



the globus alliance
www.globus.org

Optimizing Data Transport

A Tutorial on Deploying GridFTP

Michael Link
John Bresnahan

Argonne National Laboratory
University of Chicago



Obtain Installer Now

GridFTP Tutorial

- Installing to a remote machine
 - <http://www.gridftp.org/tutorials>
- Installing to laptop (linux and mac users)
 - 1 of 3 ways
 - CD
 - USB Drive
 - <http://www.gridftp.org/tutorials>



Outline

- Introduction
- Security Options
- Optimizations
- GSI Configuration
- Advanced Configurations



What is GridFTP

- Fast, secure, protocol for data transfer
- Extensions to RFC959 (standard FTP)
 - Backward compatible
 - AUTH GSSAPI
 - GSI implementation
 - Mode E
 - Data channel extension
 - Parallel TCP streams
 - Multi-host 'striped' transfers



Two Channel Protocol

- **Control Channel**
 - Command/Response
 - Used to establish data channels
 - Basic file system commands
- **Data channel**
 - Pathway over which *file* is transferred
 - Many different underlying protocols can be used
 - MODE command determines the protocol



Architecture Components

- **Control Channel (CC)**

- Path between client and server used to exchange all information needed to coordinate transfers



- **Data Channel (DC)**

- The network pathway over which the 'files' flow



- **Server Protocol Interpreter (SPI)**

- AKA: Frontend
- Server side implementation of the control channel functionality



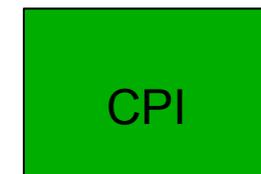
- **Data Protocol Interpreter (DPI)**

- AKA: Backend
- Handles the actual transferring of files



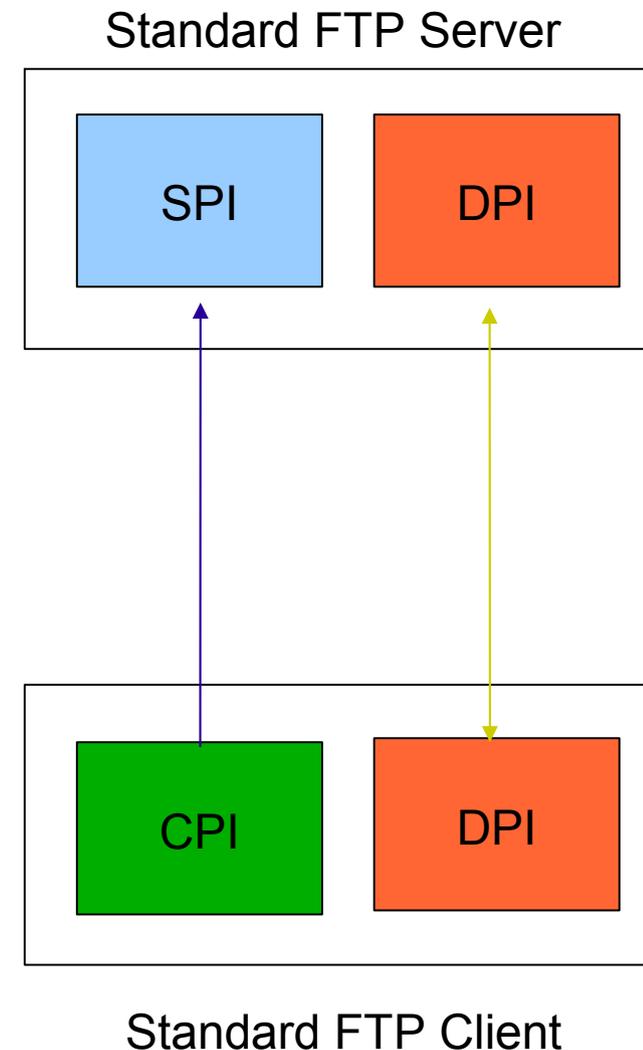
- **Client Protocol Interpreter (CPI)**

- Client side implementation of the control channel functionality



Simple Two Party Transfer

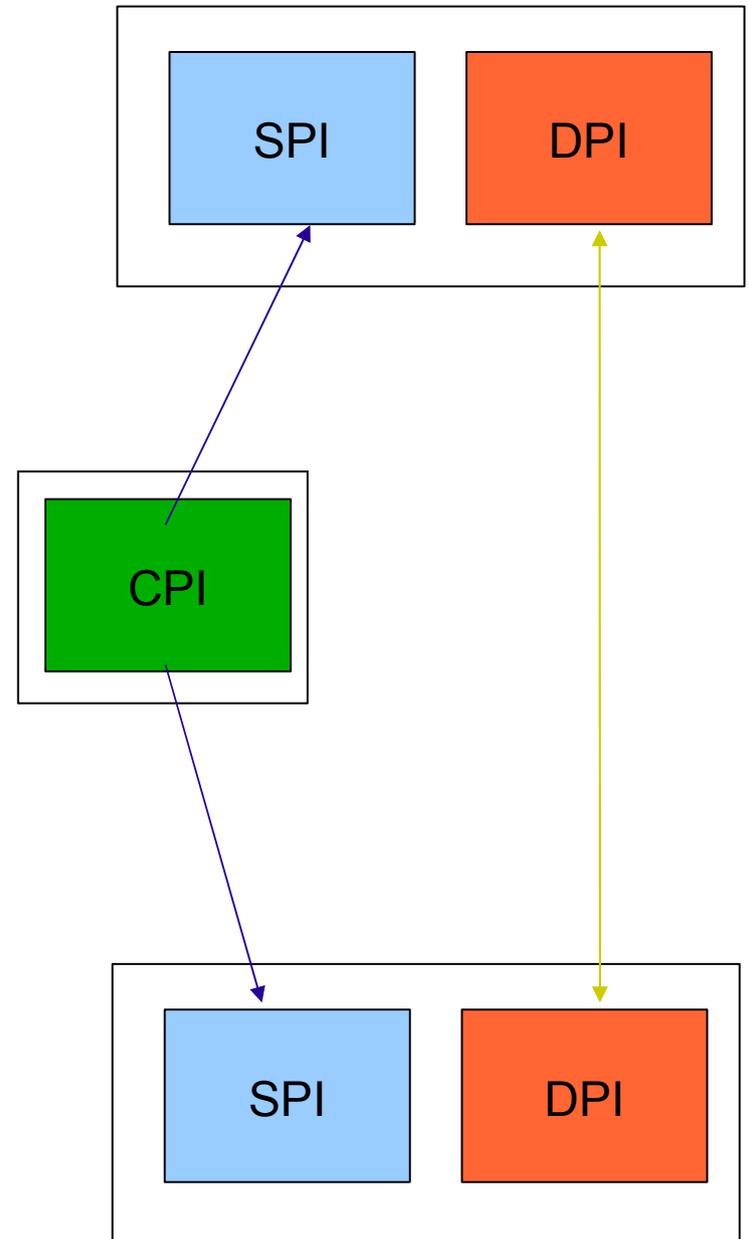
- Clear boxes represent process spaces
- The Server Side
 - SPI and DPI are co-located in the same process space
- The Client Side
 - CPI and DPI are co-located in the same process
- Interaction
 - The client connects forms a CC with the server
 - Information is exchanged to establish the DC
 - A file is transferred over the DC





