for more fruitful and dreadful outcomes. Proposed procedure can be consolidated with other resource utilize approaches to manage procure a general resource utilization show/approach.

Raza et al. [6] presented a heuristic based burden adjusted booking model for effective execution of assignments. The recommended demonstrate changes the stacks starting from a couple of customers among datacenters and in this way it offers better resource utilize and high availability as upgraded

response time and turnaround time. The proposed estimation is executed using CloudSim test framework and the result shows that the proposed figuring outmaneuvers to existing computations on tantamount targets.

Author proposed FTCloudSim [8] as a solid cloud server farm recreation framework in light of the fundamental functionalities of CloudSim. FTCloudSim gives an extensible interface to help scientists execute new cloud benefit dependability improvement components. Furthermore, FTCloudSim can likewise concentrate the conduct of the recently proposed components. They show the abilities of FTCloudSim by utilizing five unwavering quality upgrade instruments. The outcomes show the advantages of our proposed reproduction framework.

III. Study & Performance Evaluation of Scheduling Algorithms

The arranging calculation is a worldwide planning calculation that can tie any undertaking to any asset (nonetheless, the real execution arrange relies on upon the asset accessibility). It is utilized as a part of Workflow Planner. By default, the planning algorithm is not set. Map a task to its assigned resource. Some local optimization algorithms such as MIN-MIN, MAX-MIN are scheduling algorithms while some global optimization algorithms such as HEFT are planning algorithms in WorkflowSim [7].

The idea of spot instances on Clouds has been given [1]. In this, the work is composed by two scheduling heuristics that map workflow tasks onto the spot and the on-demand instance. They proposed a booking calculation that timetables assignments on Cloud.

Table 1. Simulation Parameters

Sr. no.	Parameters	Value
1	Cloudlet Image	MONTAGE-1000
2	No. of Cloudlets	1000
3	No. of Virtual Machine	200
4	Scheduling Algorithms	(a) Round Robin (b) MCT (c) MAX-MIN (d) FCFS (e) MIN-MIN
5	Planning Algorithm	HEFT
6	File System	Shared
7	Clustering	No
8	Fault Tolerant Parameters (a) FTC Monitor (b) FTC Failure	 (a) Monitor VM (b) Failure VM

There are following brief introduction about planning algorithm:

A. Round Robin Algorithm

Round robin utilizes the time cutting system. The name of the algorithmic program recommends that it work inside the round way wherever every hub is appointed with a time cut and should anticipate their flip. The time is part and interim is doled out to each hub. Every hub is allocated with a period cut inside which they have to play out their errand. Round robin employments the duration of the time cutting system.

So, Average number of failed cloudlets = 94.45,
Average total cost = 35627.941 and
Average total completion time = 70405.0675 seconds

B. Min-Min Algorithm

In this calculation, little errand is executed first with the goal that huge assignment delays for quite a while. Calculation starts with by sorting the arrangement of every unmapped assignment in expanding request of their fruition time. At that point, the undertakings having the base finishing is booked from the undiscovered errand set and the mapped assignment has been expelled from the unmapped assignment list, and the procedure rehashes until every one of the errands of unmapped rundown are mapped to the comparing accessible assets.

So, Average number of failed cloudlets = 66.95, Average total cost = 35796.6405 and Average total completion time = 78748.3745 seconds

C. Minimum Completion Time (MCT) Algorithm

Minimum Completion Time assigns each job in an arbitrary order to the available resource with the best-expected completion time of that job.

In this calculation, an undertaking that sets aside less opportunity to finish is apportioned to a machine haphazardly. Therefore, MCT carries on to some degree like a Min - min. Notwithstanding, Min-min calculation considers all the unmapped errands amid each mapping choice however then again MCT considers just a single assignment at any given moment.

So, Average number of failed cloudlets = 102.15,
Average total cost = 35483.7445 and
Average total completion time = 68819.336 seconds

D. Max Min Algorithm

Like Min-Min, Max-Min gets the employment with the extreme fulfillment time and allots it to its best accessible asset. The instinct of Max-Min is to stay away from a punishment from long running employments.

