the globus alliance
www.globus.org

Configuring and Deploying GridFTP for Managing Data Movement in Grid/HPC Environments

John Bresnahan
Michael Link
Raj Kettimuthu

Argonne National Laboratory
University of Chicago

Obtain Installer Now

GridFTP Tutorial

- Installing to a remote machine
  - http://www.gridftp.org/tutorials
  - Handout on build instructions and exercises available here
- Installing to laptop (linux and mac users)
  - 1 of 2 ways
    - USB Drive
    - http://www.gridftp.org/tutorials

User accounts

- For those with no access to a unix machine, we have 2 VMs for the exercises
  - tp-x001.ci.uchicago.edu
  - tp-x002.ci.uchicago.edu

Outline

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

What is GridFTP?

- High-performance, reliable data transfer protocol optimized for high-bandwidth wide-area networks
- Based on FTP protocol - defines extensions for high-performance operation and security
- Standardized through Open Grid Forum (OGF)
- GridFTP is the OGF recommended data movement protocol

GridFTP

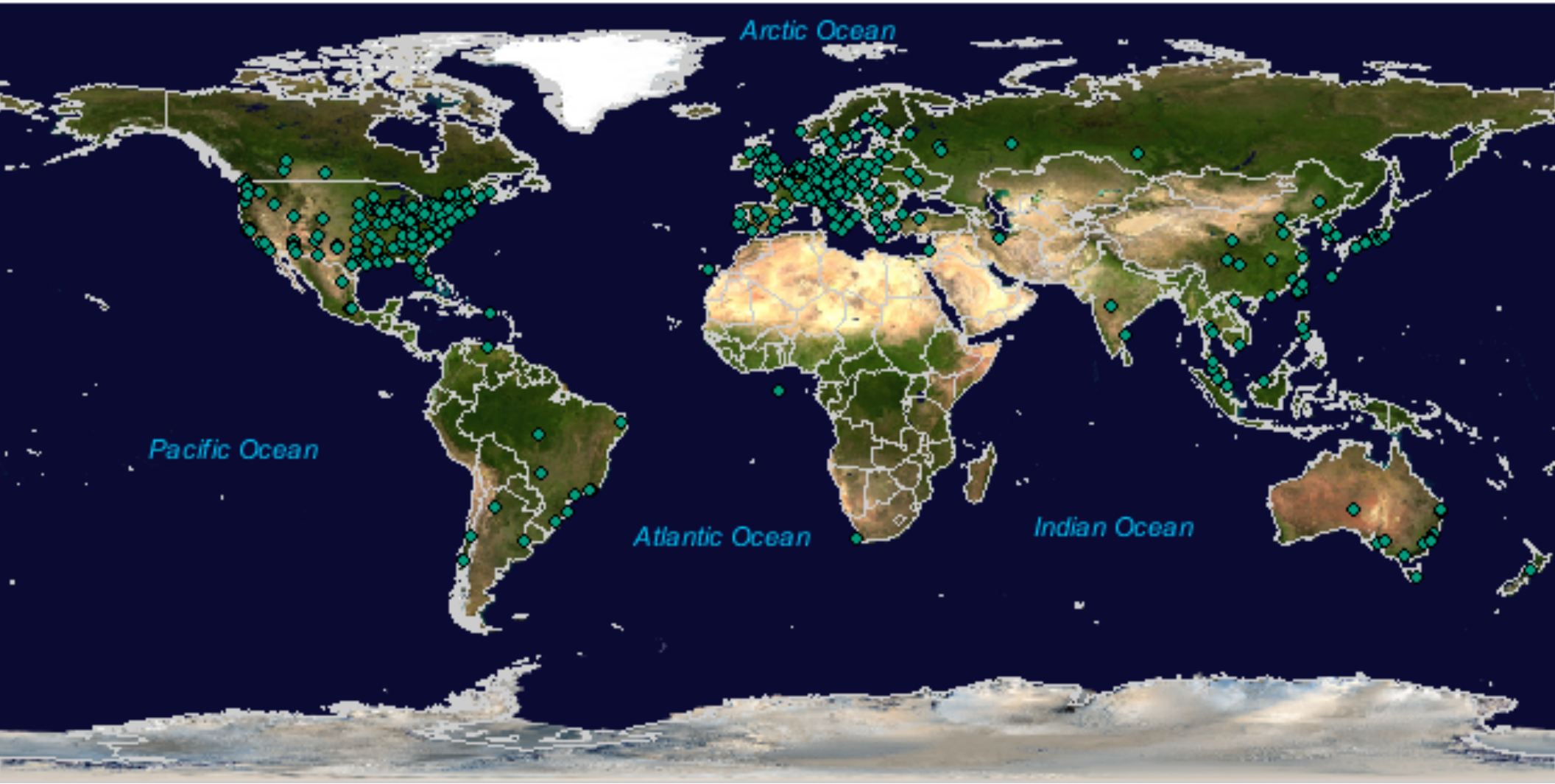
- We (Globus Alliance) provide a reference implementation:
  - Server
  - Client tools (globus-url-copy)
  - Development Libraries
- Multiple independent implementations can interoperate
  - Fermi Lab and U. Virginia have home grown servers that work with ours

Globus GridFTP

- Performance
  - Parallel TCP streams, optimal TCP buffer
  - Non TCP protocol such as UDT
- Cluster-to-cluster data movement
- Multicasting, Overlay routing
- Multiple security options
  - Anonymous, password, SSH, GSI
- Support for reliable and restartable transfers

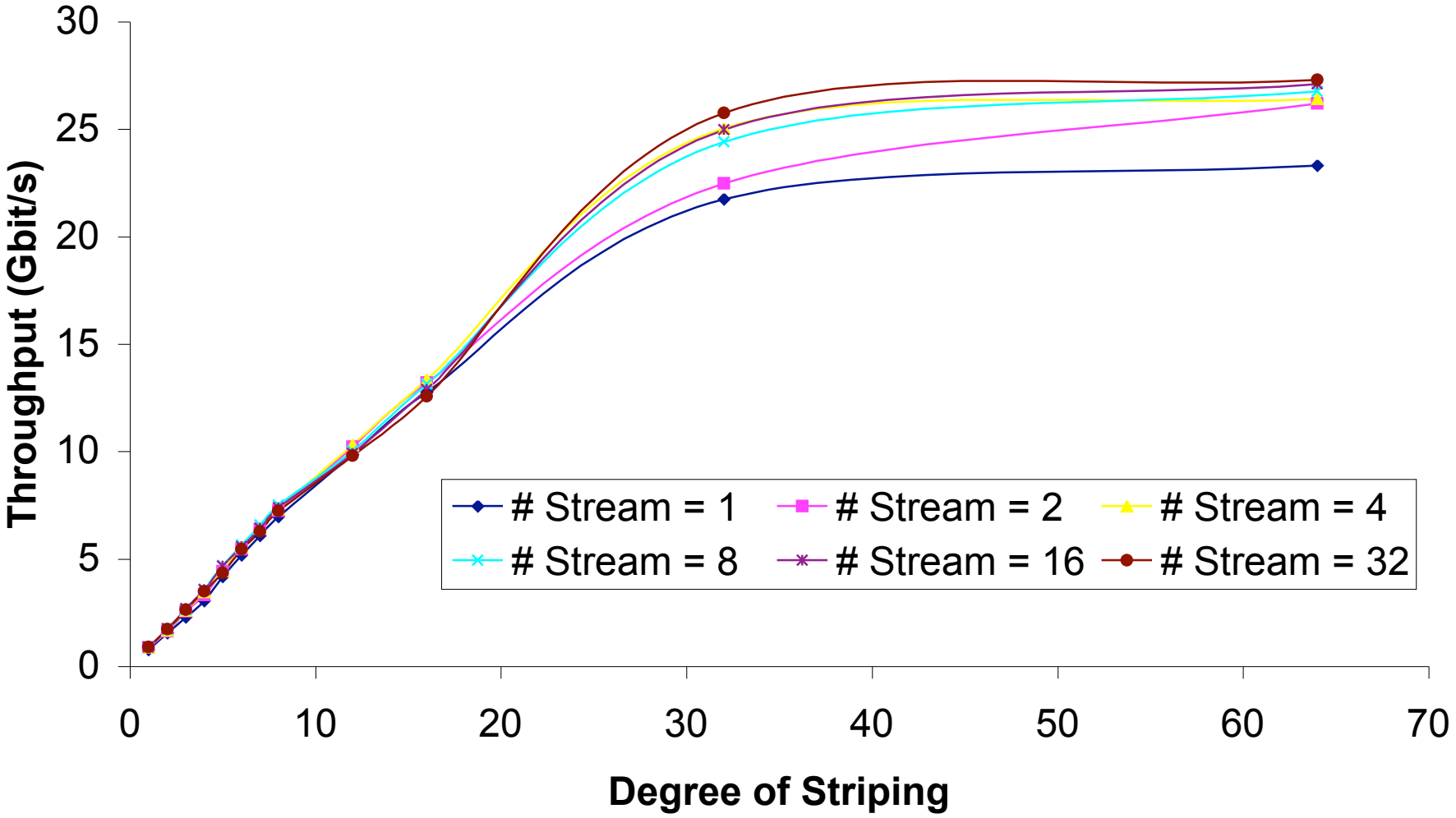
the globus alliance
www.globus.org

GridFTP Servers Around the World

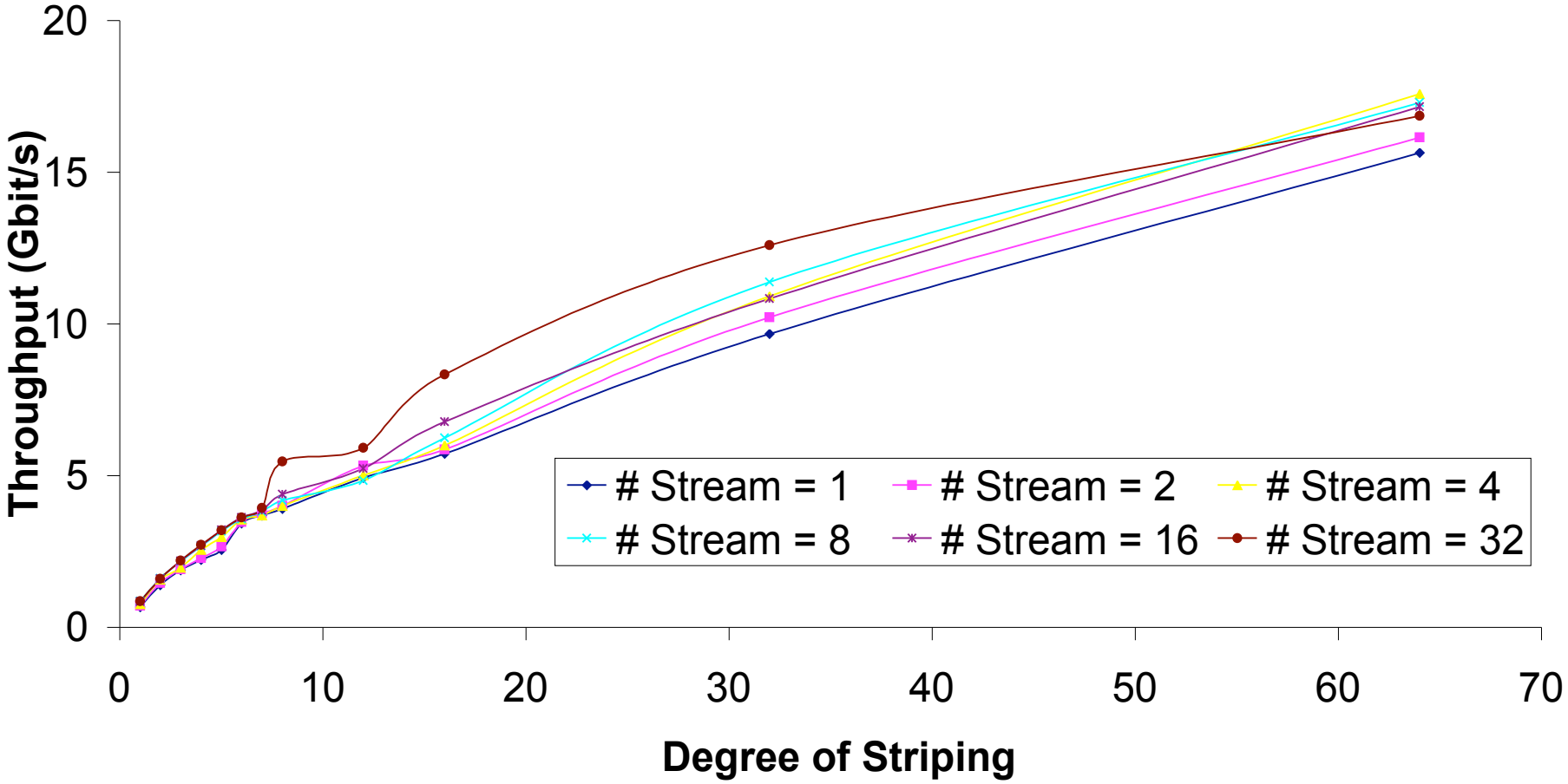


Created by Lydia Prieto ; G. Zarrate; Anda Imanitchi (Florida State University) using MaxMind's GeoIP technology (http://www.maxmind.com/app/ip-locate).