Simple Third Party Transfer

- Client forms CC with 2 servers.
- Servers are co-located SPI and DPI
- Information is routed through the client to establish DC between the two servers.
 - including delegated security credentials
- Data flows directly between servers
 - client is notified by each server SPI when the transfer is complete





Control Channel Establishment

- TCP Connection to known port
- 220 banner message
- Authentication
 - Anonymous
 - Clear text USER <username>/PASS <pw>
 - Base 64 encoded GSI handshake
- 230 Accepted/530 Rejected



Data Channel Establishment

- **PASV command**
 - Sent to the *passive* side of the transfer
 - Listen for a connection and reply with the contact information (IP address and port)
- **PORT IP,PORT**
 - Actively establish a connection to a given passive listener
- **Firewall**
 - Port side must be able to connect to PASV side
 - Can configure to bypass

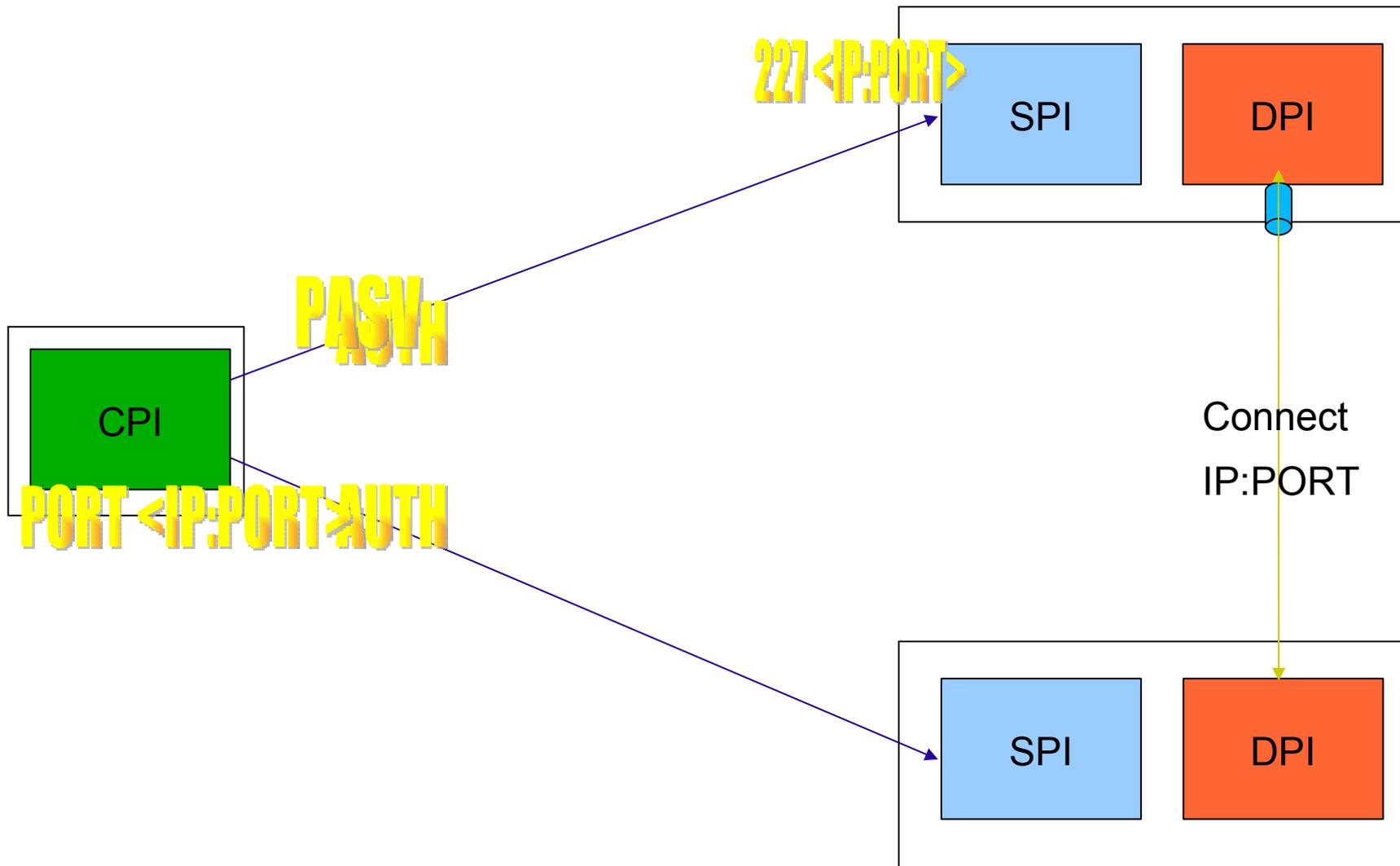


Data Channel Protocols

- **MODE Command**
 - Allows the client to select the data channel protocol
- **MODE S**
 - Stream mode, no framing
 - Legacy RFC959
- **MODE E**
 - GridFTP extension
 - Parallel TCP streams
 - Data channel caching



Data Channel Establishment





Exercise 1

Anonymous Transfer

- **Install the GridFTP Server**
 - `http://www.gridftp.org/tutorials/`
 - `tar xvfz gt-gridftp*.tar.gz`
 - `cd gt-gridftp-binary-installer`
 - `./configure -prefix /path/to/install`
 - *ignore any java/ant warnings*
 - `make gridftp install`
- **Setup the environment (repeat for all globus sessions)**
 - `export GLOBUS_LOCATION=/path/to/install`
 - `source $GLOBUS_LOCATION/etc/globus-user-env.sh`



Exercise 1

- **globus-gridftp-server options**
 - `globus-gridftp-server --help`
- **Start the server in anonymous mode**
 - `globus-gridftp-server --control-interface 127.0.0.1 -aa -p 5000`
- **Run a two party transfer**
 - `globus-url-copy -v file:///etc/group ftp://localhost:5000/tmp/group`
- **Run 3rd party transfer**
 - `globus-url-copy -v ftp://localhost:<port>/etc/group ftp://localhost:<port>/tmp/group2`
- **Experiment with -dbg, -vb -fast options**
 - `globus-url-copy -dbg file:///etc/group ftp://localhost:5000/tmp/group`
 - `globus-url-copy -vb file:///dev/zero ftp://localhost:5000/dev/null`
- **Kill the server**



Exercise 1

Examine debug output

- TCP connection formed from client to server
- Control connection authenticated
- Several session establishment options sent
- Data channel established
 - MODE E command
 - PASV sent to server
 - Server begins listening and replies to client with contact info
 - Client connected to the listener
 - File is sent across data connection



Security Options

- **Clear text (RFC 959)**
 - Username/password
 - Anonymous mode (anonymous/<email addr>)
 - Password file
- **SSHFTP**
 - Use ssh/sshd to form the control connection
- **GSIFTP**
 - Authenticate control and data channels with GSI

