International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER SSRN

2023 IJIEMR. Personal use of this material is permitted. Permission from IJIEMR must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. No Reprint should be done to this paper, all copy right is authenticated to Paper Authors

IJIEMR Transactions, online available on 5th Jan 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=Issue 01

DOI: 10.48047/IJIEMR/V12/ISSUE 01/10

Title **Reliability Analysis of Various Types of Intelligent Agents in A Multi Intelligent Agent Model Over Distributed Communication Networks by Changing Number of User Requests**

Volume 12, ISSUE 01, Pages: 95-105

Paper Authors
M.Veeresh Babu, Dr.Pushpneel Verma, Zabi Ur Rahaman K

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Reliability Analysis of Various Types of Intelligent Agents in A Multi Intelligent Agent Model Over Distributed Communication Networks by Changing Number of User Requests

M.Veeresh Babu[1], Dr.Pushpneel Verma[2], Zabi Ur Rahaman K[3]
[1]Research Scholar, Dept. of Computer Science, Bagawanth University, Ajmir, mveereshbabu@gmail.com
[2]Professor, Dept. of Computer Science, Bagawanth University, Ajmir, pushpneelverma@gmail.com
[3]Academic Dean, Paradise International School, K.R. Nagara, rahman.naju@gmail.com

Abstract
At present distribute networks are used in number of applications. Conventional networks are not in a position to balance load, shows less performance, and also reliability is less. To overcome this problem, we proposed multi intelligent agent model, here five different types of intelligent agents are created to solve user requests, and all these intelligent agents are working in a parallel manner. In this paper we proposed Multi intelligent agent model, in a network user queries in the form requests are accepted, for each request on network intelligent agent is allotted, these agents are working with mixed mode configurations, intelligent agents will identify shortest path over agents to resolve user requests, and the main objective is to execute all user requests within stipulated time period. The complete distrusted network is under observation to check overall reliability. We observed performance of multi intelligent agent model, reliability of agents, and number of tasks executed by agents, how agents are migrating from one node to other, how shortest path is established between intelligent agents, and how user requests are resolved by these agents. Performances of a multi intelligent agent model are high when compared with conventional networks.

Keywords: *Intelligent Agents, Reliability, Network Failure, Network Performance, mixed mode network configuration models, distributed communication networks.*

1. Introduction

As of late the ideas and innovations related with smart specialists have become well known and issues, for example, dependable investigation and plan strategies are progressively developed. Conveyed specialists stand out enough to be noticed, as they are utilized in getting to circulated assets in an organization. To get

to the assets proficiently, a dependable Distributed Communication Network (DCN) is required. For meeting steadily changing business necessities, associations need to put more in time and spending plan for increasing IT foundations. The accompanying basic issues are presented by cloud plans of action and advancements including load adjusting, security, energy-productivity, work process booking, information/administration accessibility, permit the executives, information secure and API plan. High Availability (HA) and unwavering quality in distributed computing administrations are a portion of the hot difficulties.

This research paper targets breaking down the dependability of DCN, which is figured out utilizing dispersed specialists. Specialists are independent elements and are called programming robots and their collaborations are helpful. The unwavering quality of the DCN is a variable that might rely upon the exhibition, accessibility and procedure of circulated specialist based framework and it gives a disappointment free climate to correspondence. A disseminated portable specialist framework is characterized as one, which comprises of different specialists, every one of which achieves a free errand. These undertaking goes from going for an irregular stroll, process the data at the picked hub and returns the outcome to the requestor. The circulated specialists go about as tokens, which goes over the organization from one hub to another. It is fundamental for the specialist to finish its walk effectively by a predetermined cutoff time t. The specialist needs an associated course during the period t to keep up with versatility. This characterizes the idea of dispersed specialist visit or walk, which is utilized to appraise the dependability of the disseminated specialists. Subsequently, the issue is diminished to the issue of effective fruition of specialist's visit, in light of the situation with DCN. With such a framework, power of DCN is stated and consequently, the expansion in unwavering quality.

