the globus alliance
www.globus.org

Globus Data Storage Interface (DSI) - Enabling Easy Access to Grid Datasets

Raj Kettimuthu, ANL and U. Chicago

DIALOGUE Workshop

August 2, 2005

What is GridFTP?

- In Grid environments, access to distributed data is very important
- Distributed scientific and engineering applications require:
  - Transfers of large amounts of data between storage systems, and
  - Access to large amounts of data by many geographically distributed applications and users for analysis, visualization etc
- GridFTP - a secure, robust, efficient, standards based data transfer protocol
- Features
  - Standard FTP get/put etc., Third-party control of data transfer

the globus alliance
www.globus.org

What is GridFTP?

- Grid Security Infrastructure and Kerberos support
- Parallel data transfer (multiple transport streams between 2 network endpoints)
- Striped data transfer (1 or more transport streams between m network endpoints on the sending side and n network endpoints on the receiving side)
- Partial file transfer
- Manual/Automatic control of TCP buffer sizes
- Support for reliable and restartable data transfer

New GT4 GridFTP Implementation

- NOT based on wuftpd
- Striping support has been added
- Has IPV6 support included (EPRT, EPSV)
- Extremely modular to allow integration with a variety of data sources (files, mass stores, etc.)
- Based on Globus eXtensible Input/Output System (XIO)
  - Simple OCRW API for byte-stream IO

New Server Architecture

- GridFTP (and normal FTP) use (at least) two separate socket connections:
  - A control channel for carrying the commands and responses
  - A Data Channel for actually moving the data
- GridFTP (and normal FTP) has 3 distinct components:
  - Client and server protocol interpreters which handle control channel protocol
  - Data Transfer Process which handles the accessing of actual data and its movement via the data channel

the globus alliance
www.globus.org

New Server Architecture

- Protocol Interpreter and Data Transfer Process can be (optionally) completely separate processes.
- A single protocol interpreter can have multiple data transfer processes behind it.
  - This is how a striped server works.
- Data Transfer Process is architecturally, 3 distinct pieces:
  - Protocol handler, Data Storage Interface and Data processing module

New Server Architecture

- The protocol handler - talks to the network and understands the data channel protocol
- The Data Storage Interface (DSI) - provides an interface to data sources and sinks.
- The data processing module - provides ability to manipulate the data prior to transmission.
  - currently handled via the DSI
  - In future we plan to make this a separate module

The Data Storage Interface (DSI)

- Number of storage systems in use by the scientific and engineering community
  - Distributed Parallel Storage System (DPSS)
  - High Performance Storage System (HPSS)
  - Distributed File System (DFS)
  - Storage Resource Broker (SRB)
  - HDF5
- Use incompatible protocols for accessing data and require the use of their own clients

the globus alliance
www.globus.org

The Data Storage Interface (DSI)

- It provides a modular pluggable interface to data storage systems.
- Conceptually, the DSI is very simple.
- DSI consist of several function signatures and a set of semantics.
- When a new DSI is created, programmer implements the functions to provide the semantics associated with them.

The Data Storage Interface (DSI)

- The DSI author is not expected to know the intimate details involved in a GridFTP transfer.
- There are a set of API functions provided that allow the DSI to interact with the server itself.
- This API provides functions for reading and writing data to and from the network.

The Data Storage Interface (DSI)

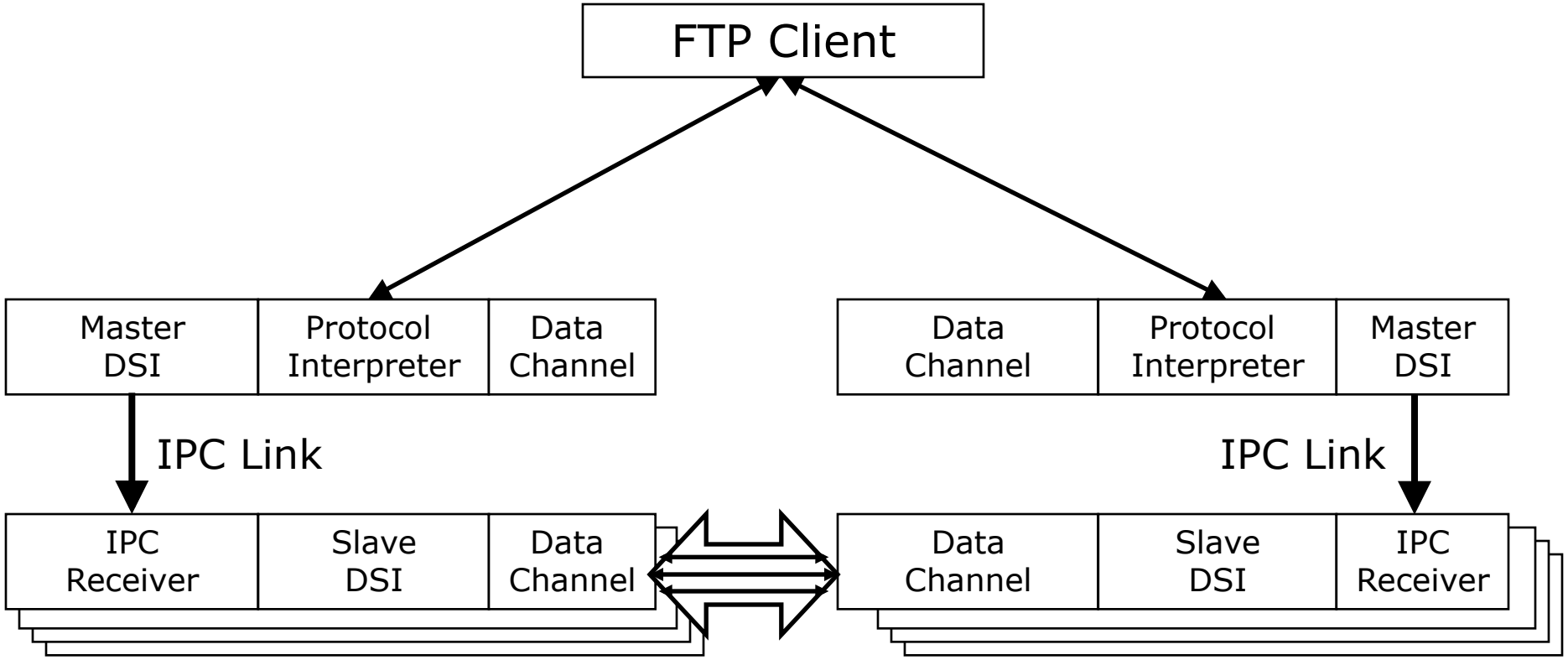
- DSI could be given significant functionality, such as caching, proxy, backend allocation, etc..
- DSIs can be loaded and switched at runtime.
- When the GridFTP server requires action from the storage system (be it data, meta-data, directory creation, etc) it passes a request to the loaded DSI module.
- The DSI then services that request and notifies the server when it is finished.

