A High-Performance GriFTP Server at Desktop Cost

Samer Al-Kiswany [1], Armin Bahramshahry[1], Hesam Ghasemi[1],
Matei Ripeanu [1], Sudharshan S. Vazhkudai [2]
[1] *University of British Columbia, {matei, samera}@ece.ubc.ca*
[2] *Oak Ridge National Laboratory, vazhkudaiss@ornl.gov*

Abstract **--** We prototype a storage system that provides the access performance of a well-endowed GridFTP deployment (e.g., using a cluster and a parallel file-system) at the modest cost of single desktop.

To this end, we integrate GridFTP and a combination of dedicated but low-bandwidth (thus cheap) storage nodes and scavenged storage from LAN-connected desktops that participate intermittently to the storage pool. The main advantage of this setup is that it alleviates the server I/O access bottleneck. Additionally, the specific data access pattern of GridFTP, that is, the fact that data accesses are mostly sequential, allows for optimizations that result in a high-performance storage system. To provide data durability when facing intermittent participation of the storage resources, we use an intelligent replication scheme that minimizes the volume of internal transfers that impact the low-bandwidth storage nodes.

The Problem. GridFTP [1] extends the FTP protocol with new features such as striping and partial file access. GridFTP has become the data access and management protocol of choice for Grid deployments in data-intensive scientific communities. As a result, significant efforts have been made to optimize the protocol itself and the software stack implementing it, and, more relevant to our work, often, GridFTP deployments are supported by expensive hardware resources that enable high storage I/O access rates (e.g., clusters and parallel file systems). The protocol has been itself modified in order to be able to take advantage of the parallel I/O paths offered by these expensive deployments.

The goal of this project is to lower the cost of GridFTP deployments while maintaining their performance.

The Solution. The key to reduce GridFTP deployment costs while maintaining the same performance is to observe that, the main bottleneck, and thus the cost driver for these deployments is the storage I/O bandwidth. Thus, to obtain high I/O bandwidth while keeping costs under control, we propose to integrate two types of storage nodes in a GridFTP deployment: First, stable, dedicated yet with low I/O bandwidth (thus cheap) nodes to provide persistent storage (that is, data durability). Second, a large number of LAN-connected, opportunistic, volatile nodes that, on aggregate, provide high access throughput to stored data. The nodes, for example, could be a subset of the desktops available in a company or research institution.

This proposed setup brings two main challenges which we address in turn. First, is finding a data placement and replication solution that is able to guarantee data durability and is optimized for the particular conditions of this deployment: the external system load and the internal data replication load need generate only a low I/O demand one the low-bandwidth, stable nodes while exploiting at maximum the volatile, high-bandwidth storage nodes.

The second challenge, relates to the transparent integration with a GridFTP implementation. For our prototype, we have chosen Globus Project's GridFTP implementation as it is the most popular. We are currently testing multiple approaches. First, we have completely isolated the GridFTP server and the data management layers as follows. We have deployed the GridFTP on top of a user-level file system [2] that harnesses FreeLoader [3] storage scavenging system to integrate all participating storage resources. This solution has the advantage that it requires limited development effort and changes to the GridFTP code but its performance also impacted by limited parallelism: all data transfers need to go through the FTP server node. We are currently exploring an alternative solution that directly integrates Freeloader

components (the data storage nodes) as GridFTP data sources (DTPs – Data Transfer Processes in Globus's GridFTP architecture). This solution offers full I/O parallelism as, once the data transfer is negotiated between the GridFTP server and the client, and the data placement is located with the aid of the FreeLoader manager component, data transfer channels are setup directly between the client(s) and each storage node using the striped transfer version of the protocol.

References

[1] *The Globus Striped GridFTP Framework and Server*. W. Allcock, J. Bresnahan, R. Kettimuthu, M. Link, C. Dumitrescu, I. Raicu, I. Foster. *Proceedings of Super Computing 2005 (SC05)*, November 2005.

[2] *A Checkpoint Storage System for Desktop Grid Computing*, Samer Al Kiswany, Matei Ripeanu, Sudharshan S. Vazhkudai, NetSysLab-TR-2007-04.

[3] *Constructing collaborative desktop storage caches for large scientific datasets*, Vazhkudai, S.S., Ma, X., Freeh, V.W., Strickland, J.W., et al.,. ACM Transaction on Storage (TOS), 2006. 2(3): p. 221 - 254.