ARPN Journal of Engineering and Applied Sciences

AN ADAPTIVE FAULT TOLERANCE MECHANISM IN GRID COMPUTING

P. S. Nithyadharshini[1], M. Pounambal[2], D. Nagaraju[3] and V. Saritha[4]
[1]SCSE, VIT University, Chennai, Tamil Nadu, India
[2]SITE. VIT University, Vellore, Tamil Nadu, India
[3]Department of Computer Science Engineering, SSJ Engineering College, Hyderabad, India
[4]Department of Computer Science Engineering, Sri Vidyanikethan Engineering College, Tirupathi, Andhra Pradesh, India
E-Mail: pounambal@rediffmail.com

ABSTRACT

Grid computing environment consist group of heterogeneous resources which provides different services to the user. Grid computing applications require lot of computation power and will perform longer tasks. Present grid computing system has a shape which is inflexible in terms of setting up an ad hoc component. To overcome this herewe proposed an approach introducing Agents with Gridsim simulation toolkit (Grid computing environment).Computational grid has objectives that are to coordinate, share, and distribute different resources to work on one computational job. Gridsim tool kit has the flexibility and extensibility. Agent is hidden in the computer system that is situated in grid environment. This paper improves performance of 75% users and increase transfer rate in grid information service.

Keywords: grid computing, fault tolerance, gridlets, agents, bokers.

1. INTRODUCTION

The objective of computational grid is to coordinate, divide, and distribute different resources to work on a single computational job [1-3]. It is used by few computational models such a mathematical model in the solution of computational problems. Grid information service is very important in the computational grid, because of the resource registration. Grid computing environment consist of group of different resources which provides services to the user. Grid applications [4] require lot of computation power and will perform long tasks which may run for many days or months on different boundaries and heterogeneous nodes require powerful mechanism to handle failures such as dynamic nature, physical damage, accidentally user shutdown node, network complexity. In this we consider the partial failures. Detecting the root cause of these problems in a massive environment is very difficult [15]. In Grid environment, if failure occurs at the time of execution other nodes may not be active because this is the drawback in grid system. We overcome the problem based on the gridsim because we use the multiple regional networks. Now day's heterogeneous computing environment very important in worldwide (organizations). Heterogeneous computing consists of grid, cloud, utility computing. Agent is encapsulated computer system that is situated in few environments, i.e., capacity of flexible, autonomous action in that environment in sequence to meet its design objectives [5]. Agents are exactly found problem solving entities with good defined boundaries and interfaces. Agents act upon the middleware build by grid information service [6-7]. The grid tools are improved using agent dependent technology. The middleware is the encapsulated complexity of grid. Agents are mainly focus on the community. Agent is based approach to transfer the task. In the proposed model, agents integrated with gridsim for grid environment. Registering agent as service in grid information service is a crucial task. Fault tolerance is a property that actives computer system to perform accurately among the period of failures of few of its component [9]. Fault tolerance is very important in gridsim (grid based) according to failures. There are several Grid projects implemented today, with different objectives, implementation issues, target Applications, and computer infrastructure. A fault tolerance framework for Grids is presented in [4]. Grid computing have the property that is dependability of own resource In the recent years, the tremendous increase in the data storage capacities, processing facilities and information carrying capabilities of the interconnecting networks made Grid Computing is a vital area of research and computing platform of the future [11-14]. It is argued that the notion of an agent as a self-contained problem-solving system capable of autonomous, reactive, proactive, social behavior represents yet another tool for the software engineer. we proposed an approach introducing Agents in the Gridsim simulation toolkit (Grid computing environment).Because present grid computing system have shape or format and inflexible in terms of setup of adhoc component. The co-allocators agents have two simulation factors. First one is providing advance reservation. Second one is reinforcement learning agents. Computational grid has objectives that are to coordinate shared, distributed different resource to work on one computational resource. Gridsim tool kit has the flexibility extensibility. Agent is hided the computer system that is situated in few environment. Grid middleware contains 4 layers first one is applications, grid core middleware, portals layers and user level middleware. A fault tolerance framework for Grids is presented in [16]. It consists of failure detection service (FDS) and a failure handle service (Grid-WFS). The FDS enables the detection of both application crashes and user-defined exceptions. The FDS enables the detection of both application crashes and user-defined exceptions. The Grid-WFS allow users to achieve failure recovery in a variety of ways, depending on requirements and constraints of their applications. Resource management [8] is an important infrastructure in the grid computing environment. In grid computing fault tolerance is very important. Gridsim toolkit easily faces the fault tolerance. A fault tolerance framework for Grids is presented in [10] which is called as MAG (Mobile Agents for Grid Computing Environments) is a research project that explores the mobile agent paradigm as a way to overcome the design and implementation challenges of developing a Grid middleware. In this total service register in grid information service and all resource transfer between client and server. In grid system environment maintain the broker [17-18]. Here Client and server are main players for job submission and job execution. Computational grid must need the hard ware and software infrastructures to give consistent, pervasive and inexpensive access to high end computational capable [8]. The fault tolerance main advantage is network independent and flexible.