Memory to Memory over 30 Gigabit/s Network (San Diego — Urbana)



Disk to Disk over 30 Gigabit/s Network (San Diego — Urbana)

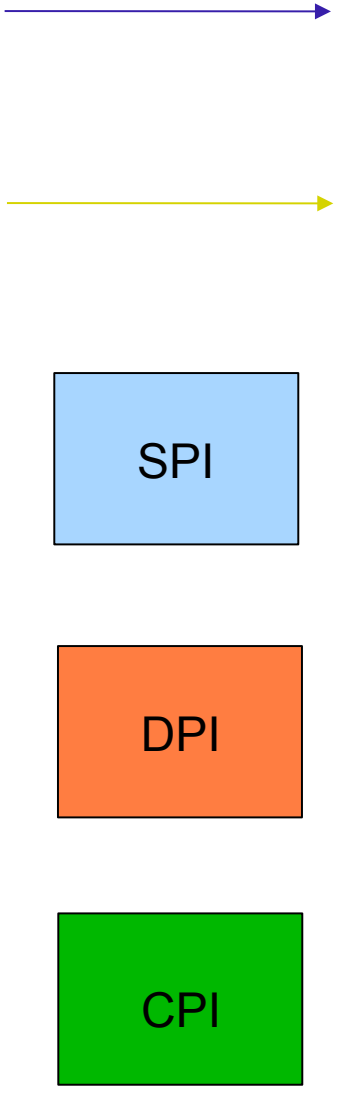


Understanding GridFTP

- Two channel protocol like FTP
- Control Channel
  - Command/Response
  - Used to establish data channels
  - Basic file system operations eg. mkdir, delete etc
- Data channel
  - Pathway over which *file* is transferred
  - Many different underlying protocols can be used
    - MODE command determines the protocol

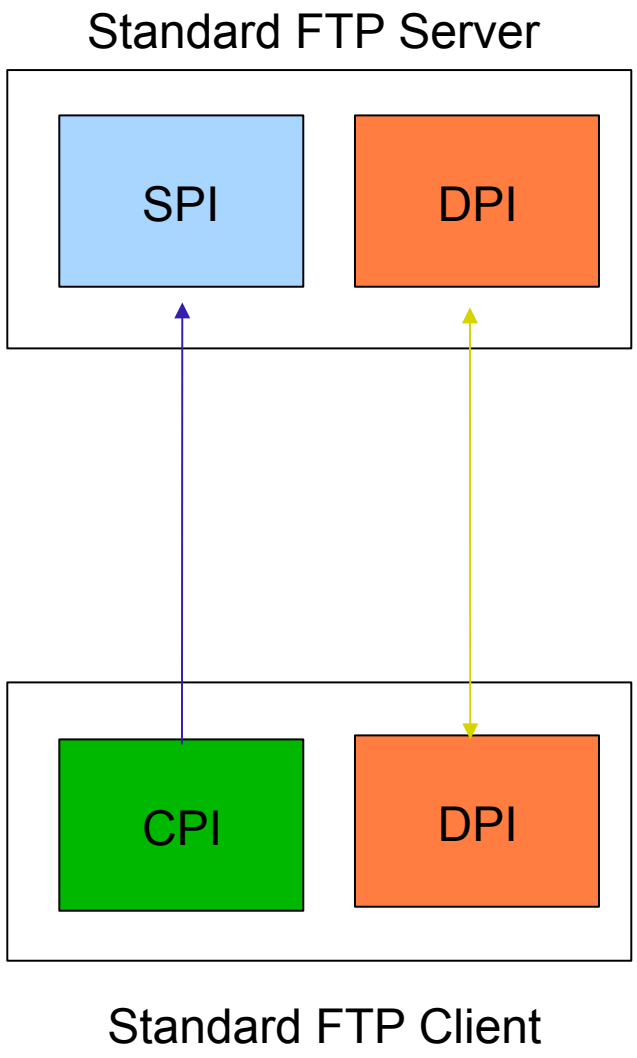
Architecture Components

- 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



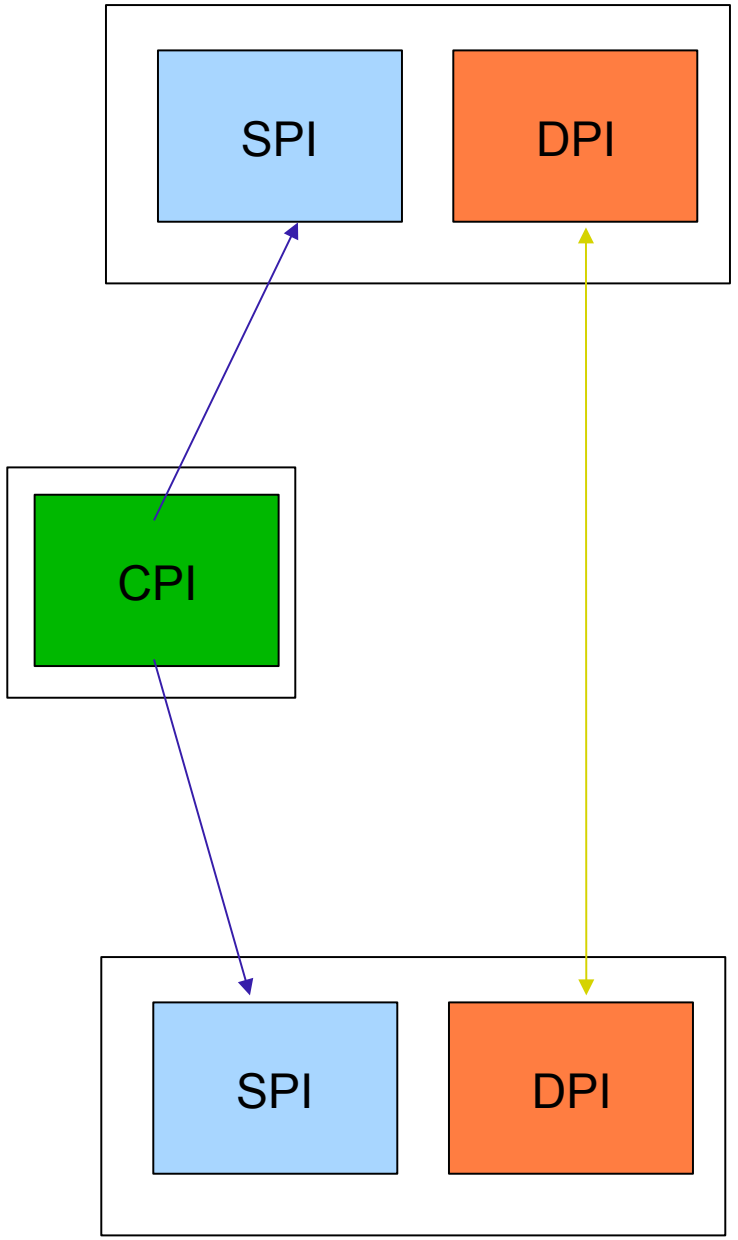
Simple Two Party Transfer

- 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 and forms a CC with the server
  - Information is exchanged to establish the DC
  - A file is transferred over the DC



Simple Third Party Transfer

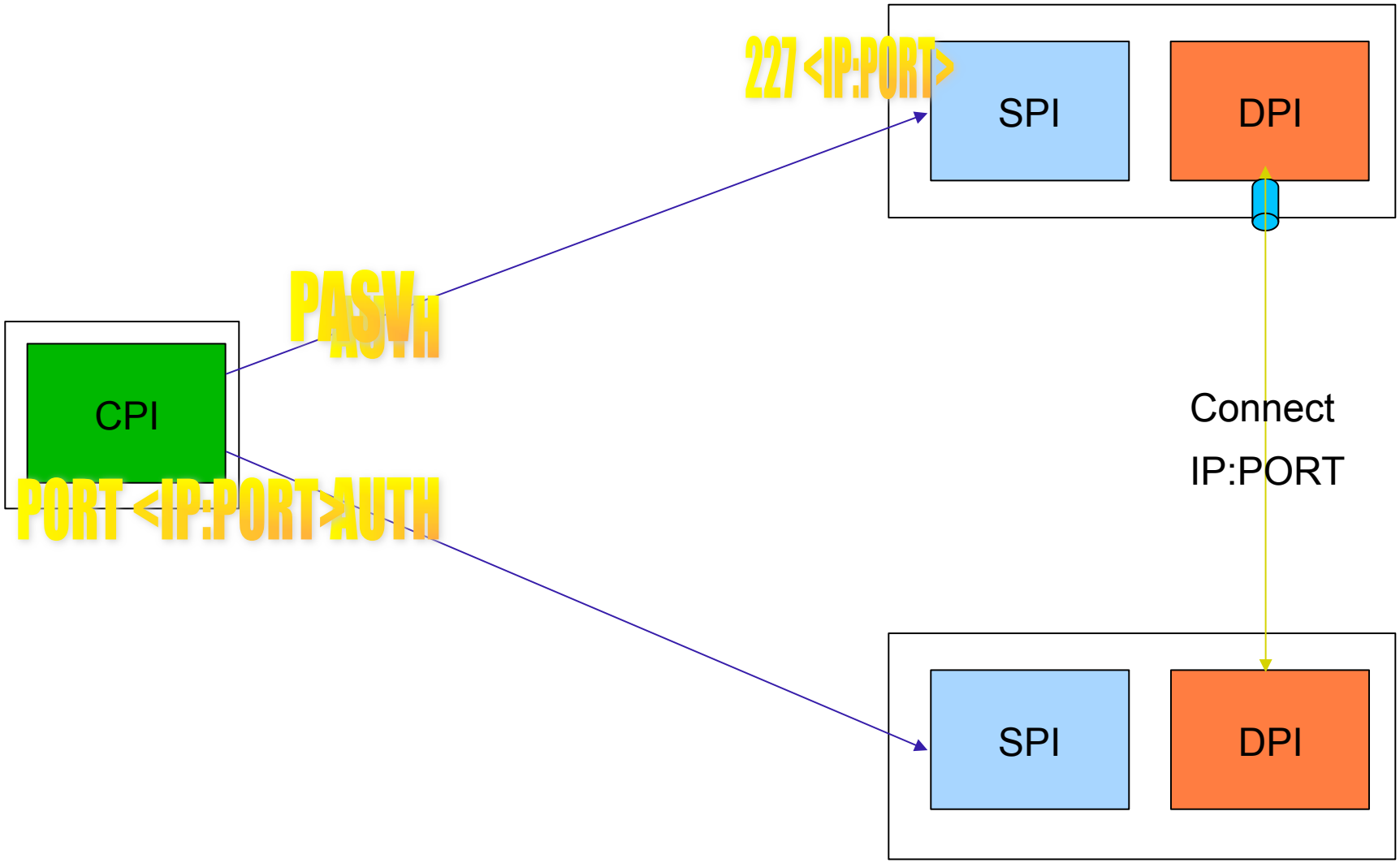
- Client initiates data transfer between 2 servers
- Servers have co-located SPI and DPI
- Client forms CC with 2 servers.
- Information is routed through the client to establish DC between the two servers.
- Data flows directly between servers
  - Client is notified by each server SPI when the transfer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