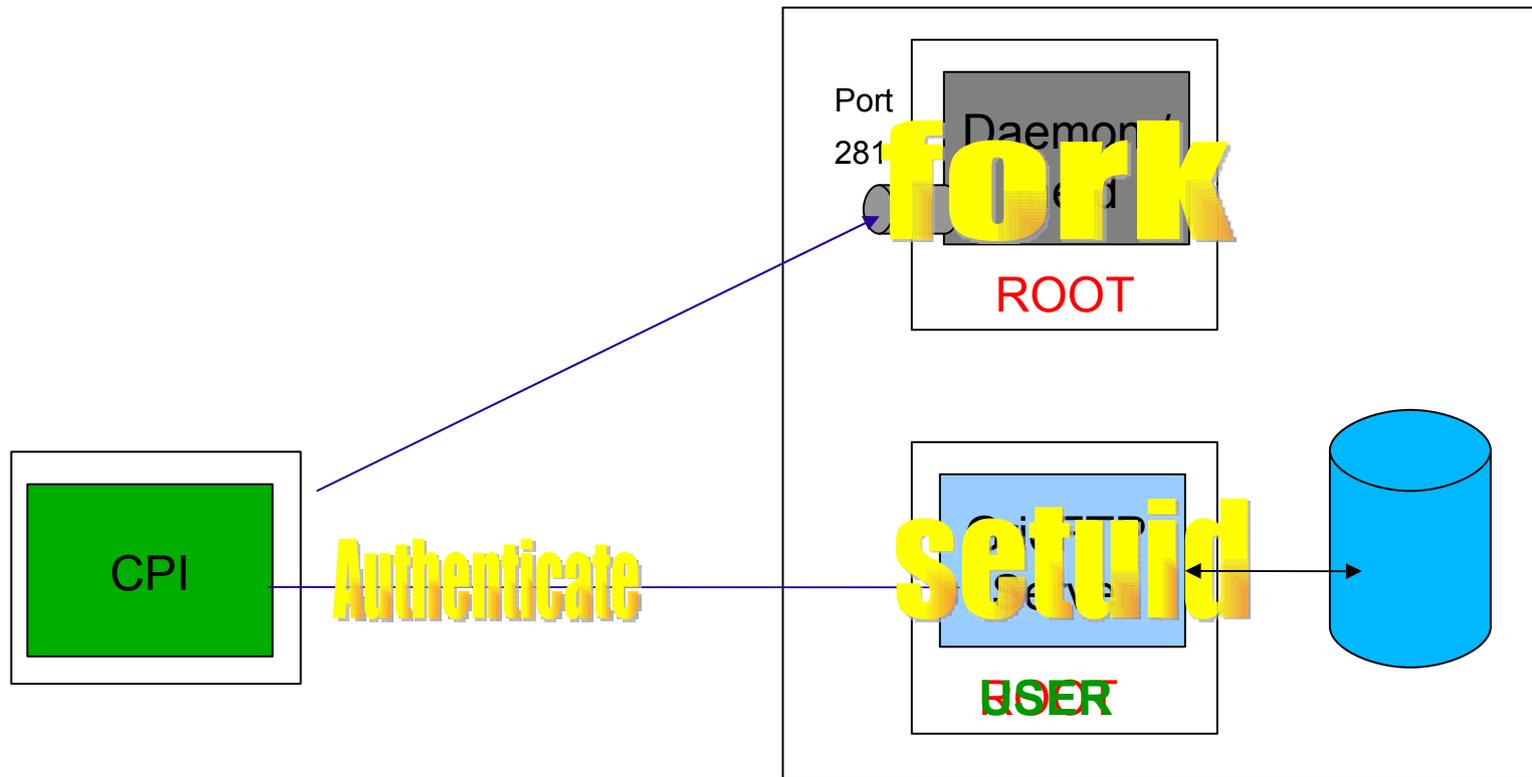


User Permissions

- File permissions are handled by the OS
- inetd or daemon mode
 - Fork on TCP accept
- User is mapped to a local account
 - Client connects and the server forks
 - The child process authenticates the connection
 - The child process `setuid()` to a local user



inetd/daemon Interactions





(x)inetd Entry Examples

- xinetd

```
service gsiftp
{
  socket_type = stream
  protocol = tcp
  wait = no
  user = root
  env += GLOBUS_LOCATION=<GLOBUS_LOCATION>
  env += LD_LIBRARY_PATH=<GLOBUS_LOCATION>/lib
  server = <GLOBUS_LOCATION>/sbin/globus-gridftp-server
  server_args = -i
  disable = no
}
```

- inetd

```
gsiftp stream tcp nowait root /usr/bin/env env \
  GLOBUS_LOCATION=<GLOBUS_LOCATION> \
  LD_LIBRARY_PATH=<GLOBUS_LOCATION>/lib \
  <GLOBUS_LOCATION>/sbin/globus-gridftp-server -i
```

- Remember to add 'gsiftp' to /etc/services with port 2811.



Exercise 2

Password file

- Create a password file
 - `gridftp-password.pl > pwfile`
- Run the server in password mode
 - `globus-gridftp-server -p 5000 -password-file /full/path/of/pwfile`
- Connect with standard ftp program
 - `ftp localhost 5000`
 - `ls, pwd, cd, etc...`
- Transfer with `globus-url-copy`
 - `globus-url-copy file:///etc/group ftp://username:pw@localhost:5000/tmp/group`
 - `globus-url-copy -list ftp://username:pw@localhost:5000/`

