Unusual Framework for Fault Tolerant in a Cloud Habitat utilizing ACO Algorithm

Virendra Singh Kushwah and Sandip Kumar Goyal

Department of Computer Science and Engineering
Maharishi Markandeshwar University, Mullana-Ambala-133207, India

***Abstract*— *In today's time, a distributed computing model requires overseeing loads from the different datacenters. In any case, it is difficult assignment to manage deal with the heaps. Since, there are many issues, which are prepared to welcome issues into the mists or datacenters. It is very much characterized that there are numerous predefined calculations to deal with the issues and those are perceived by fault tolerant calculations. Yet, there are couple of calculations, which cannot manage the deficiencies. In spite of the way that, there are various frameworks which are used to get the best possible occurs for persevering through the issues. Adaptation to non-critical failure implies a technique to framework outline that allows a framework to continue performing really, when one of its parts crashes and burns or it can be characterized as limit of a framework to respond agilely to a surprising gear or programming separate. If not operational, adaptation to non-critical failure arrangements may permit a framework to keep working at lessened limit as opposed to closing down totally taking after a disappointment. Along these lines, to keep from such issues or faults, this paper concentrated on showing a structure of new fault tolerant calculation, which depends on Ant Colony Optimization strategy. ACO takes after its working in light of insect's practices and which is finding the briefest way between their settlements and a wellspring of sustenance. That is by; ACO turns into a simple and valuable method.*

Index Terms*— *Framework, Cloud Computing, Fault Tolerant, ACO, Algorithm

I. INTRODUCTION

A. Cloud Computing

Distributed computing, as next form of the network frameworks, has every one of the characteristics of parallel framework assembling a gathering of virtualized hubs, powerfully provisioned and introduced as one brought together figuring asset. The assets are apportioned through the tenets of administration level assertions and arranged lies in the specialist organization and buyer. Investigating and testing the execution of a circulated framework, for example, an open cloud has turned out to be largely a test. Distributed computing situations are putting forth a powerfully expansive pool of assets, configurable and alternatively rebalanced. A full trial of an open cloud can bring about a noteworthy cost and time, with the likelihood to go to a huge number of handling center included. The most reasonable decision to test the organization divulgence execution, arranging, checking of these systems without an adaptable circumstance is a multiplication gadget. This instrument should have the capacity to duplicate the pertinent tests and conduct of a genuine framework.

B. Fault Tolerant

As a result of the application many-sided quality, the greater part of the circumstances faults in the framework are unavoidable and must be viewed as an ordinary part of nature. Keeping in mind the end goal to legitimately manage shortcomings we should comprehend them and the chain of occasions that lead from deformity to fault, blunder and disappointments [1]. A fault speaks to a peculiarity in the framework that makes a reason it to carry on in a capricious and sudden matter. Shortcomings can be sorted into the advancement stage they show up in, as improvement deficiencies or operational flaws. They can be outer or inner in light of the limits secured by the framework, equipment or programming, as per what causes them and they can be purposeful or inadvertent, regardless of whether they show up in the testing stage or underway.

Regardless of what the fault is, the issues that it may bring about are meant a mistake in the framework. This will not occur and regardless of the possibility that it does it may have a specific inactivity before a fault show itself as a blunder. In the event that a mistake is distinguished, the framework may at present keep on working of course however; measures must be taken to keep the framework filling in as some time recently. In the event that the blunder cannot be dealt with, it will bring about a disappointment, which is a circumstance in which the outcomes delivered by the framework are changed somehow. Just incomplete disappointments may happen, yet it will at present cause either a misfortune in execution or even total shutdown of the framework [2].

Keeping in mind the end goal to counteract disappointments in the application, three methods can be utilized at various levels of the fault blunder disappointment chain: fault evasion, fault evacuation, adaptation to internal failure. Fault evasion and fault evacuation both allude to the way that issues ought to be managed as quickly as time permits and not permit them to form into mistakes, as blunders are harder to deal with and can undoubtedly prompt to disappointments.