Control Channel Establishment

- Server listens on a well-known port (2811)
- Client form a TCP Connection to server
- 220 banner message
- Authentication
  - Anonymous
  - Clear text USER <username>/PASS <pw>
  - Base 64 encoded GSI handshake
- 230 Accepted/530 Rejected

Data Channel Establishment



Data Channel Establishment

- Firewall
  - Port side must be able to connect to PASV side
  - Configure port range for the PASV side

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

Descriptor (8 bits)	Size (64 bits)	Offset (64 bits)

Exercise 1
Anonymous Transfer

- Install the GridFTP Server
  - http://www.gridftp.org/tutorials/
  - tar xvfz gt-gridftp*.tar.gz
  - cd gt-gridftp-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
  - 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

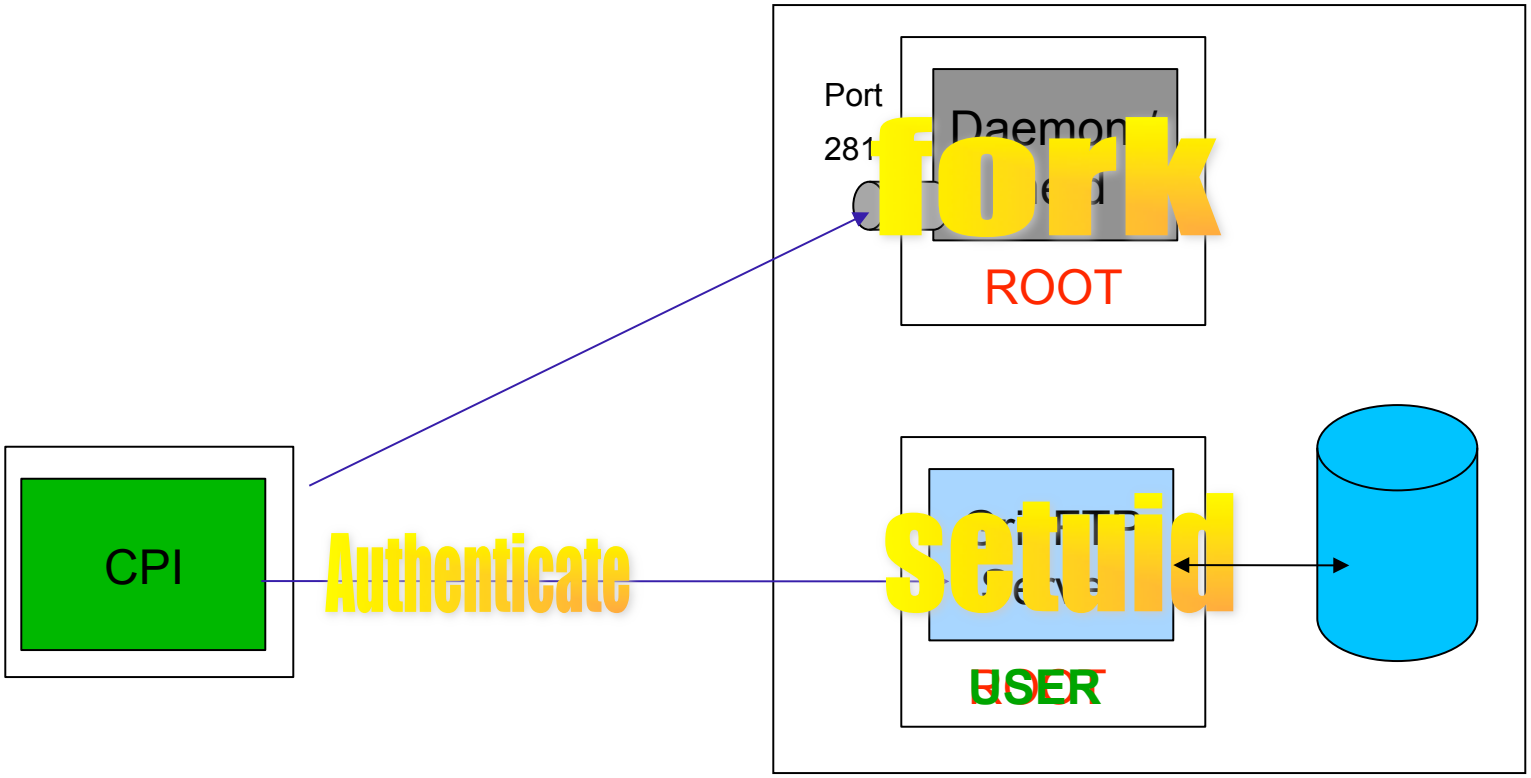
Security Options

- 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

- User is mapped to a local account and file permissions are handled by the OS
- inetd or daemon mode
  - Daemon mode - GridFTP server is started by hand and listens for connections on port 2811
  - Inetd/xinetd - super server daemon that manages internet services
  - Inetd can be configured to start up a GridFTP server upon receiving a connection on port 2811

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.

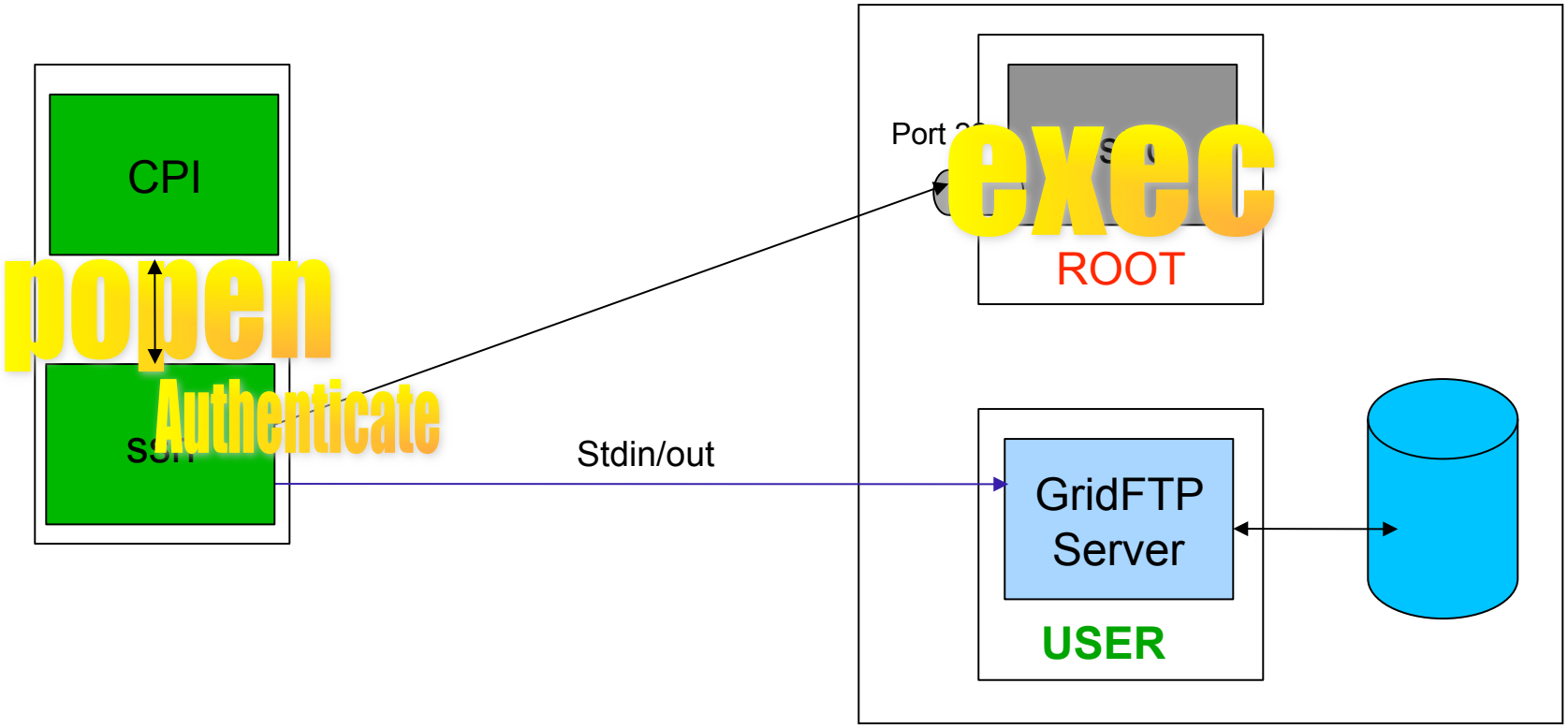
Exercise 2
Password file

- 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 authenicates 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/

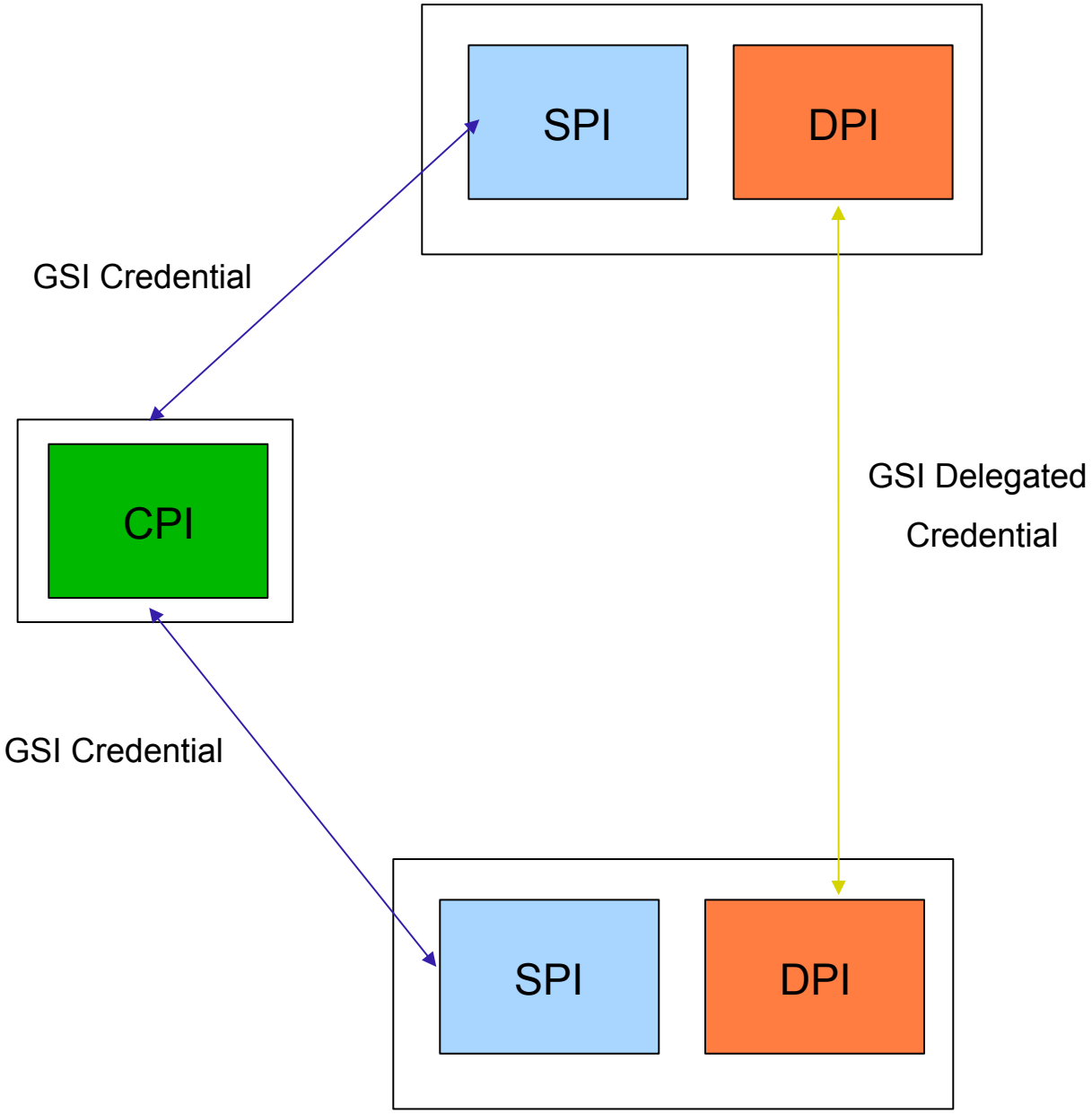
Exercise 3
What happened?

