IJCSC Volume 4 • Number 2 September 2013 pp.171-173 ISSN-0973-7391

Major Developments in Grid Computing to Improve the Throughput

Ritu Baniwal
Department of Computer Sc. & Engg, GJUS&T Hisar
baniwalritu@gmail.com

Abstract: The growth of the internet and the revolutions in computing and storage technologies have led to the major developments in grid computing. Grid computing provides an environment for the resource sharing which includes direct access to computers, software, data and other resources. The computational services provide the users with different levels of expertise in diverse areas to perform the specified tasks efficiently. Recent milestone development in the history of grid computing is in the LHC experiment.
Keywords: Grid Computing, Multi-Hop Path Splitting, Multi-Pathing, GridFTP

1. INTRODUCTION

Computers are used for various scientific and engineering problems e.g. diagnosing medical conditions, forecasting the weather, and many others. But the evolutionary changes in the VLSI technology and microprocessor architecture, increasing utilization of idle capacity and greater sharing of computational resources have led the foudation for the grid computing. Grid computing provides the environment where the researchers and scientists can perform their research without regard to geographical location. They can interact with colleagues and can access and share data and computational resources.

Grid computing technologies have enabled scientists to share the data distributed across multiple sites. An example is the LHC [9] experiment at CERN. The data which is generated by a CMS experiment at LHC needs to be transferred to a Tier-1 site in the US where it is processed and then multicast onto many domestic US tier-2 sites [9]. NOWs (Network of Workstations) were in regular use for parallel computation. In a grid based computing, heterogeneous as well as homogeneous machines are used.

2. BASIC GRID ARCHITECTURE

The basic grid architecture consists of 5 layers shown in fig 1: the fabric, connectivity, resource, collective and application layers. The Fabric layer gives the information about the resources on which the sharing is to done. For this the Resource and Connectivity protocols are there.

The Connectivity layer defines core communication and authentication protocols network transactions [3]. Communication protocols deals with the exchange of data between the resources. Authentication protocols provide secure mechanisms for verifying the identity of users which is based on cryptography.

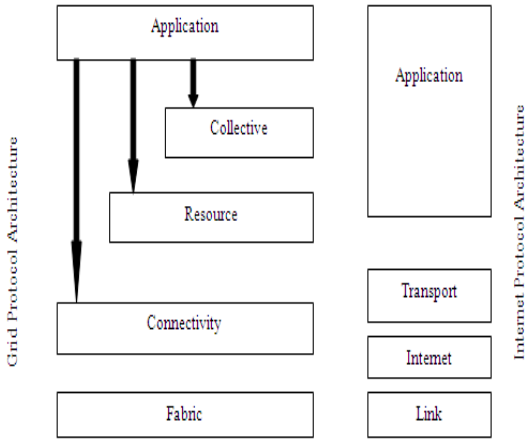

Fig 1: The Basic Grid architecture [3]

The Resource layer deals only with a single resource. The Collective layer which is not associated with any specific resource deals with the interactions across collections of resources.

3. IMPROVEMENTS

a. SCALABLE VIRTUAL ORGANIZATIONS

The real problem in distributed environment is resource sharing that underlies the Grid concept which solves the problem in virtual organizations. The sharing that we are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources [3]. A set of individuals and/or institutions which define such sharing rules form a virtual organization (VO) e.g. the application service providers.
The internet technologies address communication and information exchange among computers but these technologies do not provide approaches to the coordinated use of resources at multiple sites for computation. Enterprise distributed computing technologies such as CORBA and Enterprise Java enable resource sharing within a single organization. The Open Group's Distributed Computing Environment (DCE) supports secure resource sharing across sites [3].

b. DYNAMIC SCHEDULING APPROACH USING GridFTP

In many scientific applications there is a need to transfer very large amount of data from one set of machines to another set of machines in a wide-area network. In such a transfer the heterogeneity of the environment and the dynamic availability of shared resources must be handled. Dynamic scheduling approach minimizes the overall transfer time of data transfer requests. In dynamic scheduling approach, scheduling is done per chunk basis. Chunk is a portion of the file being staged to a destination machine. Transfer of chunks for a file can be inter-leaved with transfer of chunks for other files [4]. Also the simultaneous transfer of files from multiple file replicas can be done.
Grid computing technologies have enabled scientists to generate, store, and share data distributed across multiple sites. An example is the LHC [9] experiment at CERN. The data which is generated by a CMS experiment at LHC needs to be transferred to a Tier-1 site in the US where it is processed and then multi-cast onto many domestic US tier-2 sites. GridFTP protocol enables secure, reliable and high performance data movement [10].

c. MULTI-HOP PATH SPLITTING AND MULTI-PATHING OPTIMIZATIONS FOR DATA TRANSFERS

To improve the performance of data transfers over shared public networks the two optimization method –multi-hop path splitting and multi-pathing have been developed. As demand for handling very large data is increasing rapidly, the research is becoming data driven [5]. High-end Grid computing involves use of distributed collections of computational systems to analyze, process and visualize data resident in data repositories.
Multi-hop path splitting improves performance by replacing a direct TCP connection between the source and destination by multi-hop chain. It divides a TCP connection into a set of shorter connections by splitting it at multiple intermediate nodes.
Multi-pathing involves striping the data at the source and sending it across multiple overlay paths which lead better throughput because disjoint chunks of a file can be tranfered simultaneously to its destination. But there is a disadvantage of using this approach because independent overlay paths may share bottlenecks due to physical sharing of links and routers.
This approach uses a Path Determination Algorithm that computes a set of paths which can be collectively used to transfer a file from its source node to its destination node.

