Available online at www.sciencedirect.com
ScienceDirect
Future Generation Computer Systems 25 (2009) 695–703
FGCS
FUTURE GENERATION COMPUTER SYSTEMS
www.elsevier.com/locate/fgcs

A Recursively-Adjusting Co-allocation scheme with a Cyber-Transformer in Data Grids☆

Chao-Tung Yang[a,*], I-Hsien Yang[a], Shih-Yu Wang[a], Ching-Hsien Hsu[b], Kuan-Ching Li[c]

[a] *High Performance Computing Laboratory, Department of Computer Science and Information Engineering, Tunghai University, Taichung 40704, Taiwan, ROC*
[b] *Department of Computer Science and Information Engineering, Chung Hua University, Hsinchu 300, Taiwan*
[c] *Parallel and Distributed Processing Center, Department of Computer Science and Information Management, Providence University, Taichung 43301, Taiwan, ROC*

Received 16 February 2006; received in revised form 4 November 2006; accepted 16 November 2006
Available online 21 January 2007

Abstract

A co-allocation architecture was developed in order to enable parallel downloads of datasets from multiple servers. Several co-allocation strategies have been coupled and used to exploit rate differences among various client–server links and to address dynamic rate fluctuations by dividing files into multiple blocks of equal sizes. However, a major obstacle, the idle time of faster servers having to wait for the slowest server to deliver the final block, makes it important to reduce differences in finish times among replica servers. In this paper, we propose *Recursively-Adjusting Co-Allocation*, a dynamic co-allocation scheme for improving data transfer performance in Data Grids. The experimental results show that our approach can reduce the idle time spent waiting for the slowest server and decrease data transfer completion times. We developed Cyber-Transformer, a new toolkit with a friendly GUI interface that makes it easy for inexperienced users to manage replicas and download files in Data Grid environments. We also provide an effective scheme for reducing the cost of reassembling data blocks.

Keywords: Data Grid; Globus; GridFTP; Co-allocation; Recursively-Adjusting; Data transfer

1. Introduction

Data Grids aggregate distributed resources for solving large-size dataset management problems [1,2,4,7,9,25–30]. Most Data Grid applications execute simultaneously and access large numbers of data files in Grid environments. Certain data-intensive scientific applications, such as high-energy physics, bioinformatics applications and virtual astrophysical observatories, entail huge amounts of data that require data file management systems to replicate files and manage transfers and distributed data access. The Data Grid infrastructure integrates data storage devices and data management services into a grid environment that consists of scattered computing and storage resources, perhaps located in different countries/regions yet accessible to users [2,9,27–30].

In Data Grid environments, access to distributed data is typically as important as access to distributed computational resources. Distributed scientific and engineering applications require transfers of large amounts of data between storage systems, and access to large amounts of data generated by many geographically distributed applications and users for analysis and visualization, among others. Recently, large-scale data-sharing scientific communities such as those described in [1,4] have begun using this technology to replicate their large datasets over several sites. Replicating popular content in distributed servers is widely used in practice [11,13,15].

One way to improve download speeds is to determine the best replica locations using replica selection techniques [15]. This method selects servers that will provide optimum transfer rates because bandwidth quality can vary unpredictably due to the shared nature of the Internet. Another way is to use co-allocation technology [13] to download data. Co-allocation

☆ This paper is supported in part by National Science Council, Taiwan, ROC, under grants no. NSC94-2213-E-029-002 and NSC95-2221-E-029-004.
* Corresponding author. Tel.: +886 4 23590415; fax: +886 4 23591567.
E-mail addresses: ctyang@thu.edu.tw (C.-T. Yang), g922906@thu.edu.tw (I.-H. Yang), g932813@thu.edu.tw (S.-Y. Wang), chh@chu.edu.tw (C.-H. Hsu), kuancli@pu.edu.tw (K.-C. Li).

0167-739X/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2006.11.005

of data transfers enables clients to download data from multiple locations by establishing multiple connections in parallel. Parallel downloading [21–24] is a technique used to fetch and download files from multiple sources including Web servers, file servers, P2P nodes, etc. Parallel downloading has been integrated into many Internet applications and has become the core of many P2P systems. It speeds up download time and eliminates the server selection problem [21,22,24].

