International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July-2013 1070
ISSN 2229-5518

A Priori Data Replica Placement Strategy in Grid Computing

Gutta Subramanyam, Gudivada Lokesh, Bygani Jalaja Kumari

Abstract— Grid is a network of lines that cross each other to form a series squares or rectangles.Grid is also defined as a parallel or distributed system. Prior means in the earlier stage and the Replica is defined as the copying the duplicate data at desired location. And Now-a-Days the Grid computing is becoming very important and can be used in the areas requiring large quantity of data and calculation.To achieve the best access time and fault tolerance in such systems, the replication is the main issue to provide the above best access time and the fault tolerance.The effectiveness of a replication model depends on several factors including the replicas placement strategy. In this paper, we proposed a priori replica placement strategy which is used to optimize the distances between the data hosted on the grid in order to increase the grid performances. This approach shows a considerable improvement in job execution time without degradation of communication and storage costs. The gain in the execution time is mainly due to the improvement in the file transfer system.Experimental results arepresented to illustrate the feasibility and merits of this apporoach on placing replica prior to the clent request in the data grid.

Index Terms— Acess Time, Communication Cost, Execution Time, Fault Tolerance, File Transfer. Grid, Grid Computing, Job Execution, Prior, Replica, Storage Cost,

1 Introduction

Grid computing is a technology meeting the needs of scientific applications characterized by a high performance computing and large quantity of data. The main idea of this technology is the use of underutilized resources by distributing calculations and/or data on large scale dispersed under used resources. However, in such systems, it is difficult to guarantee the availability of all the sites at any time. In addition, the user access to remote data, can lead to undesirable wait time.

Data replication is an issue to satisfy availability constraint as well as access performance. By creating several copies of a data in different sites, replication ensures the running of the system even if some sites containing data are out of order. Moreover, it hangs over data from sites which improve access delay. To take advantage of that, it's important to choose carefully where and when placing replicas.

The replica placement problem was studied in the context of file assignment problem and considered as a complex optimization problem. Several strategies and heuristics are proposed in the literature. Among these works, we are interested in those of Ko and al [2], [3], [4]. In their works, the authors consider a network as a graph with colored nodes. The latter represent the network sites. When they have the same color, they represent replicas of the same data. In their strategy, they intend to color the nodes so that nodes of the same color are as far as possible from each other and nodes of different color are as close as possible from each other. In our work, we treat the problem of replicas placement in a P2P data grid environment.

We are guided by the work of Ko and al to do an a priori replicas placement. It means that we will place replicas on the appropriate sites before launching any work on the grid.We evaluate the performance of the strategy by simulation on the OptorSim simulator [5], [6] and study the effect of the proposed a priori replicas placement on the economic model [7] and other models implemented on OptorSim. We show that results are very promising. The rest of the paper is organized as follows. In section 2, we present some strategies and heuristics related to the replication with in grid environments. Section 3 describes the strategy we propose. In section 4, we give a brief description of OptorSim, the context of simulation. In section 5 the obtained results. In section 6 we give conclusion about this paper. Finally, We discuss some directions for future works in section 7.

2 Related works

Replication problems are widely studied in distributed systems [8]. Proposed models mainly aim at minimizing the access time and communications cost, and increase the availability of data. In the context of data grid, most models are naturally dynamic [9]. They fit to the changes that occur on the grid (unpredictable login and logout of sites) and to the jobs running on.

In [7] an economic model is proposed with two main objectives: to maximize profits and minimize management costs. Computation and storage units purchase files from other storage units following auction protocol. The deletion of replicas is also based on an economic model; a replica is deleted only if the deletion provides gain. In [10], the authors used the economic model for database replicas placement adapted to support database specificities. Several other works [11], [12] proposed cost models with various sizes of replicas, access frequency, storage cost, etc. Some works propose strategies for hierarchical data grid like in [13], [14], others for hybrid architecture like in [12]. These kinds of solutions present restrictions in communication among the network compared to P2P grid architecture [15].

