International Journal of Engineering & Technology, 7 (4.12) (2018) 39-41

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Measuring Throughput for Fault Tolerant Based ACO Algorithm under Cloud Computing: A Comparison Study

Virendra Singh Kushwah[1]*, Sandip K. Goyal[2], Avinash Sharma[3]

[1,2,3]Maharishi Markandeswar University, Mullana, Ambala, INDIA
**Corresponding author E-mail:kushwah.virendra248@gmail.com*

Abstract

Any technical problem can be main cause for any fault. Due to any fault, system would be suffered the work and enhance the system cost in term of money and others. There are many algorithms for fault tolerant in cloud computing and make comparison with fault tolerant based ant colony optimization and which is used to minimize fault during load balancing. In this paper, throughput is measured by such kind of fault tolerant based algorithms and determines that which algorithm is better. It has been compared with ACO. After such comparison, it is clearly determined that ACO has good functionalities to have better throughput. Comparative study is shown by the graphically and finally described that ACO is better than others in context of throughput calculation are. ACO is itself a meta-heuristic algorithm and better optimization technique.

Keywords: *ACO; Fault Tolerant; Throughput; Cloud Computing; Cloudsim;*

1. Introduction

Because of the dynamic idea of the distributed computing condition, extra sources failures are probably to arise within the surroundings, which may also have an effect on the genuine execution time required to execute officially planned employments and in this manner degrading the overall performance of the device and system. Cloud compute in depth programs or jobs as the case can be and often require an awful lot longer execution time a good way to clear up an immature problem. The gigantic processing capability of cloud structures for the most part keep on being unexploited because of their weakness to disappointments, which incorporates system disappointments, gadget crashes, and community disasters [5]. In distributed computing, joining shortcoming tolerant calculations inside the course of errand planning process is routinely suggested. It is far on this mellow; an expansion of the work proposed in [11] is prolonged in this paper through consolidating a blame tolerant booking calculation into the swarm wise lattice process planning set of principles proposed by method for the writer.

2. Fault Tolerant

Adaptation to non-critical failure is chargeable for taking care of the unwavering quality and accessibility of dispensed frameworks [6]. Adaptation to non-critical failure is a usefulness advanced inside the device all together that it might play out its capacity effectively even inside the nearness of help disappointment. Its miles created to hit upon immediately the occurrence of deficiencies and recoup the executable task without support of any outer merchants, subsequently, making the contraption more solid. In fault tolerance, in step with garg and kumar [5], failure is encountered when a machine drifts far away from its everyday behavior. The motive of a failure is called blunders, which additionally in the end depicts some type of fault or illness in that system. Fault is the actual purpose of a failure, and mistakes is simply an illustration or sign of a fault. More than one mistakes will be due to a fault, and even an immature mistake might be the purpose of more than one screw-ups. With many independent sources cooperating as one, the threat of failure of a character aid will increase significantly, specifically if the resources are very bodily dispersed and related the use of network links. With the opportunity of many thousands of computing assets running collectively, the chances of a long jogging manner not failing on as a minimum one resource is almost zero [10]. In addition, in line with townend and xu [15], it was set that as projects scale to exploit network assets, their size and multifaceted nature development essentially. Those frameworks with complex o beat and cooperating sports are exceptionally powerless against mix-ups and disappointments due to their extreme multifaceted nature.

3. Literature Survey

Kumar et al. [8] proposed calculation flourishes to solidness the workload of cloud foundation while bringing down the reaction time for the given number of obligations. The proposed stack adjusting methodology has been recreated utilizing the cloud analyst test system. Reproduction outcomes show that the proposed approach out ranked the present strategies like RR, FCFS and meta-heuristic ACO.

Y. Gao et al. [4] proposed a multi-target subterranean insect province framework calculation for the virtual machine situation issue. Their solution is compared with other existing algorithms and authors also proposed two single objective approaches. Lastly, they have tried to achieve better scalability.

Kushwah et al. [9] concentrated on conceptualized contemplate on adaptation to internal failure with ACO calculation. Different calculations are pro-posed by others creators however this paper is

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

a coordination of all. By utilizing the normal best time, creators upgraded or outlined another calculation to enhance the solidness and unwavering quality of the cloud hubs.

Mahjoub et al. [13] were cautioned an evaluation of various cloud-computing solutions could leverage the cloud computing research area presenting a good start line to analyze organizations and involved readers to higher select the most appropriate one. They have been also given various tools that are used to simulate cloud-computing work.