2. SYSTEM MODEL

In this agents acts as service and then registered grid information service. Grid environment have client, sever, grid broker. The Figure-1 shows the block diagram to create a grid resource that consists of one or more machines. Each machine contains one or more processing elements. The major module in the architecture is grid broker which acts like middleware.

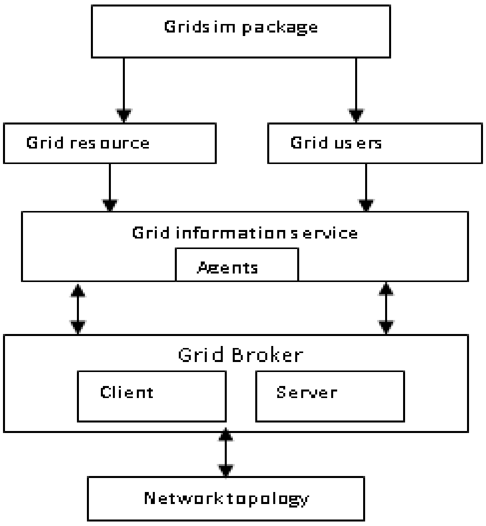

Figure-1. Block diagram of system model.

In this we have taken three machines, each machine contains three processing element. Each resource has number of processes, speed of processing and internal process scheduling policies. Grid user contains one or more gridlets or jobs to be processed. Gridlet contains the job information or data. In this we can consider the three gridlets. And we create one or more grid resource. Each grid resource contains single or many machines and every machine have one or more processing elements. Grid user entities have one or more Gridlets or jobs to be processed. Gridlets have the job description or the data. We create the network topology by connecting the grid user and resource entities. In this aim is to display gridsim ability to simulate adequate size grid test bed. The grid brokers are very important in grid environment. In this service registered in grid information services. It provides comprehensive facility for creating different class of heterogeneous resource that can be aggregated using resource brokers for solving compute and data intensive applications. Gridsim Toolkit is used for Modeling and Scheduling Resources in Distributed Environments. Grid computing has shape or format or rigid.

Issues of planning and implementing:

- Planning agent based approach to execute the job that transfer from one node to other node which is accessible in the grid environment (gridsim).
- In grid environment agents as a service are registered into the Grid Information Service (GIS). By allocating the unique ID by grouping source or destination IP address of the CPU or processing elements and gridsim classes.
- Observing the source or destination machine (processing elements and CPU) and report the information to observing agent routine (software).

3.1 Mathematical model of Service found Performance

Service found performance is very important in the grid environment. The service found parameters are: discovery speed, system efficiency, load balancing, success rate. Discovery speed is depending on the number of routing links, A_1 is the average service discovery speed and R is the total number of requests. N is the total number of links made for the discovery.

$$A_1 = R/N$$

System efficiency service contains links that performs service advertisement and data maintenance. Service advertisement has more workload on the system. Agents handle the request for the particular service will find the discovery processes d and advertising processes A_2. Efficiency of the system is defined as the ratio of total number of requests n for a certain period with the total number of links. E= n/N.

Load balancing load balancing is very important consideration in the grid environment. Here agents are not used for service discovery because no agent can have more discovery workload compared to other agent.