The replica placement problem is also viewed as an optimization problem for which different heuristics are proposed [1]. In [16], the solution is based on ant behaviors. Ants are considered as agents moving in the grid and placing replicas according to probabilistic functions. In Ko and al works [2], from which we get our idea, the authors proposed a replication protocol for large scale networks. The network is represented by a graph with colored nodes. Nodes with identical replicas have the same color. The coloring is done such as nodes with different color are close to each other and nodes with the same color are far away from each other as possible. This placement allows each site to reach the different data through the network in a short distance. The problem was formulated by two objective functions which define an NP-hard problem. The authors proposed a decentralized solution where each node learns about its nearby nodes colors, selects a color and informs its near by nodes about its new color.

Nodes change their color dynamically to fit changes on the network. Updates about color changing are continually propagated. The authors proved the convergence of their protocol to a fixed coloring. The evaluation of the performance of the protocol focused on the time of convergence to a stable coloring, the number of exchanged message and the quality of coloring (distances between colors).In [4], the same authors propose a similar modeling with a centralized solution. The solution is less consistent since it relies on assumptions that do not reflect a real environment. They also present in [3] a decentralized protocol to large-scale client-server networks. The difference with [2] is that, unlike in [2] where all the nodes hold replicas, in [3] there are two types of nodes: servers which hold replicas and clients that demand them. The aim is similar: reduce network distance between each client and the nearby server hosting the resources needed by the client.

In all works presented here, replicas are created and well placed during runtime. Sometimes replication is combined to job scheduling or load balancing protocols to achieve better results. But no importance is given to a priori replica placement where replicas are created and placed before starting jobs. Thus, the placement is independent from files requested by jobs, from their scheduling and from the sites load. This is our purpose in this paper. Also, we will show that combining our strategy to a strategy that replicates during the execution of jobs (like the economic model strategy) improves the performance of the grid.

3 Proposed Strategy

We deploy our approach on a P2P data grid model which is the most flexible architecture in terms of communication between grid sites [15]. Several files are dispersed on the grid (original copies). Our purpose is to proceed to an adequate replication of these copies in order to improve their availability. The copies replication is performed at once after the original copies are created and before any file request. We propose to place the replicas in such a way that identical ones are as far away as possible from each other. This let different replicas to be as close as possible from each other. So that each site can find the different files in its vicinity. This guarantees short access delays. The two objectives can be simply formulated as follow:

Maximize the distance between identical replicas.

Minimize the distance between different replicas.

The distance between two sites is evaluated by means of the bandwidth of the shortest path connecting the two sites. The shortest path is obtained using Dijkstra's algorithm.

We propose a decentralized strategy. Each site of the grid performs the strategy to choose the only replica to hold. We assume that the grid is static during the execution of the strategy.We do not tolerate connection/disconnection of sites nor add/delete of original files when doing the a priori replicas placement.

The site executing the strategy collects the list of all original files and their replicas on the network and then, it groups them per color. This results in several sets each one composed of one original file and zero or several replicas. We notice that all the set files have the same color. After that, it firstly, selects for each composed set, the best potential replica. The best potential replica for a given color is the nearest one to the site. The best potential replicas per color compose the Best-Replicas set. Secondly, the furthest replica from the site is selected among the Best-Replica set. It is added to the replicas of the site.

The choice of the nearest file for each color and afterwards the selection of the furthest file among the nearest colored files ensure the maximization of the distance between identical replicas. Indeed the choice of the nearest file of each color insures that there is no closer file with the same color. Finally, the choice and replication of the furthest file among the nearest ones selected previously maximize the distance. We illustrate the strategy by the following example. In figure 1(a), Site 1 will choose which file to host (red or blue) and from which site it will replicate (site 2, site 3, site 4 or site 5)

