Journal of Computer Science 8 (6): 978-982, 2012
ISSN 1549-3636

A Fault Tolerant Resource Allocation Architecture for Mobile Grid

[1]Thenmozhi, S., [2]A. Tamilarasi and [3]P.T. Vanathi
[1]Department of Computer Applications,
Chettinad College of Engineering and Technology, Karur, Tamilnadu, India
[2]Department of Computer Applications,
Kongu Engineering College, Perundurai, Tamilnadu, India
[3]Department of Electronics and Communication Engineering,
PSG College of Technology, Coimbatore, India

Abstract: Problem statement: In order to achieve high level of reliability and availability, the grid infrastructure should be fault tolerant. Since the failure of resources affects job execution fatally, fault tolerance service is essential to satisfy QoS requirement in grid computing with respect to mobile nodes. **Approach:** We propose a fault tolerant technique for improving reliability in mobile grid environment considering the node mobility. The Cluster head and monitoring agent was designed in such a way it addresses both resource and network failure and present recovery techniques for overcoming the faults. **Results:** The proposed model achieves a identifiable performance when compared to the previous model (HRAA). By simulation results, we analyze the node and link failures on parameters such as delivery ratio, throughput and delay against the rate of success. **Conclusion:** The proposed fault tolerant approach checks for availability of the nodes with least work load for transferring the executed job to cluster head providing an alternate path in case of failure thereby enhancing the reliability of the grid environment.

Key words: Grid computing, mobility, fault tolerant, multi-tree, resource allocation

INTRODUCTION

Following the grid and mobile computing, mobile grid is the successor of the Grid along with extra characteristics in order to maintain the mobile users and resources in a perfect, evident, secure and self-organized manner (Litke *et al*., 2004; Farooq *et al*., 2006). It entertains the organization of basic ad hoc networks and offers self-configuring grid systems of mobile resources linked through wireless links and creating random topologies (Nandagopal and Uthariaraj, 2010; Senthilnathan and Purusothaman, 2012). The fault may occur due to the distributed system or computing resources or time (Townend and Xu, 2003). Fault tolerance is the ability of a system to deliver its function correctly even in the presence of internal faults in order to increase the system reliability (Garg and Singh, 2011). The distributed system fault may be due to failure in service, process, media and network. The computing resource fault may be due to omission of nodes, response failure and crash failure. The Timing failure may be permanent, intermittent or transient. To overcome these failures fault tolerant techniques has to be employed. In this study, we propose a fault tolerant approach for improving reliability in grid environment.

Related work: Nandagopal and Uthariaraj (2010) proposed the fault tolerant scheduling techique for grid environment. Their technique upholds the records of resource faults within the fault index manager. By utilizing the recorded resource fault data, the check point manager adjusts the distinct intensity of check pointing prior to job scheduling. The checkpoint files existence is enhanced via replication. Their approach does not embed the scheduling strategy into real world grid computing environments.

Jiang *et al*. (2010) proposed a security-aware parallel and independent job scheduling algorithm based on adaptive job replications to make sure the job scheduling decision secure, reliable and fault tolerant. In risky and failure-prone grids, the replication number is changed according to the current security conditions and the end-user settings.

Kandaswamy *et al*. (2008) proposed a fault tolerance and recovery technique for scientific workflows in grid environment. For accompolishing the workflows in consistent way, they utilized over-provisioning and migration approach. Also in order to estimate the improved fault tolerant techique, they considered application performance models, network

Corresponding Author: Thenmozhi, S., Department of Computer Applications, Chettinad College of Engineering and Technology, Karur, Tamilnadu, India Tel: +91(0)4324 250940

latency and bandwidth, batch-queue wait time, user specified deadline and success probability of applications, resource reliability models and availability of a few core grid services.

Khanli *et al*. (2010) proposed a genetic algorithm in computational grid. Their technique utilizes the resource Fault Occurrence History (RFOH) for scheduling job in consistent manner. RFOH is sustained in the Grid Information Server (GIS). Their technique also enhances the job execution percentage within the particular deadline.