3.1.1 Co-Allocation model

VOL. 13, NO. 7, APRIL 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Algorithm 1: Co-allocation model

```
Inputs: resource agents, regulatory agent, co-allocator agent.
offers or providing list ← ψ
For b=1 ; b ≤ M ; b← b+1 do
P_b← queryidleslot(b) ;
   providing list← providing list U p_b;
end
if providing list = ψ then
Return failures;
end
gouped providing list ← sort (Providing list)_complete time;
good providing ←ψ number(M) processing element←0;
while number (M) processing element< Q do
H_(k, j,T)^(h,m)←min (grouped providing list)_complete time;
good providing list←good providing U {H_(k,j,T)^(h,m)};
numer(M) processing element(PE)←number processing element(PE) + m
grouped providing list← grouped providing – {H_(k,j,T)^(h,m)};
end
while good providing list ≠ψ do
H_(k,j,T)^(h,m)← get (good providing list);
good providing list ← good providing list - {H_(k,j,T)^(h,m)};
if send AR(h,m,T,J_L^P,b)= failed then
request regagt(h,m,T,J_L^P, this agent ID);
end
end
Return successes
```

Providing p_b contains of list of slots. A slot$H_{(k,j,T)}^{(h,m)}$ widows for resource agent b, h is time availability. k contain if no.of processors m at time s. T is duration time (execute job). The resource agent. Sid could follow different policies to generate offers. For instance, a resource agent could generate offers that are more profitable. The regulator agent.is responsible for requesting emergency offers from the different resource agents to meet a sub-jobs requirement that was sanded by a co-allocator agent.

$$\text{Makespan} = \sum_{j=1}^{\max\ jobs} (F_j - E_j)$$

Here F_j, $E_j$are respective the performance start time and the performance finish time of the job i; In order to generate an providing p_b for ajob J_L^P agent b. M is the resource agent. The co-allocators agents have two modules in simulation. First one is providing advance reservation. Second one is reinforcement learning agents. The architecture of the grid broker is given in Figure-2.

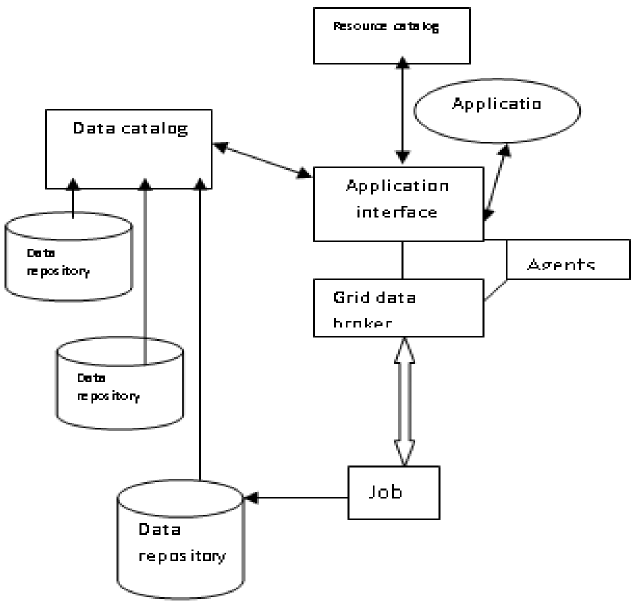

Figure-2. Grid broker.

3.3 Performance of the agent model on each node

The proposed system deals with Gridsim toolkit and agent-based approach to distribute the computation. Figure-3 involves three kinds of participants: Clients, Broker, and servers. In its most basic form, a consumer

ARPN Journal of Engineering and Applied Sciences

decomposes a job into discrete tasks with a unique Agent ID. Each task represents one unit of work that may be performed in parallel with other units of work. Tasks are associated with objects written in the Java™ programming language ("Java objects") that can encapsulate both data and executable code required to complete the task.

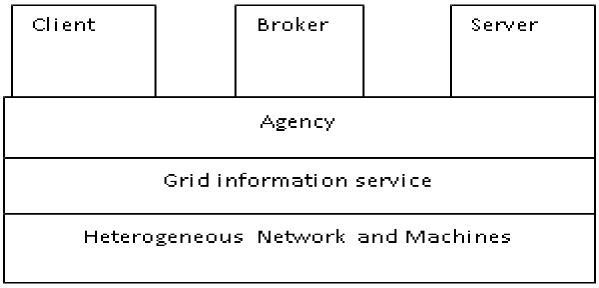

Figure-3. Performance of the proposed agent model.