- globus-url-copy popen'ed ssh
- ssh authenticates with sshd
- ssh remotely starts globus-gridftp-server
- guc reads/writes control channel messages from/to ssh
- ssh reads/writes control channel messages from/to stdin/out
- server reads/writes control channel messages from/to 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
  - Enables encryption, integrity on data channel

GSI Authentication



Certificates

- Central concept in GSI
  - Information vital to identifying and authenticating user/service
- Certificate Authority (CA)
  - Trusted 3rd party that confirms identity
- Host credential
  - Long term credential
  - Allows a client to verify the host is what they expect
- 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

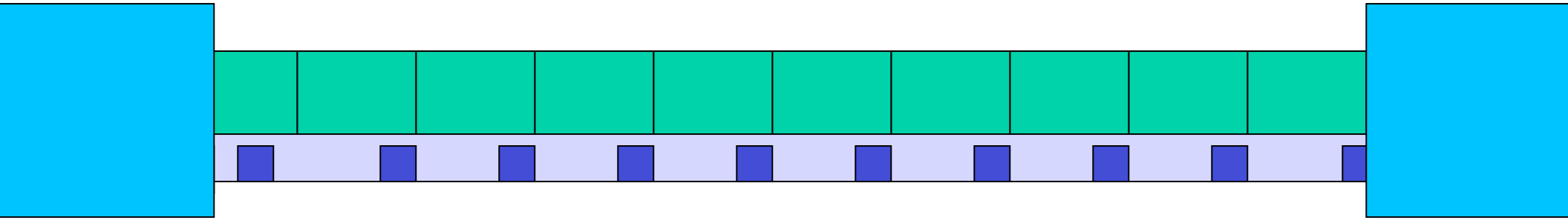
- TCP buffer size
- Disk block size
- Parallel streams
- Cached connections
- Partial file transfers

TCP Buffer Size

- Most important tuning parameter for TCP
  - Memory the kernel *allocates* for retransmits/reordering
  - Affects the maximum window size
  - Amount of data that can be sent before receiving an acknowledgment (ACK)
- Bandwidth Delay Product (BWDP)
  - BWDP = latency * bandwidth
  - The optimal number of bytes that is needed to keep the network pipe full

Window and ACKs

Small TCP Buffer Size

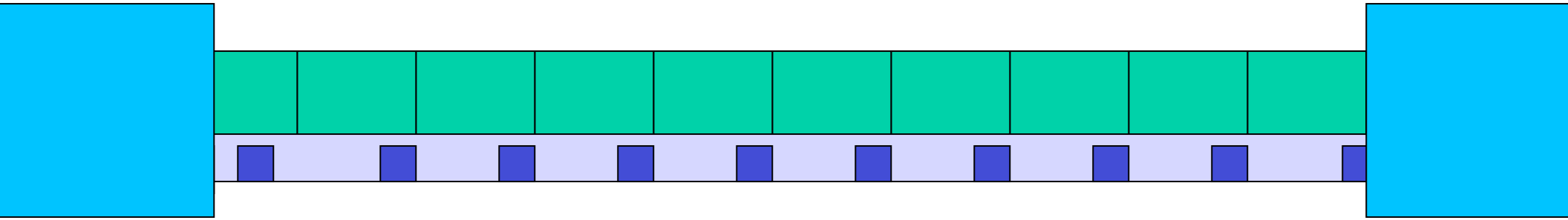


Data Packet

Acknowledgement

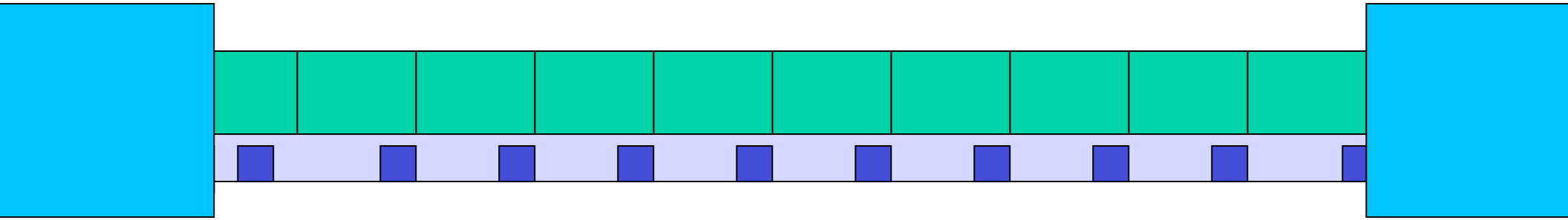
Window and ACKs

Half Full (1 trip)



Window and ACKs

Optimized TCP Buffer Size



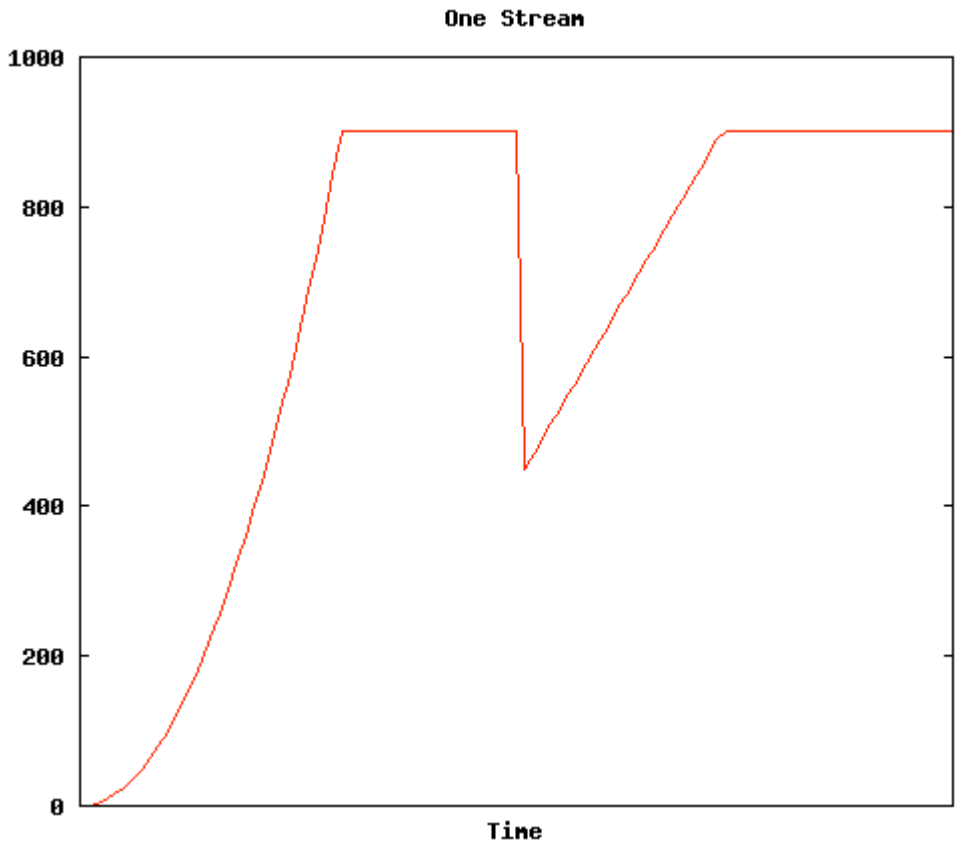
AIMD

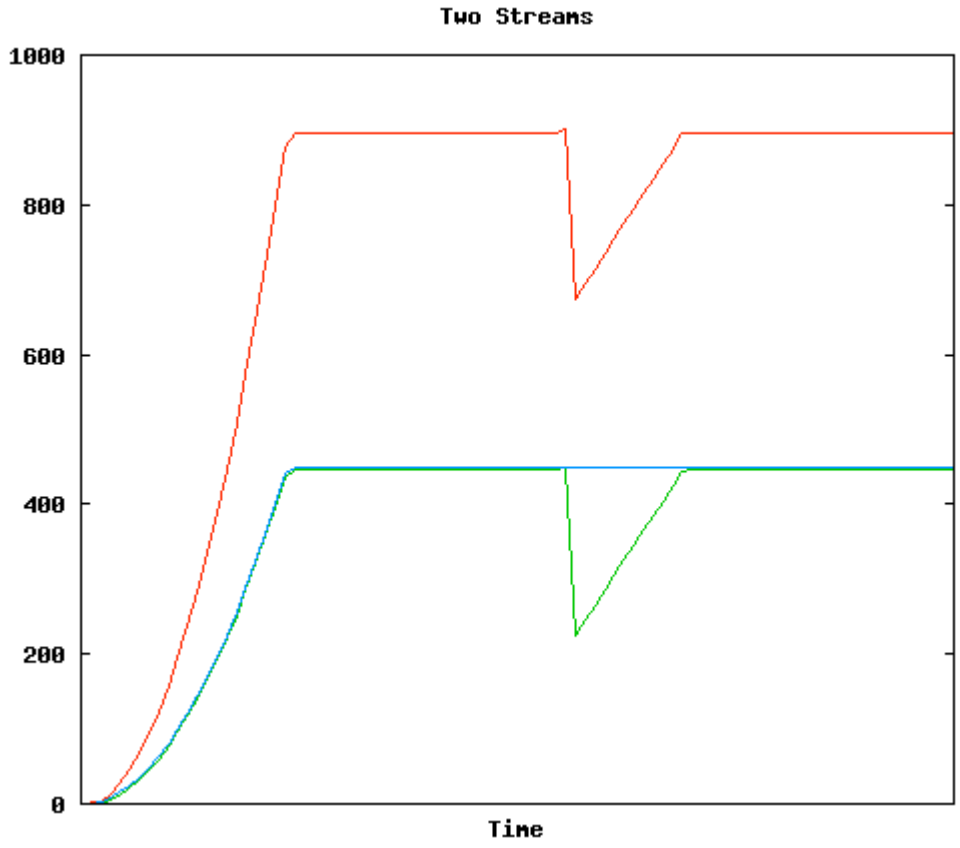
- Additive Increase Multiplicative Decrease
  1. Window size increases exponentially
  2. A congestion event occurs
  3. Window size is cut in half
  4. Window linearly increases
- Conclusion
  - Dropped packets are costly

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 limit

Lost Packets





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

Descriptor (8 bits)	Size (64 bits)	Offset (64 bits)