Zhu et al. [16] did the elaboration at the distributed computing age, and investigates the distributed computing device structure and the conviction of system, sources booking methodology is the key innovation in distributed computing. subsequently utilizing subterranean insect province set of principles for the key adaptation, extraordinary assessment and outline of the cloud help planning the solid acknowledgment, and inside the recreation programming program cloudsim reenactment explore, from the outcomes we will see that, the arrangement of tenets for ascertaining hub dissemination and load adjusting has redress execution.

Mahapatra et al. [12] discussed the load balancing method this is based to-tally on swarm intelligence and also discussed that how the mobile agents can stability the weight of a cloud the use of the concept of ant colony optimization. The quandary of this method is that it is going to be extra green if we form cluster in our cloud.

4. Existing Fault Tolerant Algorithms

4.1. Round Robin

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.

4.2. FCFS

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.

4.3. Min Min

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.

4.4 Max Min

Like Min-Min, Max-Min gets the employment with the extreme fulfilment 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.

4.5 Ant Colony Optimization

Ant Colony Optimization (ACO) Meta heuristic is inspired by the behaviour of real ants finding the shortest path between their colonies and a source of food. Dorigo M. [2] displayed the creepy crawly computation in light of the direct of certified ants in 1996; it is another heuristic count for the plan of combinatorial improvement issues. Insect has the limit of finding a perfect route from home to nourishment.

On the strategy for ants moving, they lay some emanation on the ground; while a dis-associated insect encounter a past fundamental laid trail, this sub-terrane an insect can recognize it and pick with high probability to tail it. Consequently, the trail is fortified with its own air. The probability of subterranean insect picks a way is degree to the union of far quality. To a way, the more ants pick, the way has denser quality, and the denser air attracts more ants. Through this positive information part, underground creepy crawly can find a perfect route finally.

Every single subterranean insect is awkward creepy crawly by typically. They have to a great degree compelled capacity and show particular direct that appears to keep a tremendous subjective part. Going about as a total on the other hand, ants make sense of how to play out a variety of perplexed endeavours with remarkable faithful quality and consistency. Regardless of the way this is essentially self-relationship as opposed to learning, ants need to adjust to a ponder that looks all that much like over getting ready in fortress learning techniques. The many-sided social practices of the sum total of what ants have been very thought by science-based subject, and PC scientists are as of now looking through that these direct cases can offer models to dealing with inconvenience some combinatory upgrade issues. The undertaking to make counts pushed by one a player in underground bug lead, the ability to find what PC analysts would call briefest ways, has transformed into the region of Ant Colony Optimization (ACO), the best and comprehensively saw algorithmic strategy in perspective of creepy crawly direct [14].

5. Simulation and Performance Evaluation

Keeping in mind the end goal to assess the execution of the calculation exhibited in this paper, we actualize the work by the CloudSim toolkit [1].

CloudSim is an extensible simulation framework that enables seamless modeling and simulation of cloud infrastructures and application services. With the help of this tool, entire cloud infrastructures can be easily simulated, creating the physical topology that mimics the one that could to be used in real datacenters. The infrastructure is specified by key infrastructure components like data centers or racks, which makes the simulation easy to create and maintain while still being relevant to the scenario that is being tested. CloudSim can provide an extensible simulation framework generalized by the most properties of the cloud concept.

The various fault tolerant algorithm under the load balancing are used to simulate and compare with ant colony algorithm (ACO).

Important metrics this is required to degree the performance of the new device is the throughput. Throughput is one of the most important standard metrics used to measure the performance of any fault tolerant systems. [7] [3]. Here, throughput is defined as-

Throughput = n/T ET…… …….. (1)

Where n is the aggregate number of employments submitted and TET is the aggregate execution time important to finish the n occupations. The throughput measurements is likewise used to gauge the capacity of the cloud framework to suit employments.

Here we are simply setting the parameters for ACO with cloud computing for fault tolerant. The following parameters are based on basic need for creating cloud-computing environment under faults.

Table 1: Measuring Throughput under various algorithms with no. of cloudlets

Sr. No.	Name of Algorithm	No. of Cloudlets	Total Execution Time (Seconds)	Throughput (Job/Second)
1	Round Robin	25 50 100	706.32 1611.15 5606.59	0.035 0.031 0.017
2	FCFS	25 50 100	706.32 1611.15 5672.38	0.035 0.031 0.018
3	Min Min	25 50 100	706.32 1610.94 5723.01	0.035 0.031 0.017
4	Max Min	25 50 100	1474.74 1609.74 5604.60	0.017 0.031 0.018
5	ACO	25 50 100	706.32 1679.42 5536.32	0.035 0.030 0.018