4. RESULTS AND DISCUSSIONS

Figure-4 shows the multiple regional networks of Grid information service and it is providing the service and registering. In this, number of jobs request to grid information service is shown and the jobs are allocated to the resources based on the priority. From Figures 5 and 6, it is observed that the make space will increase based on the size of the jobs. Here we observe the increase in makespan value by using the advanced reservation and RLA. In this, jobs are waiting in queue (grid). Agent is hidden in the computer system that is situated in the environment. With the proposed algorithm, we improved the performance of 75% user and increased the transfer rate in grid information service. The proposed algorithm also provides the multicast among the nodes while finding the GIS (gird information service).

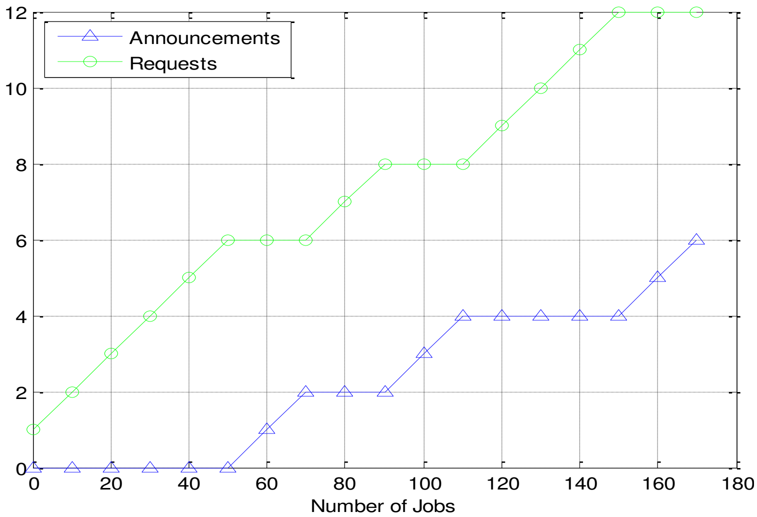

Figure-4. Comparison of Requests and Announcements in GIS.

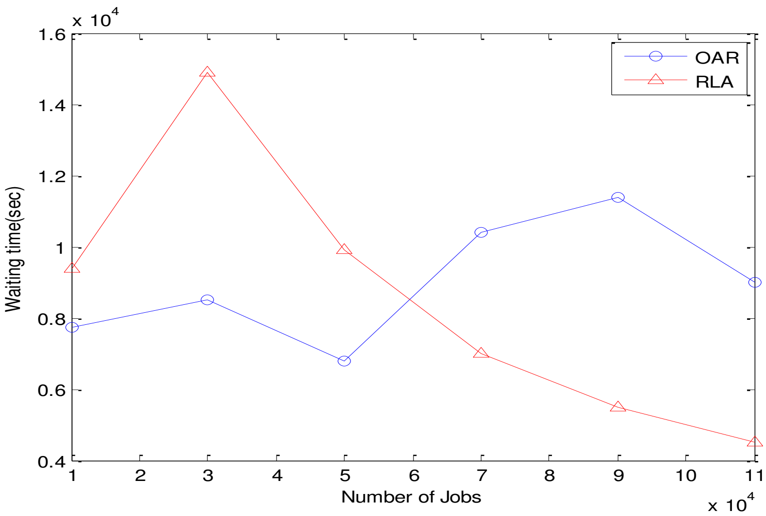

Figure-5. Time taken to execute all the jobs.

ARPN Journal of Engineering and Applied Sciences

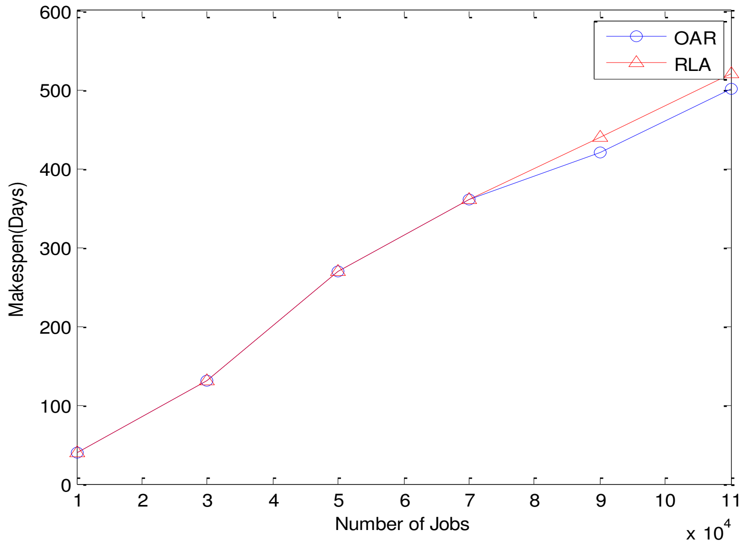

Figure-6. Average waiting time of all the submitted jobs.

