IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-6, Issue-1 E-ISSN: 2347-2693

Efficient Resource Allocation Algorithm in Dependable Distributed Computing Systems Using A Colony Optimization

Manas Kumar Yogi[1], G. Kumari[2], L. Yamuna[3]

[1]Computer Science & Engineering, Pragati Engineering College, Surampalem, Kakinada, India
[2] Computer Science & Engineering,Pragati Engineering College,Surampalem,Kakinada, India
[3] Computer Science & Engineering,Pragati Engineering College,Surampalem,Kakinada, India

*Corresponding Author: manas.yogi@gmail.com , Tel.: 09966979279

Available online at: www.ijcseonline.org

Received: 07/Dec/2017, Revised: 20/Dec/2017, Accepted: 10/Jan/2018, Published: 31/Jan/2018

***Abstract*—** In this paper we present efficient resource allocation algorithm which give rise to economic models for job scheduling in distributed computing environmental. Existing schemes which schedule jobs in such a environment have their routes in searching of time slots in resource occupancy schedules which consider only the time slot sets. Our algorithm proposes a hybrid time slot search algorithm and configures each job in an efficient schedule.

***Index Terms*—** Distributed, Reliability System, Ant Colony Optimization, Multi Criteria Decision

I. INTRODUCTION

Distributed computing is a model in which components of software systems are shared among multiple computers to improve efficiency and performance. The configuration of a distributed computing system involves a set of cooperating processors communicating over the communication links to maximize the outcomes and to reduce the consumption of time the process is allocated to nearby processor for fast response by using ACO. This will decreases the fault tolerance by allocating resources. distributed computing architecture consists of a number of client machines with very lightweight software agents installed with one or more distributed computing management servers (DCMS) the agents running on the client machine usually checks for the idle system and send notification to the management server that the system is not in use and available for processing job .the agents then requests an application software packages .whenever the client receives the package from DCMS it start processing when cpu cycles are free and send the result back to the DCMS when user returns the resource is allocated to some other agents like food is completed ant chooses other location in the same if same user requires resource once again then it again checks for the availability of the other resources and process the Information for acknowledgement but it does not checks for optimal path this is solved by choosing most effective ANT COLONY OPTIMIZATION ALGORITHM(ACO).

Among others, system cost and reliability are two of the most concerned objectives to improve the performance of the distributed system .In this execution cost place major role from its allocated processor. This is done by the ACO as described below in order to provide solution.

The following is the in detail regarding the problem. The task is to process allocation that minimizes the system cost and maximizes the system reliability which leads to DRS(Distributed reliability System) for better understanding network topology is render it uses the graph to indicate the processor interaction graph (PIG).

P-processor

L-communication links

G(P,L)

$P= \{p_i\}_{i=1,2,3,4\ldots.n}$

$L= \{LF_i\}_{i=1,2,3,\ldots.n}$

For better resources allocation mesh topology is used to indicate the graph .The nodes represent communication links.

II. PRINCIPLE

Consider a job j_i which has to be scheduled on a CPU p_i whenever p_i accepts j_i its load compared to previous processing load increases. We introduced a load factor l_f which indicates amount of load present currently on a processor. The following illustration expands the principle involved.

Load Factor	Processor	Jobs
Lf_1	P_1<----------	J_1
Lf_2	P_2<----------	J_2
Lf_3	P_3<----------	J_3
Lf_4	P_4<----------	J_4

In case after time t,j_2,j_3 are also assigned to p_1 then Lf_2,Lf_3 becomes Zero and Lf for p_1 is now $Lf_1+Lf_2+Lf_3$.

If we calculate the load density of each processor we can observe that few processors will have huge load and few processors will have no load at all. This decreases the efficiency of the scheduling algorithm we consider an environment where jobs are to homogeneous and processor. Speeds are also not uniform It becomes more challenging scenario. To tackle this challenge, we apply ant-colony optimization which is based on choosing a shortest path from ant nest to a food source depending on a path where phenomenon density is high.

In distributed dependable computing each job be processed fully or partly on a cpu. so, a job which is computationally intensive can be scheduled to execute on multiple processors based load factor of the processors. Now, the illustration is based on ACO mechanism.

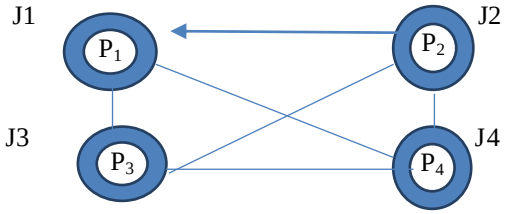

Figure 1.Allocation of jobs to processors

As shown above ,all processors can communicate with each other and 4 jobs J_1,J_2,J_3,J_4 are currently assigned to 4 CPU's p_1,p_2,p_3,p_4.

