Metadata Control Agent approach for Replication in Grid Environments

P. SunilGavaskar¹, Dr.Ch D V Subba Rao²
¹² Department of Computer Science and Engineering
¹ Sri Venkateswara University, Tirupathi, Andhra Pradesh, India
² Sri Venkateswara University, Tirupathi, Andhra Pradesh, India

Email- Sunil079@gmail.com

Abstract- since grid environment is dynamic, network latency and user requests may change. In order to provide better communication, access time and fault tolerant in decentralized systems, the replication is a technique to reduce access time, storage space. The objective of the work is to propose an agent control approach for Heterogeneous environments using the Agents for storing objects as replicas in decentralized environments. Our idea minimizes the more replicas (i.e. causes overhead on response time and update cost), therefore maintaining suitable number of replicas is important. Fixed replicas provides file access structure to identify the esteem files and gives optimal replication location, which minimize replication issues like access time and update cost by assuming a given traffic pattern. In this context we present the Agents as replicas to maintain a suitable scalable architecture. The solution uses fewer replicas, which lead to fewer agents as a result of that frequent updating is possible. Our tests show that the proposed strategy outperforms previous solutions in terms of replication issues.

Keywords – Agents, Metadata, Lookup Service, Replicas, Data Grid

I. INTRODUCTION

Grid [1] is a basic infrastructure of national high performance computing and Information services, it achieves the integration and interconnection of many types of high performance Computers, Data servers, large-scale storage systems which are distributed heterogeneous in nature, and the important application research queries which are lack of effective research approach. Data grid [2] provides services for supporting the discovery of resources and enables computing in heterogeneous storage resource by storage of resource agent. Data grid [3] allows sharing data among different environments with appropriate permission from storage space of data. Replica utilization in fault tolerance strategy is challenging research in data grid. By placing multiple replicas at different locations, replica management [4] can reduces network delay and bandwidth consuming of remote data access. Check pointing frequencies are increased as the fault frequencies increases [5], thereby diskless check pointing has been introduced as solution to avoid I/O bottleneck of disk- based check pointing. According to replica creation method, fewer replicas [6] would increase the response time of data access, while more replica would increase data update cost and waste storage space. Therefore, maintaining adequate number of replicas is important. In addition, due to the dynamic change of network bandwidth and storage resource, a fixed number of replica is neither feasible.

The rest of the paper is organized as follows. Related works are presented in section II. Proposed System model and strategies are explained in section III. Simulation and Analysis are presented in section IV. Concluding and future works are given in section V.

II. RELATED WORK

The replica management strategy can be divided into simple replica management strategy, this strategy again divided into No Replica Strategy (NR), Least Recently Used Strategy (LRU), Least Frequently Used Strategy (LFU). Hierarchical model based replica management strategy can be divided into best Client Strategy, Cascading
Strategy, Plain Caching Strategy, Caching and Cascading Strategy and Fast Spread Strategy. The Economic model based replica management strategy uses evaluation function to evaluate replica value, then decides whether to create local replica or not. This strategy uses antivickrey auction model to choose best replica. [2][3].The simple replica management strategy would make data frequently replicated, thus is not suitable to the dynamic grid environment. Second strategy hierarchical model is designed for Europe data grid and could not accommodate P2P and hybrid network structure. By considering third strategy an economic model with two main objectives: to maximize profits and minimize management costs [9]. Computation and Storage units purchase files from other storage units following auction protocol. The replicas deletion is also based on an economic model; a replica is deleted only if the deletion gives gain. In [9], the authors used economic model for database replica placement adapted to support database applications.

Issues with Replication are widely studied in Distributed systems [7]. Our Proposed models mainly aim at minimizing search time of replica with the help of agents as result of that reduces access time, communication cost and increase availability of data. In order to implement that the number of replicas become larger as the load of accessed node increases, to overcome this issue multiple object states are set. Each time creating replica for heavy load nodes and puts these replicas in the relative light load nodes, thus achieves load-balance. Root nodes always maintains metadata catalogue to perform mapping of objects and agent replicas.

III. SYSTEM MODEL

The proposed architecture is demonstrated in Figure 2: which contains the following components called child node or S Agents and superiors are called Magents. 1. Sagents: The roles of agents are storing local replicas of the currently registered distributed objects. 2. Magents: They also act as replicas and contain local copies and Meta-data catalogues. There by subset of Agents becomes under the control of Magents, the Agents are connected to form a complete graph. All the messages are routed between Magents there by Magents connect together several connected clusters of Sagents. Here clusters are formed on the basis of a Calendar Queue [10], which divides the queue into class types. In The proposed model we consider that we have numbers of Magents are equivalent of classes. Sagents are having length of class i.e. Number of objects supporting in it. Here we have several lookup discovery services, which stores information regarding the currently existing Magents. There by a new cluster is formed, its Magents dynamically registers to closest lookup service and then Sagents uses one lookup service to obtain information. The forming of agents into cluster is based on their network bandwidth and also on the basis of set of condition metrics.

A. Meta-Data Catalogue—

The Catalogue is treated as hash table where the keys describes specific information, Such as the Class name and access Pattern or Bean with Lifecycle of information’s to create and maintain new objects when required. In accordance with Architecture proposed in Figure 1 the Sagents are converted into Magents based on Election algorithm that happens among Sagents, due to this Phenomenon the Catalogues will avoid Complexity in Mapping of Replica, there by reduces job execution time. The bean lifecycle will maintains object with various states, where more than one request is required by different process on a same replica.
B. Replicas Management.