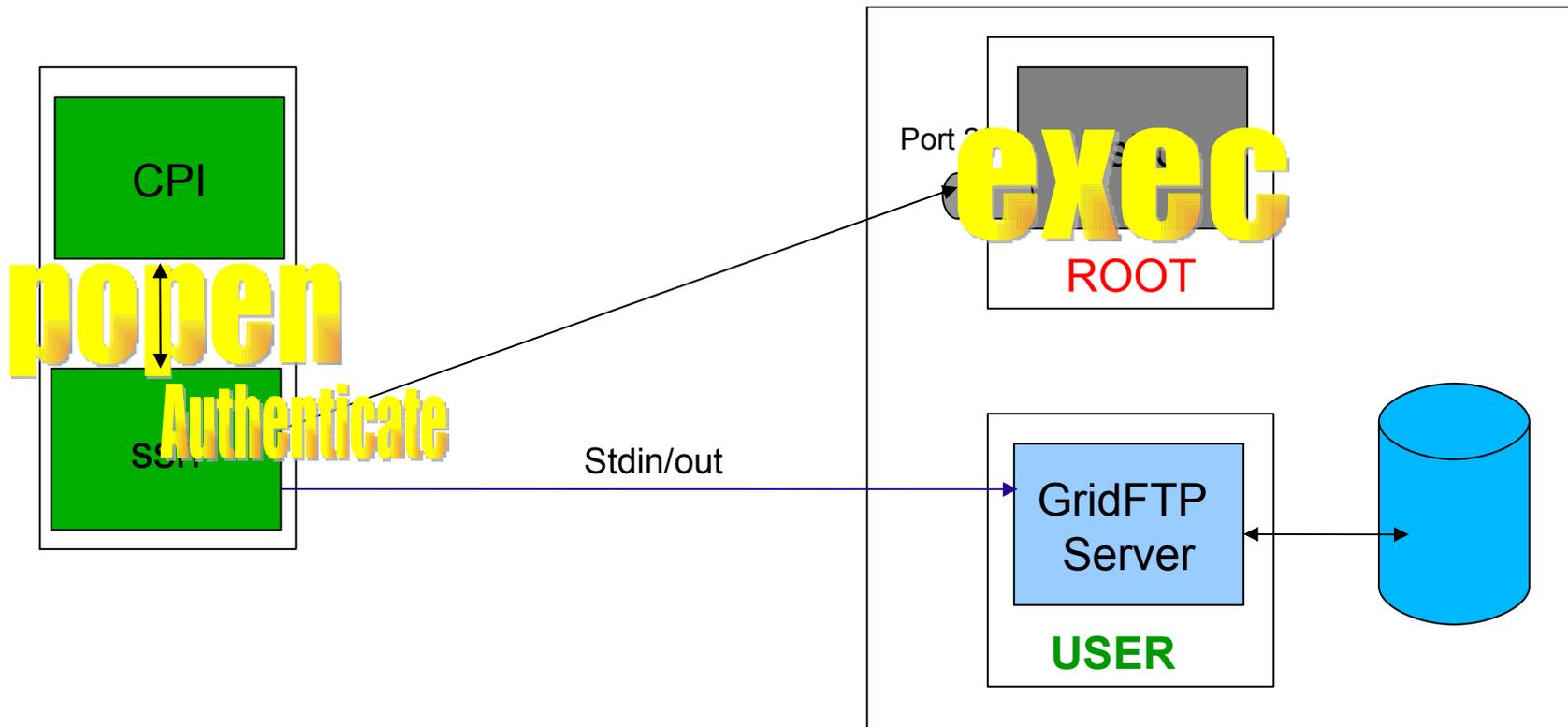


GridFTP Over SSH

- sshd acts similar to inetd
- control channel is routed over ssh
 - globus-url-copy *popens* ssh
 - ssh authenticates with sshd
 - ssh/sshd remotely starts the GridFTP server as user
 - stdin/out becomes the control channel



sshftp:// Interactions





Exercise 3

sshftp

- Configure SSHFTP
 - `$GLOBUS_LOCATION/setup/globus/setup-globus-gridftp-sshftp`
 - Enables **client** support for `sshftp://` urls for this `$GLOBUS_LOCATION`
 - `$GLOBUS_LOCATION/setup/globus/setup-globus-gridftp-sshftp -server -nonroot`
 - Enables **server** support for `sshftp://` connections **for this user only**.
 - To enable for all users run as root and remove `-nonroot`.
- `globus-url-copy` transfers
 - `globus-url-copy -v file:///etc/group sshftp://localhost/tmp/group`
 - `globus-url-copy -list sshftp://localhost/tmp/`



Exercise 3

What happened?

- globus-url-copy popen'ed ssh
 - guc read/writes control channel messages to ssh
 - ssh writes CC messages to stdin/out
- ssh authenticates with sshd
- ssh remotely starts globus-gridftp-server
 - server is expecting control channel messages on stdin/out
- control channel messaging is routed through ssh via stdin/stdout

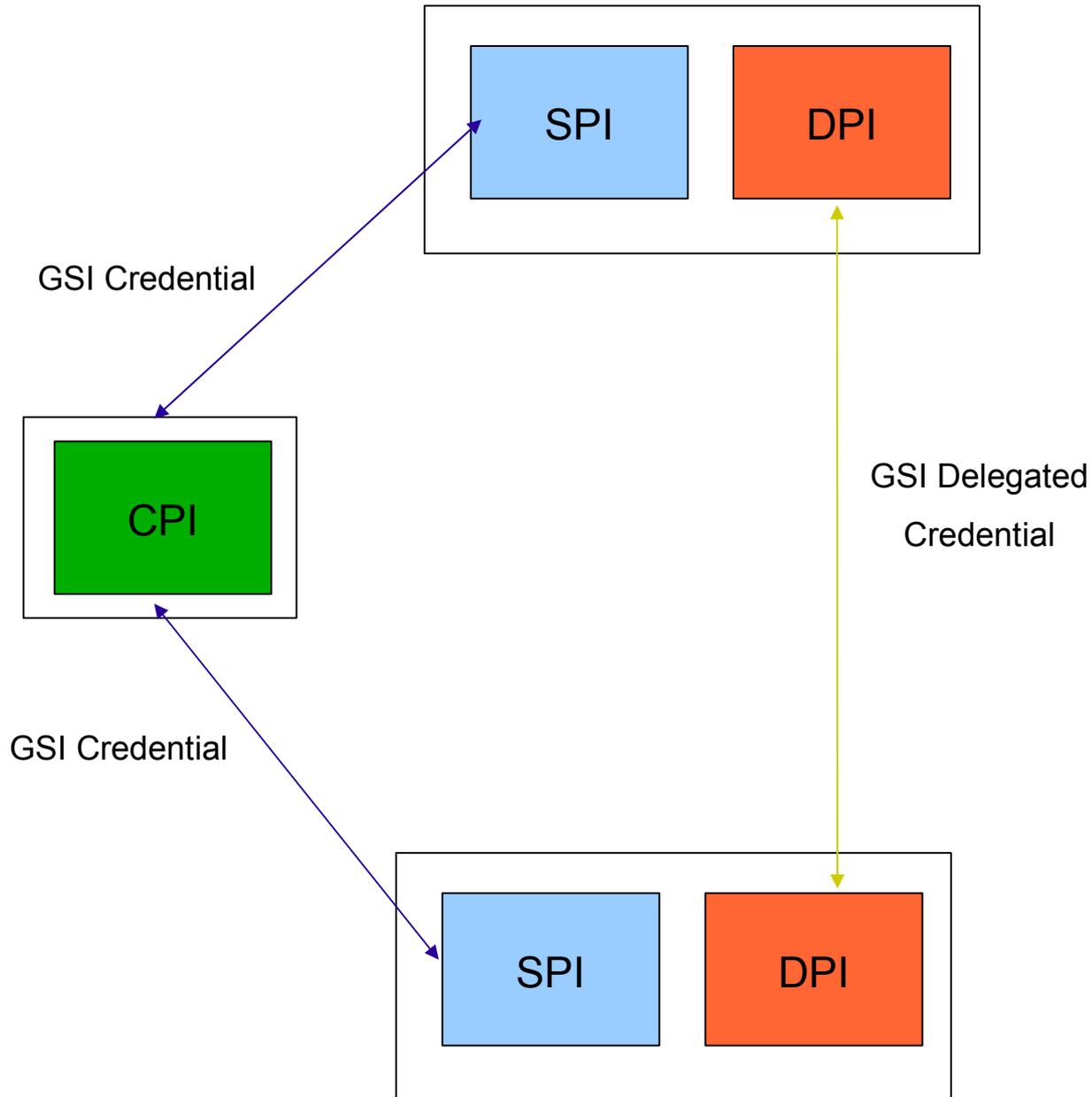


GSI Authentication

- **Strong security on both channels**
 - SSH does not give us data channel security
- **Delegation**
 - Authenticates DC on clients behalf
 - Flexibility for grid services such as RFT
 - Agents can authenticate to GridFTP servers on users behalf