In our technique we express the probability of choosing the next processor j from current processor i as ,

$$P_{ij}^{k}(t)=T^{\alpha}_{ij}\,\eta^{\beta}_{ij}/\Sigma_k T^{\alpha}_{ij}\,\eta^{\beta}_{ij} \quad (1)$$

Where t_{ij} denotes the load density along the edge n_{ij} denotes the heuristic information like free slots available on a processor currently Once ll jobs are scheduled on all processors, the load density a teach processor can be updated as

$$T_{ij}(t+1)=(1-P)T_{ij}(t)+ T_{ij} \quad (2)$$

Here, P denotes the rate at which the cpu completes a job, to avoid accumulation of jobs at a single processor,Where L_k is the total length of a Job J_i so, if a job is not scheduled on many processors before its completion we infer that it's a less computationally intensive job and such jobs can be scheduled on processors with low speed .on the contrary ,if a job j_i is unable to execute fully even after scheduling on all available CPU's in one our (or) cycle we infer that it should be scheduled first on the cpu with highest speed .It is obvious that job switching from one processor to another is costly .so there should be a mechanism to observe that jobs should be transformed from one processor to another with minimum cost .In ACO the pheromone density determines which path should be selected by an ant to reach the food source. In our proposed mechanism a job has to be scheduled onto a processor which has the right mix of two factors .One factor is amount of time slot available and second factor is amount of time needed by the job to complete execution. So, if amount of time needed to complete execution is greater than amount of timeslot available, we may bypass the scheduling of that job to a processor in the current cycle.

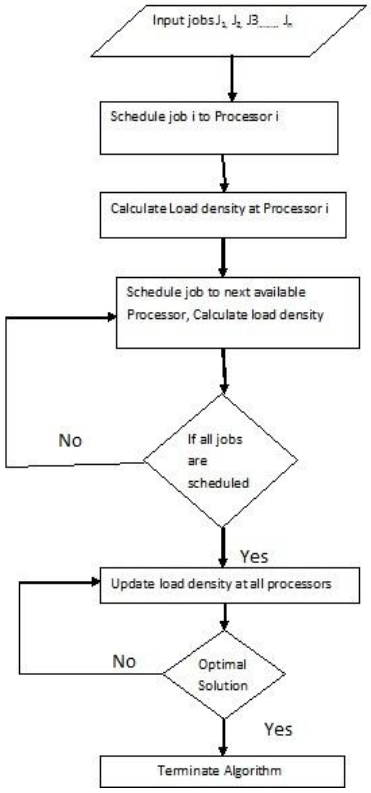

Figure 2 . Flowchart for proposed mechanism

III. THREATS TO PROPOSED MODEL

1. Not every time each job is computationally expensive for input output operation. Some cases involve input of a job to be given by output of previous job. In such cases serializability becomes a threat. Consistency of system will be affected adversely in such situations.
2. Secondly, CPU speeds of all machines are practically unequal in distributed computing environment more so if its an dependable computing environment. In such environment each cluster has multiple processors with different processing speeds. But for our technique we assumed equal CPU speed in a cluster.
3. The final threat to validity for our proposed mechanism is we did not give much importance to the density of interactions between the processors. Suppose out of 3 CPU's P_1, P_2, P_3 both P_2, P_3

interact with P_1 then P_1 has highest density of interaction, there by contributing to its load density also. We have not attributed its significance assuming such interactions do not heavily add to load density. But in a real time scenario this cannot be ruled out.

We assume all processors speed is equal for theoretical considerations. The following flowchart represents the proposed technique concisely.

In above flowchart optimal solution is the scheduling of all jobs on every available processor so that load density of all processors P1, P_2, P_3,…… P_n are almost similar there by producing a system throughput at acceptable limit.

IV. CONCLUSION

Quality of service in dependable computer service has become an indispensable part of a schedule design systems. So the need of efficient scheduling mechanism is developed. We have proposed a hybrid mechanism for time slot search. This approach features searching for multiple job executions considering ant colony optimization.

V. FUTURE WORK

We would like to executed the current mechanism with multi criteria decision making algorithm as the complexity of the dependable computing architecture increases job scheduling efficiency arises as a challenge to the system designers .In future we intent to put our scenario efforts in the concerned direction so as to enhance the through put of all the involving elements in a dependable computing system.

REFERENCES

[1].Zuberek," W.M.: Checking compatibility and substitutability of software components.In: Models and Methodology of System Dependability", ch.14, pp. 175–186. Oficyna Wydawnicza Politechniki Wroclawskiej, Wroclaw (2010).