Related Work

All Unscathed element that is shortcoming which decides the likelihood of events assaults and furthermore play a hitting liability with different variables of safety. Issues are normal even today. Shortcoming is the reason for programming disappointment while disappointment is the impact that happens because of the issue. Ability of shortcoming is connoted by the break of unstable programming. A

security assessment process gives a precise clue to estimation of programming shortcomings. The flaws are found during the testing and the disappointment is the point at which the framework quits working. Issue assaults can be sent in programming which by and large assistance to keep away from, recognize and address deficiencies. During advancement of programming, blames and defects are presented either from the execution or from the plan of the product. It depends on programming credits that how much framework's product is safeguarded; the very shortcoming chooses for how longer programming will be secured. Shortcoming isn't just the component that makes things hard to see yet with sufficient trouble anything can become more diligently to comprehend.

The interest in programming adaptation to non-critical failure has been developing as programming issues have turned into the significant supporter of framework disappointments. Investigation of safety, specialists says that issue, soundness, intricacy and dependability are a fundamental ascribes which encroach on the profundity of safety. An extensive stretch of time adaptation to internal failure has been utilized as a mean of improving and controlling the security of programming framework. Adaptation to non-critical failure is the powerful technique that is utilized to hold the interconnected frameworks together, support security and accessibility in frameworks. Adaptation to non-critical failure can be sorted in three level; Hardware, Software, and System flaws. Specialists much of the time upholds and support that adaptation to non-critical failure hypothesis ought to be applied at beginning phase in the improvement cycle of programming. Programming adaptation to non-critical failure is frightened with procedures expected to work with a framework to endure programming 'bugs' or blames. Adaptation to internal failure is in many cases old equivalently with effortless corruption which means to recognize, discrete and resolve issues prudently. Not all of the security flaws existing in programming frameworks are recognizable during the issue examination.

The model created improved results that are contrasted and another generally existing model that utilizes same strategies for example replication and resubmission. The proposed calculation reschedules the bombed errand to some profoundly solid Virtual Machine (VM). An issue location and moderation approach. It distinguish shortcoming in VM at beginning phase

and stay away from it or shortcoming at runtime .If shortcoming in VM is recognized then essential activity are taken like restarting position at new VM or moving position to some other solid VM. Both proactive and receptive methodology is utilized for shortcoming recuperation of component of organization by displaying them as Multidimensional Knapsack issue. Point is to work on the dependability and accessibility of the VM. At first for proactive methodology two tests are given for disappointment expectation for example Acknowledgment test and Reliability assessor and if disappointment happens in both the test then receptive methodology likewise gave to bargain the disappointment of VM.

The design of a dispersed figuring framework includes a bunch of collaborating processors imparting over the correspondence joins. A disseminated program running comprises of a few modules that should be dispensed to the processors and between convey through the connections for the culmination of program execution. To work on the presentation of a it, a few issues emerge, for example, the minimization of execution and correspondence cost, the expansion of framework unwavering quality and security, and the accomplishment of better adaptation to non-critical failure utilizing programming and equipment overt repetitiveness. The likelihood for the effective culmination of dispersed programs which expects that every one of the designated processors and involved correspondence joins is functional during the execution lifetime. The conveyed program unwavering quality via looking through the entire insignificant document crossing trees, which give openness to the expected information records for the program. Nonetheless, some framework boundaries, for example, the execution seasons of projects and correspondence loads on the connections are not viewed as in this model. They accept that all processors and correspondence joins have steady dependability.

Application suppliers utilize load-adjusting replication for their applications. Replication disperses responsibility across numerous application cases, each frequently running on its own different VM. Albeit an application can work — perhaps with corrupted execution — in light of utilization occasion disappointments, replication is worthless in the event that appropriate systems are not set up to distinguish disappointments and to recuperate from them. Associations

frequently prefer to utilize free shortcoming the board framework that accompanies promptly accessible observing and alert help for application parts. Such independent administration framework decoupled from application facilitates improvement cycle while advancing programming reuse. However, this considerate situation tragically makes the recuperation of utilizations subject to the shortcoming the board design also, the situation with its chiefs — we utilize the term, recuperation, to mean disengaging and eliminating the bombed application cases.

At the point when the disappointment of an application occasion happens, the impacted positions are either lost, or resubmitted by load adjusting module to the excess functional application cases once the adjusting module is familiar with the disappointment. The resubmission might cause transient over-burden in the framework prompting execution debasement. The over-burden might be taken care of by utilizing an affirmation control instrument to fulfill the exhibition goals.

