Setting up and using a Globus Toolkit 5 based Grid

Raj Kettimuthu
Stuart Martin
Bill Mihalo

Argonne National Laboratory and The University of Chicago
Outline

- Introduction
  - Grid and Globus Toolkit
- Grid Security Infrastructure
- GT5 Installation and Configuration
- GridFTP
- GRAM
The Grid

- Resource sharing & coordinated problem solving in dynamic, multi-institutional virtual organizations
  - “On-demand” access to ubiquitous distributed computing
  - Transparent access to distributed data
  - Easy to plug resources into
  - Complexity of the infrastructure is hidden
Virtual Organization

Site A

Site B

Site C

Site D
“Coordinating multiple resources”: ubiquitous infrastructure services, app-specific distributed services

“Sharing single resources”: negotiating access, controlling use

“Talking to things”: communication (Internet protocols) & security

“Controlling things locally”: Access to, & control of, resources
The Globus Toolkit centers around

- Connectivity layer:
  - Security: Grid Security Infrastructure (GSI) - allows collaborators to share resources without blind trust

- Resource layer:
  - Resource Management: Grid Resource Allocation Management (GRAM)
  - Data Transfer: Grid File Transfer Protocol (GridFTP)

- Also collective layer protocol
  - Replica Management (RLS)

- Focuses on simplifying heterogeneity for application developers
Grid Security Infrastructure (GSI)

- Open source libraries, tools and standards which provide security functionality of the Globus Toolkit
- Goal is to support VO
- Provides for cross-organizational:
  - Authentication
  - Authorization
  - Single sign-on
Terminology

Authentication: Proving who you are.

Authorization: What are you allowed to do?

Delegation: Granting a right to another entity.

John Doe @ NCSA
GSI

- Based on asymmetric cryptography
  - Private and Public Key - allows for two entities to authenticate with minimal cross-organizational support

- Certificates - Central concept in GSI
  - Information vital to identifying and authenticating user/service
  - Distinguished Name – unique Grid id for user/service
  - "/DC=org/DC=doegrids/OU=People/CN=Raj Kettimuthu 227852"

- Certificate Authority (CA)
  - Trusted 3rd party that confirms identity

- Host credential
  - Long term credential

- User credential
  - Passphrase protected
Digital Signatures

- Used to determine if the data has been tampered
- Also, identify who signed the data
- Digital signatures are generated by
  - Creating secure hash of the data
  - Encrypting the hash with private key
- The resulting encrypted data is the signature
- This hash can then be decrypted only by the corresponding public key
Certificates

- Allow for binding of an Identity (John Doe) to a key or person
Proxy Certificates

- X.509 Proxy Certificates are our extension
- Standardized in IETF
- Allow for dynamic delegation
- Proxy credentials are short-lived credentials created by user
  - Proxy signed by user certificate private key
- Stored unencrypted for easy repeated access
Delegation

- Enabling another entity to run on behalf of you
- E.g. Service that runs a job needs to transfer files.
- Ensure
  - Limited lifetime
  - Limited capability
- GSI uses proxy certificates for delegation
Authorization

- Establishing rights of an identity
  - Can user do some action on some resource
- Identity based authorization
  - Establish identity using authentication
  - Check policy to see what identity can do
  - Eg: Gridmap authorization a list of mappings from allowed DNs to user name
    - "/DC=org/DC=doegrids/OU=People/CN=Raj Kettimuthu 227852" kettimut
  - Identity based authorization may not scale
- Attribute based authorization
  - Attributes are information about an entity
    - Employee of Argonne National Lab
    - Member of virtual organization ABC
GSI Stack

• GSI uses a standard PKI for identity certificates.
• Each entity (user, service) has an X.509 certificate from a CA that uniquely names it.
GSI Stack

SSL

PKI (Certs, CAs)
GSI Stack

- X.509 Proxy Certificates
- SSL
- PKI (Certs, CAs)

- X.509 Proxy Certificates are our extension
- Standardized in IETF (pkix)
- Allow for dynamic delegation
GSI Stack

- Grid-Mapfile
- X.509 Proxy Certificates
- SSL
- PKI (Certs, CAs)

- Grid-Mapfile maps Grid users (identified by certificates) to local users (e.g. Unix account)
- Allows authorization using normal local methods (e.g. filesystem perms, quotas)
GSI-Enabled Coordination

