International Journal of Modern Trends in Engineering and Research
www.ijmter.com
e-ISSN No.:2349-9745, Date: 2-4 July, 2015

DATA REPLICATION OPTIMIZATION IN DATA GRID USING CENTRALIZED ALGORITHM STRATEGY

Prof. D.S.Rajnor[1], Prateek Laddha [2], Akshaykumar Jain[3], Pranay Gothi[4], Rohit Tatiya[5]

[1]*Department of Computer Engineering, S.N.J.B's KBJ COE, Chandwad, dsrajnor13@gmail.com*
[2]*Department of Computer Engineering, S.N.J.B's KBJ COE, Chandwad, prateekladdha.laddha80@gmail.com*
[3]*Department of Computer Engineering, S.N.J.B's KBJ COE, Chandwad, akshayjain0311@gmail.com*
[4]*Department of Computer Engineering, S.N.J.B's KBJ COE, Chandwad, pranaygothi@gmail.com*
[5]*Department of Computer Engineering, S.N.J.B's KBJ COE, Chandwad, rohittatiya1993@gmail.com*

Abstract— Data replication has been well adopted in data intensive scientific applications to reduce data file transfer time and bandwidth expenditure. However, the problem of data replication in Data Grids, an enable technology for data intensive applications, has confirmed to be NP-hard and still non approximable, creation this problem not easy to solve. Meanwhile, most of the previous research in this field is either theoretical survey without practical concern, or heuristics-based with little or no theoretical presentation assurance. In this paper, we propose a data replication algorithm that not only has a provable theoretical act guarantee, but also can be implemented in a distributed and sensible manner. Specifically, a polynomial time centralized replication algorithm is designed that reduces the total data file access hold-up by at least half of that reduced by the optimal replication solution. Based on this centralized algorithm, and also design a distributed caching algorithm, which can be simply adopted in a distributed surroundings such as Data Grids. Broad simulations are performed to legalize the effectiveness of our proposed algorithms .Using our own simulator show that our centralized replication algorithm performs comparably to the optimal algorithm and other intuitive heuristics under different network parameters. Using GridSim as well as flash builder , a well-liked distributed Grid simulator or alternative by using adobe flash, we express that the distributed caching technique considerably outperforms an live popular file caching technique in Data Grids, and it is more scalable as well adaptive to the dynamic change of file right to use pattern in Data Grids.

Keywords— Data intensive application, Data Grids, data replication, algorithm design and analysis, simulation.

I. INTRODUCTION

Data intensive scientific applications, which mainly aim to answer some of most fundamental questions facing human beings, are becoming increasingly prevalent in a wide range of scientific and engineering research domains for ex. high particle physics. In such applications data sets are generated, accessed and analyzed. The data grid is an enabling technology for data intensive scientific applications which is composed of hundreds of geographically distributed computation, storage and networking resources to facilitate data sharing and management in data intensive scientific applications. One distinct feature of data grids is that they produce and manage very large amount of data sets in order of terabytes and petabytes. For example The Large Hadrons Collider (LHC) at the European organization for nuclear research near Geneva, Switzerland, is the largest scientific instrument on the planet.

We study how to replicate the data files onto the grid sites with limited storage space in order to minimize the overall job execution time. In our model, the scientific information, in the form of files, are produced on some grid sites as the result of the scientific experiments, simulation, or computation. Each grid site executes a sequence of scientific jobs. To execute each job, grid site usually needs some scientific data as its input files. If these input files are not in the restricted storage resource of grid sites, they will be accessed and transferred from other sites. Each grid node can cache multiple data files subjects to its storage capacity limitation. The objective of our file replication problem is to minimize the overall job execution time.
Replication is valuable mechanism to reduce file transfer time and bandwidth consumption in data grids placing most accessed data at the right locations can greatly improve the performance of data across from a user's perspective.