A fault tolerant framework is a kind of structure which can tweak itself when a product or equipment based blunder happened. As a plainly obvious reality, in a defect tolerant system if a mistake happens the structure will have the capacity to repair itself and proceed with its occupation with no intrusion. Dependability is the guideline point in these structures is faithful quality component, which is so crucial. Various game plans can grow relentless quality in these systems. One of these game plans is excess technique.

Adaptation to internal failure incorporates every one of the strategies fundamental for heartiness and constancy. The principle favorable circumstances of utilizing adaptation to non-critical failure in distributed computing incorporates disappointment recuperation, bring down expenses, and enhanced gauges in execution. Plainly distributed computing has turned into a middle to run conveyed applications by misusing the distinctive layers of virtualization, and can give more adaptability in planning applications. The constancy of a few parts of the QoS gave by the framework comprises of components, for example, reliability and accessibility [3].

II. ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) metaheuristic is roused by the conduct of genuine ants finding the briefest way between their provinces and a wellspring of nourishment. Dorigo M. [4] displayed the creepy crawly figuring in light of the direct of veritable ants in 1996; it is another heuristic computation for the game plan of combinatorial improvement issues. Insect has the limit of finding a perfect path from home to sustenance.

On the technique for ants moving, they lay some emanation on the ground; while a detached subterranean insect encounter a past fundamental laid trail, this subterranean insect can recognize it and pick with high probability to tail it. In this manner, the trail is reinforced with its own atmosphere. The probability of subterranean insect picks a way is degree to the merging of a ways emanation. To a way, the more ants pick, the way has denser atmosphere, and the denser quality attracts more ants. Through this positive info part, underground creepy crawly can find a perfect route finally.

Every single subterranean insect is awkward creepy crawly by behaviorally. They have a greatly obliged capacity and show solitary lead that appears to keep a tremendous discretionary part. Going about as a total of course, ants make sense of how to play out a grouping of jumbled endeavors with phenomenal unfaltering quality and consistency. Regardless of the way this is fundamentally self-relationship as opposed to learning, ants need to adjust to a ponder that looks all that much like over get ready in stronghold learning strategies. The unpredictable social practices of the sum total of what ants have been very focused by science based subject, and PC analysts are as of now seeking that these direct illustrations can offer models to dealing with troublesome combinatory upgrade issues.

The attempt to make estimations impelled by one a player in underground bug direct, the ability to find what PC scientists would call briefest ways, has transformed into the territory of Ant Colony Optimization (ACO), the best and extensively saw algorithmic methodology in perspective of bug lead [5].

III. CLOUD-BASED FRAMEWORK FOR ACO

To make ACO calculations have the viable abuse of the inquiry encounter gathered up until now and keep adequately investigation all the while, by taking utilization of the haphazardness and vulnerability of cloud model, we display a structure for ACO, called Cloud-based system for Ant Colony Optimization.

```
Algorithm Cloud-based Framework for ACO
Procedure CFACO
    Initialize parameters;
    While (termination criterion not satisfied)
        Construct Solution;
        Apply Local Search; // optional
        Global Pheromone Update;
        Evaluate Pheromone State;
        Apply Self adaptive Mechanism;
        Suboptimal Solutions Pheromone
    Update;
    End While
End Procedure
```

For ACO, the calculation first introduces the execution showed by pheromone in Fig. 1. and after that continues developing era by era. In every era, to develop an answer for every insect, a typical procedure is to view every cloud undertaking Ti as a subterranean insect venture, as appeared in Fig. 2. By then, execution pheromone and heuristic information are gotten to pick the most sensible cloud resource Rj to execute the task. After each one of the game plans are constructed, pheromone neighborhood update and overall upgrade are performed. ACO will end when an end condition is met.

A basic framework of using ACO to schedule user tasks is illustrated in Fig. 2. where each ant uses M steps to construct a solution. In the ith step to schedule the ith task T_i, the insect utilizes pheromone and heuristic data to pick the appropriate asset R_j. After M steps, all the M tasks have been scheduled on different resources.