In this calculation, expansive undertaking is executed first with the goal that little assignment delays for quite a while. This calculation is fundamentally the same as Min-min calculation, rather than sorting the assignment in the expanding request of finish time.

So, Average number of failed cloudlets = 117.1, Average total cost = 35761.5925 and Average total completion time = 77867.3295 seconds

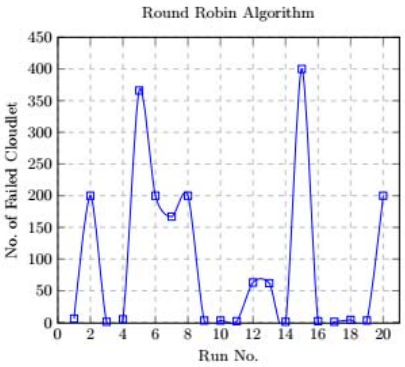

Fig. 1. Failed Cloudlets by RR Algo.

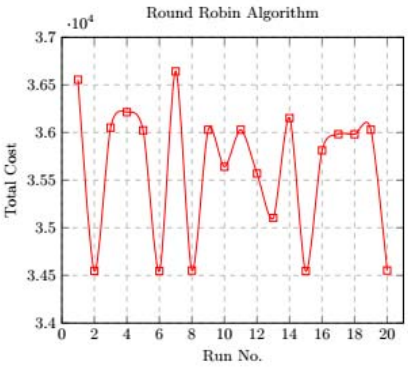

Fig. 2. Total Cost by RR Algo.

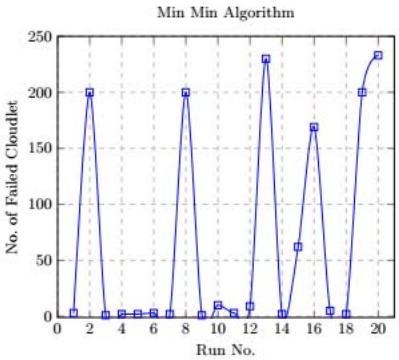

Fig. 3. Failed Cloudlets by MIN-MIN Algo.

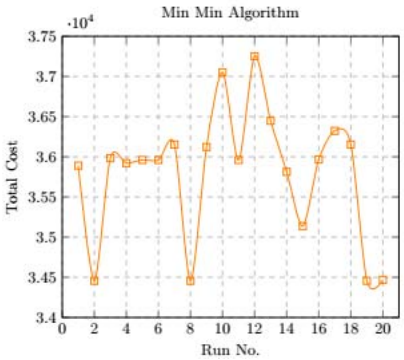

Fig. 4. Total Cost by MIN-MIN Algo.

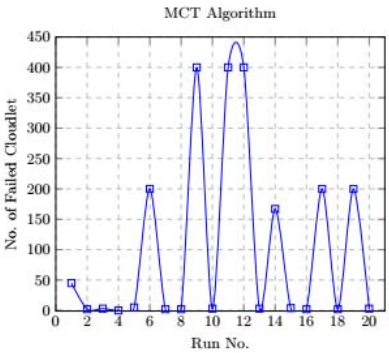

Fig. 5. Failed Cloudlets by MCT Algo.

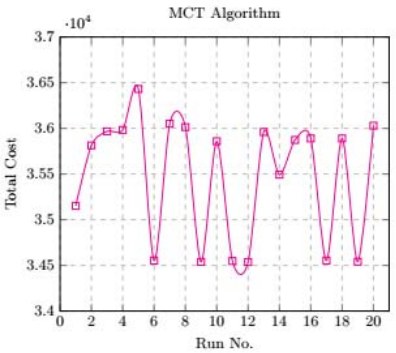

Fig. 6. Total Cost by MCT Algo.

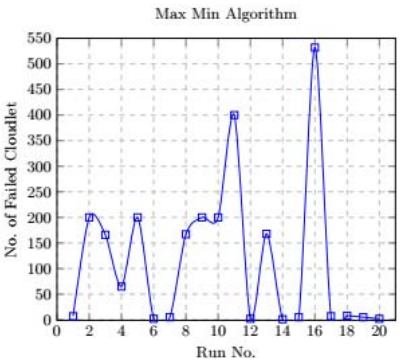

Fig. 7. Failed Cloudlets by MAX-MIN Algo.