[2].A. Gaspar-Cunha, R. Takahashi, Schaefer, and L. Costa (Eds.) "Soft Computing in Industrial Applications", 2011 ISBN 978-3-642-20504-0,Vol. 96.

[3]. W. Zamojski, J. Kacprzyk,J. Mazurkiewicz, J. Sugier, and T. Walkowiak (Eds.) Dependable Computer Systems, 2011 ISBN 978-3-642-21392-2, Vol. 97.

[4].Wojciech Zamojski, Janusz Kacprzyk, Jacek Mazurkiewicz, Jarosław Sugier, and Tomasz Walkowiak (Eds.) "Dependable Computer Systems Advances in Intelligent and Soft Computing ".ISSN 1867-5662 Library of Congress Control Number: 2011928737 _c 2011 Springer-Verlag Berlin Heidelberg.

[5].Y. Adil, , A. A. Hanan, and A. A. Atahar, "A bidding-based grid resource selection algorithm using single reservation mechanism", Int. J. Comp. Appl., vol. 16, no. 4, pp. 39-43, 2011.

[6] S. Dawei, C. Guiran, J. Lizhong and W. Xingwei, "optimizing grid resource allocation by combining fuzzy clustering with application preference", Int Conf. on Advanced Computer Control (ICACC), pp: 22-27, 2010.

[7] G. Tibor, "A resource allocation protocol for providing quality of service in grid computing, using a policy-based approach", AICT-ICIW '06 Proc. of the Advanced Int'l Conf. on Telecommunications and Int'l Conf. on Internet and Web Applications and Services, IEEE Computer Society Washington, DC, USA, 2006.

[8] S., Zikos, and H. D. Karatza, "Resource allocation strategies in a 2-level hierarchical grid system", Proc. of the 41st Annual Simulation Symp. (ANSS), April 13–16, IEEE Computer Society Press, SCS, pp. 157–164, 2008.

[9] Y. ZHU, "A survey on grid scheduling systems", Tech. Report, Department of Computer Science, Hong Kong University of Science and Technology, 2003.

[10] R. Buyya, "A grid simulation toolkit for resource modelling and application scheduling for parallel and distributed computing".

[11] J. Balasangameshwara, and N. Raju, "A decentralized recent neighbour load balancing algorithm for computational grid", Int. J. of ACM Jordan, vol. 1, no. 3, pp. 128-133, 2010.

[12]. C. W. Tan, D. P. Palomar, and M. Chiang, "Distributed Optimization of Coupled Systems With Applications to Network Utility Maximization," in Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, Toulouse, France, 2006.

[13].C. W. Tan, D. P. Palomar, and M. Chiang, "Distributed Optimization of Coupled Systems With Applications to Network Utility Maximization,"in Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, Toulouse, France, 2006.

[14].Bhardwaj, D., Jain, S. and Singh, M.P., "Estimation of network reliability for a fully connected network with unreliable nodes and unreliable edges using neuro optimization", International Journal of Engineering-Transactions A: Basics, Vol. 22, No. 4, (2009), 317.

Authors Profile

Mr. Manas Kumar Yogi pursued Bachelor of Technology from VR Siddhartha Engineering College,Vijayawada,A.P. in 2006 and Master of Technology From Malla Reddy College Of Engineering And Technology in year 2012. He is currently working as Assistant Professor in Department of Computer Science Engineering , Pragati Engineering College (Autonomous), Surampalem, East Godavari District, since 2014. He is a member of IEEE & ACM since 2014. He has published more than 40 review,research papers in reputed international journals ,conferences including IETE sponsered conferences. His main research work focuses on Software Engineering, Distributed Computing, Cloud Security and Privacy, Big Data Analytics, , IoT and Computational Intelligence based optimisations. He has 8 years of teaching experience and 2 years of software industry Experience.

Mrs.G.Kumari pursued Bachelor of Technology from Aditya Institute of Technology and Management, JNTUH in year 2005, and Master of Technology from Godavari Institute of Engineering & Technology, JNTUK Kakinada in the year 2013. She is Currently working as Assistant Professor in Pragati Engineering College,Surampalem, kakinada since 2011. She is a member of ACM since 2015. She has published 10 research papers in reputed international journals and conferences. Her main research work focuses on Cloud Security

and Privacy, Big Data Analytics and Machine Learning.She has 10 years of teaching experience.

Miss *L.Yamuna* pursued Bachelor of Technology from Aditya Engineering College under JNTUK in year 2013 and Master of Technology from V.S Lakshmi Engineering college under JNTUK in year 2016.Currently working as Assistant Professor in Pragati Engineering College (Autonomous), Surampalem, East Godavari District, since 2016 .She has published 6 research paper and one conference paper.

