Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 2, February 2014, pg.80 – 85

RESEARCH ARTICLE

A NOVEL ON FAST PARALLEL FILE TRANSFER USING REPLICATION

K.Sabarigirivason[1]
M.E Computer science and Engineering,
Sri Eshwar college of Engineering,
Coimbatore, Tamilnadu, India.
sabari1151991@gmail.com

R. Giridharan[2]
M.E Computer science and Engineering,
Sri Eshwar college of Engineering,
Coimbatore, Tamilnadu, India.
giridharanmecse@gmail.com

Abstract- Data replication is the most critical component of data-intensive grid computing environment. The need for data replication arises in various areas of data analysis such as high-energy physics, bio-informatics, climate modeling and astronomy. In addition to grid data environments, data replication is the key part of various data sharing applications such as digital libraries, persistent archival environment and content distribution. Parallel file replication where a large file needs to be simultaneously replicated to multiple sites is an integral part of data-intensive grid environment. Propose a tool that creates multiple distribution trees by pipelining point-to-point transfer and optimizes the file replication time to multiple sites. One of the key parts in data replication is the replica catalog that manages the mappings for files from the hierarchical namespace to one or more physical file locations, thus providing an efficient and transparent file sharing on a Grid. Managing and coordinating the data movement process is the crucial performance issue.

Keywords— Data replication, data intensive, grid computing, pipelining, replica

I. INTRODUCTION

Designing cost-efficient, secure network for transmitting data in parallel from one place to another it is a challenging problem because sending data fastly and data security will lead which will prevent from identification of data loss and using of GridFTP(File Transfer Protocol). Timely data replication is one of the most critical components of data-intensive grid computing environment. The need for this component arises in various areas of data analysis such as high-energy physics, bio-informatics, climate modeling and astronomy. For example, terabytes and petabytes of data produced by CERN have to be shared and accessed by the high-energy physics community around the world. In addition to grid data environments, data replication is the key part of various data-sharing applications such as digital libraries, persistent archival environment and content distribution. In addition to these strategies, the network (transport) mechanism used in the actual movement of the data plays an equally important role in the overall performance. The access time in data replication in general depends upon how the network resources are utilized by the data transport mechanism.

GridFTP is designed for point-to-point reliable data transport based on file splitting and opening multiple parallel TCP streams.

I. Some data replication scenarios require point-to-multipoint data distribution. For example, in parallel data analysis, a file is often replicated from its source to multiple cluster nodes in parallel. Digital libraries, persistent archives and content distribution also require this mode of distribution. Fast Parallel File Replication of the Data Grid architecture (FPFR) tool that can significantly reduce the overall time required for parallel file replication to multiple sites with efficient coordination of any point-to-point transport mechanisms. FPFR uses the information about the network resources (e.g., from Network Weather Service (NWS) to spatially diversify multiple transport sessions over the network. FPFR(Fast Parallel File Replication) tool can be a part of the resource management layer of the Data Grid architecture. A data grid is an architecture or set of services that gives individuals or groups of users the ability to access, modify and transfer extremely large amount of geographically distributed data.

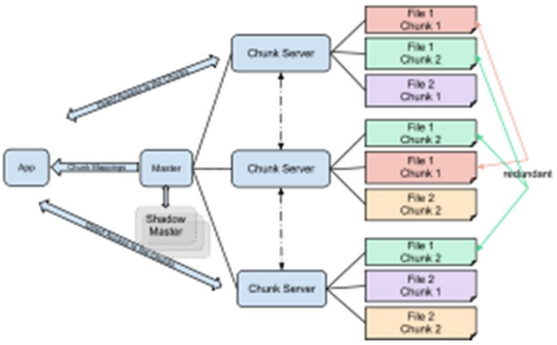

Fig.1 Data grid

II. OBJECTIVE

File Replication provides data which needs to be parallely transmitted by Using Grid FTP which provides Reliable Fast and Security and along with it we have proposed a key System where which also Provides the Security and this kind of Transmission is in which it will involve to the advancement of Multithreading and along with it is mostly Concerned with Servers

III. EXISTING SYSTEM

Development in multimedia processing and network technologies has facilitated the distribution and sharing of multimedia through networks, and the security demands increase with the rowing of the network and multimedia technologies. The creation, coding, and delivery of multimedia data constitute a unique data path.