GSI Authentication





Certificates

- **Certificate Authority (CA)**
 - Trusted 3rd party that confirms identity
- **Host credential**
 - Long term credential
 - Distinguished name is based on hostname
 - predictable
- **User credential**
 - Passphrase protected
 - Used to activate a short term proxy



Exercise 4

GSI Security

- Setup simpleCA
- Create a user credential
- Create proxy
 - grid-proxy-init
- Create gridmap file
- Run GridFTP server
- Perform a GSI authenticated transfer
- Evaluate results



Optimizations

- TCP buffer size
- Disk block size
- Parallel streams
- Cached connections
- Partial file transfers



TCP Buffer Size

- Most important tuning parameter for TCP
 - Memory the kernel *allocates* for retransmits/reordering
 - Effects the maximum window size
- Bandwidth Delay Product (BWDP)
 - $BWDP = \text{latency} * \text{bandwidth}$
 - The optimal number of bytes that can be sent before waiting on an acknowledgement (ACK)
 - The optimal number of *in-flight* bytes



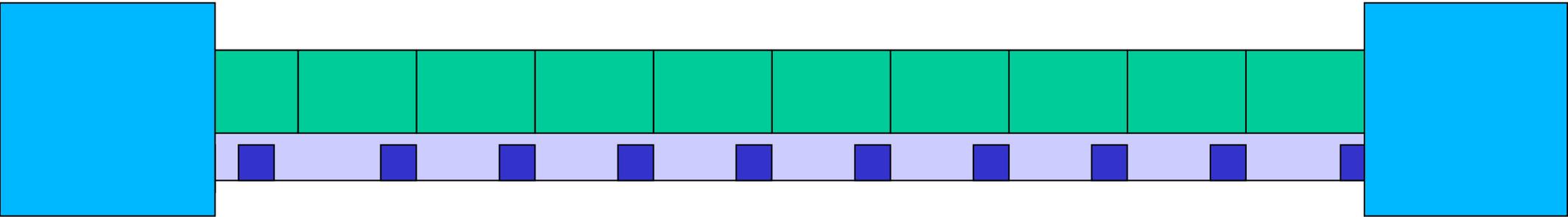
BWDP

- Send |window| of bytes
- When receiver gets first byte it sends ACK
- ACK takes 1 trip to get back to sender
- Ideally sender is never stalled out waiting for an ACK
 - thus the window size must be large enough so that we are sending for one entire RTT



Window and ACKs

Small TCP Buffer Size



Data Packet

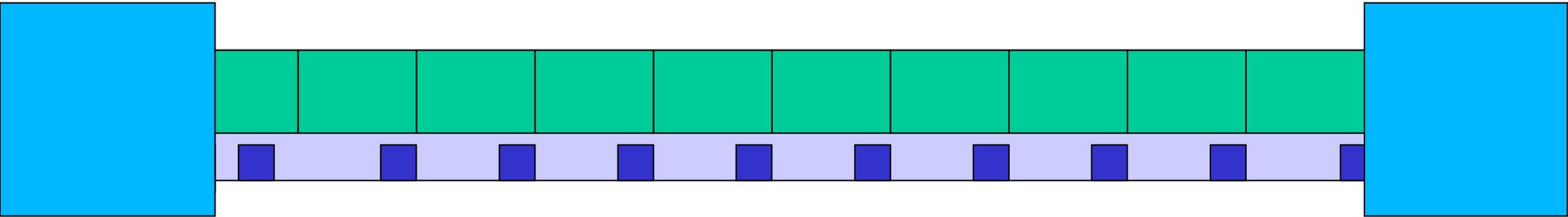


Acknowledgement



Window and ACKs

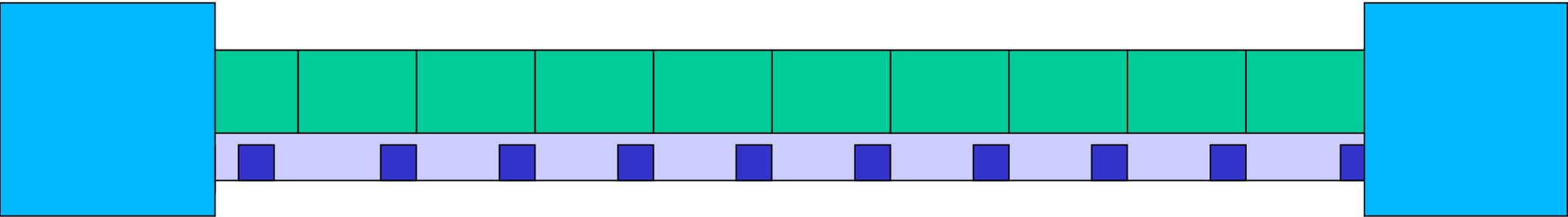
Half Full (1 trip)





Window and ACKs

Optimized TCP Buffer Size





AIMD

- **Additive Increase Multiplicative Decrease**
 1. Window size increases exponentially
 2. A congestion event occurs
 3. Window size is cut in half
 4. Window linearly increases
- **Conclusion**
 - Dropped packets are costly



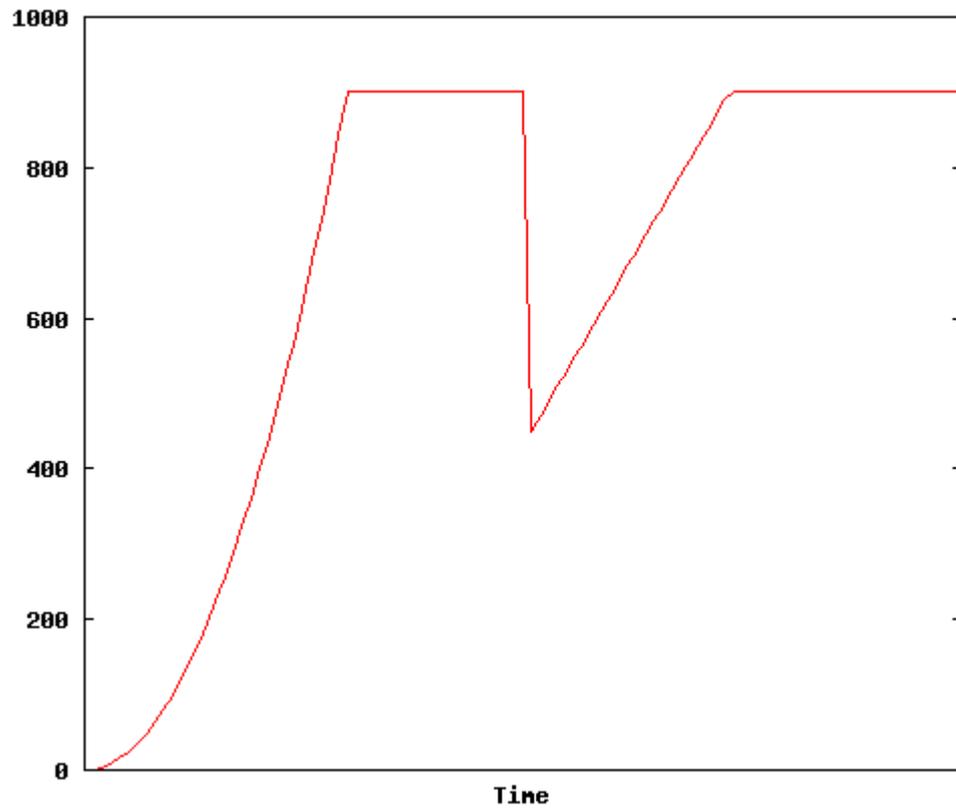
Why Parallel TCP?