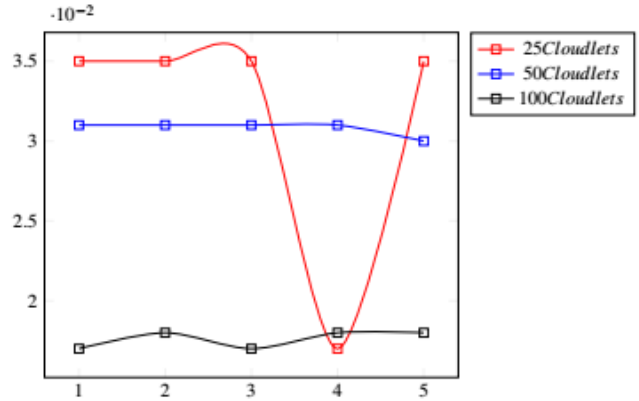

Fig. 1. Measuring Throughput under various algorithms with no. of cloudlets

6. Conclusion and Future Work

Comparison between two algorithms can be done by various measurement factors. Here, in this paper, such type of fault tolerant based algorithms, which are used to manage by fault tolerant, measures throughput. A compassion work is done under cloud computing environment with ACO. Results are satisfied the recognized criteria. It is shown in graphically supported by number of cloudlets used by the algorithm.

Future enhancement can be carried out to design an algorithm with all constraint of ACO. This algorithm will also try to achieve higher throughput and reduce the cost, which are to be calculated by various existing algorithms.

Acknowledgement

Authors are grateful to Department of Computer Science and Engineering at Maharishi Markandeshwar University, Mullana, and Ambala for giving profoundly motivational backings.

References

[1] Rodrigo N Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar AF ´ De Rose, and Rajkumar Buyya. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1):23–50, 2011.

[2] Marco Dorigo and Christian Blum. Ant colony optimization theory: A survey. *Theoretical computer science*, 344(2):243–278, 2005.

[3] Absalom E Ezugwu, Seyed M Buhari, and Sahalu B Junaidu. Virtual machine allocation in cloud computing environment. *International Journal of Cloud Applications and Computing (IJCAC)*, 3(2):47–60, 2013.

[4] Yongqiang Gao, Haibing Guan, Zhengwei Qi, Yang Hou, and Liang Liu. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. *Journal of Computer and System Sciences*, 79(8):1230–1242, 2013.

[5] Ritu Garg and A Kumar Singh. Fault tolerance in grid computing: state of the art and open issues. *International Journal of Computer Science & Engineering Survey (IJCSES)*, 2(1):88–97, 2011.

[6] Sajjad Haider, Naveed Riaz Ansari, Muhammad Akbar, Mohammad Raza Perwez, and KM Ghori. Fault tolerance in distributed paradigms. In *In2011 International Conference on Computer Communication and Management, Proc. of CSIT*, volume 5, 2011.

[7] Fiaz Gul Khan, Kalim Qureshi, and Babar Nazir. Performance evaluation of fault tolerance techniques in grid computing system. *Computers & Electrical Engineering*, 36(6):1110–1122, 2010.

[8] Rajeev Kumar and Tanya Prashar. Performance analysis of load balancing algorithms in cloud computing. *International Journal of Computer Applications*, 120(7), 2015.

[9] Virendra Singh Kushwah and Sandip Kumar Goyal. A basic simulation of aco algorithm under cloud computing for fault tolerant. In *Proceedings of the International Conference on Data Engineering and Communication Technology*, pages 465–472. Springer, 2017.

[10] Muhammad Shafie Abd Latiff, Syed Hamid Hussain Madni, Mohammed Abdullahi, et al. Fault tolerance aware scheduling technique for cloud computing environment using dynamic clustering algorithm. *Neural Computing and Applications*, 29(1):279–293, 2018.

[11] Simone A Ludwig and Azin Moallem. Swarm intelligence approaches for grid load balancing. *Journal of Grid Computing*, 9(3):279–301, 2011.

[12] Deepak Mahapatra, Gaurav Kumar Saini, Himanshu Goyal, and Amit Bhati. Ant colony optimization: A solution of load balancing in cloud.

[13] Meriam Mahjoub, Afef Mdhaffar, Riadh Ben Halima, and Mohamed Jmaiel. A comparative study of the current cloud computing technologies and offers. In *Network Cloud Computing and Applications (NCCA), 2011 First International Symposium on*, pages 131–134. IEEE, 2011.

[14] 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.

[15] Paul Townend and Jie Xu. Fault tolerance within a grid environment. *Time-out*, 1(S2):S3, 2003.

[16] Linan Zhu, Qingshui Li, and Lingna He. Study on cloud computing resource scheduling strategy based on the ant colony optimization algorithm. *IJCSI International Journal of Computer Science Issues*, 9(5):1694–0814, 2012.