A GridFTP Interface for iRODS

Shunde Zhang (shunde.zhang@arcs.org.au)
The Australian Research Collaboration Service (ARCS)

Abstract

This paper describes the design and implementation of a GridFTP interface for iRODS, the Integrated Rule-Oriented Data System. Users of the previous Data Intensive Cyber Environments Center (DICE Center) Storage Resource Broker (SRB) may know that there is a GridFTP interface for SRB, written in C as a DSI plugin for the Globus GridFTP server. However, there is no such plugin on Globus GridFTP for iRODS yet. The implementation of this GridFTP interface for iRODS is to provide an OS-independent and standalone solution that doesn't rely on Globus but is compatible with existing Globus clients.

1. Introduction

GridFTP is the de facto standard protocol for data transfer in the grid world. It is based on the traditional FTP protocol with enhanced features for better performance and reliable transfer. Our implementation is a standalone Java-based GridFTP server with Jargon to connect to iRODS. It is self-contained and doesn't require the installation of Globus, which makes it easier to install and run.

2. Features

The current version of this interface mainly follows the specification of GridFTP version 1, and 51 GridFTP commands have been implemented. The main features are listed below.

2.1. Parallel Transfer

In GridFTP version 1, data always flows from the source to the destination. When uploading a file (STOR), it has to be in passive mode (PASV followed by STOR), where the server listens on a port waiting for the client to open multiple connections to it; then the client starts sending data. While doing a download (RETR), it has to be in active mode (PORT, OPTS parallelism, then RETR). In this mode, the client listens on a port and the server opens a number of connections to the client, where the number is specified in the OPTS parallelism command. Then data flows from the server to the client.

The scenario we face is data going from a GridFTP client to a GridFTP server, and then to an iRODS server, or vise versa. We assume that the GridFTP server and the iRODS server are on the same VM/machine, or in the same local network with a fast network connection, while the GridFTP client and the GridFTP server are in different cities with a WAN connection, which is much slower than the GridFTP server to iRODS server connection. Therefore, the bottleneck is the lag between the GridFTP client and the GridFTP server. In this implementation, we use one connection between the GridFTP server and the iRODS server, and allow multiple connections between the GridFTP client and the GridFTP server. The iRODS protocol does support parallel transfers, but Jargon currently lacks support for that. However, our experiments show that the single connection doesn't affect the whole transfer, based on our assumptions.

2.2. File Operations

Most basic file operations are implemented, including: creating a new folder (MKD), deleting a folder (RMD), file listing (LIST), listing only names (NLST), listing for machine process (MLST and MLSD), renaming a file (RNFR and RNTO), deleting a file (DELE), changing working directory (CWD), printing current directory (PWD), etc.

2.3. Other Transfer Features

Data channel authentication is supported to protect file contents during the transfer. In particular, DCAU A (self authentication) is used by default. It can be turned off by sending a "DCAU N" command. Disabling DCAU can produce better performance.

Performance markers are sent back from the server in extended block mode to monitor the performance. They are also useful for keeping the control channel socket alive as they are sent every 5 seconds, so that the control channel socket will never timeout, even during large file transfers.

2.4. Work in Progress

UDT support is currently under development. UDT has been added to the latest GridFTP 5.0, and has proved to be faster than TCP transfer.

3. Experiments

A simple test of transferring twenty-one 320 Mb files (6.7G in total) from Hobart to Melbourne (using a 310 Mbps connection) shows it performing well.

Test	Time
Globus GridFTP 5 on disk (UDT, 2 FTP connections, 2 threads each)	10.5 mins
Globus GridFTP 5 on disk (TCP, 2 FTP connections, 2 threads each)	15 mins
Griffin to iRODS (TCP, 2 FTP connections, 2 threads each)	14 mins
iput	13 mins