III. Multi Intelligent Agent Model

In multi intelligent agent model multiple agents are created and these agents move from one node to other over a network to solve user requests within stipulated time. These agents should be connected with each other to form distributed network and also useful to manage mobility of network. These agents are move around from one node to other node over a wireless network and when it is move around from node to node it will carry info like AID, its data, and status of execution to transfer. Middleware is used to perform all these operations. These agents are working on behalf of users to solve their requests.

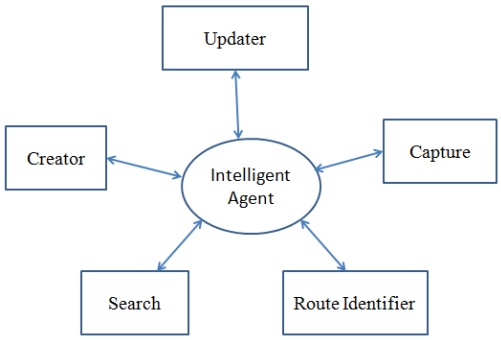

Figure 1: Multi Intelligent Agent Model

Intelligent agents are created and network is formed with help of intelligent agents. Number of intelligent agents was created to solve user requirements and number of agents may be changing dynamically based on number of user requests. Algorithm for searching elements, updating elements, capturing, and shortest path using multi intelligent agents was described below.

Algorithm Intelligent_Agent_Search( )

Step 1: Number of intelligent agents to be created must be initialized.

Step 2: Intelligent agents were created.

Step 3: users may request an element or data or file and these are searching by using intelligent agents.

Step 4: count number of user requests and to resolve each user request one intelligent agent created.

Step 5: Reliability of these intelligent agents are calculated and these values are varied from zero to one.

Step 6: if an intelligent agent working without errors then intelligent agent reliability is equal to one.

Step 7: if an intelligent agent is not working then intelligent agent reliability is equal to zero.

Step 8: if the intelligent agent reliability is equal to 0.25 then agent will not continue with requests.

Step 9: if is there any searching faults then reliability are in between 0.25 to 0.85.

Step 10: all multi intelligent agent reliabilities are recorded for analysis purpose.

Algorithm Intelligent_Agent_updater ( )

Step 1: Update request must be generate by the user.

Step 2: Intelligent agents are created to resolve update request.

Step 3: first intelligent agents perform search operation to identify the required file to be updated.

Step 4: count number of user requests and to resolve each user request one intelligent agent created.

Step 5: Reliability of these intelligent agents are calculated and these values are varied from zero to one.

Step 6: if an intelligent agent working without errors then intelligent agent reliability is equal to one.

Step 7: if an intelligent agent is not working then intelligent agent reliability is equal to zero.

Step 8: if the intelligent agent reliability is equal to 0.25 then agent will not continue with requests.

Step 9: if is there any searching faults then reliability are in between 0.25 to 0.85.

Step 10: all multi intelligent agent reliabilities are recorded for analysis purpose.

Algorithm Capture_Intelligent_Agent ( )

Step 1: Network systems are used to capture intelligent agents.

Step 2: if capture request is initiated then call Intelligent_Agent_Search ( ) to search intelligent agent.

Step 3: intelligent agents are creating desktop intelligent agent and as when required it is migrated to remote desktop system to capture.

Step 4: Reliability of these intelligent agents are calculated and these values are varied from zero to one.

Step 5: if an intelligent agent working without errors then intelligent agent reliability is equal to one.

Step 6: if an intelligent agent is not working then intelligent agent reliability is equal to zero.

Step 7: if the intelligent agent reliability is equal to 0.25 then agent will not continue with requests.

Step 8: if is there any searching faults then reliability are in between 0.25 to 0.85.

Step 9: all multi intelligent agent reliabilities are recorded for analysis purpose.

Algorithm Intelligent_Agent_Path( )

Step 1: To resolve user request using multi intelligent agent required shortest path to perform user tasks.

Step 2: each intelligent agent follow dijkstra method to establish route from source to destination.

Step 3: same dijkstra technique is used to reroute when an intelligent agent migrating.

Step 4: Reliability of these intelligent agents are calculated and these values are varied from zero to one.

Step 5: if an intelligent agent working without errors then intelligent agent reliability is equal to one.

Step 6: if an intelligent agent is not working then intelligent agent reliability is equal to zero.

Step 7: if the intelligent agent reliability is equal to 0.25 then agent will not continue with requests.