In visible watermarking of images, a secondary image (the watermark) is embedded in a primary (host) image such that watermark is intentionally perceptible to a human observer whereas in the case of invisible watermarking the embedded data is not perceptible, but may be extracted/detected by a computer program.

This raises serious security concerns since the receivers/subscribers do not know what processes have been applied to multimedia data, and neither do they know whether this copy comes from a trusted source. Therefore, it is critical to provide forensic tools to identify the history of operations applied to multimedia data.

IV. DRAWBACKS

- Propagating false data or redundant data are costly
- Depleting limited network resources
- Wasting response efforts
- Time Consuming

V. PROPOSED SYSTEM

Various architectures are being proposed and developed to manage data replication (Data Grid[1], European Data Grid[2] Datafarm[3]. Data Grid Reference Architecture (DGRA) covers important architectural components and their functionalities. One of the key parts in data replication is the replica catalog that manages the mappings for files from the hierarchical namespace to one or more physical file locations, thus providing an efficient and transparent file sharing on a Grid. Managing and coordinating the data movement process is the crucial performance issue. Current strategies use data locality, access time and pattern to decide whether to move computation to data source or vice versa.

We thus propose a Fast Parallel File Replication (FPFR) tool that can significantly reduce the overall time required for parallel file replication to multiple sites with efficient coordination of any point-to-point transport mechanisms. FPFR uses the information about the network resources (e.g., from Network Weather Service (NWS) [9]) to spatially diversify multiple transport sessions over the network. FPFR can be a part of the resource management layer of the Data Grid architecture. We have implemented FPFR and evaluated its performance benefit on (i) LAN connecting a few PCs in a cluster and (ii) also on Internet (on a worldwide scale) connecting over 50 PCs in a cluster.