5. CONCLUSIONS

Grid environment is very important for organizations to execute jobs. Fault tolerance is one of the major issues in grid computing. The proposed model developed an agent based mechanism for handling the fault tolerance issue in grid computing. It integrates the agents with grid communities to allocate the jobs to the resources. The co-allocation algorithm is introduced to distribute the jobs to the idle resource and also to reduce the burden on the nodes. GridSim is used to simulate the environment. The efficiency of the proposed model is tested with makespan and waiting time and it is proved that the algorithm is more efficient.

REFERENCES

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

[2] Latchoumy P. and P. Sheik Abdul Khader. 2011. Survey on fault tolerance in grid computing. International Journal of Computer Science and Engineering Survey. 2(4): 97.

[3] Jain Sumant and Jyoti Chaudhary. 2014. New fault tolerant scheduling algorithm implemented using check pointing in grid computing environment. In Optimization, Reliabilty, and Information Technology (ICROIT), 2014 International Conference on, pp. 393-396. IEEE

[4] Li. Chulin, Lilayuna. 2004. Agent Framework to support the computational grid. The Journal of Systems and Software. pp. 177-187.

[5] Wooldridge, M. and Jennings, N.R. 1999. Software Engineering with Agents: Pitfalls and Pratfalls. IEEE Internet Computing. 3(3): 20-27.

[6] Altameem T. 2013. Fault tolerance techniques in grid computing systems. International Journal of Computer Science and Information Technologies. 4(6): 858-862.

[7] Devi K. Nirmala and A. Tamilarasi. 2014. Improving fault tolerant resource optimized aware job scheduling for grid computing. Journal of Computer Science. 10(5): 763.

[8] Cao J., Kerbyson D.J, Nudd G.R. Australia. 2001. Performance evaluation of an agent-base resource management infrastructure for grid computing. In Proceedings of 1st IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 311-318, Brisbane.

[9] Maheshwari, Ketan, Marcus Lim, Lydia Wang, Ken Birman, and Robbert van Renesse. 2013. Toward a reliable, secure and fault tolerant smart grid state estimation in the cloud. In Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES, pp. 1-6.IEEE.

[10] Ding C.H., Nutanong S. and Buyya, R. Dec. 2003. P2P Networks for Content Sharing. Technical Report, GRIDS-TR-2003-7, University of Melbourne.

[11] Reddy T., Sunil Kumar, J. Jannet, K. N. Sivabalan, and DasariNagaraju. 2016. An Enhanced Routing Mechanism for Energy Reduction in Wireless Sensor Networks.

ARPN Journal of Engineering and Applied Sciences

[12] Vijaya kumar, Singanamalla, Nagaraju Dasari, Bharath Bhushan and Rajasekhar Reddy. 2017. Semantic Web-Based Framework for Scientific Workflows in E-Science. In Web Semantics for Textual and Visual Information Retrieval. pp. 187-202. IGI Global.

[13] DASARI, NAGARAJU, and V. Saritha. 2016. A Survey on Communicational Issues in Mobile Cloud Computing. Walailak Journal of Science and Technology (WJST). 14(10).

[14] P. Pownend and J-xu. 2003. Fault Tolerance with a Grid Environment. In proceedings of AHM 2003, p. 272.

[15] R. Medeiros, W. Crine, F. Brasilerio and J. Saure. Nov 2003. Faults in Grids: Why are they so bad and what can be done about it in Grid Computing. In Proceedings of Fourth International Workshop. pp. 18-24.

[16] S. Hwang and C. Kesselman. 2003. A flexible Framework for Fault Tolerance in Grid. In j-J-GRID-COMP. pp. 251-272.

[17] Wooldridge M. 1997. Agent-based software engineering. IEEE, in Proceedings of Software Engineering. 144: 26-37.

[18] Blatecky A. 07 September 2006. Grid Computing. Research Bulletin of EDUCAUSE Center for Applied Research. Issue 17, 2002.