II. LITERATURE SURVEY

Replication has been an active research topic for many years in World Wide Web [1], peer-to-peer networks [3].In Data Grids, the gigantic scientific data and complex scientific applications call for new replication algorithms, which have introduced lots of research recently that are discussed below.
Cibej et. al. study data replication on Data Grids as a static optimization problem [4]. They only consider static data replication for the purpose of formal analysis. The limitation of static approach is that the replication cannot adjust to the dynamically changing use access pattern.
Tang et al. worked on dynamic replication algorithm for multi-tier Data Grids [5]. He proposed two dynamic replica algorithms (Single Bottom Up and Aggregate Bottom Up). Performance results show both algorithms reduce the average responses time of data access compared to a static replication strategy in a multi-tier Data Grid.
Park et al. proposed a dynamic replica strategy called BHR [2], which benefits from network-level locality to reduce data execution time by avoiding networking congestion in a Data Grid. To execute the submitted jobs, each Grid site either gets the required input data files to its local compute resource, or schedules the job close to the sites where the needed input data files are store, or transfers both the data and job to a third site that performs the computation and returns the result.
Identify the limitations of current research of data replication in data grid: they are either theoretical investigation without practical consideration or heuristic based implementations without provable performance guarantees. Develop a data replication algorithm that not only has provable theoretical performance guarantee but can also be implemented in distributed manner as well.
Via simulation, our proposed replication strategies perform comparably with the optimal algorithm and significantly outperform previous existing replication technique. Via simulation, our replication strategies adapt well to the dynamic access pattern change in data grids. For the simulation purpose we use action script of adobe flash (flash builder4.7)

III. SYSTEM ARCHITECTURE

DATA GRID MODEL:

Data Grid model considered for the implementation is as shown in Fig. 1 [6]. Data Grid consists of a set of sites such as institutional sites, top level sites.

The data grid consist of number of sites such as
1. Institutional site
2. Top level Site

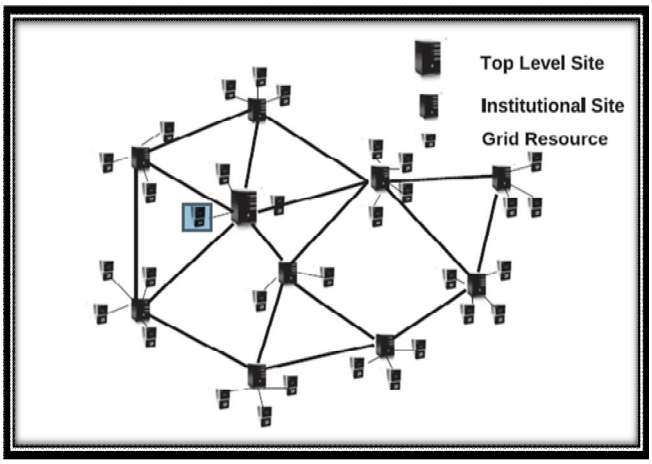

Fig1. Data Grid Model

These correspond to different scientific institutions participating in the scientific project. There is one top level site, which is the centralized management entity in the entire Data Grid environment, and its major role is to manage the Centralized Replica Catalogue (CRC). CRC provides location information about each data file and its replicas, and it is essentially a mapping between each data file and all the institutional sites where the data is replicated. Each site (top level site or institutional site) may contain multiple grid resources. A grid resource could be either a computing resource, which allows users to submit and finish job, or a storage resource, which allow users to store data files. We assume that each site has both computing and storage capacities, and that within each site, the bandwidth is high enough that the communication delay inside the site is negligible. For the data file replication problem addressed in this article, there are multiple data files, and each data file is produced by its source site (the top level site or the institutional site may act as a source site for more than one data files). Each Grid site has limited storage capacity and can cache/store multiple data files subject to its storage capacity constraint.
We have used four algorithms for this strategy.