d. GRIDFTP PIPELINING

GridFTP pipelining addresses the lots of small files (LOSF) which persisted in the conventional implementations of GridFTP. Pipelining approach minimizes the amount of time between transfers. Pipelining allows the client to process multiple transfer commands in parallel.
The server processes the requests in the same order in which they are sent and acknowledgments are returned back to the client in the same order. The latency of transfer requests can be hidden by overlapping them with data transfers. After the first transfer request is sent the data begins to flow across the data channel. While the file transfer is in progress, the client sends the next 'n' files transfer requests in parallel to these requests [6]. Fig 2.1 and 2.2 shows the difference between the two approaches [6].

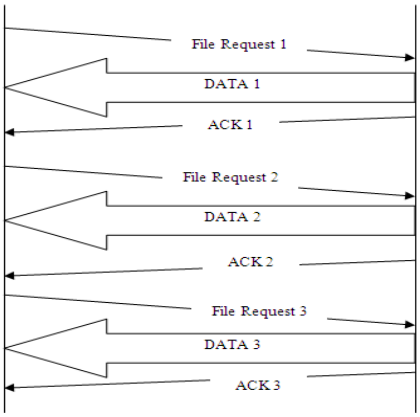

Fig 2.1: GridFTP file transfers with no pipelining

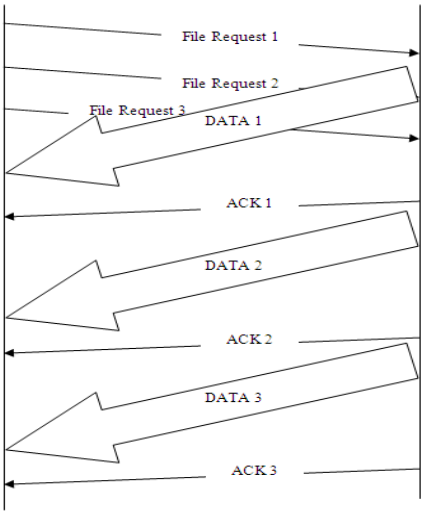

Fig 2.2: GridFTP file transfers with pipelining

4. CONCLUSION

Distributed computing applications use grids to tackle problems that cannot be solved on a single system. In high performance computing, the grid is used to schedule large numbers of loosely coupled or independent tasks which makes unused processor cycles e.g. idle workstations to work. Some on-demand applications use grid concept to access the resources that cannot be located locally in a convenient and efficient way. In data-intensive applications, data is maintained in geographically distributed databases, repositories and digital libraries. Collaborative applications are concerned with the shared use of computational resources such as data archives and simulations. To address such diverse demands in the high performance computing environment grid computing concept have emerged. Different methodologies have been adopted to achieve high-throughput. In GridFTP pipelining the transfer of requests are overlapped with the data transfers.

Recent milestone development in the history of grid computing is in the LHC experiment. Thousands of physicists at hundreds of laboratories and universities worldwide came together to design, create, operate, and analyze the products of a major detector at CERN, the European high energy physics laboratory [9]. Grid uses homogeneous as well as the heterogeneous set of machines.

REFERENCES

[1] Ian Foster, Carl Kesselman, "Computational Grids".
[2] Manish Parashar, Craig A. Lee "Grid Computing: Introduction and Overview.
[3] Ian Foster, Carl Kesselman, Steven Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", Intl J. Supercomputer Applications, 2001.
[4] Gaurav Khanna, Umit Catalyurek, Tahsin Kurc, Rajkumar Kettimuthu, P. Sadayappan, Joel Saltz, "A Dynamic Scheduling Approach for Coordinated Wide-Area Data Transfers using GridFTP".
[5] Gaurav Khanna, Umit Catalyurek, Tahsin Kurc, Rajkumar Kettimuthu, P. Sadayappan, Joel Saltz, Ian Foster "Multi-Hop Path Splitting and Multi-Pathing Optimizations for Data Transfers over Shared Wide-Area Networks using GridFTP", HPDC'08, June 23–27, 2008.
[6] John Bresnahan, Michael Link, Rajkumar Kettimuthu, Dan Fraser, Ian Foster, "GridFTP Pipelining", TERAGRID 2007 conference, Madison.
[7] Henri Casanova and Jack Dongarra, "Netsolve: A network server for solving computational science problems". Technical Report CS-95-313, University of Tennessee, November 1995.
[8] F. Douglis and J. Ousterhout, "Transparent process migration: Design alternatives and the Sprite implementation". Software-Practice and Experience, 21(8):757-85, 1991.
[9] "The Large Haldron Collider (LHC)", http://lhc.web.cern.ch/lhc/.
[10] W. Allcock, J. Bresnahan, R. Kettimuthu, and M. Link, "The globus striped gridftp framework and server", In SC '05: Proceedings of the 2005 ACM/IEEE conference on Supercomputing, Washington, DC, USA, 2005.