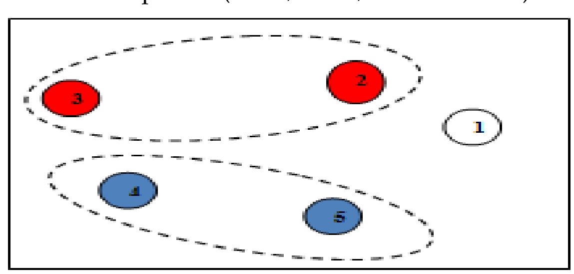

Fig.1 (a): Replica Placement Strategy-1

ISSN 2229-5518

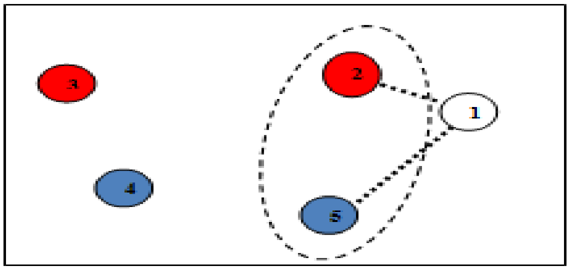

Fig.1 (b): Replica Placement Strategy-2

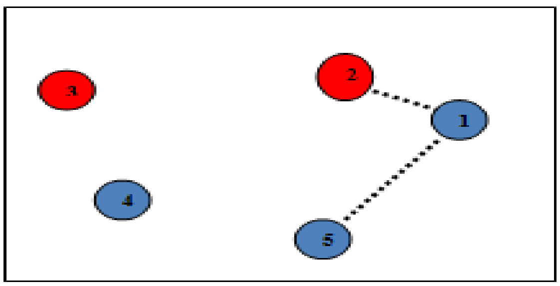

Fig.1 (c): Replica Placement Strategy-3

Step 1: Site 1 collects the set of replicas and original copy of each file. See figure 1(a). For red file, the set is composed of the file hosted in site 2 and the one hosted in site 3. For blue file, the set is composed of the file hosted in site 4 and the one hosted in site 5.

Step 2: From each set constructed in step 1, site 1 chooses the nearest file. Thus, the nearest red file is the one hosted in site 2. The nearest blue file is the one hosted in site 5. These two selected files are added to the Best-Replicas set. See figure 1(b).

Step 3: Site 1 chooses the furthest file from Best-Replicas set. It chooses the blue one. It replicates it from site 5. See figure 1(c).

The strategy executed by the sites is described in the following Pseudo code:

for each file f from list of original files **do**
Replicas= listReplicas (f);
BestReplicas[f]= nearestReplica(Replicas);
end for
return (furthestReplica (BestReplicas));

Before the selected replica is replicated on the site executing the strategy, storage and computation capacity constraints should be taken into account. Storage and computing resources on the grid are designated by Storage Element (SE) and Computing Element (CE). Each site in the grid has zero or more SE and zero or more CE with limited capacities. In our current work, we just consider storage element limitations. We will consider computation constraints in future works.

The global replication process is described in the following pseudo code:

for all sites of the grid **do**
if the site has at least one SE **then**
replica=selectReplica(list _of_ original_ files);
search SE with sufficient storage space to hold "replica";
if SE exists **the**
Replicate (replica, SE)
end if
end if
end for

4 Simulation and Tests

We implement our strategy in OptorSim simulator. The latter was developed to simulate dynamic replication strategies. OptorSim runs the jobs submitted on the grid using a grid configuration, a replica optimizer and schedule algorithms. To be executed, a job needs a set of files which can be located in different sites. The job scheduling controller known as

