AN ADAPTIVE FAULT TOLERANCE MECHANISM IN GRID COMPUTING

P. S. Nithyadharshini¹, M. Pounambal¹, D. Nagaraju³ and V. Saritha⁴
¹SCSE, VIT University, Chennai, Tamil Nadu, India
²SITE, VIT University, Vellore, Tamil Nadu, India
³Department of Computer Science Engineering, SSJ Engineering College, Hyderabad, India
⁴Department 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 here we 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 hidden in 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.](image)

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, \( A_1 = \frac{R}{N} \) 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.

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. \( E = \frac{n}{N} \).

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
Providing $p_b$ contains of list of slots. A slot $H_{(k,l,T)}$ 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.

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

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 a job $J_p$ 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.

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
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.

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 (grid information service).

Figure-3. Performance of the proposed agent model.

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

Figure-5. Time taken to execute all the 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