Problem statement and system architecture: In study (Thenmozhi and Tamilarasi, 2010), we proposed Hierarchical Resource Allocation Architecture (HRAA) which includes resource monitoring and scheduling operations for mobile grid. In this architecture, the Mobile Grid is divided into clusters. Each cluster has one Cluster Head (CH). A Master Server (MS) controls each local clusters and has frequent updates of all the CH information. Each CH has a Monitoring Agent (MA) which will periodically predict the mobility of the cluster nodes and monitor the resource availability and update their values. When the MS forwards the job request of a user to the ideal CH, the CH schedules the jobs based on the predicted time for resource availability and sufficiency of the resources. This study addresses the issue of fault tolerance which is required for effectiveness of the system.

MATERIALS AND METHODS

USER 1, USER 2, USER 3… USERn: The users who submits job for getting done which consists of specific nodes of the fixed grid.

- MS: MS represents Master Server which acts like a access point and schedules the job for different clusters
- MA: MA represents the monitoring agent embedded with the cluster head for execution of jobs by different nodes
- CH1, CH2, CH3…CHn: CH represents Cluster head selected among various mobile nodes and manages the mobile group
- n0, n1,….. nn: Mobile nodes used as service provider or service recipients

The proposed model helps to tolerate the fault that occur at the mobile node level or cluster head level.

Functioning model: Let n_0, n1, n2 and n3 are nodes in the given cluster and Ack0, Ack1, Ack2, Ack3 are acknowledgement values from corresponding nodes. Let CH be the corresponding cluster head. JE represents the job been executed in the resource. MA represents the monitoring agent of the particular cluster:

- T_1 be time taken for the acknowledgement to reach monitoring agent from the nodes
- T_2 be time taken for the executed job to reach CH1 from the nodes

The specified time interval is based on the resources speed S_r, communication latency L_c and queue length Q_l of the resource.

Case 1:

Resource failure notice: The job is assigned to every node in the cluster.

Within time T_1, the acknowledgment about the job completion should reach MA.

In case MA does not receive any acknowledgement, it decides that there is some resource failure.

Fault recovery: After notification of resource failure, the monitoring agent informs its corresponding cluster head regarding the failure and the cluster head can choose alternative nodes in same cluster or cluster head can reschedule the job through master server.

From Fig. 1, the fault notice and recovery is as follows.

After job assignment, when the Ack 1, Ack 2 and Ack 3 are received by the monitoring agent and if Ack0 is not received, this result in some resource failure in n_0 and CH1 try to schedule job to any other node from the same cluster. If it is not feasible, CH1 will reschedule the job through MS.