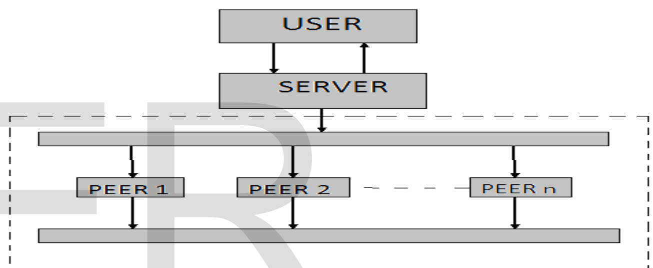

Fig.2 (a): System's Block Diagram - 1

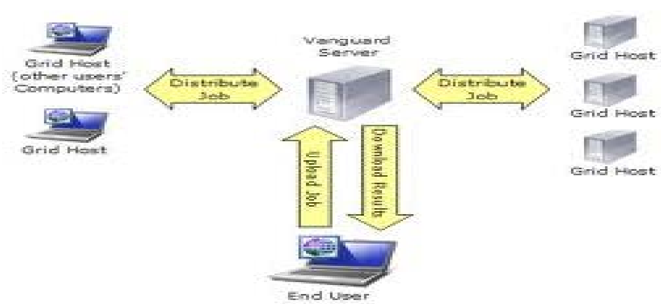

Fig. 2(b): System's Block Diagram - 2

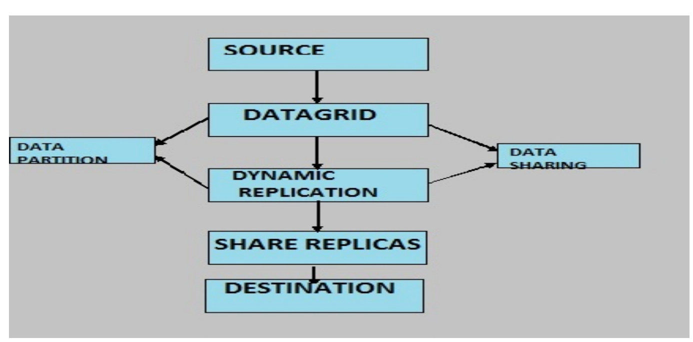

Fig.3: Data Flow Diagram

ISSN 2229-5518

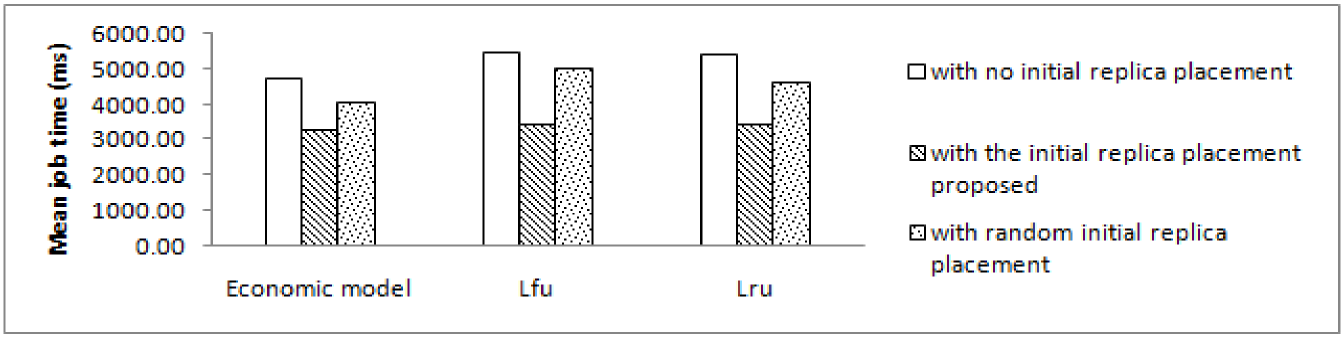

Fig. 4. Evaluation of mean job time.