In this unique circumstance, [6, 7] have booked the M undertakings one by one to the cloud assets, utilizing the plan in Fig. 2., in light of the fact that at every progression the undertaking can be planned on any asset of an asset set. The pheromone upgrade plan was adjusted by various schedule openings of cloud administration in [6] and the heuristic data in light of the clients QoS on client cost, framework dependability, reaction, or security was utilized to direct the subterranean insect to choose the ideal asset in [7]. It has been [8] additionally utilized an ACO-based way to deal with calendar cloud asset by considering makespan, client cost, arrange transmission capacity, and framework unwavering quality. The assignments were initially grouped into various classes as per distinctive QoS measurements, and after that, undertakings in various classifications were bound to the cloud assets through the ACO streamlining.

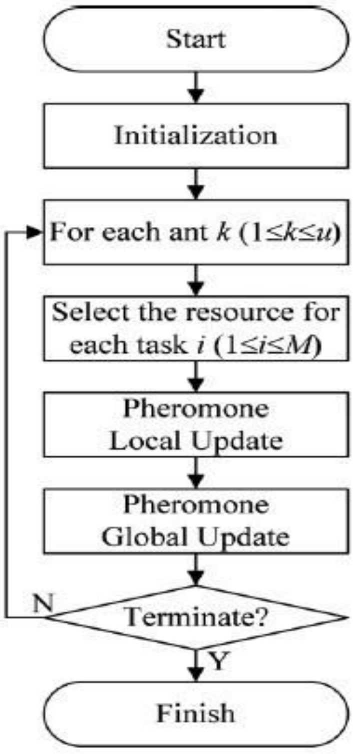
Start
Initialization
For each ant k ($1\leq k\leq u$)
Select the resource for each task i ($1\leq i\leq M$)
Pheromone Local Update
Pheromone Global Update
Terminate?
N
Y
Finish

Fig. 1. Framework of Ant Colony Optimization.

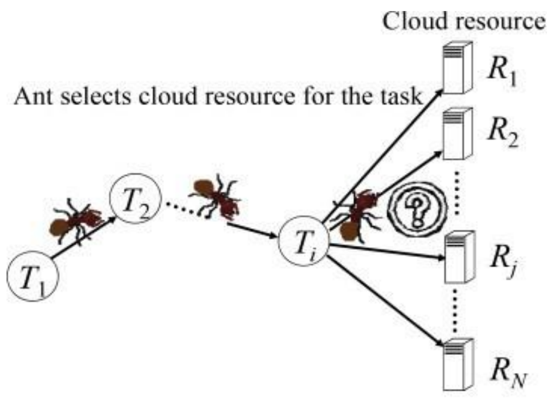

Fig. 2. Illustration of an ant search fashion in ACO for cloud resource scheduling. [21]

IV. LITERATURE REVIEW

Tawfeek et al. [9] have taken minimization of makespan as the goal work. They have taken an imperative of going to each Virtual Machine (VM) once for every subterranean insect and heuristic capacity depends on expected execution time and exchange time of assignment ti on VM vmj. The calculation is mimicked in Cloudsim with the quantity of undertakings shifting from 100 to 1000. ACO is contrasted and Round Robin and FCFS calculations and exploratory outcomes demonstrate that with the expansion in number of undertakings, ACO takes less time than RR and FCFS. For 1000 errands, there is around 29-32% decrease in makespan in correlation with RR and FCFS.

To enhance the execution of ACO and to make it more proficient, pheromone upgrading methodologies are proposed in [10, 11]. Liu and Wang [12] introduced an errand planning calculation for lattices by adaptively changing the estimation of pheromone. The estimation of dissipation rate is adaptively changed and never permitted to lessen to zero. It quickens the joining rate, spares the seeking time and dodges the prior stagnation.

MadadyarAdeh and Bagherzadeh [13] have presented the idea of one-sided beginning ants to enhance ACO. Their approach utilizes the aftereffects of deterministic calculations for one-sided beginning ants. Creators have additionally viewed as standard deviation of employments notwithstanding pheromone, heuristic data and anticipated that time would execute a vocation on a given machine. Their trial comes about show makespan decrease of 33% and 20% in correlation with MaxMin and MinMin individually in predictable, low undertaking and low machine heterogeneity environment.

Encourage, planning calculations in light of ACO can be enhanced by applying neighborhood look procedures to the yield of the ACO calculation [14]. The neighborhood seek procedures proposed in are centered around finding the issue asset i.e. the asset whose aggregate execution time is equivalent to the makespan of the arrangement and moving or swapping occupations from issue asset to another asset that has least makespan.