- Taking advantage of loopholes in the system
 - *Cheat* TCP out of intended fair backoff
- Reduces the severity of a congestion event
 - Only effects $1/p$ of the overall transfer
- Faster recovery
 - Smaller size to recover
- Work around for low TCP buffer limits
- *Almost a side effect of striping*

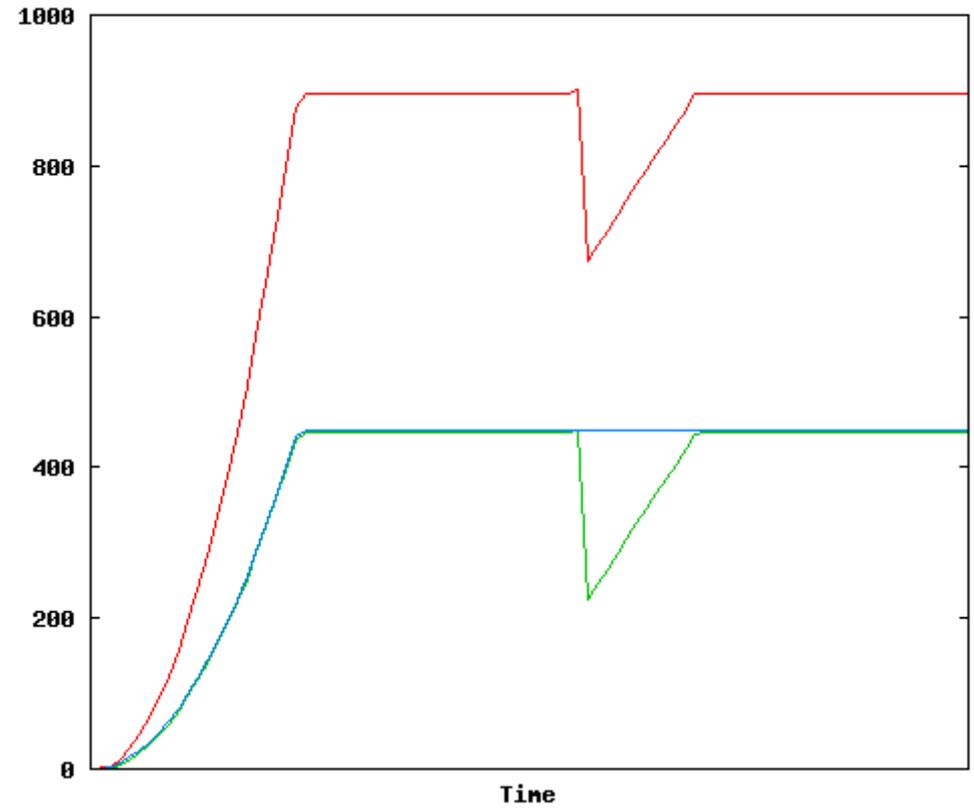


Lost Packets

One Stream



Two Streams





Data channel caching

- Establishing a data channel can be expensive
 - Round trips over high latency links
 - Security handshake can be expensive
- Mode E introduces data channel caching
 - Mode S closes the connection to indicate end of data
 - Mode E uses meta data to indicate file barriers
 - Doesn't need to close



Demonstration 1

Performance

- Transfer on a real network
 - Show performance markers
 - Show transfer rate
- Calculate the BWDP
- Vary -tcp-bs
- Vary -p



Partial File Transfer

- **Large file transfer fails**
 - We don't want to start completely over
 - Ideally we start where we left off
- **Restart markers sent periodically**
 - Contain blocks written to disk
 - Sent every 5s by default
 - In worst case recovery sends 5s of redundant data
- **Reliable File Transfer Service**

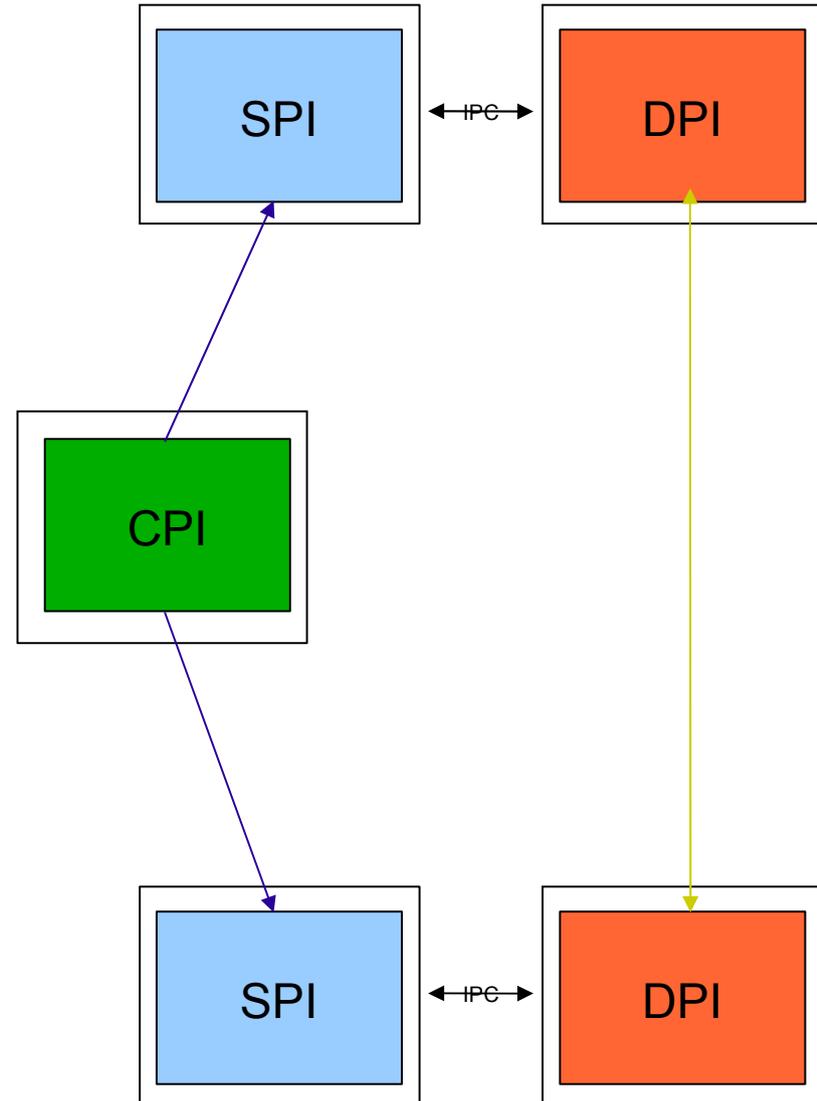


Advanced Configurations

- Separation of processes
- Proxy server
- Striping
- Data Storage Interface (DSI)
- Alternative data channel stack