1. Centralized Data Replication Algorithm:
Our centralized data replication algorithm be a greedy algorithm. First, all Grid sites have all unfilled storage space (except for sites that initially produce and store some files). Then, at each step, it places one data file into the Storage space of one site such that the decrease of total access cost in the Data Grid is maximize at that step. The Algorithm terminate when all storage space space of the sites has been replicated with data files. It manages all the data at its centralized System.

2. Distributed Data Catching Algorithm:
In the distributed algorithm, each Grid site observes the limited Data Grid traffic to make an intelligent caching decision. Our distributed caching algorithm is beneficial since it does not require global information such as the network topology of the Grid, and it is additional reactive to network states such as the file sharing, user right to use pattern, and job distribution in the Data Grids. Therefore, our distributed algorithm can adopt fine to such dynamic changes in the Data Grids. The distributed algorithm is collected of two important Components: nearest replica catalog (NRC) maintain at each site and a localized data caching algorithm running at each site.

3. Nearest Replica Catalog (NRC):

Each site i in the Grid maintain an NRC, and each entrance in the NRC is of the form (Dj, Nj), where Nj is the adjacent site that has a replica of Dj. When a site execute a work, from its NRC, it determine the nearest replicate location for each of its input data files and go to it directly to obtain the file (provided the input file is not in its local storage). As the initialization step, the source sites forward messages to the top level site informing it about their unique data files. Thus, the centralized replica catalog originally records each data file and its source site. The top level site then broadcast the replica list to the entire Data Grid. Each Grid site initializes its NRC to the source site of every data file. note down that if i is the source site of Dj or has cached Dj, then Nj is interpret as the second nearest replica site, The second nearest replica site in order is useful when site i decides to eliminate the cached file Dj. If there is a cache fail to see, the request is redirected to the top level site, which send the location replica site catalogue for that data file. After delivery such information, the site will update perfectly its NRC table and sends the request to the site's adjacent cache site for that data file.

4. Localized Data Catching Algorithm:

In Localized catching Algorithm the two important factors are consider:

4.1. Reduction in access cost of caching a data file:

Reduction in access cost as the effect of caching a data file at a site is the decrease in access cost given by the following: access occurrence in limited access traffic observed by the site × distance to the adjacent replica site.

4.2. Increase in access cost of deleting a data file:

Increase in access cost as the effect of deleting a data file at a site is the increase in right to use cost given by the following: access frequency in limited access traffic observed by the site × distance to the second-nearest Replica site.

CONCLUSION

Our goal is to replicate data files in data Intensive systematic application, to decrease the file access time with the concern of limited storage space of Grid sites. We suggest a centralized algorithm with performance assurance, we also suggest a distributed algorithm where in Grids sites react strongly to the Grid status and create intelligent caching decisions. Using Action script simulator of adobe flash, a distributed Grid simulator, we show that the distributed replication technique considerably outperforms a well-liked existing replication technique.

REFERENCES

[1] L.Qiu, V.N.Padmanabhan and G.M.Voelkar. On the placement of web server replicas. In proc. of IEEE Conference on Computer Communications (INFOCOM),2001.
[2] S. M. Park, J. H. Kim, Y. B. Lo, and W. S. Yoon. Dynamic data grid replication strategy based on internet hierarchy. In Proc. of Second International
Workshop on Grid and Cooprative Computing at GCC 2003.
[3] A. Aazami, S. Ghandeharizadeh, and T. Helmi. Near optimal number of replicas for continuous media in ad-hoc networks of wireless device . In Proc International Workshop on Multimedia Information System , 2004.
[4] U. Cibej, B. Slivnik, and B. Robic. The complexity of static data replication in data grids. Parallel Computing , 31(8-9):900-912, 2005.
[5] M. Tang, B.-S. Lee, C.-K. Yeo, and X. Tang. Dynamic replication algorithms for the multi-tier grid . Future Generation Computer Systems,
21:775-790, 2005.
[6]Dharma Teja Nukarapu et. al. On data replication . In proc.of IEEE Conference on data replication in data intensive scientific application with performance guarantee in 2011.