Site A

Allows for standard authentication method

Site B

Proxy Certificate

Site C

Allows for delegation to allow for coordinated resource usage
GSI in Action: “Create Processes at A and B that Communicate & Access Files at C”

- Single sign-on via “grid-id” & generation of proxy cred.
- Or: retrieval of proxy cred. from online repository

**User**

- User Proxy
  - Proxy credential
  - Remote process creation requests*
  - Remote file access request*

**GSI-enabled GRAM server**

- Process
  - Local id
  - Delegated Proxy cred

**GSI-enabled GRAM server**

- Process
  - Local id

**Site C**

- GSI-enabled GridFTP server
  - Authorize
  - Map to local id
  - Access file

**Site A**

- GSI-enabled GRAM server
  - Authorize
  - Map to local id
  - Create process
  - Generate credentials

**Site B**

- GSI-enabled GRAM server
  - Ditto

* With mutual authentication
MyProxy – credential repository

- Allows users to acquire Grid credentials from Username/Password
- Enables mobility

MyProxy server

Web Server

The Grid
Globus Toolkit 5
Installation Demo
Installation Steps

**Installing Globus**

- `wget http://www.globus.org/ftppub/gt5/5.0/5.0.0/installers/src/gt5.0.0-all-source-installer.tar.bz2`
- `tar xvfz gt5.0.0-all-source-installer.tar.bz2`
- `cd gt5.0.0-all-source-installer`
- `./configure -prefix /path/to/install`
- `make`
- `make install`
Fetching User and Host Caps

- https://pki1.doegrids.org/ca/
- download the DOE support CA files tarball from http://pki1.doegrids.org/Other/doegrids.tar
- untar it into /etc/grid-security/certificates
- cp /etc/grid-security/doegrids/globus-host-ssl.conf.1c3f2ca8 /etc/grid-security/globus-host-ssl.conf
  cp /etc/grid-security/doegrids/globus-user-ssl.conf.1c3f2ca8 /etc/grid-security/globus-user-ssl.conf
  cp /etc/grid-security/doegrids/grid-security.conf.1c3f2ca8/etc/grid-security/grid-security.conf
- run 'grid-cert-request -host <hostname>' from your Globus install
- Go to http://pki1.doegrids.org/ca/
  Select "Grid or SSL Server". Paste the Certificate Signing Request into the "PKCS#10 Request" text box. Fill out the rest of the form and "Submit".
GridFTP
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**

- **Multiple security options**
  - Anonymous, password, SSH, GSI

- **Support for reliable and restartable transfers**
GridFTP Servers Around the World

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
Client/Server and 3rd Party Transfers

- Two party transfer
  - The client connects and forms a CC with the server
  - Information is exchanged to establish the DC
  - A file is transferred over the DC

- Third party transfer
  - Client initiates data transfer between 2 servers
  - 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

GridFTP Client

GridFTP Server

Connect
IP:PORT

GridFTP Server

IP:PORT

PASV

PORT

AUTH

227 <IP:PORT>
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

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Size</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>(8 bits)</td>
<td>(64 bits)</td>
<td>(64 bits)</td>
</tr>
</tbody>
</table>
Globus-url-copy

- Command line scriptable client
- Globus does not provide an interactive client
- Commonly used client for GridFTP
- Syntax overview
  - `globus-url-copy [options] srcURL dstURL`
  - `guc gsiftp://localhost/foo file:///bar`
    - Client/server, using FTP stream mode
  - `guc -vb -dbg -tcp-bs 1048576 -p 8 gsiftp://localhost/foo gsiftp://localhost/bar`
    - 3rd party transfer, MODE E
- URL rules
  - `protocol://[user:pass@][host]/path`
  - host can be anything resolvable - IP address, localhost, DNS name
Demonstration

- **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**

- **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**
Demonstration
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**

```plaintext
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**

```plaintext
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.
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

- CPI
- SSH
- Authenticate
- Popen
- GridFTP Server
- Port 22
- USE
- ROOT
- Stdin/out

GlobusWorld 2010
GSI Authentication

bullet Strong security on both channels
  - SSH does not give us data channel security

bullet 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

- GridFTP Client
- GridFTP Server
  - GSI Credential
  - GSI Delegated Credential
**Troubleshooting**