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

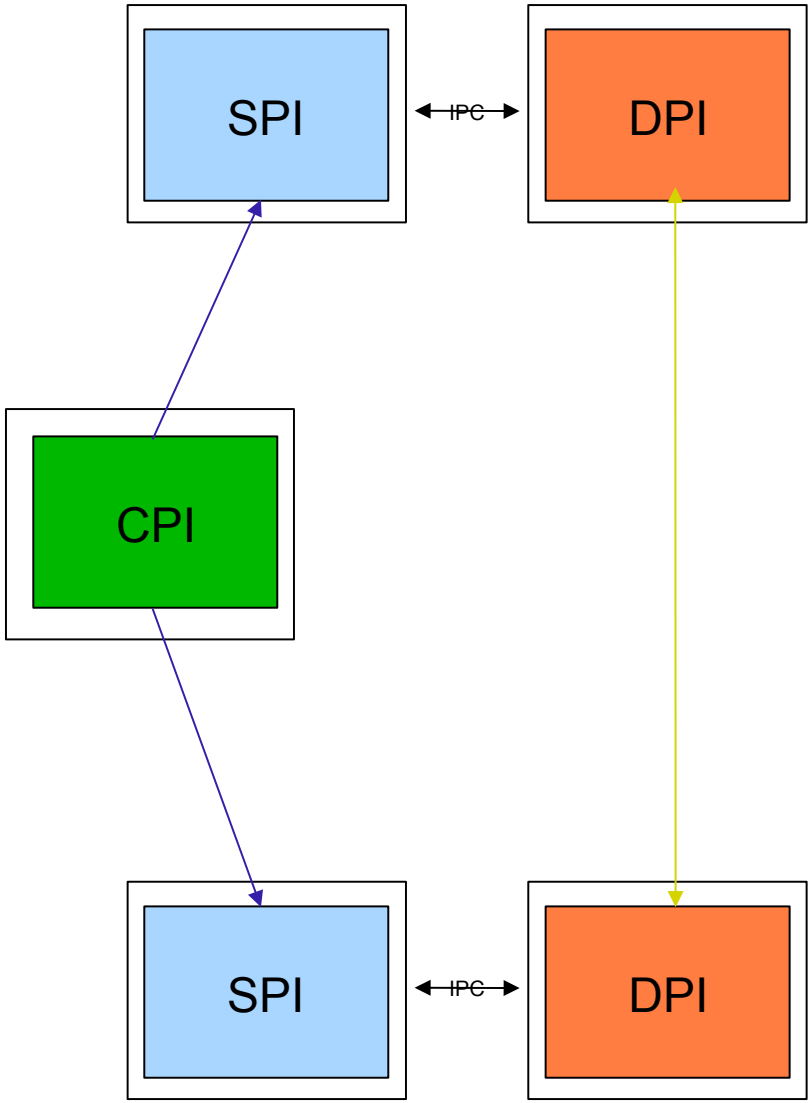
- Large file transfer fails
  - We don't want to start completely over
  - Ideally we start where we left off
- Restart markers sent periodically
  - Contain blocks written to disk
  - Sent every 5s by default
  - In worst case recovery sends 5s of redundant data
- Reliable File Transfer Service

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

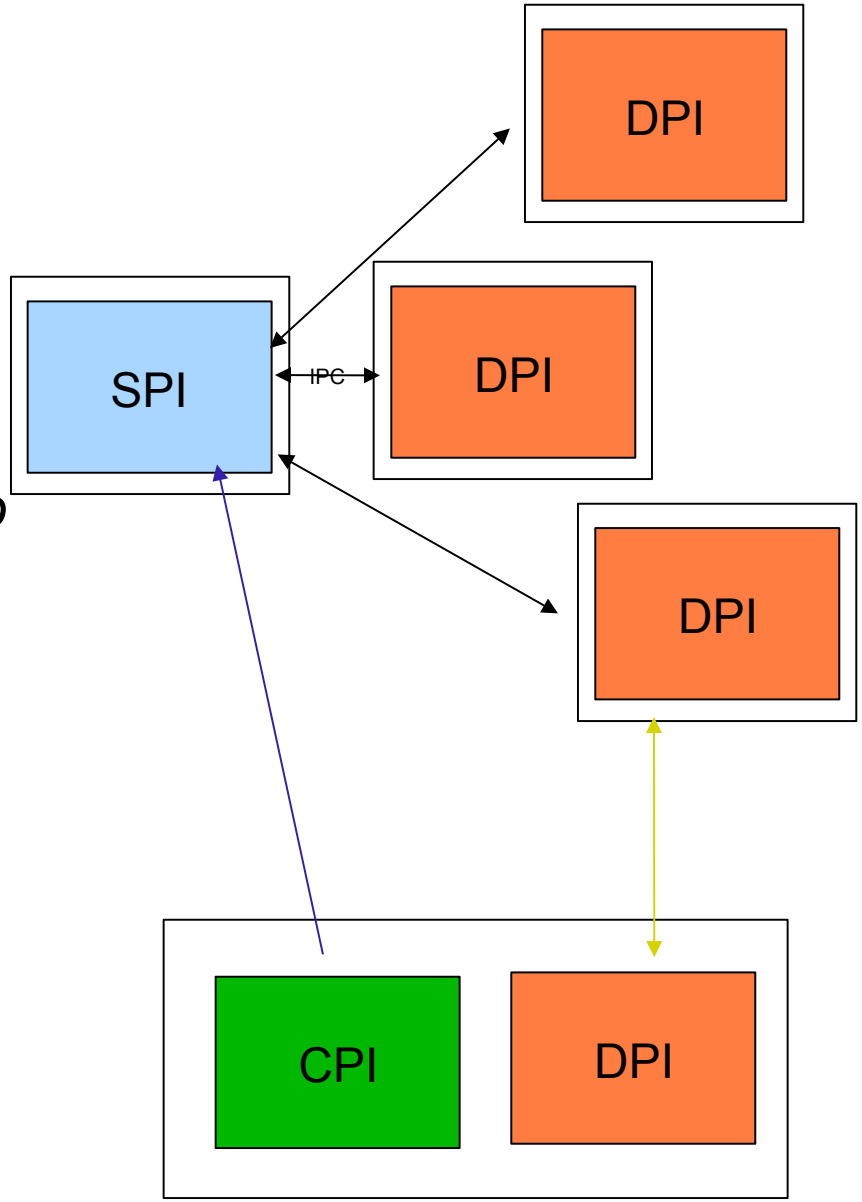


Separation of Process Advantage

- 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

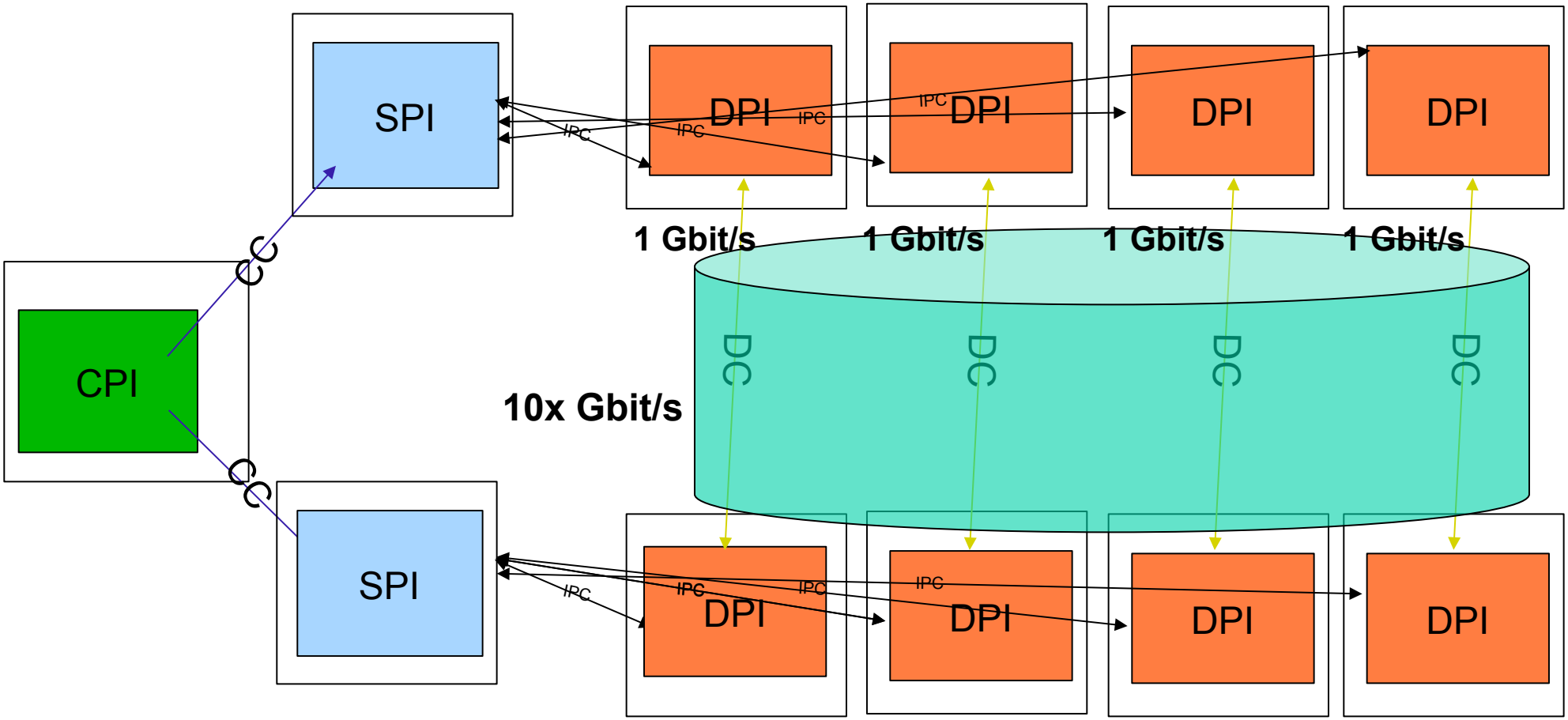
- The separation of processes buys the ability to proxy
  - *Allows for load balancing*
  - *SPI can choose from a pool of DPIs to service a client request*



Striping or Cluster-to-cluster transfer

- A coordinated transfer between multiple nodes at end of the transfer
  - 1 SPI at each end
  - Many DPIs per SPI
  - Each DPI transfers a portion of the file
  - Allows for fast transfers
  - Many NICs per transfer

Cluster-to-cluster transfer

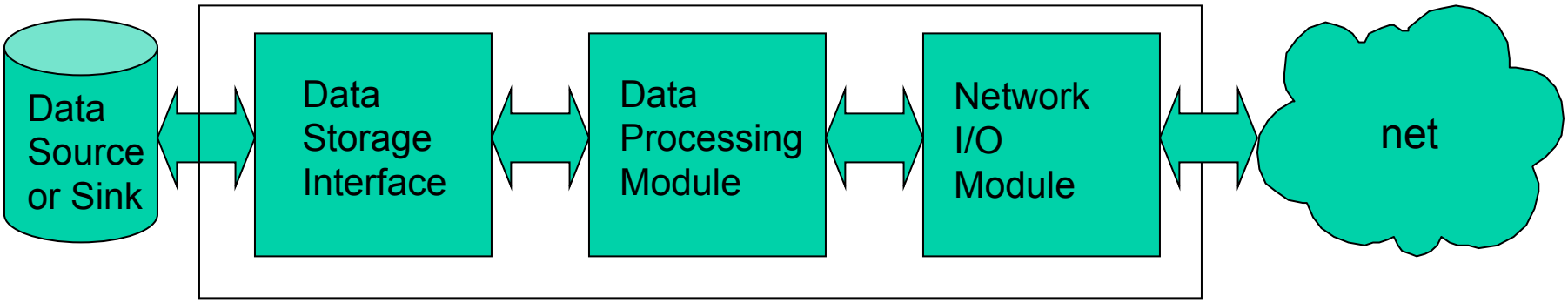


Demonstration 2
Striping

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

Modular

- Globus GridFTP is based on XIO and is modular
- Well-defined interfaces

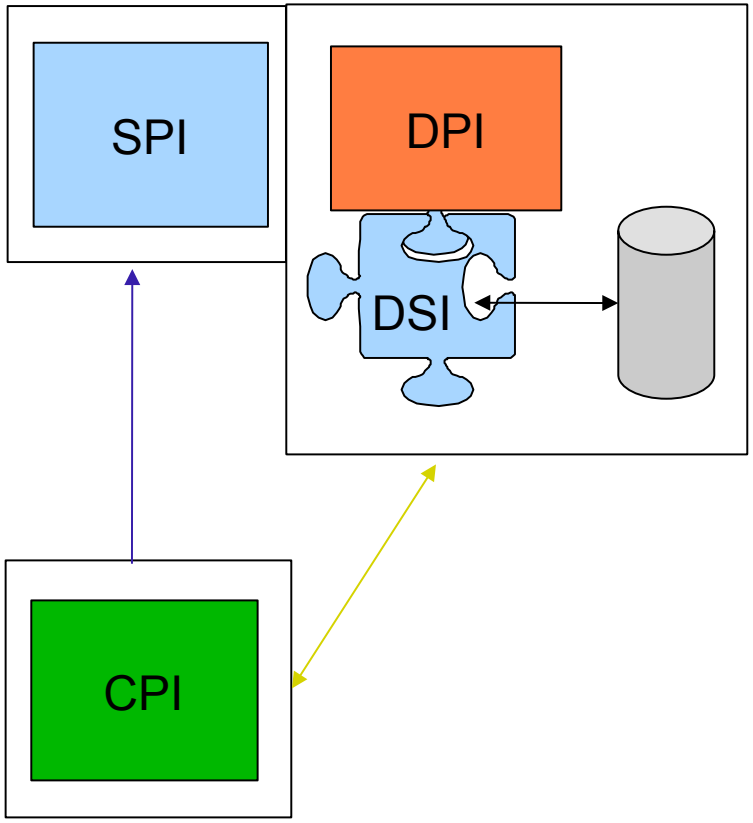


Data Storage Interface (DSI)

