A Fair Replica Placement for Parallel Download on Cluster Grid

Chih-Ming Wang[1], Chu-Sing Yang[2], and Ming-Chao Chiang[1]

[1] Department of Computer Science and Engineering,
National Sun Yat-sen University, Kaohsiung, Taiwan, R.O.C.
{wangcm,mcchiang}@cse.nsysu.edu.tw
[2] Department of Electrical Engineering,
National Cheng Kung University, Tainan, Taiwan, R.O.C.
csyang@mail.ee.ncku.edu.tw

Abstract. Grid technologies congregate numerous computers to provide powerful computing and massive storage. In data-intense applications, Data Grids are developed to cope with the efficiency of data access. Replication is one of the methods to elevate the access performance. When a file is replicated, it can be downloaded from all the nodes with that file in parallel, thus reducing the access latency. Therefore, a fair and adaptive replication strategy for high speed transmission is important. In this paper, we design such a strategy to duplicate popular files to beneficial nodes in the grid networks. All the users deserve the same quality of transmission. The contributions of our mechanism are to average the transmission cost and evenly distribute the workload for download. Simulation results are also given to demonstrate the performance of our replication strategy.

1 Introduction

Grid computing is an emerging technology for scientific, physical, and biological applications, focusing on distributed computing and resource sharing. In recent year, the most widely used software for constructing grid networks is certainly the Globus Toolkit developed by the Globus Alliance [1]. All kinds of resources are congregated via the broadband network to cooperate to get the work done quickly. Nowadays, the exploiting empires include IBM, Sun, SGI, NEC, Entropia, and so on. They almost dominate the development of grid.

Cluster takes advantage of the local resources to cooperate, but grid breaks this restriction. It connects all the resources available on all the networks worldwide by the supporting software. The differences between Grid and cluster are primarily the dynamic and heterogeneous resources, which can be located anywhere for scalability. In a grid system, a job can be divided into many sub-jobs and dispatched to multiple nodes to get it computed in parallel. When the results are reassembled in one node, the job is accomplished. The goals are to reduce the job completion time and balance the workload between all the grid nodes. Moreover, not only the computing power, but also all kinds of resources can be shared, which include both hardware and

T. Enokido, L. Barolli, and M. Takizawa (Eds.): NBiS 2007, LNCS 4658, pp. 268–277, 2007.

software such as CPU, memory, storage, files, databases, and so on. Grid efficiently integrates the various resources via the networks [2]. In a storage system, files are usually replicated to resolve the access latency problem. Identical files may be stored in different storage subsystems for fault tolerance, parallel download, or fast discovery. Grids continue all the characteristics of distributed systems and storage systems. Consequently, research on grids can be divided into that on the computing grid and that on the data grid [3]. The computing grid offers a model for solving massive computational problem. The data grid focuses on three issues: (1) The replica management, which manages the creation or deletion of replica on a massive storage [9][11][12][13]. (2) The replica selection [4], which chooses the best nodes to obtain the resources for download or cooperation [7][10]. (3) The replica catalog, which maps logical file name to physical location address for discovery [8]. In this paper, our focus will be on the replica management.

Bandwidth is certainly one of the most important resources in a data grid [5][6]. A popular file may become a hotspot and thus cause bottleneck at a node because there are a lot of downlinks occurring at the same time. Obviously, the number of user requests is also a crucial factor to affect the bandwidth. Replication is one of the methods to elevate the access performance. When the bottleneck occurs at a node, it means that that node is overloaded and requires to be alleviated. On the other hand, when a file is requested by a lot of users, it indicates that the file is very popular and should be duplicated. But the question is where the replica should be placed? In order to reduce the transmission cost, we must know that a high bandwidth node may not have numerous requests whereas a node with numerous requests may not have sufficient bandwidth. Bandwidth and user requests need to be considered simultaneously as far as downloading is concerned. Therefore, designing an efficient replication strategy to put the replica to an appropriate node is very important. In this paper, we propose a replica placement scheme to overcome the bottleneck problem and to average the access latency for fairness on cluster grid system.

The rest of this paper is organized as follows. Section 2 is the related work about replica placement. In Section 3, we state the problem of replication on cluster grid. Our algorithm is proposed in Section 4. Section 5 shows the simulation results and gives analysis. Section 6 concludes the paper and suggests some possible future research works.

2 Related Works

As previous studies show, replica placement problems are usually handled based on either the centralized or hierarchical network structure. In [11], NoReplication, BestClient, Cascading, PlainCaching, Caching plus Cascading and FastSpread are proposed to estimate the performance. Among these schemes, Cascading and FastSpread perform the best, but with a tradeoff. FastSpread performs better than Cascading in terms of the bandwidth consumption and response time when the access patterns are totally random, but the storage overhead is higher than Cascading. In [13], the authors propose a new placement algorithm for the replicas so that the workload of user requests among these replicas is balanced. In [11] and [13], only the