- **Can I get connected?**
  - telnet to the port: `telnet hostname port`
  - 2811 is the default port
- **You should get something like this:**
  - `<add GridFTP banner>`
- **If not, you have firewall problems, or server config problems.**
Troubleshooting

- no proxy
  - grid-proxy-destroy
  - guc gsiftp://localhost/dev/zero file:///dev/null
  - add –dbg
  - grid-proxy-init
  - guc gsiftp://localhost/dev/zero file:///dev/null
  - add –dbg
Setting TCP buffer sizes

- It is critical to use the optimal TCP send and receive socket buffer sizes for the link you are using.
  - Recommended size to fill the pipe
    - $2 \times \text{Bandwidth Delay Product (BDP)}$
  - Recommended size to leave some bandwidth for others
    - around 20% of $(2 \times \text{BDP}) = .4 \times \text{BDP}$
Setting TCP buffer sizes

- Default TCP buffer sizes are way too small for today’s high speed networks
  - Until recently, default TCP send/receive buffers were typically 64 KB
  - Tuned buffer to fill Argonne to LBL link: 8 MB
    - 125X bigger than the default buffer size
  - With default TCP buffers, you can only get a small % of the available bandwidth!
TCP tuning

- Many OS’s now include TCP autotuning
  - TCP send buffer starts at 64 KB
  - As the data transfer takes place, the buffer size is continuously re-adjusted up to max autotuning size
- Default autotuning maximum buffers on Linux 2.6: 256K to 1MB, depending on version
  - net.core.rmem_max = 16777216
  - net.core.wmem_max = 16777216
  # autotuning min, default, and max number of bytes to use
  - net.ipv4.tcp_rmem = 4096 87380 16777216
  - net.ipv4.tcp_wmem = 4096 65536 16777216
- http://fasterdata.es.net/TCP-tuning/
Parallel Streams

![Graph showing differences between One Stream and Two Streams](image-url)
Parallel TCP Streams

- Potentially unfair
- Reduces the severity of a congestion event
  - Only effects 1/p of the overall transfer
- Faster recovery
  - Smaller size to recover
- But they are necessary when you don’t have root access, and can’t convince the sysadmin to increase the max TCP buffers

Graph from Tom Dunigan, ORNL
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

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Size</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>(8 bits)</td>
<td>(64 bits)</td>
<td>(64 bits)</td>
</tr>
</tbody>
</table>
Demonstration
Performance

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

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Size</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>(8 bits)</td>
<td>(64 bits)</td>
<td>(64 bits)</td>
</tr>
</tbody>
</table>

GlobusWorld 2010
Firewall

- Control channel port is statically assigned
- Data channel ports dynamically assigned
- Mode E requires that the data sender make an active connection
Firewall

- Outgoing allowed at sender, incoming blocked at receiver

![Diagram showing firewall principles]
Firewall

- Outgoing allowed at sender, incoming blocked at receiver

Mode S

- Outgoing allowed at sender, incoming blocked at receiver
Firewall

- Outgoing allowed at sender, incoming blocked at receiver

Mode E
Firewall

- Open a port range on the receiver’s ends firewall and set GLOBUS_TCP_PORT_RANGE to that open range
- 50000-51000 is the recommended port range for data channel connections
- `export GLOBUS_TCP_PORT_RANGE = 50000,51000`
Firewall

- **Outgoing blocked at sender**
  - Can open a range of ports for outgoing connections to specific set of remote hosts (any remote port)
  - Use `GLOBUS_TCP_SOURCE_RANGE` to make the local end bound to a specified range
  - If outgoing connections can be opened up only for specific remote port range at specific remote hosts
    - firewall rule needs to modified on a case-by-case basis
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
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

10x Gbit/s

1 Gbit/s

DPI

CPI

SPI

DPI

SPI

DPI

DPI

DPI

DPI

DPI

DC

DC

DC

DC

10x Gbit/s
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
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
Lots of Small Files (LOSF) Problem
Concurrency

- Use concurrency optimization for transferring lots of small files
- What is a small file?
  - Depends on the network bandwidth and latency
  - Files of size $\leq 100$ MB
- Transfer multiple files concurrently
  - `globus-url-copy -cc`
GRAM
What is GRAM?