Step 8: if is there any searching faults then reliability are in between 0.25 to 0.85.

Step 9: all multi intelligent agent reliabilities are recorded for analysis purpose.

IV. Performance Evaluation

Total time of the computing environment includes time required to create intelligent agent, time required to communicate, and time required to perform task.

$$T_{IA} = T_{CR} + T_{CM} + T_T$$

Where T_{IA} is a total time required to complete user request by intelligent agent, T_{CR} is time required to create an intelligent agent, T_{CM} is time required to communicate, and T_T is time required to complete task. In a communication network, if agents are not used for completing user requests then complexity may automatically increase. So, multi intelligent agent model is used to improve overall reliability of a distributed network. In table 1 shows how much time taken by each intelligent agent to complete allotted task. From the experiments as multiple

intelligent agents are increased, they are all working in a parallel manner to accomplish tasks, and reliability of proposed system is increased. Reliability analysis of multi intelligent search is depicted in table 2. Different types of intelligent agents reliability is noted and values are plotted in figure 2. Round trip time of each multi intelligent agents are calculated and a graph is plotted and it is shown in figure 3.

Table 1. Total time taken by intelligent agents

S.No.	No. of IA	NOP	Time (ms)
1	3	3	15982
2	6	6	17952
3	9	9	23984
4	12	12	27945
5	15	15	33969
6	18	18	38891
7	21	21	41897
8	24	24	53896
9	27	27	55234
10	30	30	62534

Table 2. Intelligent search agent's reliability analysis

S.No.	No. of IA	Reliability
1	1	0
2	2	0.21
3	3	0.785
4	4	0.912
5	5	0.99

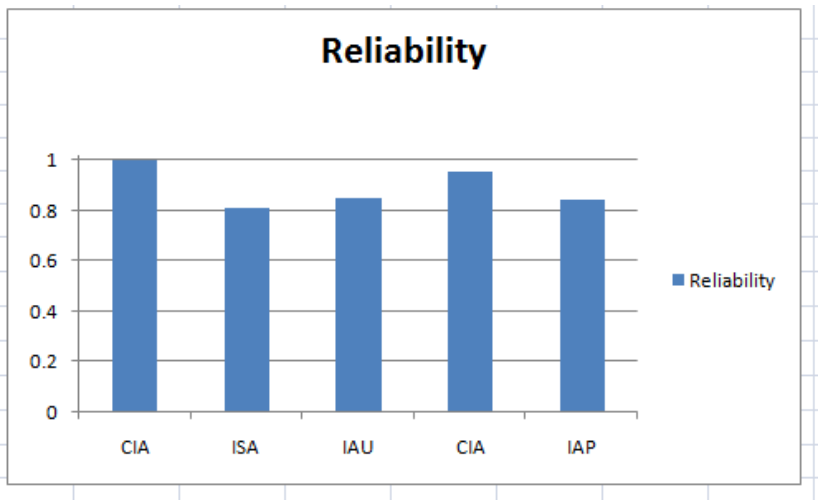

Figure 2: Intelligent search agent's reliability analysis

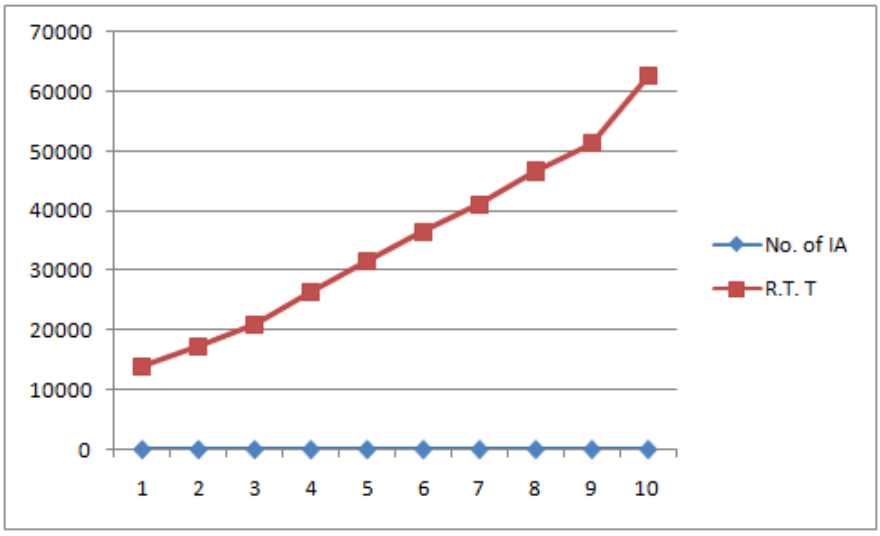

Figure 3: Number of Intelligent agents Vs. R.T.T

Proposed multi intelligent agent based distributed communication network system shows better performance and reliability over traditional networks. If the system is build with this model, it is working in mixed mode execution model, and it will execute more number of tasks within stipulated time period without errors.

