International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.3, Issue.3, May-June. 2013 pp-1307-1308 ISSN: 2249-6645

File Replication to Access Files with Reasonable Response Time in Data Grid Environment… A Review Study

Dhananjay L. Joshi[1], Asst. Prof. S. A. Hasmi[2], Dr. Mrs. A. M. Rajurkar[3]
[123]*Department of CSE, M. G. M. College of Engineering, S. R. T. M. University Nanded, India*

Abstract: *Data Grid is an integrated architecture that connects multiple computers and its resources in distributed environment. The file replication is an effective functionality in Data Grid that not only minimizes total access time by replicating most accessed data file at appropriate location but also improve data files availability. This paper mainly deals with different file replication methodologies and enlists various effective strategies proposed by earlier authors to access data file with reasonable response time in Data Grid environment.*

Keywords: *Data Grid, Data File Replication, Replication Strategies, Simulation.*

I. Introduction

Data Grid is composed of set of sites and each site contains multiple computing, storage and networking resources. All sites are geographically connected to manage and store large data files of size Gigabytes and beyond in data repositories (sites) throughout the world. Data Grid provides an important service of data and/or data file replication in multiple locations, so that, it helps user not only to speed up data file access but also increases data file availability. A community of researchers distributed worldwide can access and share these replicated data files. In Data Grid, each data files are initially produced and stored in Grid sites. A Grid site may contain multiple data files and will be replicated in appropriate location in Data Grid to reduce access cost.

Many of the time Replication is confused with caching as they have multiple copies of file, and they have some differences. Replication is a server side phenomenon whereas caching is associated with a client. A server decides when and where to replicate files. A client request for a file and stores a copy of the file locally for use. Any other nearest client can also request for that cached copy.

Replication is that, it can enhance data availability and network performance. The replication of files in Data Grid follows the full or partial replication strategy. In full replication all files are replicated to all resources where as in partial replication files are replicated to some resources in the Data Grid. There are two replication schemes depending on the use access pattern: 1. Static Replication: in which replicas are kept until it is deleted. 2. Dynamic Replication: in which replicas are created and destroyed or replaced according to variation access of the pattern or environment behavior. In data replication there are three issues: 1. Replica Management- create, delete, move & modify replica. 2. Replica Selection-selecting appropriate replica across grid. 3. Replica Location-selecting physical locations of several replicas of desired data.

II. Literature Survey

1.1 Data Replication in Data Intensive Scientific Application with Performance Guarantee [1].

This paper deals with scientific data in the form of data files are produced, stored and replicated if necessary. The author proposed a centralized data replication algorithm (Greedy), it places one data file into the storage space of one site and algorithm terminates when all storage space of sites has been replicated with data files to minimize total access cost in the Data Grid. This algorithm that not only has a provable theoretical performance guarantee, but can be implemented in distributed and practical manner

Specifically, the author designed a polynomial time centralized replication algorithm that reduces total access cost by at least half of reduced by the optimal replication solution. Based on this centralized algorithm a localized distributed data caching algorithm is designed to make intelligent caching decisions. It is composed of Centralized Replica Catalogue (CRC): maintained at top level sites, which is essentially a list of replica sites list for each data file. Nearest Replica Catalogue (NRC): maintained at each sites which contains information of replica copy and nearest sites, and any changes made to NRC will be updated in CRC by sending message to top level site. Simulation results shows centralized greedy algorithm performs quite close to optimal algorithm.

1.2 Identifying Dynamic Replication Strategies for a High-Performance Data Grid [7].

This paper discusses about dynamic replication strategies for high performance; author presents data replication in hierarchical Data Grid model (as a tree topology) and six different replication strategies: (1)No Replication and Caching-where no replication takes place. (2)Best Client-The best client is one that has generated the most number of requests for that file, and then the node creates a replica of that file. (3)Cascading Replication- Once the threshold for a file is exceeded at the root replicas are created on next level but on the path of best client. (4)Plain Caching- The client request a file stores a copy locally. (5) Caching plus Cascading Replication- this combines strategy (3) & (4). The client caches file locally. The server identifies the popular files and propagates them down the hierarchy. (6) Fast Spread-replicas are created at each site along its path.