Chiang et al. [15] proposed to incorporate two probabilistic state-move rules while applying ACO to work process planning in bunches. Creators have characterized the control for undertaking determination notwithstanding asset choice Author work has joined nearby inquiry system toward the finish of every cycle to enhance each got arrangement.

For booking of ward errands or work processes in network or cloud, a subjective N number of ants are utilized as a part of [16]. Every subterranean insect begins with a passage undertaking and chooses an asset in light of the previously mentioned probabilistic capacity. Every insect fabricates a grouping of errands that fulfills the priority limitations. The request of mapping assignments to assets depends on this succession. In this way, every insect manufactures the arrangement incrementally in N steps, where N is the no. of undertakings. Amid every emphasis, B ants construct B arrangements in parallel. Nearby pheromone, redesigning is done in the wake of mapping asset to every assignment and worldwide pheromone overhauling is done toward the finish of every emphasis.

Chen et al. [17] introduced a work process booking calculation in light of Ant Colony System (ACS) calculation with the expansion of a few new components for its change. They intended to limit the cost while meeting the due date. For this, two sorts of pheromone are characterized, one supporting the minimization of makespan and other favoring minimization of cost. They to manage the ants in discovering their heading of pursuit have characterized three sorts of heuristic data. Out of these, every subterranean insect utilized one heuristics sort and one pheromone sort in every emphasis in light of the probabilities controlled by two parameters. These two parameters are adaptively balanced in the calculation.

ACO can be upgraded by using data got from destined number of best courses of action of past accentuations [18]. Learning system is consolidated with the ACO figuring. Data system is changed by two methodologies, one is by learning sparing guideline and other is by learning vanishing standard. Learning keeping incorporates copying the data cross section with a consistent and is performed on best timetable found till then while learning dispersal is done after every accentuation.

Wen et al. [19] prescribed that ACO count can in like manner be combined with various computations, for instance, Particle Swarm Optimization (PSO) to improve its execution. The proposed estimation not simply redesigns the union speed and improves resource use extent, moreover abstains from falling into close-by perfect game plan.

Pacini et al. [20] tended to the issue of conforming throughput and response time when different customers are running their exploratory investigations on online private cloud. The course of action expects to reasonably arrange virtual machines on hosts. Throughput is related to number of customers suitably served and response time is associated with number of virtual machines appropriated.

A bit of the makers have focused on weight modifying of advantages for improve the execution of task booking in cloud environment. Li et al. [7] presented Load Balancing Ant Colony Optimization (LBACO) count, for booking of self-ruling errands with the purpose of limiting makespan and even load over every single virtual machine. They have discovered the level of clumsiness to evaluate the abnormality among virtual machines. They have figured the level of awkwardness to gauge the lopsidedness among virtual machines. Their test outcomes show that LBACO has diminished the typical makespan by 63% and level of cumbersomeness by 47% generally in examination with FCFS count for the 500 free endeavors with the expressed parameter settings in CloudSim. In this manner, it has performed much superior to FCFS calculation.

V. CONCLUSION

In this paper, we have endeavored to demonstrate another structure of the issue tolerant system using ACO framework. The motivation of this framework is to revise instrument of tolerant the blemishes. ACO considers after its working taking ants practices and which is finding the most concise route between their settlements and a wellspring of food. It is a progression technique which has the higher resolute quality for giving perfect game plan.

ACKNOWLEDGEMENT

Authors are thankful to Department of Computer Science and Engineering at Maharishi Markandeshwar University, Mullana, Ambala for giving highly motivational supports.

REFERENCES

[1]. Lavinia, C. Dobre, F. Pop, and V. Cristea, “A failure detection system for large scale distributed systems,” in *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on*. IEEE, 2010, pp.482–489.

[2]. F. Palmieri, S. Pardi, and P. Veronesi, "A fault avoidance strategy improving the reliability of the egi production grid infrastructure," in *Principles of Distributed Systems*. Springer, 2010, pp. 159–172.

[3]. P. Latchoumy and S. A. Khader, "Survey on fault tolerance in grid computing", IJCSI International Journal of Computer Science Issues, vol. 2, no. 4, 2011.