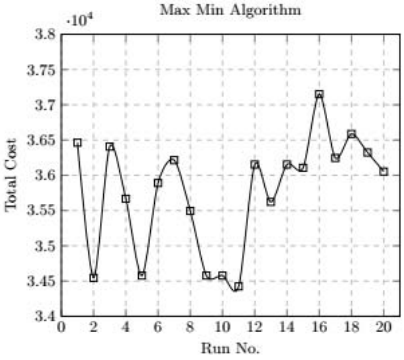

Fig. 8. Total Cost by MAX-MIN Algo.

E. First Come First Serve (FCFS) Algorithm

First Come First Serve is that the basic version of scheduling algorithmic program utilized in our simulator. It assigns each job, inside the internal request on successive on the accessible resources, regardless the parts relied upon fruition time on it attempting hub. Though there numerous assets on their availability, it haphazardly decides particular case likewise those hopeful. In this algorithm, tasks are compared because of their arrival time and the task that comes first in the ready queue is served first.

So, Average number of failed cloudlets = 108.15,
Average total cost = 35975.9885 and
Average total completion time = 73218.126 seconds

F. Result Analysis

Here we are drawing various graphs based on the below table's data.

There are following graphs are to be drawn:

1. Graph for average no. of failed cloudlets,
2. Graph for average total cost,
3. Graph for average total execution time.

Table 2. Summary of Results

Sr. No.	Algorithm Name	Average No. of Failed Cloudlets	Average Total Cost	Average Total Execution Time
1	Round Robin	94.45	35627.94	70405.07
2	Min Min	66.95	35796.64	78748.37
3	MCT	102.15	35483.74	68819.34
4	Max Min	117.1	35761.59	77867.33
5	FCFS	108.15	35975.99	73218.13

In the Fig. 11, it has been seen that MIN-MIN algorithm has the lowest average no. of failed cloudlets while MAX-MIN algorithm has the highest.

The Fig. 12 is plotted for comparison of average total cost among all the algorithms. It can easily determine that MCT algorithm has lowest cost among the all while FCFS has the highest value for the cost.

Unlike the above graphs, the following graph is shown for the average total execution time for the all specified algorithms. We can see that MCT algorithm has the lowest total execution time and this algorithm defines itself while FCFS algorithm has the highest total execution time.

ACKNOWLEDGMENT

Authors are thankful to Department of Computer Science and Engineering at Maharishi Markandeshwar University, Mullana, Ambala (*NAAC-A Grade Deemed University*) for giving highly motivational supports.

REFERENCES

[1] Deepak Poola, Kotagiri Ramamohanarao, and Rajkumar Buyya. Faulttolerant workflow scheduling using spot instances on clouds. Procedia Computer Science, 29:523-533, 2014.

[2] M-C Nita, Florin Pop, Mariana Mocanu, and Valentin Cristea. Fim-sim: Fault injection module for cloudsim based on statistical distributions. Journal of Telecommunications and Information Technology, (4):14-23, 2014.

[3] Ratan Mishra and Anant Jaiswal. Ant colony optimization: A solution of load balancing in cloud. International Journal of Web & Semantic Technology (IJWesT), 3(2):33-50, 2012.

[4] Baoyan Song, Cai Ren, Xuecheng Li, and Linlin Ding. An efficient intermediate data fault-tolerance approach in the cloud. In Web Information System and Application Conference (WISA), 2014 11th, pages 203-206. IEEE, 2014.

[5] Vijay Prakash and Anju Bala. A novel scheduling approach for workflow management in cloud computing. In Signal Propagation and Computer Technology (ICSPCT), 2014 International Conference on, pages 610-615. IEEE, 2014.

[6] Raza Abbas Haidri, CP Katti, and PC Saxena. A load balancing strategy for cloud computing environment. In Signal Propagation and Computer Technology (ICSPCT), 2014 International Conference on, pages 636-641. IEEE, 2014.

[7] Weiwei Chen and Ewa Deelman. Workflowsim: A toolkit for simulating scientific workflows in distributed environments. In E-Science (e-Science), 2012 IEEE 8th International Conference on, pages 1-8. IEEE, 2012.