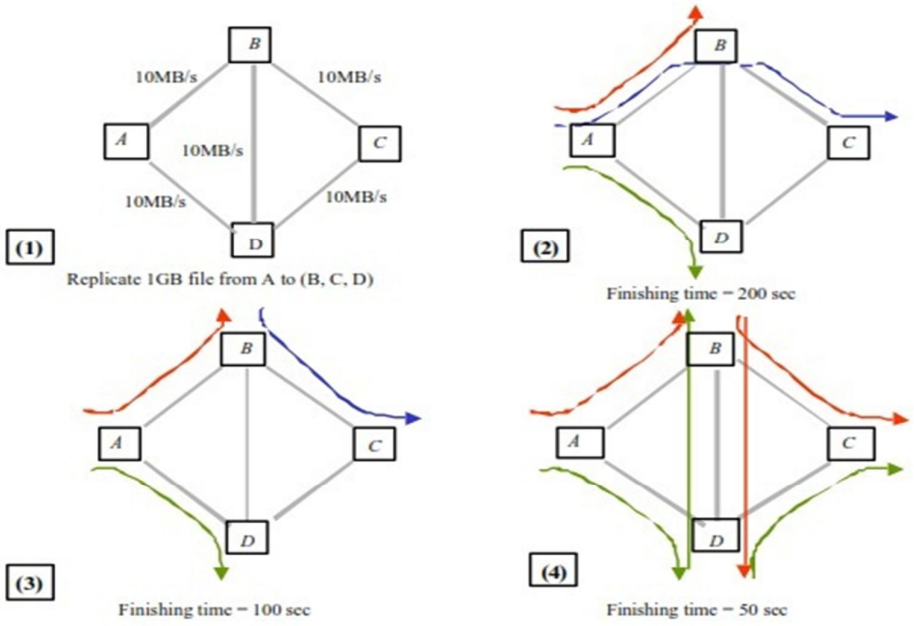

Fig 2

A. **Rationale**

This the rationale is illustrated as in fig.2. Section (1) shows four cluster nodes. The goal is to replicate a 2GB file from node A to rest of the nodes in minimum time. Section (2) shows that the first process can be applied directly using point-to-point transport tool GridFTP. Here three transfer sessions are created from A to all end points with the link A->B shared by two sessions each will gets 10MB/s peak capacity therefore maximum replication process is 200 sec. Section (3) shows an improvement by creating a single tree from A connecting all other nodes avoid sending multiple session through same link. To have concurrent session one needs to split files into small sub-files and create sessions for each sub-files in a pipelined manner. Here it can utilize peak capacity 100MB/s resulting in finish time of 100 sec. Finally Section (4) provides further improvement by utilizing the spatial diversity of the network topology. Here we use two trees as shown in fig 2. The first tree is (A->B->(C, D)) and the second tree is (A->D->(B, C)). Here the finishing time is 50 sec, whish correspond to the acceleration of distribution by a factor of 4.

VI. MODULES

MODULE DESCRIPTION

A. *USER INTERFACE DESIGN*

The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals—what is often called user-centered design. Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design may be utilized to support its usability. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs. Interface design is involved in a wide range of projects from computer systems, to cars, to commercial planes; all of these projects involve much of the same basic human interactions yet also require some unique skills and knowledge.

B. *TORRENT TRACKER*

A Torrent tracker is a server that assists in the communication between peers using the BitTorrent protocol. It is also, in the absence of extensions to the original protocol, the only major critical point, as clients are required to communicate with the tracker to initiate downloads. Clients that have already begun downloading also communicate with the tracker periodically to negotiate with newer peers and provide statistics; however, after the initial reception of peer data, peer communication can continue without a tracker.

C. *TORRENT CLIENT*

Many subsequent clients have been at least partially based on it. Not all clients were originally built for Torrent, having added support for the protocol later on. There have been attempts to package malware as Torrent clients, probably due to the availability of many legitimate clients and users' willingness to try new ones.

VII. DESCRIPTION

The system prototype is divided into control plane and data plane. The control plane is in charge of session state management, and distributed tree state managements in data transfer sessions. The data plane takes care of data replication and forwarding, congestion control on a tree and load balancing between trees. Session state is a 5-tuple of {session_id, source, destination lists, source file name list, and source-to-destination file name map}. It reflects the single -source-multiple-destinations parallel replication nature of our system.

In our data transfer session model, the destination list is fixed during the life of a session. This model is different from multicast-based streaming and peer-to-peer file download model, where the destinations can join and leave in a single data session. We choose this model because there are varieties of such applications in file replication of storage system, content distribution network and data replication in grid environments, where the destinations are chosen in advance manually or automatically by out-of-band algorithm and the destination list are fixed in a data transfer session.

The source file name is the file name in the data source and the source-to-destination file name map translates original file name to destination file names. Such file name presents the file physical location, consists of node name, directory name and file name, and is encoded as URI. This map allows the file created in each destination to be different from the original in name and directory. This is a very important feature when file systems structure on different sites are different from each other. It shares the same vision as the logic file name to physical file name map in replica catalog service. Our system can integrate with such replica catalog service with trivial translation. (Note: replica catalog service maintains a map from logic file name to target file names, while in our replication session, the session maintain a map from file name at source to file names at multiple destinations). In each data transfer session, all files share same destination node list, and all files are transferred in a batch.

The control plane distributes the tree soft state to all forwarding nodes using explicit signaling, and the state is refreshed periodically. The currently implemented signaling is similar to explicit source multicast in that the tree structure is encoded in a message and the message propagation follows the encoded tree structure.

The soft state and periodical state refresh simplified the system implementation and improved its robustness. Session manager/controller currently assumes the central role of control plane in our implementation. It accepts data feeds from monitoring system, maintains session state and orchestrates tree state signaling on each forwarding node. The session manager is an independent component; it interacts with session agent in each node through distributed messaging. Our implementation of session manager supports both centralized and distributed session managements. In centralized mode, a single session controller orchestrates all sessions from different sources. In distributed session management, a session manager associates/collocates with each node and only manages session originating from the source node, which distributes session management load uniformly on each data-source node.