Separated Third Party Transfer

- DPI and SPI do not need to be in the same process.
 - nor on the same host
- Separation is transparent to client
- Opaque communication mechanisms between SPI and DPI





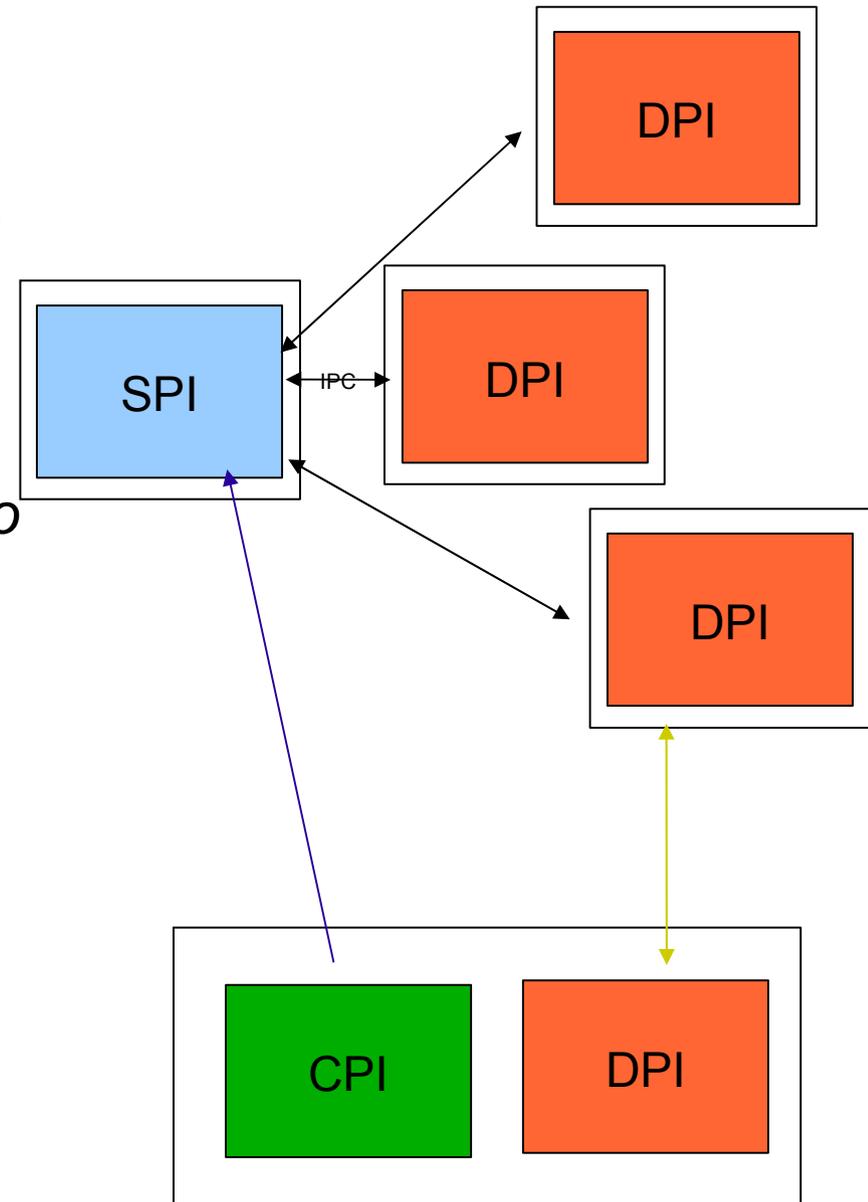
Separation of Process Advantage

- **More secure**
 - *Frontend* (SPI) no longer must be run as root
 - Client never communicates with a root process
- **Load balancing proxy server**
 - Select the least busy backend
 - Backends can come and go dynamically
- **Striping**
 - Many backends can be used for a single transfer



Proxy Server

- The separation of processes buys the ability to proxy
 - *Allows for load balancing*
 - *SPI can choose from a pool of DPIs to service a client request*



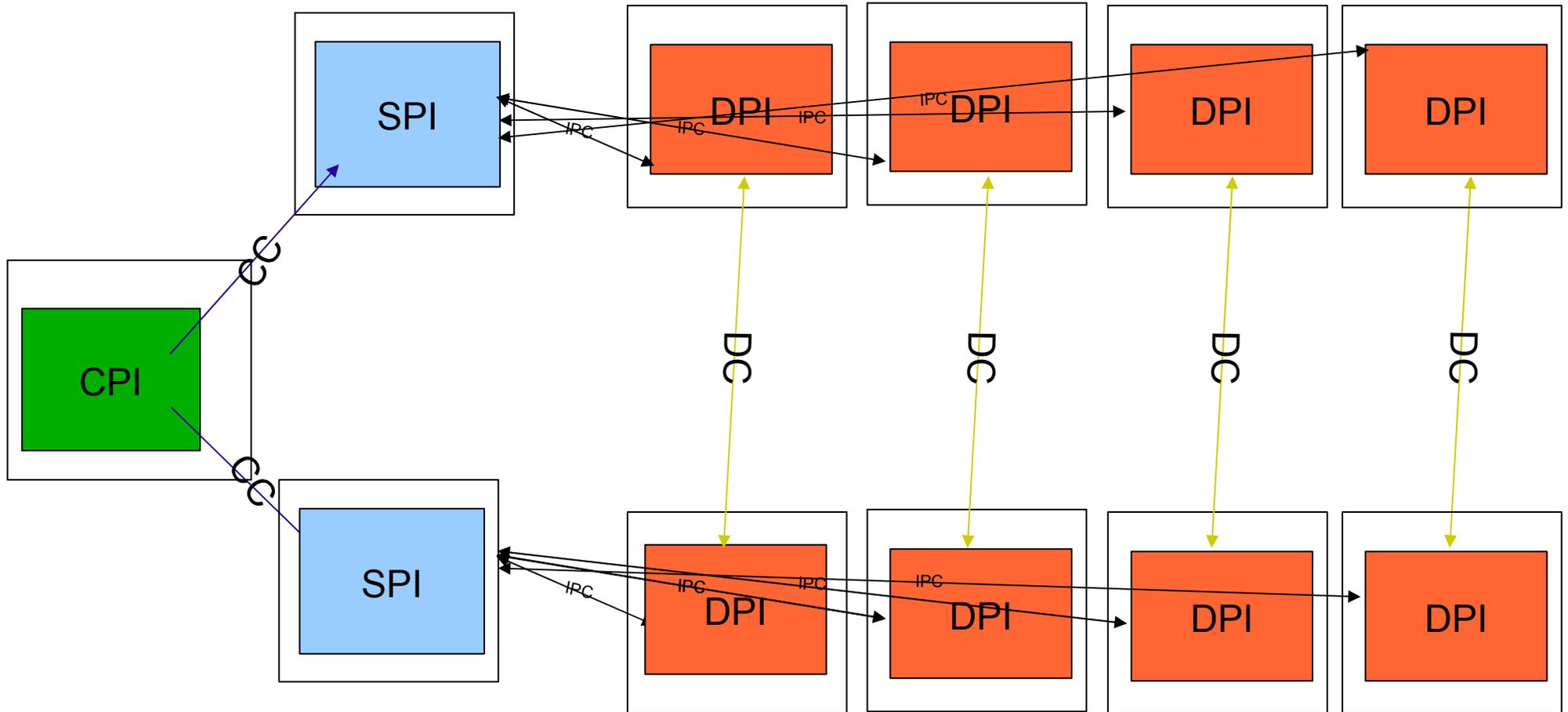


Striping

- A coordinated multi-host transfer
 - 1 SPI per server
 - Many DPIs per SPI
 - Each DPI transfers a portion of the file
 - Allows for fast transfers
 - Many NICs per transfer