- GRAM is a Globus Toolkit component
  - For Grid *job management*
- GRAM is a unifying remote interface to Resource Managers
  - Yet preserves local site security/control
- GRAM provides stateful job control
  - Reliable create operation
  - Asynchronous monitoring and control
  - Remote credential management
  - Remote file staging and file cleanup
Grid Job Management Goals

Provide a service to securely:

- Create an environment for a job
- Stage files to/from environment
- Cause execution of job process(es)
  - Via various local resource managers
- Monitor execution
- Signal important state changes to client
Traditional Interaction

- Satisfies many use cases
- TACC’s Ranger (62976 cores!) is the Costco of HTC ;-), one stop shopping, why do we need more?
GRAM Benefit

- Add remote execution capability
  - Enable clients/devices to manage jobs without logging into the cluster

![Diagram showing Local Jobs, GRAM Service, LRM (e.g. PBS), Compute Nodes, Resource A, remote GRAM Jobs, and GRAM API connections]
GRAM Benefit

- Provides LRM abstraction

GRAM API

Local Jobs

GRAM Service

LRM (e.g. PBS)

Compute Nodes

Resource A

GRAM Service

LRM (e.g. SGE)

Compute Nodes

Resource B
GRAM Benefit

- Scalable job management
- Interoperability
Users/ Applications:
Science Gateways, Portals, CLI scripts,
App Specific Web Service, etc.

Local Resource Managers:
PBS, Condor, LSF, SGE, Fork
GRAM Client Interfaces

- **CLIs**
  - `globusrun`, `globus-job-run`, `globus-job-submit`, `globus-job-clean`, `globus-job-get-output`

- **C APIs**
  - `www.globus.org/api/c-globus-5.0.0`
  - Blocking and async functions for
    - submission, RSL manipulation, callbacks, cancelling, status

- **Java CoG JGlobus APIs**
  - `www.cogkit.org/release/4_1_4/api/jglobus/`
  - Classes: Gram, GramJob, GramAttributes
GRAM Authentication Test

- `globusrun -a -r never-1`

- **Resource Names**
  - `HOST:PORT/SERVICE:SUBJECT`

- `globusrun -a -r never-1.ci.uchicago.edu:2119/jobmanager:/DC=org/DC=doegrids/OU=Services/CN=host/never-1.ci.uchicago.edu`
globus-job-*

- bourne shell scripts that call globusrun
- Hide details of RSL
globus-job-run

- Blocking CLI to gram service
  - `globus-job-run never-1 /bin/hostname`
    - Basic job
  - `globus-job-run never-1 -np 5 /bin/sleep 10`
    - Multiple processes
  - `globus-job-run never-1 /bin/sleep 90`
    - Cancel execution by CTRL-C
  - `globus-job-run never-1 -env TEST=1 -env GRID=1 /usr/bin/env`
    - Augment job environment
globus-job-run cont..

* globus-job-run never-1 -env TEST=1 -env GRID=1 /usr/bin/env
  - Augment job environment

* globus-job-run –dumprrsl never-1 -env TEST=1 -env GRID=1 /usr/bin/env –u TEST
  - (executable="/usr/bin/env")
    (environment= ("TEST" "1") ("GRID" "1")
    (arguments= "-u" "TEST")
globus-job-submit, clean, get-output

- Non-blocking CLI to gram service
- `globus-job-submit never-1 /bin/hostname`
  - Returns job contact string
    - https://never-1.ci.uchicago.edu:
      37980/16073836513828969566/7364555675185249161/
  - Service will save the output, use get-output
- `globus-job-get-output <job contact>`
  - Returns – “never-1.ci.uchicago.edu”
- `globus-job-clean <job contact>`
  - Clean up after yourself!
globus-job-status

- globus-job-submit never-1 /bin/sleep 10
  - Get your remote job running
- globus-job-status <job contact>
  ACTIVE
- globus-job-status <job contact>
  DONE
  - Monitor status
- globus-job-clean <job contact>
  - Don’t forget to cleanup
globusrun

- C program
- Takes an Resource Specification Language (RSL) as an argument
- `globusrun -p "&(executable=/bin/ls)"
  - RSL Parsed Successfully...
- `globusrun -p "&(executable=/bin/ls) (howabout=this) (eventhough=(this doesnt) (make sense))"
  - RSL Parsed Successfully...
globusrun continued