IV. Conclusion

At present distribute networks are used in number of applications. Conventional networks are not in a position to balance load, shows less performance, and also reliability is less. To overcome this problem, we proposed multi intelligent

agent model, here five different types of intelligent agents are created to solve user requests, and all these intelligent agents are working in a parallel manner. We simulated distributed network using intelligent agents, we observed performance of multi intelligent agent model, reliability of agents, and number of tasks executed by agents, how agents are migrating from one node to other, how shortest path is established between intelligent agents, and how user requests are resolved by these agents. Performances of a multi intelligent agent model are high when compared with conventional networks.

References

K. Chen and Bo Liu, "Reliability evaluation model for grid node using fuzzy logic," Ji Suan Ji Yu Xian Dai Hua, vol. 217, no. 9, pp. 27–34, 2019.

W. Tian, "Reliability and operation model analysis of distributed computer network structure," Computer Knowledge and Technology, vol. 10, no. 35, pp. 8390-8391, 2014.

W. Jing, Z. Wu, and H. Liu, "Multiole DAGs dynamic workflow reliability scheduling algorithm in a cloud computing system," Journal of Xidian University, vol. 43, no. 2, pp. 83–89, 2016.

Y. Zheng, "Problems and challenges of software reliabilities on distributed systems," Computer and Information Technology, vol. 22, no. 6, pp. 29–32, 2014.

C. Yang, J. Xiong, and L. I. Dong, "Research on reliability model for distributed system," Computer Engineering, vol. 38, no. 3, pp. 51–53, 2017.

F. Yang and Y. Shi, "Research of evaluation algorithm for operational reliability of power grid based on distributed computing," Industry and Mine Automation, vol. 40, no. 8, pp. 22–25, 2014.

J. Feng, X. Tang, and W. Wang, "Reliability verification method of numerical simulation based on grid independence and time independence," Journal of Shihezi University:Natural Science, vol. 35, no. 1, pp. 52–56, 2017.

C. Yan, Z. Hu, and Li Xi, "Reliability-cost optimization scheduling model and algorithm in grid," Computer Science, vol. 40, no. 3, pp. 136–150, 2013.

C. Yan, Z. Hu, and L Xi, "Reliable scheduling algorithm for time-constrained grid workflow," Journal of Chinese Computer Systems, vol. 33, no. 7, pp. 1528–1532, 2012.

H. Zhu, L Ting, and J. Yan, "Research on a distributed task scheduling algorithm based on dynamic load balancing," High Tech Communication, vol. 24, no. 12, pp. 1261–1269, 2014.

D. Dong, Study on Prediction and Early Warning of Gas Concentration Based on Measured Data of Safety Monitoring System, Xi'an University of science and technology, Xi'an, China, 2012.

D. Dong, S. Qu, and H. Wang, "Gas concentration early warning method based on correlation analysis of monitoring data," Industry and Mine Automation, vol. 41, no. 6, pp. 1–4, 2015.

D. Dong, S. Qu, and H. Wang, "Characteristics analysis and pretreatment of mine gas monitoring data," Industry and Mine Automation, vol. 41, no. 9, pp. 1–4, 2015.

W. Deng, "Improved reliability dynamic level for grid scheduling algorithm," Computer & Digital Engineering, vol. 42, no. 6, pp. 927–929, 2014.

Y. Zhang, Y. Liu, and Y. Chen, "Research on grid computing and its reliability," Journal of Mudanjiang University, vol. 19, no. 2, pp. 102–106, 2018.

S. Bawa, "Modified Min-Min Heuristic for Job Scheduling Based on Qos in Grid Environment," in Proceedings of the 2013 2nd International Information Management in the Knowledge Economy International Conference, pp. 166–171, IEEE, Chandigarh, India, December 2013.