Journal of Network and Computer Applications 35 (2012) 1297–1303

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

A dynamic replica management strategy in data grid

Najme Mansouri*, Gholam Hosein Dastghaibyfard

Department of Computer Science and Engineering, College of Electrical and Computer Engineering, Shiraz University, Molla Sadra Avenue, Shiraz, Iran

ARTICLE INFO

Article history:
Received 12 August 2011
Received in revised form
9 December 2011
Accepted 25 January 2012
Available online 1 February 2012

Keywords:
Data grid
Data replication
Number of requests
Simulation

ABSTRACT

Data Grid provides scalable infrastructure for storage resource and data files management, which supports several large scale applications. Due to limitation of available resources in grid, efficient use of the grid resources becomes an important challenge. Replication is a technique used in data grid to improve fault tolerance and to reduce the bandwidth consumption. This paper proposes a Dynamic Hierarchical Replication (DHR) algorithm that places replicas in appropriate sites i.e. best site that has the highest number of access for that particular replica. It also minimizes access latency by selecting the best replica when various sites hold replicas. The proposed replica selection strategy selects the best replica location for the users' running jobs by considering the replica requests that waiting in the storage and data transfer time. The simulated results with OptorSim, i.e. European Data Grid simulator show that DHR strategy gives better performance compared to the other algorithms and prevents unnecessary creation of replica which leads to efficient storage usage.

1. Introduction

In recent years, applications such as bioinformatics, climate transition, and high energy physics produce large datasets from simulations or experiments. Managing this huge amount of data in a centralized way is ineffective due to extensive access latency and load on the central server. In order to solve these kinds of problems, Grid technologies have been proposed. Data Grids aggregate a collection of distributed resources placed in different parts of the world to enable users to share data and resources (Chervenak et al., 2000; Allcock et al., 2001; Foster, 2002; Worldwide Lhc Computing Grid, 2011).

Data replication is an important technique to manage large data in a distributed manner. The general idea of replication is to place replicas of data at various locations. Data replication has been used in database systems (Wolfson and Milo, 1991), parallel and distributed systems (Bae and Bose, 1997; Loukopoulos et al., 2005; Rehn-Sonigo, 2007; Tzeng and Feng, 1996), mobile systems (Hara, 2001; Tu et al., 2006) and Data Grid systems (Abawajy, 2004; David, 2003; Rahman et al., 2008; Ranganathana and Foster, 2001; Stockinger et al., 2001). There are three key issues in all the data replication algorithms which are replica placement, replica management and replica selection. Placing the replicas in the appropriate site reduces the bandwidth consumption and reduces the job execution time.

Each grid site has its own capabilities and characteristics; so, choosing appropriate site from many sites that have the required data is an important decision. The response time is an essential parameter that influences the replica selection and thus the job turnaround time. Replica management is the process of creating or deleting replicas in Data Grid. To create a replica, we have to answer some important questions, such as, which file should be replicated? Where the file should be stored? and finally when should the replicas be created? Generally, replication algorithms are either static or dynamic. In static approaches the created replica will exist in the same place till user deletes it manually or its duration is expired. On the other hand, dynamic strategies create and delete replicas according to the changes in grid environments, i.e. users' file access pattern.

Meanwhile, even though the memory and storage size of new computers are ever increasing, they are still not keeping up with the request of storing large number of data. Hence methods needed to create replicas that increase availability without using unnecessary storage and bandwidth. In this work a novel data replication strategy, Dynamic Hierarchical Replication (DHR) is proposed. DHR extends proposed algorithm in Horri et al. (2008) and selects best replicas when various sites hold replicas of datasets. The proposed replica selection strategy selects the best replica location for the users' running jobs by considering the replica requests that waiting in the storage and data transfer time. DHR also stores each replica in an appropriate site i.e. best site in the requested region that has the highest number of access for that particular replica. The simulated results of DHR with OptorSim, show that DHR outperforms over current strategies about 37% and prevents unnecessary creation of replica which leads to efficient storage usage.

* Corresponding author. Tel.: +98 915 3624299; fax: +98 711 6474605.
E-mail addresses: najme.mansouri@gmail.com, mansouri@cse.shirazu.ac.ir (N. Mansouri).

1084-8045/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2012.01.014

The rest of the paper is organized as follows. Section 2 gives a brief introduction of previous work on data replication for grids. In Section 3, Dynamic Hierarchical Replication (DHR) algorithm is proposed. Section 4 shows simulation results. Finally, conclusions and future research works are presented in Section 5.

2. Related work

Recently, modeling Data Grid environments and simulating different data replication strategies as well as basic file replication protocols (Stockinger et al., 2001) has drawn researchers' attention. Rahman et al. (2008) presented an algorithm for replica selection using a simple technique called the k-Nearest Neighbor (KNN). The KNN rule chooses the best replica for a file using previous file transfer logs. They also suggested a predictive way to estimate the transfer time between sites. Accordingly, one site can request the replica from a site which has minimum transfer time. They showed neural network (with back propagation, to train the network) predictive technique outperforms multi regression model.