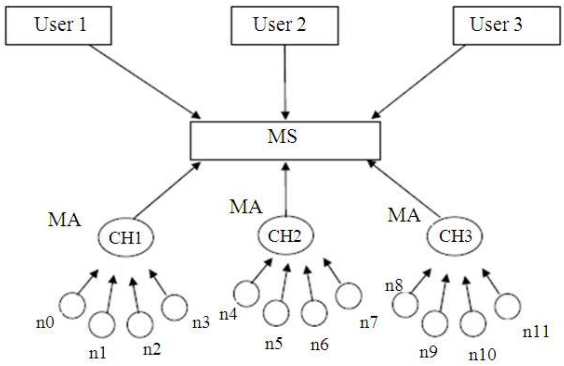

Fig. 1: Model of HRAA

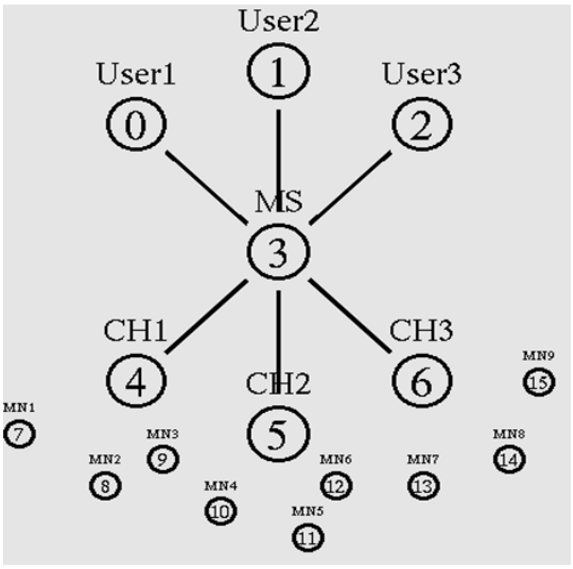

Fig. 2: Simulation setup

Table 1: Simulation settings

Mobile nodes	9
Users	3
Clusters	3
Area size	1000×1000
Mac	802.11
Radio range	250 m
Routing protocol	DSDV
Simulation time	50 sec
Traffic source	CBR
Packet size	512
Rate	250 kb, 500 kb,….1000 kb
Speed	5 m sec^{-1}
Transmit power	0.660 w
Receiving power	0.395 w
Idle power	0.335 w
Initial energy	10.1 J

Case 2:

Network failure notice: The status of JE must reach cluster heads after specified time interval T2. If not, it results in network failure.

Fault recovery: Upon failure notice, monitoring agent checks for the other nodes with least work load within the cluster. The executed job is copied to such nodes and it is possible for transferring executed job to cluster head, thus finding the alternate path.

In case all nodes within the cluster are highly loaded, then monitoring agent in neighbouring clusters checks for availability of the nodes with least work load. Further, executed job is copied to such nodes in neighbouring clusters and it is destined to clustering head, thus giving an alternate path.

From Fig. 1, the network failure notice and recovery is as follows.

The acknowledgement from node 0, node 1 and node 2 reaches the MA and JE of corresponding nodes reaches CH_1 but JE of node 3 does not destine to CH_1 which will result in the network failure. Upon fault notification, the CH chooses other alternate path.

The two events are involved in finding the alternate paths.

If n_3 fails to destine JE to CH1, then monitoring agent checks for the other nodes with least work load within the cluster. JE is copied to such nodes and it is possible for transferring JE to CH1, thus finding the alternate path.

In case all nodes within the cluster are highly loaded, then monitoring agent in neighbouring clusters cl_n checks for availability of the nodes with least work load. Further, JE is copied to such nodes in cl_n and it is destined to CH1, thus giving an alternate path.

Simulation results:

Simulation parameters and settings: Here, we examine the performance of our Hierarchical Resource Allocation Architecture (HRAA) with an extensive simulation study based upon Network simulator, version 2.36. The simulation topology is given in Fig. 2.

We compare our results with our previous Hierarchical Resource Allocation Architecture (HRAA) (Thenmozhi and Tamilarasi, 2010). Various simulation parameters are given in Table 1.

Performance metrics: In our experiments, we measure the following metrics.

- Average execution delay: It measures the average delay occurred while executing a given task
- Average success ratio: It is the ratio of the number of tasks executed successfully and the total number of tasks submitted
- Throughput: It is the number of tasks finished successfully

RESULTS AND DISCUSSION

To implement we have selected NS2 after anlalysing different implementation tools such as Glomosim, Mobigrid, gridftp. NS2 is a event simulator which provides a platform for simulation, visualization, emulation and scaling. The proposed and implemented system works properly in Linux platform.

In this experiment, we vary the execution rate as 250, 500, 750 and 1000 Kb.