All of the above strategies are evaluated with three user access patterns: (1) Random Access- No locality in Access. (2) Temporal Locality- recently accessed files are likely to be accessed again. (3) Geographical plus Temporal locality- a recently accessed files are likely to be accessed again by a close site. Their simulation result shows Cascading (with geographical plus temporal locality) and Fast Spread (with random access) works better.

1.3 Analysis of Scheduling and Replica Optimization Strategies for Data Grids Using OptorSim [2].

In this paper, author discussed and concentrated on the effect of various job scheduling and data replication strategies with optimization as follows.

Scheduling Optimization Strategies: This algorithm decides when & where job should be executed by selecting the best job location. It calculates cost of running job on each site using following cost metrics. Access Cost: based on network status for obtaining required files. Queue Size & Queue Access Cost: gives total estimated access cost for all jobs in the queue.

Replica Optimization Strategies: is useful to minimize a single job's execution cost (as low a cost as possible) and to maximize the usefulness of locally stored files (by utilizing available data resources)by performing tasks *viz* replication decision, selection and file replacement. Author also considered three specific optimization strategies, one is LFU(Least Frequently Used) algorithm and two economic strategies are similar to each other, but uses different prediction functions, one is binomial based and other is Zipf-based, to calculate file values used in replication and file replacement decisions (sites can "buy" and "sell" files by using auction protocol mechanism).Their simulation result shows scheduling optimization reduces average times to execute jobs & economic based strategy have greatest effect.

1.4 Agent Based Replica Placement in a Data Grid Environment [3].

The author proposed an agent based replica placement algorithm for making a replica decision to select 'candidate site' for replica placement to reduce access cost, network traffic, and aggregated response time for the applications. To select a candidate site for a replica, an agent is deployed at each site that holds master copies of the files for which the replicas are to be created. The agent in this approach is autonomous, self-contained software capable of making independent decisions. Replica placement strategy considers two issues in choosing replica location: (1) placing a replica at proper site so that times taken for obtaining all files required by jobs are minimized. (2) Place a replica at sites that optimizes total execution time of the jobs executed in Data Grid.

The author extended the GridSim toolkit for decision making process for selection of candidate site by implementing Replica Catalogue and Replica Manager to maintain and control all replicas.

III. Conclusion

The data file replication performance depends on a variety of factors such as replica selection, placement, network traffic and bandwidth. This paper focuses on data file replication algorithms by following different file replication strategies using simulation environments. Well suited replication strategy can improve Data Grid performance depending on data file access situation.

References

Journal Papers:

[1] Dharma Teja Nukarapu,Bin Tang, Liqiang Wang and Shiyong Lu, "Data Replication in Data Intensive Scientific Applications with Performance Guarantee", IEEE Transaction on Parallel and Distributed Systems, Vol. 22, No 8, Aug 2011.

[2] D. G. Cameron, R. Carvavajal-Schiaffino, A. P. Millar, C. Nicholson, K. Stockinger and F. Zini, "Analysis of Scheduling and Replica Optimisation Strategies for Data Grids using OPtorSim ".

[3] Ms. Shaik Naseer and Dr. K. V. Madhu Murthy, "Agent Based Replica Placement in a Data Grid Environment", First Int'l Conference on Computational Intelligence, Communication Systems and Networks 2009.

[4] Mahesh Mayura and Ketan Shah,"A Review on File Replication Algorithms", journal of Sci. & Tech. Mgt. vol 3(2), july 2011.

Books:

[5] Pradip K Sinha, " Distributed Operating Systems-concepts and design", IEEE Computer Society Press, Prentice Hall of India 1998.

Thesis:

[6] Wong Yuk Key."Performance Analysis of Data Replication in Data Grid", University of Malaya, Kuala Lumpur, MALAYSIA, 2006/2007.

Proceedings Papers:

[7] Kavita Rananathan and Ian Foster, "Identifying Dynamic Replication Strategies for High Performance Data Grid", Proc. Second Int'l Workshop Grid Computing (Grid), 2001.