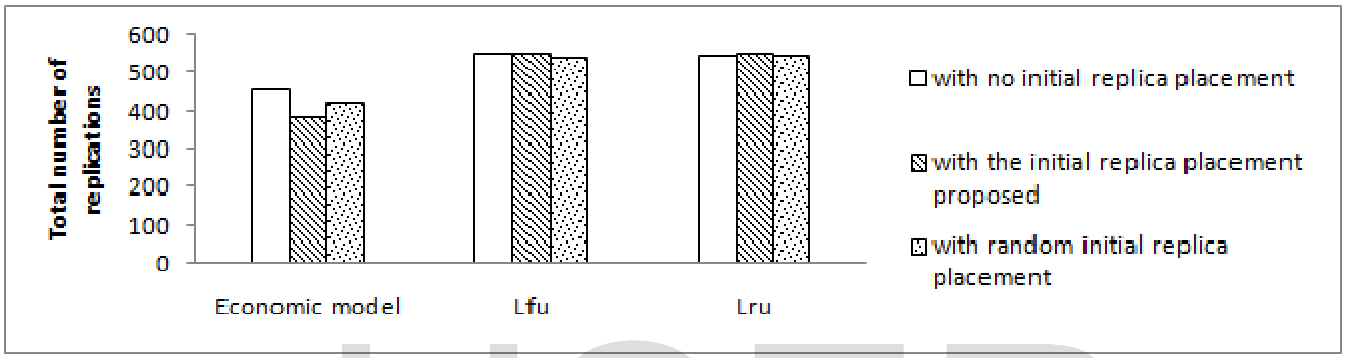

Fig. 5. Evaluation of the number of replication in runtime.

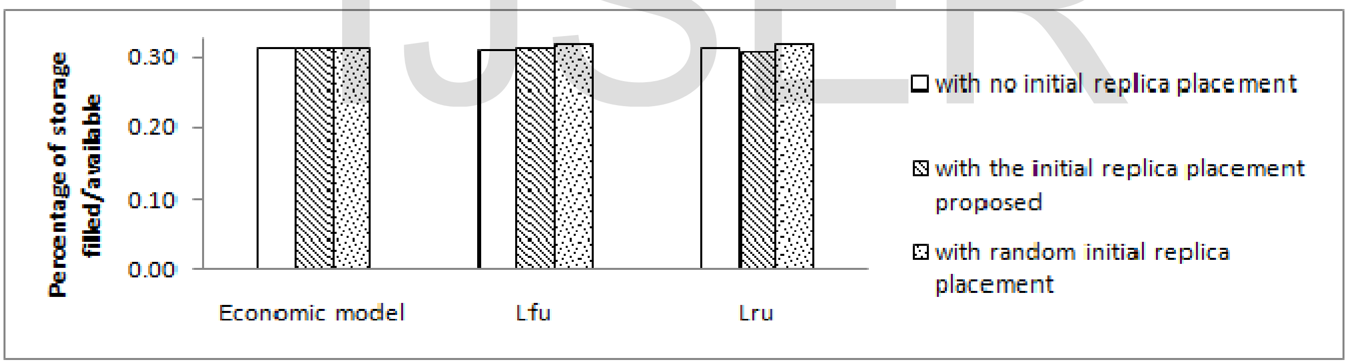

Fig. 6. Evaluation of the storage resource consumption

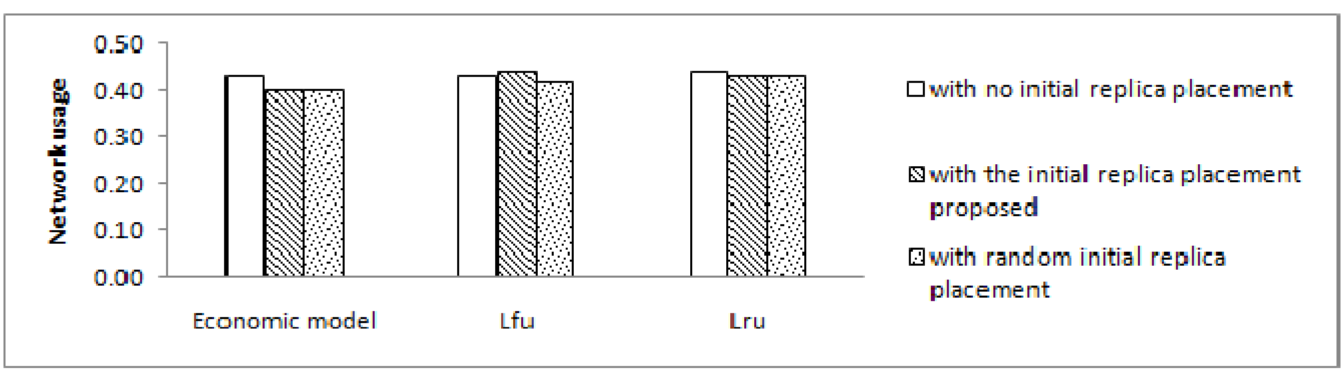

Fig. 7. Evaluation of the network usage

ISSN 2229-5518

Resource Broker [5] uses a schedule algorithm to assign the job to the adequate site. To run a job, the Replica Optimizer selects by means of the replica optimizer algorithm the files used by the job. We modified OptorSim by adding once the proposed a priori replica placement strategy and a second time, a random a priori replica placement strategy. In the latter, each site chooses randomly the replica it hosts.