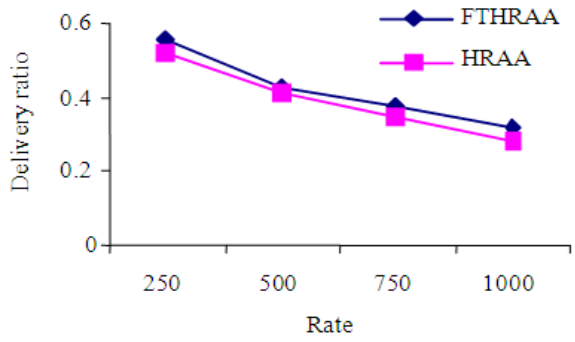

Fig. 3: Rate Vs delivery ratio

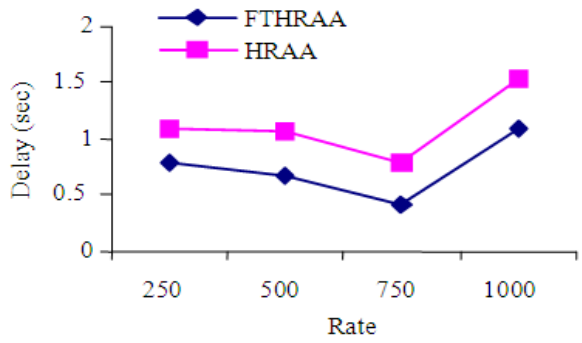

Fig. 4: Rate Vs delay

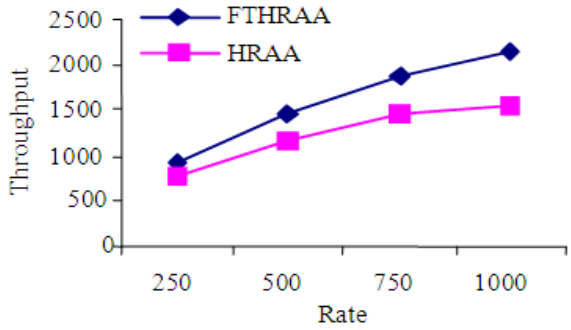

Fig. 5: Rate Vs throughput

Based on node failure: In this scenario, node failure is introduced in cluster 1. Nodes MN1, MN2 and MN3 are executing the job request of user1. After execution, MN1 and MN3 return the ACK indicating the job execution is over. MN3 which does not send ACK within that time interval, assumed to be failed. Under this scenario, we vary the execution rate as 250, 500, 750 and 1000 Kb and measure the performance of both the schemes.

Figure 3 shows that when the Delivery rate is increased then the success ratio gets decreased. Also, we can see that the FTHRAA achieves good success ratio, compared to HRAA.

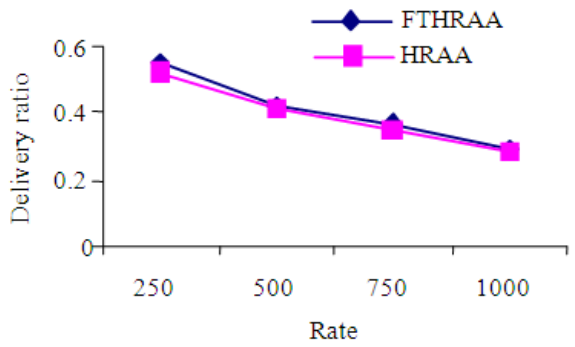

Fig. 6: Rate Vs delivery ratio

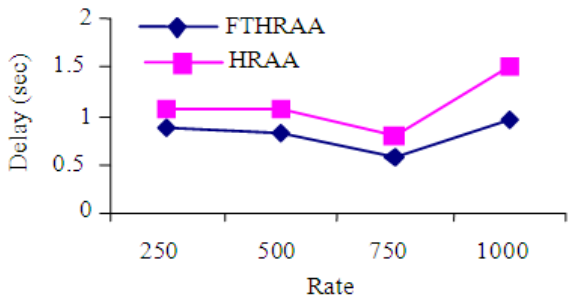

Fig. 7: Rate Vs delay

From Fig. 4, it is clear that when the execution rate is increased then the delay also increases. We can see that the average execution delay of the proposed FTHRAA algorithm is less when compared to the HRAA when the rate is increased.

Figure 5 shows that when the rate is increased the throughput also increased. When we compare the FTHRAA with HRAA the FTHRAA has high throughput than the HRAA.

Based on link failure: In this scenario, link failure is introduced in cluster 1. Nodes MN1, MN2 and MN3 are executing the job request of user1. After execution, MN1, MN2 and MN3 return the ACK indicating the job execution is over. Then the cluster head CH1 receives the completed job from MN1 and MN2. MN3 which is not able to send the executed job within that time interval, because of the link between MN3 and CH1 is failed. Under this scenario, we vary the execution rate as 250Kb, 500Kb, 750Kb and 1000Kb and measure the performance of both the schemes.