Dogan (2009) evaluated the performances of eight dynamic replication strategies under different Data Grid settings. The simulation results show that the file replication policy chosen and the file access pattern have great influence on the real-time Grid performance. Fast Spread-Enhanced was the best of the eight algorithms considered. Also, the peer-to-peer communication was indicated to be very profitable in boosting the real-time performance.

Bsoul et al. (2011) proposed a dynamic replication strategy that takes into account the number and frequency of requests, the size of the replica, and the last time the replica was requested. This algorithm is a modified version of Fast Spread replication strategy (Ranganathana and Foster 2001) that holds valuable replicas while the other less important replicas are replaced with more important replicas. A dynamic threshold is used to determine if the requested replica should be stored at each node along its path to the requester. They claim their algorithm has better performance comparing with Fast Spread with LRU and Fast Spread with LFU.

Zhong et al. (2010) have presented a dynamic replica management strategy. It consists of the creation strategy of dynamic replica that can automatically increase replica based on the frequency of the file access, the selection strategy of replica based on the GridFTP and the replacement strategy of the replica combining with the establishment time, number of access and file size. They claim the proposed replica management strategy has much better performance than the five built-in replica management strategies in OptorSim.

Ranganathana and Foster (2001) have proposed six distinct replica strategies (No Replica, Best Client, Cascading Replication, Plain Caching, Caching plus Cascading Replica and Fast Spread) for multi-tier data. They also introduced three types of localities, namely:

- Temporal locality: The files accessed recently are much possible to be requested again shortly.
- Geographical locality: The files accessed recently by a client are probably to be requested by adjacent clients, too.
- Spatial locality: The related files to recently accessed file are likely to be requested in the near future.

These strategies evaluated with different data patterns: first, access pattern with no locality. Second, data access with a small degree of temporal locality and finally data access with a small degree of temporal and geographical locality. The results of simulations indicate that different access pattern needs different replica strategies. Cascading and Fast Spread performed the best in the simulations. Also, the authors combined different scheduling and replication strategies.

Park et al. (2004) presented a Bandwidth Hierarchy based Replication (BHR) which decreases the data access time by maximizing network-level locality and avoiding network congestions. They divided the sites into several regions. Network bandwidth between the regions is lower than the bandwidth within the regions. So, if the required file is placed in the same region, its fetching time will be less. BHR strategy has two deficiencies, first if replica exists within the region it terminates, and second replicated files are placed in all the requested sites not the appropriate sites.

Nukarapu et al. (2011) have proposed a data replication strategy that has a provable theoretical performance guarantee and can be implemented in a distributed and practical manner. They also proposed a distributed caching strategy, which can be easily adopted in a distributed system such as Data Grids. The key point of their distributed strategy is that when several replicas are available, each site keeps track of the closest replica. Their simulation result show distributed replication algorithm significantly outperforms popular existing replication strategy under various network parameters.

Shorfuzzaman et al. (2010) proposed Popularity Based Replica Placement (PBRP) strategy in a hierarchical data grid which is guided by file "popularity". The "popularity" of a file is determined by its access rate by the clients. It places replicas close to clients to decrease data access. The effectiveness of PBRP depends on the determination of a threshold value related to file popularity. They also proposed Adaptive-PBRP (APBRP) that calculates this threshold dynamically based on data request arrival rates. Their simulation results show that PBRP performs better than other dynamic replication methods in terms of both job execution time and average bandwidth consumption.

Khanli et al. (2011) proposed Predictive Hierarchical Fast Spread (PHFS), a dynamic replication algorithm based on Fast Spread in multi-tier Data Grid. PHFS tries to predict user's subsequence component to adapt the replication configuration with the available condition, to increase locality in access. The conceptual basis for PHFS was the users who worked on the same context may be requested some files with high probability in the future. One of the main results is that the PHFS algorithm is appropriate for applications in which the clients work on a context for some duration of time and their requests are not random. The results of simulation also show that PHFS has better performance and lower latency comparing to common Fast Spread.

3. Proposed replication algorithm

In this section, first the 3-Level Hierarchical Algorithm is presented, and then network structure is described, and finally a novel Dynamic Hierarchical Replication (DHR) algorithm is proposed.

3.1. 3-Level hierarchical algorithm

Horri et al. (2008) considered a hierarchical network structure that has three levels. First level are Regions that are connected through internet i.e. have low bandwidth. Second level comprises LAN's (local area network) within each region that have moderately higher bandwidth comparing to the first level. Finally the third level are the nodes (sites) within each the LAN's, that are connected to each other with a high bandwidth.

3LHA first checks replica feasibility. If the requested file size is greater than SE size, file will be accessed remotely. It among the candidate replicas selects the one that has the highest bandwidth