Developer Implemented Functions

```
typedef struct globus_gfs_storage_iface_s
{
    int                                 descriptor;

    /* session initiating functions */
    globus_gfs_storage_init_t           init_func;
    globus_gfs_storage_destroy_t        destroy_func;

    /* transfer functions */
    globus_gfs_storage_transfer_t       list_func;
    globus_gfs_storage_transfer_t       send_func;
    globus_gfs_storage_transfer_t       recv_func;
    globus_gfs_storage_trev_t           trev_func;
    /* data conn funcs */
    globus_gfs_storage_data_t           active_func;
    globus_gfs_storage_data_t           passive_func;
    globus_gfs_storage_data_destroy_t   data_destroy_func;

    globus_gfs_storage_command_t        command_func;
    globus_gfs_storage_stat_t           stat_func;

    globus_gfs_storage_set_cred_t       set_cred_func;
    globus_gfs_storage_buffer_send_t    buffer_send_func;
} globus_gfs_storage_iface_t;
```

the globus alliance
www.globus.org

Striped Data Transfer



Master and Slave DSI

- If you wish to support striping, you will need two DSIs
- The Master DSI will be in the control process or front end.
  - Usually, this is relatively trivial and involves minor processing and then "passing" the command over the IPC channel to the slave DSI
- The slave DSI does the real work. It typically implements the following functions:
  - send_func: This function is used to send data from the DSI to the server (get or RETR)

Master and Slave DSI

- recv_func: This function is used to receive data from the server (put or STOR)
- stat_func: This function performs a unix stat, i.e. it returns file info. Used by the list function
- command_func: This function handles simple (succeed/fail or single line response) file system operations such as mkdir, site chmod, etc.

- The master should implement all functions. Besides the above functions, it implements:
  - active_func: This is for when the DSI will be doing a TCP connect.
    - The master figures out who gets what IP/port info and then passes it through.

Master and Slave DSI

- passive_func: The counter-part to the active_func when the DSI will be the listener
- list_func: This should be passed through and will handle LIST, NLST, MLST, etc..

- There are also some utility functions the master should implement:
  - trev_func: This handles the restart and performance markers, but should be a simple pass through

the globus alliance
www.globus.org

IPC Calls

- These calls are how the master DSI “passes” the call to the slave DSI
- These calls implement an internal protocol to transfer the necessary structures between the front end and the back end.
- The IPC receiver receives the message and then invokes the appropriate DSI call.

Helper Functions that should be used

- When implementing the DSI functions, the following helper functions should be called:
  - <function>_finished: This tells the server that a specific function (such as recv) has completed
  - register[read|write]: This is how file data is transferred between the DSI and the server.
  - bytes_written: This should be called anytime the DSI successfully completes a write to its own storage system. This allows performance and restart markers to be generated.

Helper Functions that should be used

- get_blocksize: This indicates the buffer size that you should exchange with the server via the register_[read|write].
- get_[read|write]_range: This tells the DSI which data it should be sending.
  - This handles striping (this DSI only needs to send a portion of the file), and partial files.

Existing DSIs

- DSIs do exist for:
  - File systems accessible via standard POSIX API
  - Storage Resource Broker (SRB)
  - High Performance Storage System (HPSS) and
  - NeST from the Condor team

Summary

- DSIs confer benefits to both the keepers of large datasets and the users of these datasets.
- Dataset providers would gain a broader user base, because their data would be available to any client.
- Dataset users would gain access to a broader range of storage systems and data.