To evaluate our contribution, we refer to three replica optimizer algorithms already implemented in OptorSim:
1) Lru Optimizer: that always replicates. It deletes least recently created files. 2) Lfu Optimizer: which always replicates? It deletes least frequently accessed files. 3) Economic Model Optimizer: which replicates when replication is less expensive than the remote access? It deletes the least valuable file. We test these algorithms firstly, with no a priori replica placement, secondly, with a random a priori replica placement and finally with our proposed replica placement. Considerable gain on the mean job time. Our a priori replicas placement reduces the mean job time by 30% when combined with any replica optimizer algorithms (economic model, lfu or lru) whereas the random replica placement reduces it by about 10%.This is due to a lower file transfer time since all data are closer to all sites thanks to the strategic placement of the replicas .On the other hand, the time required for the execution of the strategy does not increase the job execution time since t he strategy is executed before launching the jobs on the grid.

The improvement in the number of replicas in the economic model results from the fact that some replicas are already placed in the right place when executing the jobs. In deed, the economic model creates a replica whenever it's beneficial. Otherwise, a remote access is done. For lfu and lru models, the a priori placement of replicas has no effect on the number of replicas created during jobs execution because the replication is performed whenever a file is needed (always replicate).The percentage of filled/available storage is practically the same for the three strategies. The a priori placement of replicas increases the occupation of the storage space. This causes a slight degradation in the percentage of filled/available storage noticed in lfu model. For the economic model, the gain in number of replicas created during the runtime compensates the degradation caused initially. The latter is calculated as follows [6]:

$$\textit{Network usage} = \frac{\textit{Number of remote access} + \textit{Number of replication}}{\textit{Number of local access}}$$

The network usage parameter informs about the communication rate and the band width consumption. In fact, it has a high value in case of excess in remote access and/or replication relatively to local access. This causes a high communication rate and thus large bandwidth consumption. A slight improvement in the network usage provided by the economic model. The gain reflects the decrease in the number of replication. When there is less replication, there is less transfer of files and therefore less congestion in the network. This explains the gain in network usage. The most important result we reach is that the proposed a priori replicas placement improves the jobs execution time with no degradation in the storage resources or bandwidth consumption.

5 RESULTS

After executing the source code the following screen shots have been displayed.

Fig.8: Server Page

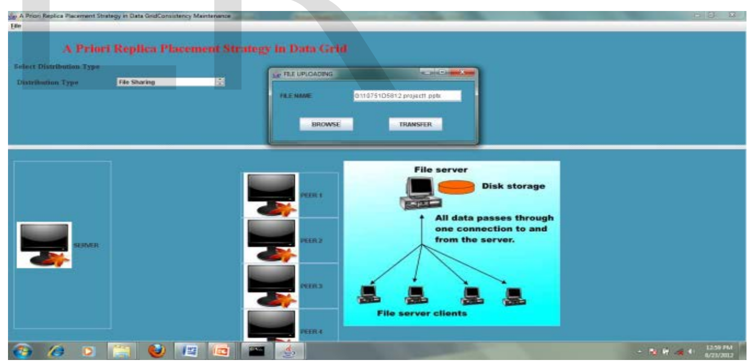

Fig.9: Loading the file into the Server

Fig.10: Sending replica to different peer(s)