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

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
- Existing DSIs
  - HPSS, SRB, POSIX FS (default)

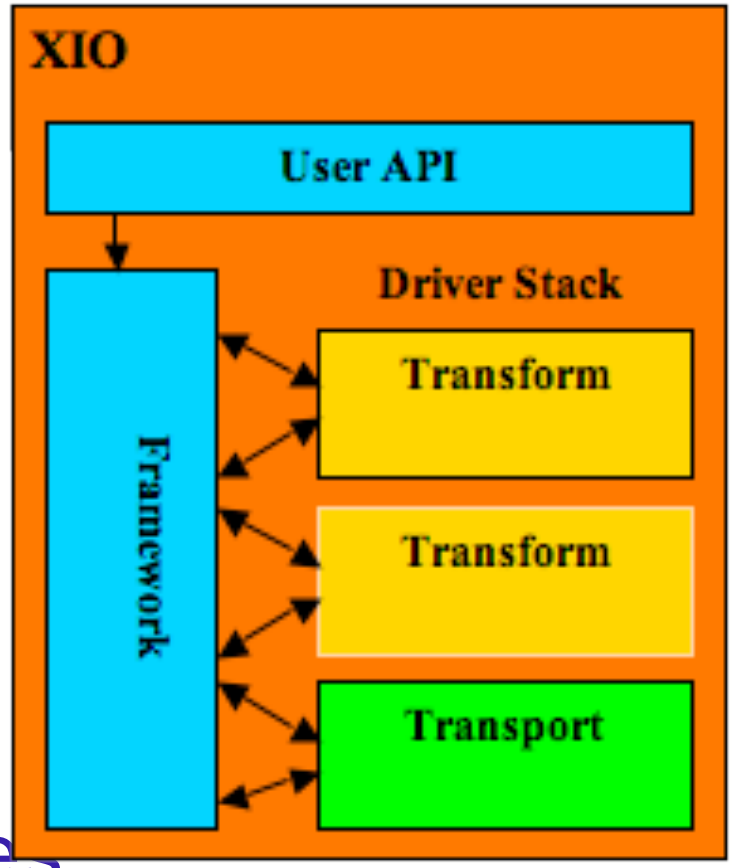


HPSS and SRB DSIs

- http://www.hpss-collaboration.org/hpss/administrators/docs/HTML/rel6.2/GridFTPHPSS.jsp
- http://www.globus.org/toolkit/docs/4.0/data/gridftp/GridFTP_SRB.html

Globus XIO

- Framework to compose different protocols
- Provides a unified interface open/close/read/write
- Driver interface to hook 3rd party protocol libraries

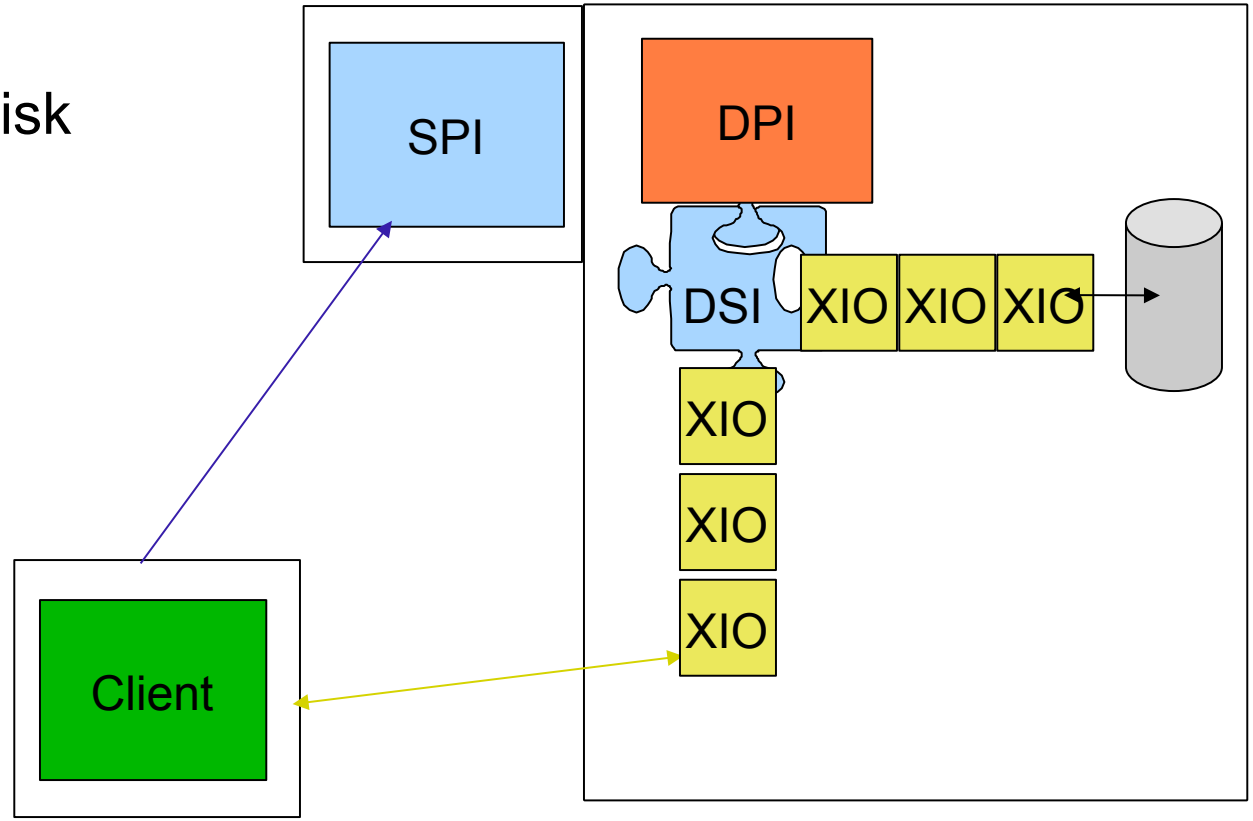


Alternative stacks

- 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: compression, 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

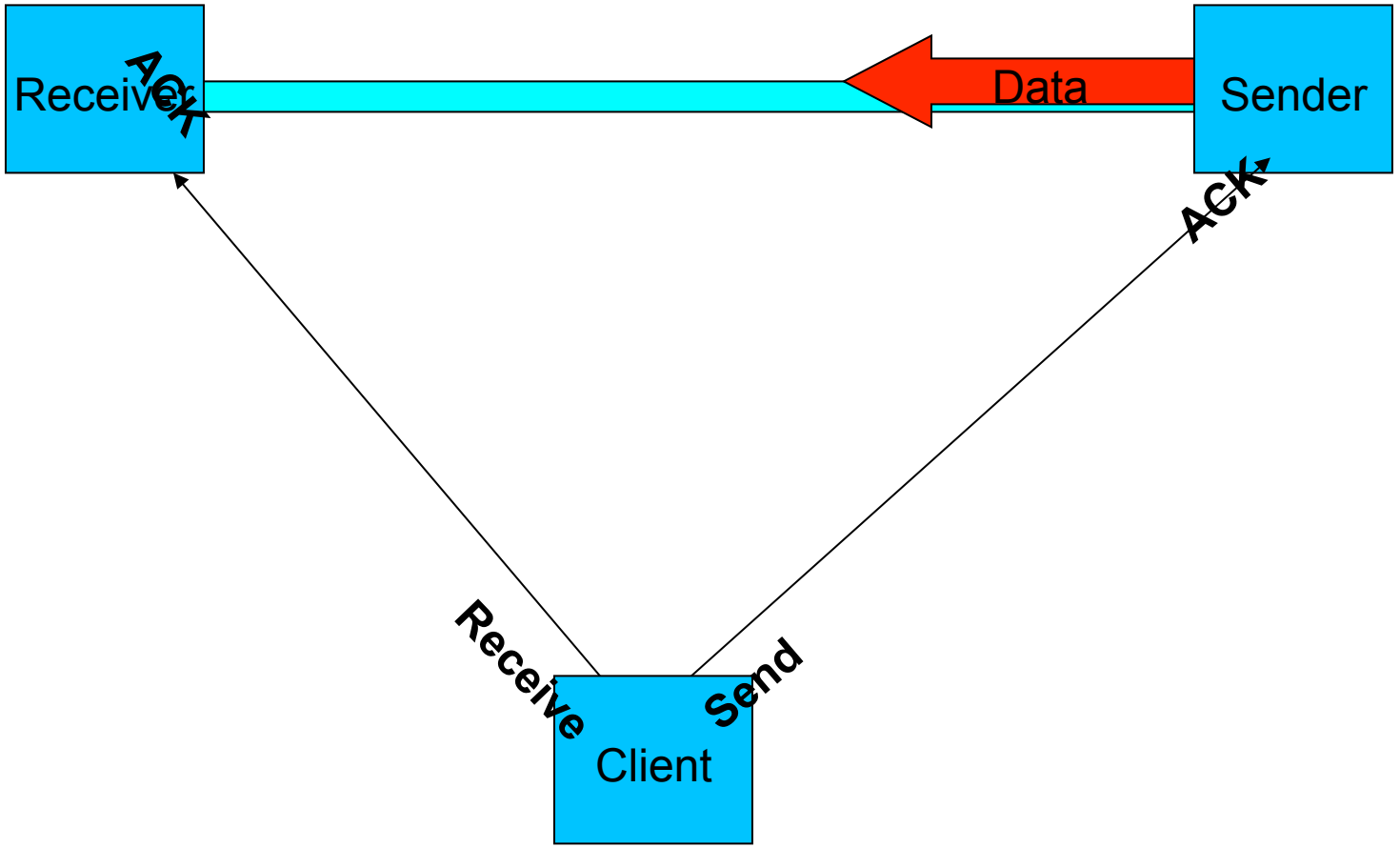


Demonstration 3

GridFTP over UDT

- Show a transfer with UDT as the transport protocol
  - Show dynamic data channel protocol stack selection
  - Show the performance increases
- Requirements
  - Threaded build of the Globus GridFTP server
  - Threaded build of globus-url-copy (for client-server transfers)
- Transferring a file
  - globus-url-copy -udt file:///etc/group ftp://localhost:5000/tmp/group