Striping





Demonstration 3

Striping

- Show a striped transfer
 - 3rd party transfers
 - show performance increases
 - `globus-url-copy -vb`



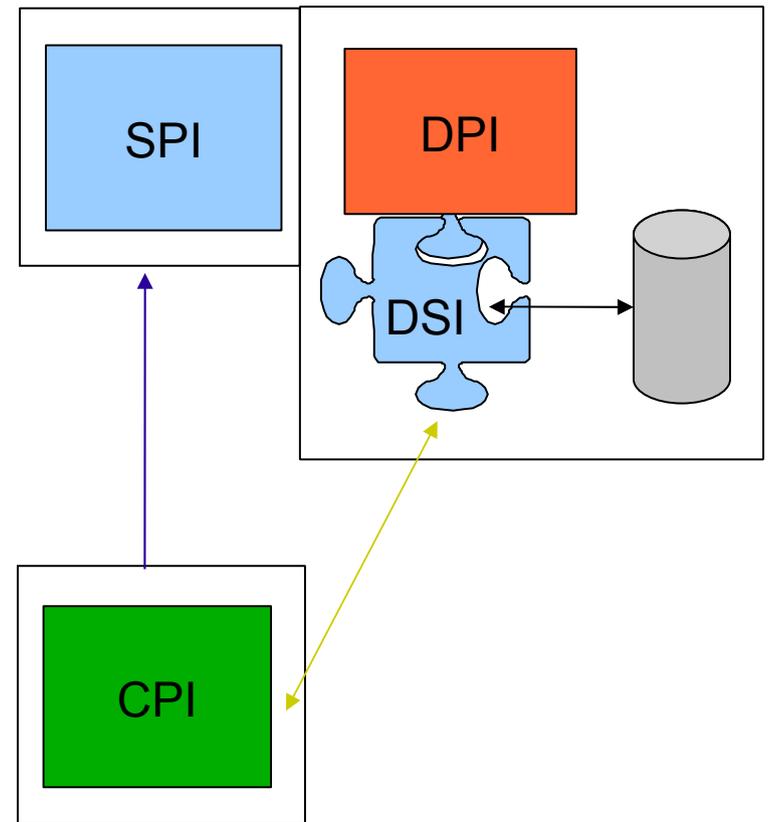
Data Storage Interface (DSI)

- Modular abstraction to storage systems
 - GridFTP is a network interface to storage
 - customize to specific storage systems
 - software, optimization, etc
- Existing DSIs
 - SRB, HPSS, posix FS, remote



DSI

- DSI plugs into DPI
 - works with stripes as well
- All interaction with storage goes through DSI
- DSI is transparent to client or remote party





Alternative stacks

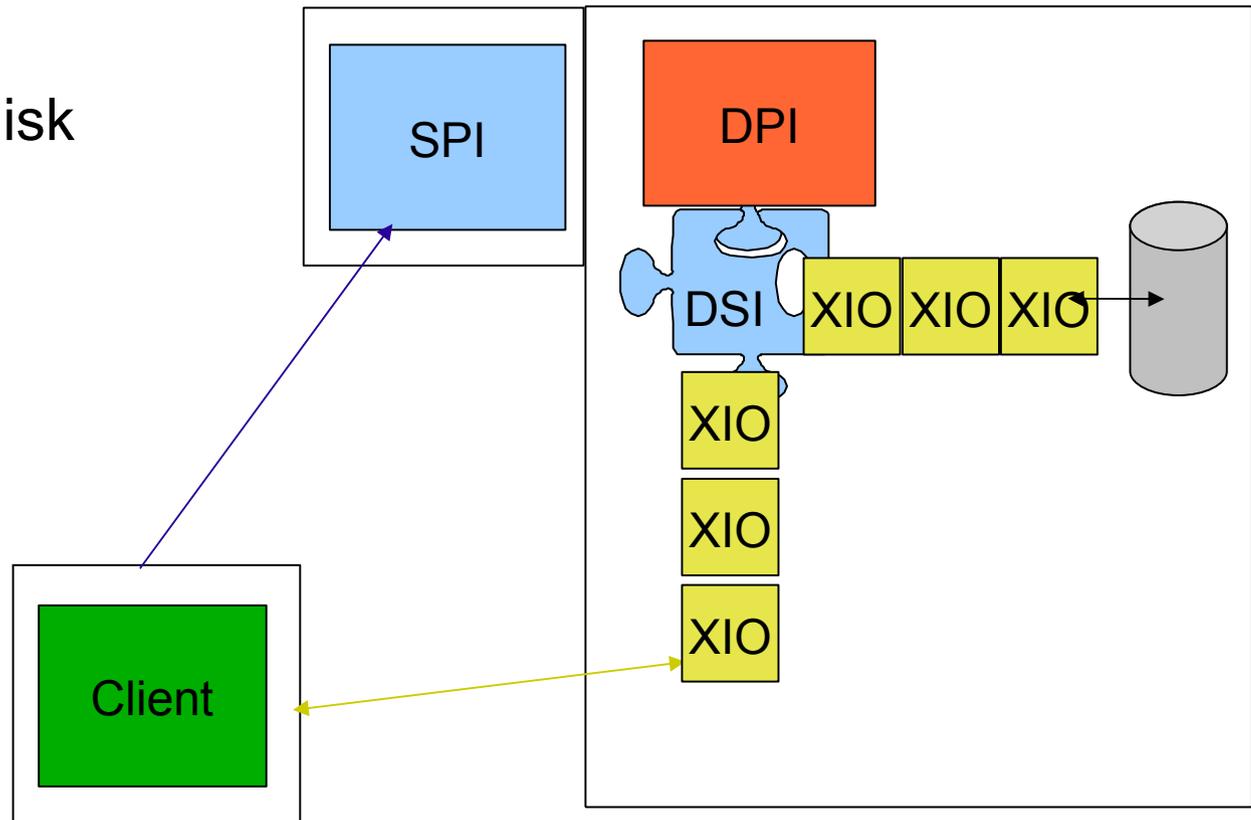
- All I/O in GridFTP is done with Globus XIO
 - data channel and disk
- XIO allows you to set an I/O software stack
 - transport and transform drivers
 - ex: mode_e,gsi,tcp
- Substitute UDT for TCP
- Add BW limiting, or netlogger



XIO Driver Stacks

- All data passes through XIO driver stacks

- to network and disk
- observe data
- change data
- change protocol





Thank you

- **More Information:**
 - <http://www.gridftp.org>
 - <http://www.globus.org/toolkit>
 - `gridftp-user@globus.org`