Downloading large datasets from several replica locations may result in varied performance rates, because the replica sites may have different architectures, system loadings and network connectivity. Bandwidth quality is the most important factor affecting transfers between clients and servers since download speeds are limited by the bandwidth traffic congestion in the links connecting the servers to the clients [17,18]. This method can improve the performance compared to the single-server cases and alleviate the internet congestion problem [13]. Several co-allocation strategies were provided in a previous work [13]. An idle-time drawback remains, since faster servers must wait for the slowest server to deliver its final block. Therefore, it is important to reduce the differences in finish time among replica servers.

In this paper we present a new toolkit: Cyber-Transformer. It is integrated with the Information Service, Replica Location Service and Data Transfer Service. Its friendly GUI interface makes it easy for inexperienced users to manage replicas and download files in Data Grid environments. And we propose *Recursively-Adjusting Co-Allocation*, a dynamic co-allocation scheme based on a Data Grid data transfer architecture, that reduces the idle time spent waiting for the slowest server and improves data transfer performance. Rather than picking a specific server to download a file, a number of servers are selected and content is downloaded in parallel from all of them, getting different portions of the file content from different servers. Experimental results show that our approach is superior to previous methods and achieved the best overall performance.

The remainder of this paper is organized as follows. The background of the co-allocation architecture and related studies is presented in Section 2. Our research approach is outlined in Section 3 and experimental results and a performance evaluation of our scheme are presented in Section 4. Section 5 concludes this research paper.

2. Background and related work

First, the concept of a Data Grid is stated in Section 2.1. Then, the co-allocation architecture and related work are described in Section 2.2.

2.1. Data Grid

Data Grids consist of scattered computing and storage resources located in different countries/regions yet accessible to users [7]. In this study we used the grid middleware, Globus Toolkit [8,10,12], as the Data Grid infrastructure. The Globus Project [8,12,16] provides software tools collectively called The Globus Toolkit that make it easier to build computational Grids and Grid-based applications. Many organizations use the Globus Toolkit to build computational Grids to support their applications. The composition of the Globus Toolkit can be pictured as three pillars: Resource Management, Information Services and Data Management. Each pillar represents a primary component of the Globus Toolkit and makes use of a common foundation of security. GRAM implements a resource management protocol, MDS implements an information services protocol and GridFTP implements a data transfer protocol. They all use the GSI security protocol at the connection layer [10,11,16].

One of its primary components is MDS [5,8,10,12,20], which is designed to provide a standard mechanism for discovering and publishing resource status and configuration information. It provides a uniform and flexible interface for data collected by lower-level information providers in two modes: static (e.g., OS, CPU types, system architectures) and dynamic data (e.g., disk availability, memory availability and loading). And it uses GridFTP [1,8,12], a reliable, secure and efficient data transport protocol to provide efficient management and transfer of terabytes or petabytes of data in a wide-area, distributed-resource environment.

The Globus Project surveyed available protocols and technologies, implemented some prototypes, settled on using FTP and its existing extensions as a base, and then extending it again to add missing required functionality. The Globus alliance proposed a common data transfer and access protocol called GridFTP that provides secure, efficient data movement in Grid environments. This protocol, which extends the standard FTP protocol, provides a superset of the features offered by the various Grid storage systems currently in use.

As datasets are replicated within Grid environments for reliability and performance, clients require the abilities to discover existing data replicas and create and register new replicas. A Replica Location Service (RLS) [3,15] provides a mechanism for discovering and registering replicas. Several prediction metrics have been developed to help replica selection. For instance, Vazhkudai and Schopf [14,16,17] used past data transfer histories to estimate current data transfer throughputs.

2.2. Co-allocation architecture

Parallel downloading is a technique used to fetch and download files from multiple sources including Web servers, file servers, P2P nodes, etc. Parallel downloading has been integrated into many Internet applications and has become the core of many P2P systems. It speeds up download time and eliminates the server selection problem. Another mechanism, striped transfer, distributes a file's data over multiple independent data nodes, and transfers over multiple data connections. In GridFTP transfers, striping refers to having multiple network endpoints at the source, destination or both participating in transferring the same file. This is normally accomplished using a cluster with a parallel shared file system. Each node in the cluster reads a section of the file and sends it over the network. This mode of transfer is necessary if the