Lots of Small Files (LOSF) Problem

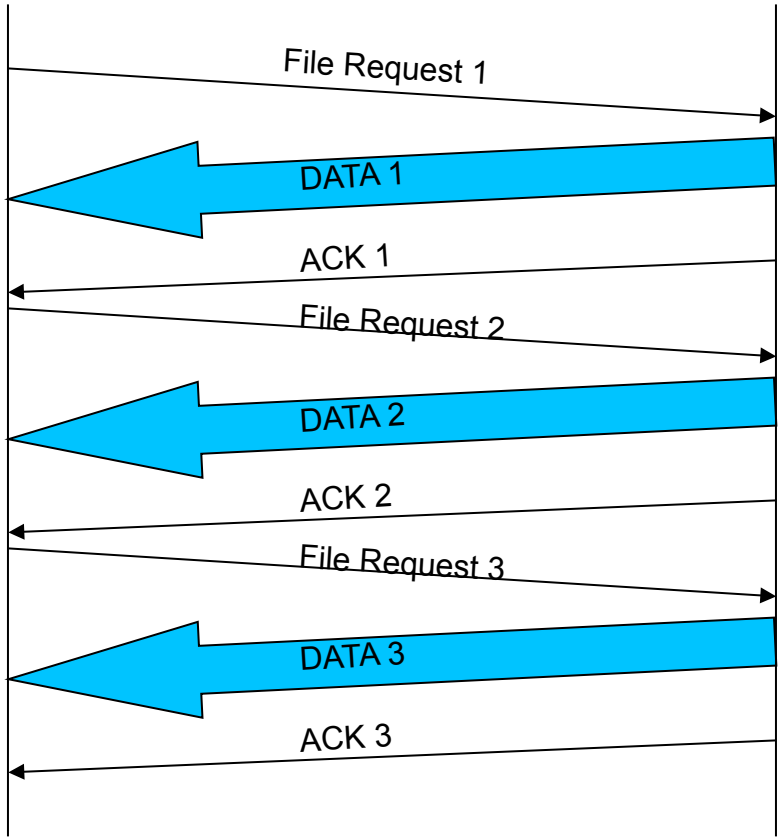


Solution - Pipelining

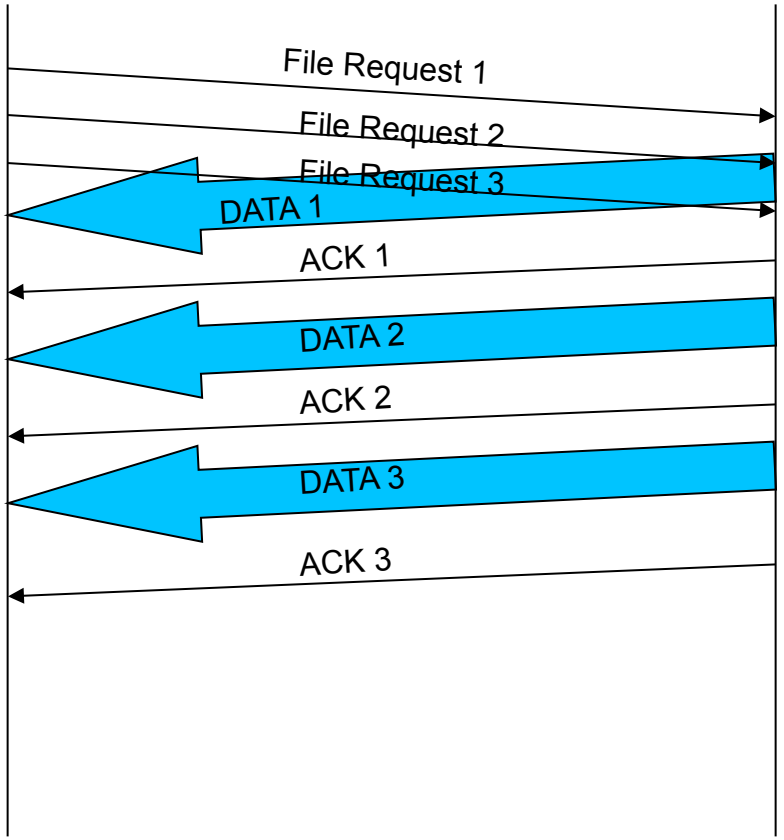
- Allow many outstanding transfer requests
- Send next request before previous completes
  - Latency is overlapped with the data transfer
- Backward compatible
  - Wire protocol doesn't change
  - Client side sends commands sooner
- '-pp' option in globus-url-copy enables pipelining

Pipelining

Traditional



Pipelining



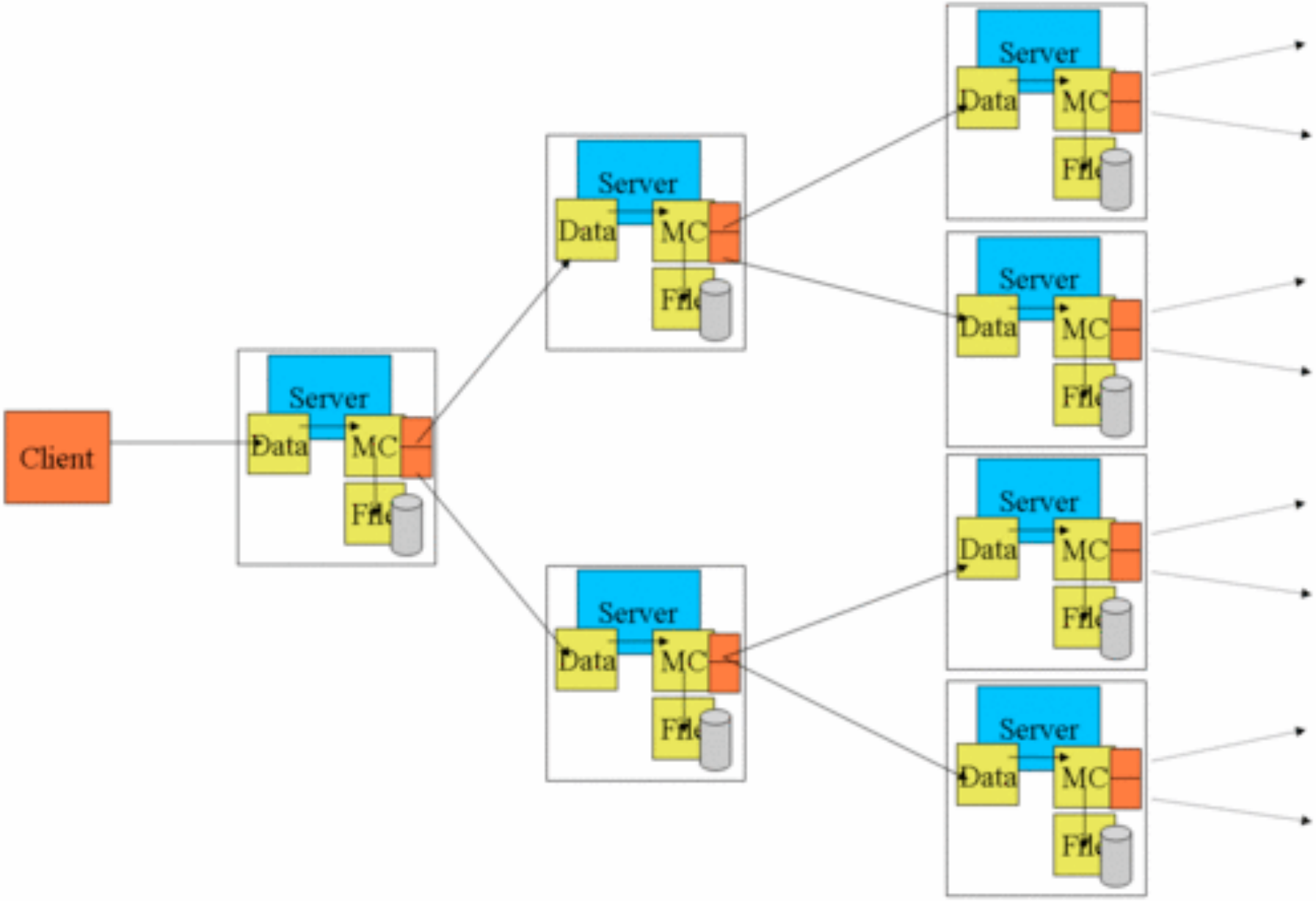
Significant performance improvement for LOSF

Demonstration 4

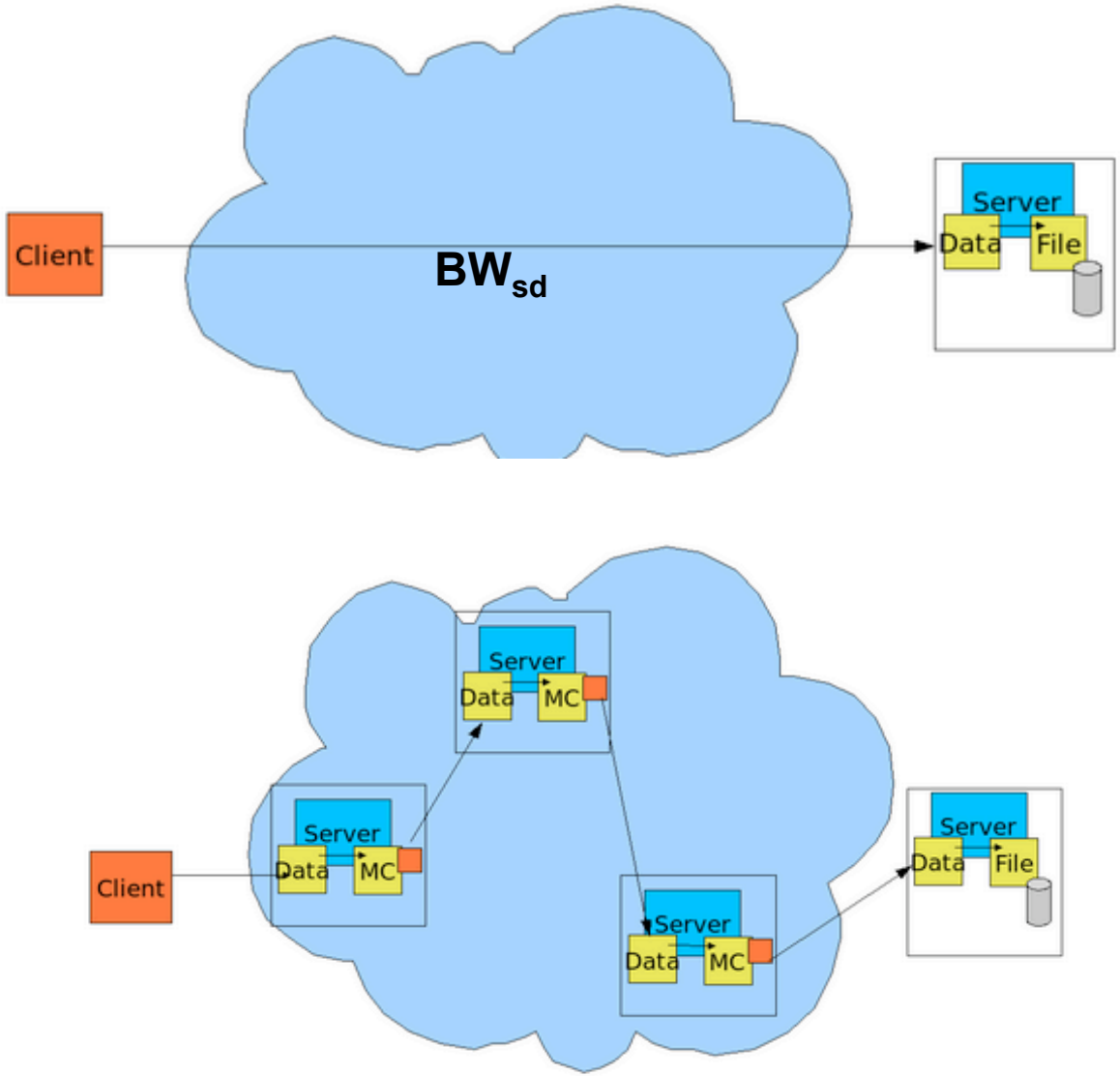
Pipelining

- Show lots of small files transfers with and without pipelining
  - Show performance increases with pipelining

GridFTP Multicasting



GridFTP Overlay Network



If Min(BW_{sa} , BW_{ab} , BW_{bc} , BW_{cd}) > BW_{sd},

Overlay route yields better performance

Demonstration 5

- Show a one-to-many transfer with and without multicasting
- Show a transfer to demonstrate how GridFTP overlay routing can improve performance