Our distributed session management scheme scales independent of network sizes. The data plane consists of data forwarding and replication, congestion control and load balancing between trees. The data plane on each node maintains a forwarding table mapping a forwarding tree to the next hops. This forwarding table is signaled by session management process in control plane. Data frame is tagged with. To avoid overflow in a tree node due to variations of network bandwidth, the congestion control described limits the tree forwarding speed to the bandwidth of slowest tree link. In order to take advantage of forwarding capacity on all trees, while sending each sub-file to a tree at source, we always choose the tree with the largest available forwarding buffer.

The data plane of each node consists of three components: sender, forwarder and receiver. Forwarder is a multicast component that can duplicate incoming data and forward it to multiple outgoing connections according to the installed multicast routing table. A forwarder can be activated on non-destination nodes. *Sender* is activated on the file source. It fragments data files in to sub-files, sends sub-files to multiple trees and performs load balancing between trees. *Receiver* is activated on each destination.

VIII. CONCLUSIONS

This tool by creating multiple distribution trees through pipelining point-to-point transfer sessions minimizes the net file replication time. We introduce *Fast* Replica for efficient and reliable replication of large files in the Internet environment. Fast Replica partitions an original file into a set of sub files and uses a diversity of Internet paths among the receiving nodes to propagate the sub files within the replication set in order to speed up the overall download time for the original content. To analyze and validate future optimization for *Fast Replica*, a large-scale Internet environment or tested is needed. In recent work, authors propose Model- Net as a comprehensive Internet emulation environment to evaluate Internet-scale distributed systems. A new initiative within the research community around PlanetLab is aiming to build a global tested for developing and accessing new network services. The introduction of such environments and large-scale testbeds will help to support interesting scalability experiments in the near future.

REFERENCES

[1]. Chervenak, A., Foster, I., Kesselman, C., Salisbury, C. and Tuecke, S." *The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Data Sets"*. J. Network and Computer Applications, 2001.
[2] EU Data Grid Project, "The Data Grid Architecture", DataGrid-12-D12.4-333671-3-0, 2001
[3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "*Grid Computing Networks: A Survey*" Computer Networks, vol. 38, no. 4, pp. 393-422, Mar. 2002.
[4] C. Vu, R. Beyah, and Y. Li, "*A Composite Event Detection in Grid Environment Networks,*" Proc. IEEE Int'l Performance, Computing, and Comm. Conf. (IPCCC '07), Apr. 2007.
[5] S. Uluagac, C. Lee, R. Beyah, and J. Copeland, "*Designing Secure Protocols for Grid Networks*" Wireless Algorithms, Systems, and Applications, vol. 5258, pp. 503-514, Springer, 2008
[6] Crossbow Technology, http://www.xbow.com, 2008. [5] G.J. Pottie and W.J. Kaiser, "*Integrated Network Sensors*" Comm. ACM, vol. 43, no. 5, pp. 51-58, 2000.
[7] R. Roman, C. Alcaraz, and J. Lopez, "*A Survey of Cryptographic Primitives and Implementations for Hardware-Constrained Sensor Network Nodes*" Mobile Networks and Applications, vol. 12,no. 4, pp. 231-244, Aug. 2007.
[8] Grid Datafarm, http://datafarm.apgrid.org/

Authors Bibliography

K.Sabarigirivason born in Coimbatore, Tamilnadu, India in 1991. He received B.E Degree in Computer Science and Engineering from Anna University, Coimbatore, India. He is pursuing M.E Degree in Computer Science and Engineering in Sri Eshwar College of Engineering Affiliated to Anna University, Chennai, Tamilnadu, India. He is a member of an IEEE Association. His research interests include, Networks and Datamining.

R.Giridharan born in Namakkal, Tamilnadu, India in 1989. He received B.E Degree in Computer Science and Engineering from Anna University, Chennai, India, He is pursuing M.E Degree in Computer Science and Engineering in Sri Eshwar College of Engineering Affiliated to Anna University, Chennai, Tamilnadu, India. He is an active member in IEEE.