[8] Ao Zhou, Shangguang Wang, Qibo Sun, Hua Zou, and Fangchun Yang. Ftcloudsim: A simulation tool for cloud service reliability enhancement mechanisms. In Proceedings Demo & Poster Track of ACM/IFIP/USENIX International Middleware Conference, page 2. ACM, 2013.

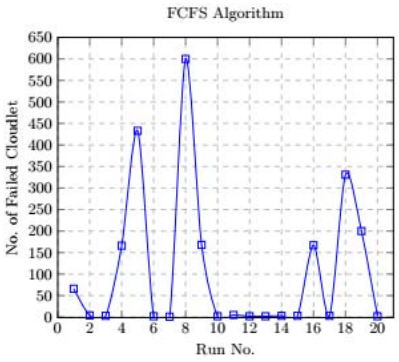

Fig. 9. Failed Cloudlets by FCFS Algo.

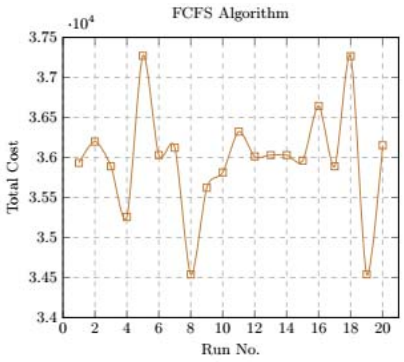

Fig. 10. Total Cost by FCFS Algo.

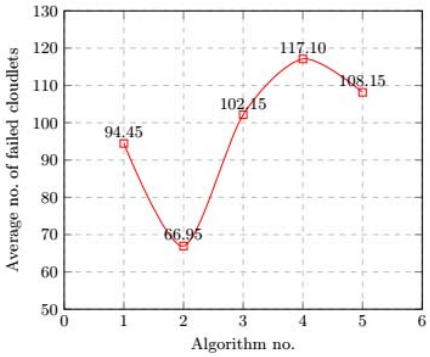

Fig. 11. Comparison of Avg. No. of Failed Cloudlets

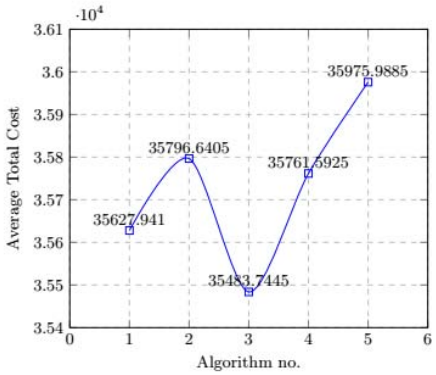

Fig. 12. Comparison of Avg. Total Cost

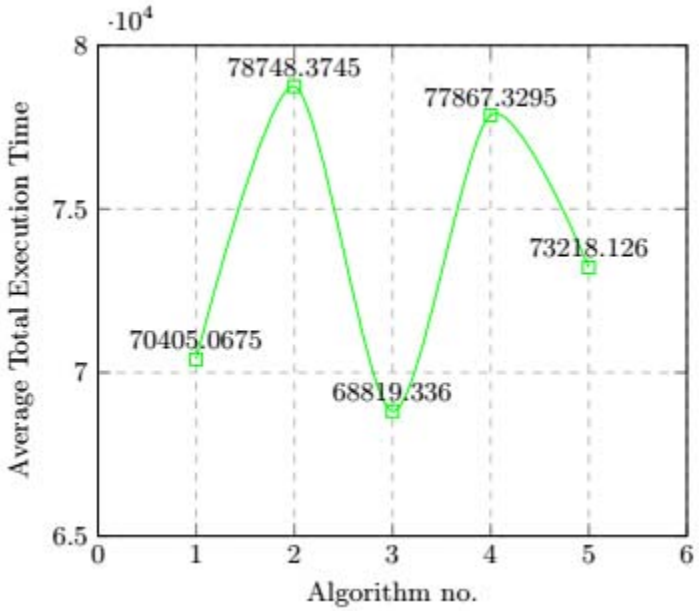

Fig. 13. Comparison of Average Total Execution Time algorithm wise