ISSN 2229-5518

Fig.11: peer(s) receiving the replica from the server

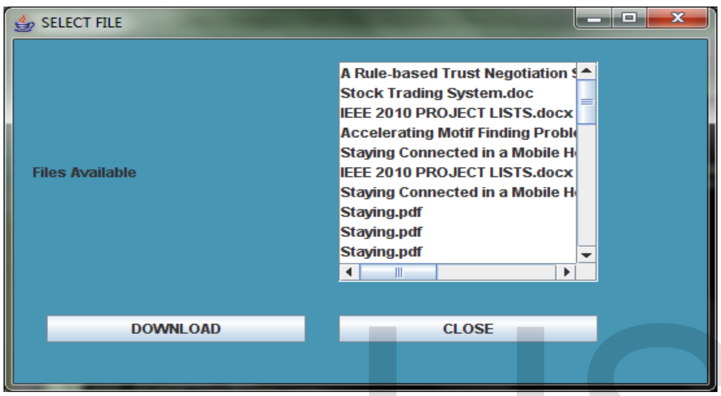

Fig.12: Requesting file from the client

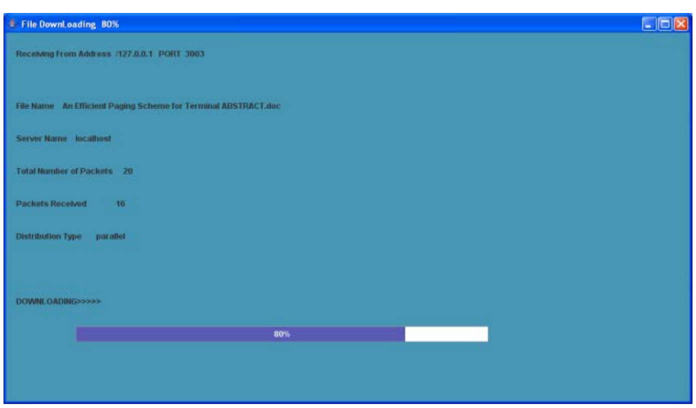

Fig.13: Client receiving the file form the peer

Fig.14: Downloading the file

6 Conclusion

We proposed in this paper an a priori replicas placement strategy in order to improve the grid performances. Indeed, the experimentation on OptorSim simulator shows a considerable improvement in job execution time without degradation of communications and storage costs. The gain in the execution time is mainly due to the improvement in the file transfer time. This is reached since each site finds the required data in its vicinity thanks to the proposed replicas placement strategy. In our current work, we test the approach with different configurations to determine the best parameters values (bandwidth, file size, storage space, etc.) for which the strategy gives the best results. This also leads us to refine our strategy.

7 Future Enhancement

In the proposed approach the replicas are generated after creating the original files and before the client request. These replicas are sent to all the local sites after replication process. Now the data is available at all the sites. So that user can access the file at all the sites. But when ever user makes any changes in any of the replica, that change cannot be uploaded to the original file. In the similar manner if the user makes the changes in any one of the replica at any site then the replicas at remaining site remains unchanged. This result in inconsistency that is accessing the same file at different sites may not yield the same data. This approach is developed only in static network but not in dynamic network. So future enhancements are conducted in order to fulfill these problems.

Acknowledgment

We deeply thank Miss S.Komala, Mrs. Y.Mohana Roopa, Miss R.Rani and Mr. J.Hemanth for their valuable help.

References

[1] M. Karlsson and C.T. Karamanolis, "*Choosing replica placement heuistics "for wide-area systems*," in 24th International Conference on Distributed Computing Systems (ICDCS 2004), 2004, pp. 350– 359.

[2] B.-J. Ko and D. Rubenstein, " *Distributed sel f- stabilizing placement of replicated resources in emergin networks*," IEEE/ACM Trans.Netw. vol.13, no. 3, pp. 476–487, 2005.

[3] —, "*Distributed server replication in large scale networks*",in Proceedings of the 14th International Workshop on Network and Operating Sys tems Support for Digital Audio and Video (NOSSDAV 2004), 2004, pp. 127– 132.

[4] —,"*A greedy approach to replicated content placement using graph coloring*," in SPIE ITCom Conference on Scalability and Traffic COntrol in IP Networks II, 2002.

[5] W. H.Bell , D. G. Cameron, A. P. Millar, L. Capozza,K. Stockinger and

ISSN 2229-5518

F. Zini, "Optorsim: *A grid simulator for studying dynamic data replication strategies*," International Journal of High Performance Computing Applications (IJHPCA03), vol 17, no. 4, pp. 403–416, 2003.