- globusrun –j –r never-1 “&(executable=/bin/ls)”
  - Toolkit version: 4.3.0-HEAD Job Manager version: 10.5 (1256257907-0)
- globusrun -b -r never-1 "&(executable=/bin/sleep)(arguments=10)"
  - globus_gram_client_callback_allow successful
  - GRAM Job submission successful
  - https://never-1.ci.uchicago.edu:34159/16073843111170748796/736455675185248438/
  - GLOBUSGRAM_PROTOCOL_JOB_STATE_ACTIVE
globusrun continued

- `globusrun -status <job contact>`
  - Getting status of a job
- `globusrun -k <job contact>`
  - Cancelling a job
globusrun expired proxy

- Create a new proxy via grid-proxy-init
- Restarting a job will cause the JM to use the new proxy for all jobs
  - globusrun -r never-1 "&(restart=<job contact>)"
File staging and RSL substitution

- Run ls on never-1, but first stage the file from never-2

  `globusrun -s -r never-1 `&(rsl_substitution = (GRIDFTP_SERVER gsiftp://never-2.ci.uchicago.edu)) (executable=/bin/ls) (arguments=/tmp/staged_file) (file_stage_in = ($(GRIDFTP_SERVER)/home/tutorial1/junk /tmp/staged_file))'
File Stage In Shared

- Run ls on never-1, but first stage the file from never-2
  
  ```
globusrun -s -r never-1 \&(rsl_substitution = (GRIDFTP_SERVER gsiftp://never-2.ci.uchicago.edu)) (executable=/bin/ls) (arguments=/tmp/staged_file) (file_stage_in = ($GRIDFTP_SERVER)/home/tutorial1/junk /tmp/staged_file))'
  ```
File stage in shared

- Run `ls` on never-1, but first stage the file from never-2 into the gass cache from globusrun’s built in GASS server
  - `globusrun -s -r never-1 ‘&(executable=/bin/ls) (arguments = -l /tmp/staged_file_link1) (file_stage_in_shared = $(GLOBUSRUN_GASS_URL)/home/tutorial1/junk /tmp/staged_file_link1))’
  - `lrwxrwxrwx 1 tutorial1 tutorial1 122 Mar  2 01:22 /tmp/staged_file_link1 -> /home/tutorial1/.globus/.gass_cache/local/md5/73/6a9ff8a069d11515f240090bf77327/md5(cb/20eadb906d8fd93d30cd6385f6703a/data`
File stage out

- Run ls on never-1, then transfer the output using the gridftp server running on never-2
  
  globusrun -r never-1 '&(executable=/bin/ls)
  (stdout=$(HOME)/results.txt)
  (file_stage_out =
   $(HOME)/results.txt
   gsiftp://never-2.ci.uchicago.edu/home/tutorial1/
   never-1-ls-results.txt))'
file clean up

- Same thing only remove the results.txt file on never-1 after the contents have been staged out.
  - globusrun -r never-1 '
    (executable=/bin/ls)
    (stdout=$(HOME)/results.txt)
    (file_stage_out =
      $(HOME)/results.txt
      gsiftp://never-2.ci.uchicago.edu/home/tutorial1/
      never-1-ls-results.txt))
  (file_clean_up=$(HOME)/
   results.txt)'
GRAM5 Architecture

Job Submission

Client → Gatekeeper → Job Manager
1 process

Job Manager

RM adapter
RM adapter submit
throttled (default 5)

Resource Manager
User Job(s)

Job Monitoring

Job Manager
1 process

SEG
1 process

Resource Manager
User Job(s)

SEG log
RM log
Running the SEG

- By Default, jobs are monitored via polling
- But, SEG can be used and is more scalable and provides better performance

- For Fork, add "-seg-module fork" to $GLOBUS_LOCATION/etc/grid-services/jobmanager-fork

- Start the SEG
  - $GLOBUS_LOCATION/sbin/globus-job-manager-event-generator -scheduler fork -background -pidfile $GLOBUS_LOCATION/var/fork-pid
Examples of Production Scientific Grids

- APAC (Australia)
- China Grid
- DGrid (Germany)
- EGEE
- NAREGI (Japan)
- Open Science Grid
- Taiwan Grid
- TeraGrid
- ThaiGrid
- UK Nat’l Grid Service
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
- gt-user@globus.org