Figure 6 shows that when the Delivery rate is increased then the success ratio gets decreased. From the Fig. 6, we can see that the FTHRAA achieves good success ratio, compared to HRAA.

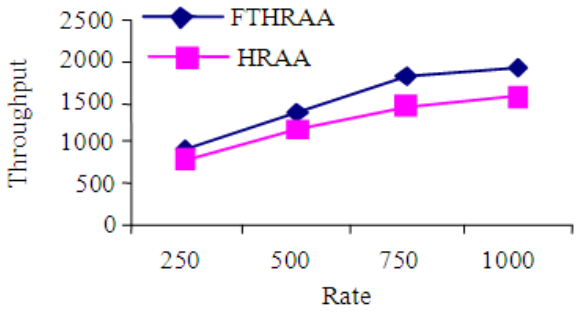

Fig. 8: Rate Vs throughput

From Fig. 7, it is clear that when the execution rate is increased then the delay also increases. We can see that the average execution delay of the proposed FTHRAA algorithm is less when compared to the HRAA when the rate is increased.

Figure 8 shows that when the rate is increased the throughput also increased. When we compare the FTHRAA with HRAA the FTHRAA has high throughput than the HRAA.

CONCLUSION

In this study, we have developed a fault tolerant technique to improve reliability in mobile grid environment. The proposed technique handles both resource and network failure and present recovery techniques also. When the job is assigned to the node, acknowledgement must reach monitoring agent within certain time interval else it results in some resource failure. Hence cluster head try to schedule job to any other node from the same cluster or cluster head will reschedule the job through master server. When executed job does not reach cluster heads after specified time interval it results in network failure. To overcome this fault, the monitoring agent checks for the other nodes within the cluster or in neighbouring clusters cl_n checks for availability of the nodes with least work load for transferring the executed job to cluster head thus giving an alternate path. By simulation results, we have shown the proposed fault tolerant approach enhances the reliability of the grid environment.

REFERENCES

Farooq, U., S. Mahfooz and W. Khalil, 2006. An efficient resource prediction model for mobile grid environments. Punjab University College of Information Technology Lahore, Pakistan.

Garg, R. and A.K. Singh, 2011. Fault tolerance in grid computing state of the art and open issues. Int. J. Comput. Sci. Eng. Survey, 2: 88-97. DOI: 10.5121/ijcses.2011.2107

Jiang, C., X. Xu and J. Wan, 2010. Replication based job scheduling in grids with security assurance. Proceedings of the 3rd International Symposium on Electronic Commerce and Security Workshops, Jul. 29-31, Guangzhou, P. R. China, pp: 156-159.

Kandaswamy, G., A. Mandal and D.A. Reed, 2008. Fault tolerance and recovery of scientific workflows on computational grids. Proceedings of the 8th IEEE Conference Symposium on Cluster Computing and the Grid, May 19-22, IEEE Xplore Press, Lyon, pp: 777-782. DOI: 10.1109/CCGRID.2008.79

Khanli, L.M., M.E. Far and A. Ghaffari, 2010. Reliable job scheduler using RFOH in grid computing. J. Emerg. Trends Comput. Inform. Sci., 1: 43-47.

Litke, A., D. Skoutas and T. Varvarigou, 2004. Mobile grid computing: Changes and challenges of resource management in a μobile grid environment. National Technical University of Athens, Greece.

Nandagopal, M. and V.R. Uthariaraj, 2010. Fault tolerant scheduling strategy for computational grid environment. Int. J. Eng. Sci. Technol., 2: 4361-4372.

Senthilnathan, T. and T. Purusothaman, 2012. A multi-agent based data replication mechanism for mobile grid. Am. J. Applied Sci., 9: 542-552. DOI: 10.3844/ajassp.2012.542.552

Thenmozhi S. and A. Tamilarasi, 2010. A hierarchical resource allocation architecture for mobile grid environments. Int. J. Comput. Sci. Inform. Security, 8: 6-11.

Townend, P. and J. Xu, 2003. Fault tolerance within a grid environment. Proceedings of the UK e-Science All Hands Meeting, (UKSAHM' 03), The Pennsylvania State University.