[6] D.G.Cameron, R.Carvajal-Schiaffino, A.P.Millar,C. Nicholson K.Stock-Inger, and F. Zini,"*Evaluating scheduling and replica optimisation strategies in optorsim*," in GRID, 2003, pp.52–59

[7] U. Cibej, B. Slivnik, and B. Robic, "*The complexity of static data replication in data grids,"Parallel Computing*, vol.31, no. 8-9, pp 900– 912, 2005.

[8] C. Haddad and Y. Slimani, " *Economic model for replicated Database placement in grid,* " in 7th IEEE International Symposium on Cluster Computing and the Grid (CC Grid ' 07), 2007, pp. 283–292.

[9] M. Garmehi and Y.Mansouri, " *Optimal placement replication on Data grid environments,* in 10th International Conference on Information Technology (ICIT 2007), 2007, pp. 190– 195.

[10] H. Lamehamedi, B.Szymanski, Z. Shentu, and E.Deelman "*Data replication strategies in grid environments* ," in Proceedings of the Fifth International Conference on Algorithms and Architecture for Parallel Processing (ICA3PP'02), 2002, pp. 378– 383.

[11] J. H. Abawajy, " *Placement of file replicas in data grid environments*,in Proceedings of the Programming Paradigms Grids and Meta computing Systems Workshop (PPGaMS 2004),pp.66-73.

[12] P. Liu and J.-J. Wu, " *Optimal replica placement strategy for Hierarchical data grid systems*," in International Symposium On Cluster, Cloud and Grid Computing (CCGRID 2006), 2006, pp.417–420.

[13] Q. Rasool, J. Li, G. S. Oreku, E. U. Munir, and D.Yang, "*A Comparative study of replica placement strategies in data grids*,"in APWeb / WAIM Workshops, 2007, pp. 135–143.

[14] A. Forestiero, C. Mastroianni and G. Spezzano , "*Qos-based Dissemination of content in grids*, " Future Generation Comp. Syst., vol. 24, no. 3, pp. 235–244, 2008.

[15] A.Chakrabarti, R. A.Dheepak, and S. Sengupta, "*Integration of Scheduling and replication in data grids*,"in Proceedings of 11th International Conference on High Performance Computing (HiPC2004) , 2004, pp. 375–385.

[16] F. Desprez and A. Vernois, "*Semi-static algorithms for data Replication and scheduling over the grid*," 2003.

[17] M. Tang, B.-S. Lee, X. Tang, and C. K. Yeo, "*The impact of data replication on job scheduling performance in the data grid* ,"Future Generation Comp. Syst.ol.3,pp. 254 – 268, 2006.

About Authors

1) GUTTA SUBRAMANYAM received the B.Tech Degree in 2005 from Jawaharlal Nehru Technological University, Hyderabad, India and M.E. Degree in 2010 from Sathyabama University, Chennai,India.Currently he studying M.Tech Degree at Annamacharya Institute of Technology & Sciences, Tirupati-517520, Andhra Pradesh, India and will receive his M.Tech Degree in august/ September 2013 from Jawaharlal Nehru Technological University, Anantapur. And he has a total teaching experience about 8 years and worked as Assistant Professor and Associate Professor in various engineering Colleges.

2) GUDIVADA LOKESH received the B.Tech Degree in 2010 from Jawaharlal Nehru Technological University, Anantapur, India and M.E. Degree in 2012 from Sathyabama University, Chennai,India. And he has a teaching experience about 01 year and now working as Assistant Professor in Vemu Inst. of Tech., Chittoor , A,P., India.

3) BYGANI JALAJA KUMARI received the B.Tech Degree in 2006 from Jawaharlal Nehru Technological University, Hyderabad, India and M.Tech. Degree in 2009 from Jawaharlal Nehru Technological University, Anantapur, India. And she has a teaching experience about 05 years and now working as Assistant Professor in Vemu Inst. of Tech., Chittoor , A,P., India..