[4]. M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," *Theoretical computer science*, vol. 344, no. 2, pp. 243–278, 2005.

[5]. R. Mishra and A. Jaiswal, "Ant colony optimization: A solution of load balancing in cloud," *International Journal of Web & Semantic Technology (IJWesT)*, vol. 3, no. 2, pp. 33–50, 2012.

[6]. S. Banerjee, I. Mukherjee, and P. Mahanti, "Cloud computing initiative using modified ant colony framework," *World academy of science, engineering and technology*, vol. 56, pp. 221–224, 2009.

[7]. K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, "Cloud task scheduling based on load balancing ant colony optimization," in *Chinagrid Conference (ChinaGrid), 2011 Sixth Annual*. IEEE, 2011, pp. 3–9.

[8]. A. Zhou, S. Wang, Q. Sun, H. Zou, and F. Yang, "Ftcloudsim: A simulation tool for cloud service reliability enhancement mechanisms," in *Proceedings Demo & Poster Track of ACM/IFIP/USENIX International Middleware Conference*. ACM, 2013, p. 2.

[9]. M. Tawfeek, A. El-Sisi, A. E. Keshk, F. Torkey *et al.*, "Cloud task scheduling based on ant colony optimization," in *Computer Engineering & Systems (ICCES), 2013 8th International Conference on*. IEEE, 2013, pp. 64–69.

[10]. J. Sun, S.-W. Xiong, and F.-M. Guo, "A new pheromone updating strategy in ant colony optimization," in *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, vol. 1. IEEE, 2004, pp. 620–625.

[11]. P. Mathiyalagan, S. Suriya, and S. Sivanandam, "Modified ant colony algorithm for grid scheduling," *IJCSE) International Journal on Computer Science and Engineering*, vol. 2, no. 02, pp. 132–139, 2010.

[12]. A. Liu and Z. Wang, "Grid task scheduling based on adaptive ant colony algorithm," in *Management of e-Commerce and e-Government, 2008. ICMECG'08. International Conference on*. IEEE, 2008, pp. 415–418.

[13]. M. MadadyarAdeh and J. Bagherzadeh, "An improved ant algorithm for grid scheduling problem using biased initial ants," in *Computer Research and Development (ICCRD), 2011 3rd International Conference on*, vol. 2. IEEE, 2011, pp. 373–378

[14]. K. Kousalya and P. Balasubramanie, "To improve ant algorithm's grid scheduling using local search." *International Journal of Intelligent Information Technology Application*, vol. 2, no. 2, 2009.

[15]. C.-W. Chiang, Y.-C. Lee, C.-N. Lee, and T.-Y. Chou, "Ant colony optimisation for task matching and scheduling," in *Computers and Digital Techniques, IEE Proceedings-*, vol. 153, no. 6. IET, 2006, pp. 373–380.

[16]. W.-N. Chen and J. Zhang, "An ant colony optimization approach to a grid workflow scheduling problem with various qos requirements," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, no. 1, pp. 29–43, 2009.

[17]. W.-N. Chen, J. Zhang, and Y. Yu, "Workflow scheduling in grids: an ant colony optimization approach," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2007, pp. 3308–3315.

[18]. Y. Hu, L. Xing, W. Zhang, W. Xiao, and D. Tang, "A knowledge-based ant colony optimization for a grid workflow scheduling problem," in *Advances in Swarm Intelligence*. Springer, 2010, pp. 241–248.

[19]. X. Wen, M. Huang, and J. Shi, "Study on resources scheduling based on aco allgorithm and pso algorithm in cloud computing," in *Distributed Computing and Applications to Business, Engineering & Science (DCABES), 2012 11th International Symposium on*. IEEE, 2012, pp. 219–222.

[20]. E. Pacinia, C. Mateosb, and C. G. Garinoa, "Balancing throughput and response time in online scientific clouds via ant colony optimization," *Advances in Engineering Software. In press. Elsevier*, 2014.

[21]. Zhan, Zhi-Hui, et al. "Cloud computing resource scheduling and a survey of its evolutionary approaches." *ACM Computing Surveys (CSUR)* 47.4 (2015): 63.