Here all the agents maintain copy of several replicated objects, these are java serializable objects. Once Clusters are grouped then it contains Master Privileges and information regarding the current objects are maintained with in the metadata catalogue, there by the metadata catalogue is updatable when Sagents are converted into Magents. The Replica mapping techniques is on the basis of hashing, which uses two values (Hash key, HashfieldValue). Hash Key of replica represents a Unique identification of object identifier with the particular mapping pattern Key (id0, id1, id2, id3…idn-1). HashfieldValue (id_agent1,id_agent2,…id_agentsN) identifies an Agent’s identifier, where N depends on class inheritance. In order to access the particular agent’s replica the request passes from Magents to other agents.

Previously replication is combined to job scheduling or at runtime [11]. In our model the replicas are created and placed before job execution / job starts, therefore agents contain replicas which are independent of the files requested by job (i.e. Copies of agents replication is happens at once after the original copies are created and before any file request, but modification replica allows on basis of mapping objects and agentsID present in metadata catalogue). This model performs to add, remove, rejoin of agents so that replica is dynamic in nature based on its size. From figure 1 the following steps will demonstrates our strategy.

**Step1:** Sagents joins in the network by Identifying list of Magents based on the lookup services and then estimate best Magent.
**Step2:** Magent is chosen based on size of its class id (i.e. load on it). and then Number of Sagents connected to Magents.
**Step3:** Sagents finds Magent which having fewer Sagents and then Computes metric (Distance between Magents is evaluated by means of shortest path connecting two Magents using shortest path routing algorithm.).

Thereby we reduced total Sagents connected to Magents by converting Sagents into Magents. In order to avoid remote access, Enhance availability of replicas and to control metadata catalogue growth rate.
Step4: If the Number of Sagents is greater when compare to minimum length of Magent catalogue size.
Step5: Among Sagents promote one Sagent into Magent based on the basis of election procedure, as a result of that metadata also changes.
Step6: Sagents connected are lesser than when Compare to Minimum Length of Magent catalogue size. Then the Catalogue destroys and that catalogue connected Sagents joins with the one from adjacent Magent or shortest path connecting it.
In the above said architecture such kind of hierarchical interconnection topology ensures scalability and optimal replication in critical data-intensive applications.

IV. SIMULATIONS

OptorSim is a salable, configurable and programmable simulation tool for grid. We extend OptorSim by adding the Java implementation of our strategy.

Table 1. Simulation Configuration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of files</td>
<td>97</td>
</tr>
<tr>
<td>Size of files</td>
<td>1GB</td>
</tr>
<tr>
<td>Number of jobs</td>
<td>100</td>
</tr>
<tr>
<td>Job Execution time</td>
<td>5</td>
</tr>
<tr>
<td>Job delay</td>
<td>2500</td>
</tr>
<tr>
<td>Background bandwidth</td>
<td>NO</td>
</tr>
<tr>
<td>Number of sites</td>
<td>20</td>
</tr>
<tr>
<td>Number of SE per site</td>
<td>1</td>
</tr>
<tr>
<td>Capacity of SE</td>
<td>50-100GB</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Lru Optimizer</td>
</tr>
<tr>
<td></td>
<td>Lfu Optimizer</td>
</tr>
<tr>
<td></td>
<td>EcoModelOptimizer</td>
</tr>
<tr>
<td>Number of worker node per site</td>
<td>100</td>
</tr>
</tbody>
</table>

Figure 2. Average job time
In figure 2. The simulation results show that gain on the average job time, our proposed strategy reduces average job time by 25%, here the replicas are placed according to pattern mapping using metadata catalogue, therefore updating of replicas are faster.

From the figure 3. It shows that numbers of replicas are generated, in our proposed strategy model, it reduces maintaining unwanted replicas (agents) and the accesses of all the replicas as local access.

By considering figure 4. The percentage of Storage space changes slightly when compare to other strategies. Our strategy uses replicas (Sagnets) based on the number of object patterns present in it. Storage space is compensating at runtime by means of connection and disconnection of agents (Number of objects can accept request on same replica, Thereby it depends on the objects states). This strategy performs little worse over percentage of storage. Through the analysis of simulation results, it can reduce number of replicas as a result of that less network traffic and improvement over average job time. Due to limitations in simulation tool kit the performance is not achieved fully, but we believe that our agents strategy achieve good performance.

V. CONCLUSION
In this paper, we proposed the concept of object oriented Replicas; we have introduced the strategy called agents activation of replica creation, selection, replacement and maintenance. The operations are optimized in terms of number of replicas used and messages exchanged between agents. With objects as replicas two processes can demand for the same resource at same time, due to objects state maintenance. Our work is useful to users where the decentralized environments facing challenges like load balance, storage space, access rate. Simulation results shows that, our concept of replicas optimization can significantly reduces total number of replicas. Thereby gain in the execution time is mainly due to enhancement in availability of replicas (objects) and its transfer time. In future we plan to further extend ideas with service level fault tolerance apart from resource level fault tolerance. The service level fault tolerance should focus on QoS issues and global behavior. It can benefit for Representation of grid global state in service oriented format.

REFERENCE