New Features

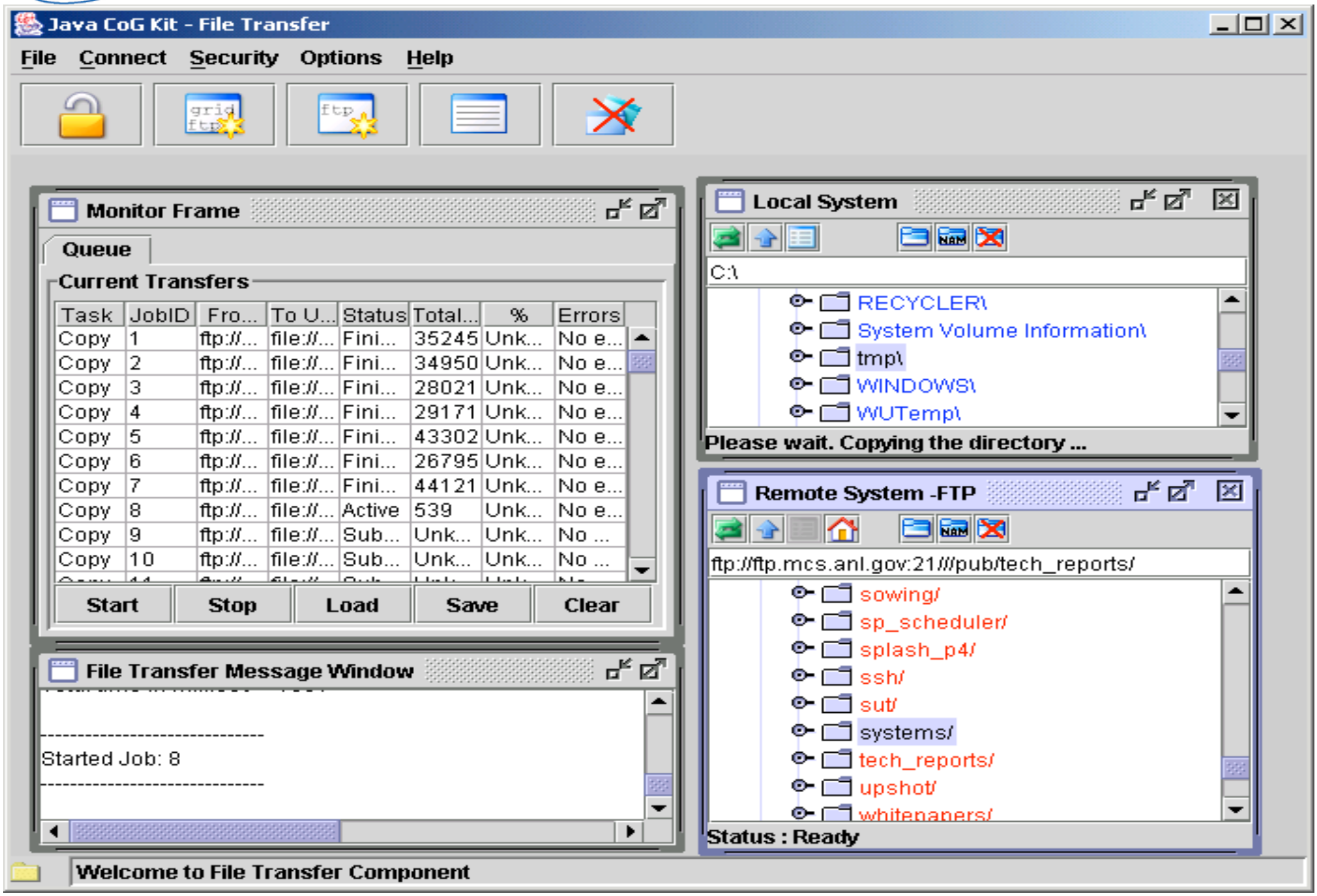
- GUI client
- GFork
  - Resource management
  - Dynamic backends
- Popen

GUI Client

- An alpha version is available at http://www.globus.org/cog/demo/
- Java web start application
- Integrated with myproxy-logon
  - Certificates can be completely hidden from the user
- If certificates are in place, proxy can be generated through the GUI
- Provides support for RFT as well

the globus alliance
www.globus.org

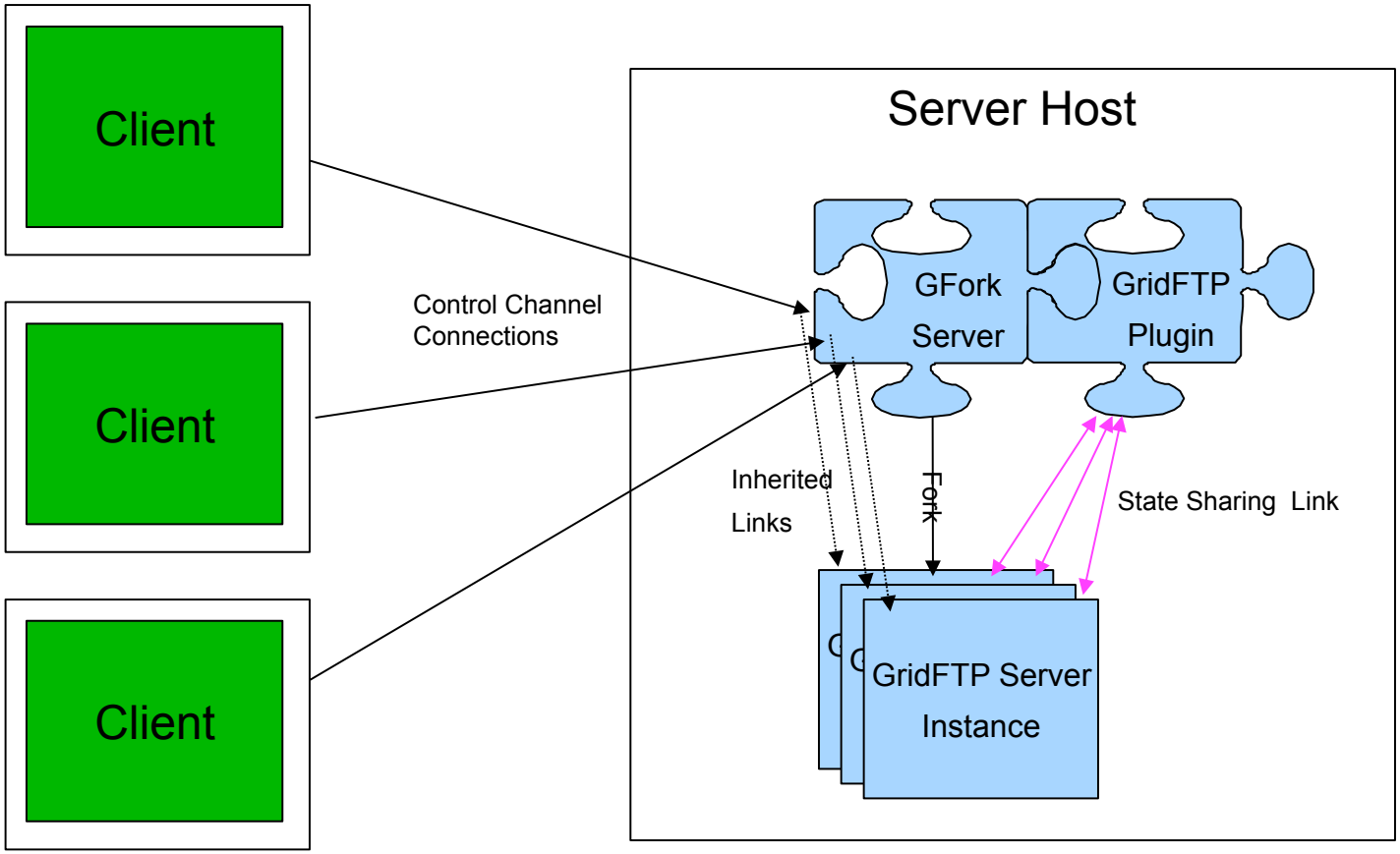
GUI Client



GFork

- Super-server daemon very similar to xinetd
- Xinetd - no way to maintain long-term and shared state between sessions
  - Required to manage the resources on a GridFTP node
- GFork is designed to address this situation

GFork


Client
Client
Client
Control Channel
Connections
Server Host
GFork
Server
GridFTP
Plugin
Inherited
Links
Fork
State Sharing Link
GridFTP Server
Instance

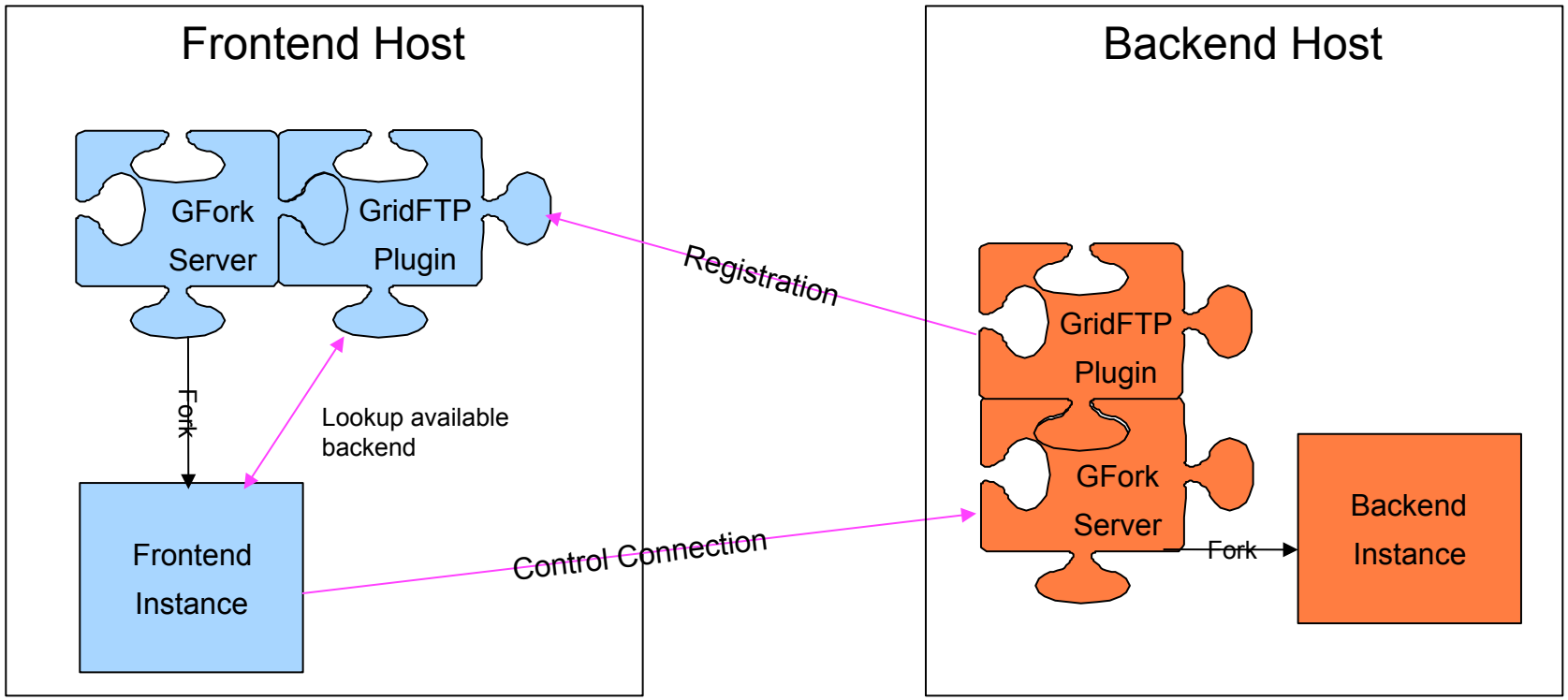
Dynamic Backends

- Dynamic list of available backends (DPIs)
- Frontend (SPI) listens for registration
  - Backends register (and timeout)
  - Select backend(s) to use for a transfer
- Backend failure is not system failure
- Resources can be provisioned to suit load

Dynamic Backends

- Dynamic list of available backends (DPIs)
- Frontend (SPI) listens for registration
  - Backends register (and timeout)
  - Select backend(s) to use for a transfer
- Backend failure is not system failure
- Resources can be provisioned to suit load

Dynamic Backends



Popen

- Popen XIO driver
  - allows users to open pipes to the standard IO of existing programs
  - leverage programs like you can with UNIX pipes
  - globus-gridftp-server -p 5000 -fs-whitelist popen,file,ordering -aa
  - globus-url-copy -dst-fsstack popen:argv=#/usr/bin/zip#/home/bresnaha/text.txt.zip#-,ordering ftp://localhost:5000/home/bresnaha/text.txt ftp://localhost:5000/y

Feedback

- Comments welcome
- If you need any specific functionality requirement, please let us know

Thank you

- More Information:
  - http://www.gridftp.org
  - http://www.globus.org/toolkit
  - gridftp-user@globus.org