

GT 4.2.1 GridFTP: User's Guide

GT 4.2.1 GridFTP: User's Guide

Introduction

The GridFTP User's Guide provides general end user-oriented information.

Table of Contents

1. Managing Files on a Grid (GridFTP Quickstart)	1
1. Basic procedure for using GridFTP (globus-url-copy)	1
2. Accessing data from other data interfaces	2
3. Pipelining	4
4. GridFTP Where There Is FTP (GWTFTP)	4
5. Multicasting	4
2. GridFTP Client Tool	7
globus-url-copy	8
3. Graphical User Interface	18
4. Security Considerations	19
1. Security Considerations	19
5. Troubleshooting	21
1. Error Codes in GridFTP	22
2. Establish control channel connection	22
3. Try running globus-url-copy	23
4. If your server starts... ..	24
5. High latency for GridFTP server connections	24
6. Usage statistics collection by the Globus Alliance	25
1. GridFTP-specific usage statistics	25
Glossary	26
Index	28

List of Figures

2.1. Effect of Parallel Streams in GridFTP	16
--	----

List of Tables

2.1. URL formats	10
5.1. GridFTP Errors	22

Chapter 1. Managing Files on a Grid (GridFTP Quickstart)

1. Basic procedure for using GridFTP (globus-url-copy)

If you just want the "rules of thumb" on getting started (without all the details), the following options using `globus-url-copy` will normally give acceptable performance:

```
globus-url-copy -vb -tcp-bs 2097152 -p 4 source_url destination_url
```

where:

`-vb` specifies verbose mode and displays:

- number of bytes transferred,
- performance since the last update (currently every 5 seconds), and
- average performance for the whole transfer.

`-tcp-bs` specifies the size (in bytes) of the TCP buffer to be used by the underlying ftp data channels. This is critical to good performance over the WAN.

How do I pick a value?

`-p` Specifies the number of parallel data connections that should be used. This is one of the most commonly used options.

How do I pick a value?

The source/destination URLs will normally be one of the following:

- `file:///path/to/my/file` if you are accessing a file on a file system accessible by the host on which you are running your *client*.
- `gsiftp://hostname/path/to/remote/file` if you are accessing a file from a GridFTP *server*.

1.1. Putting files

One of the most basic tasks in GridFTP is to "put" files, i.e., moving a file from your file system to the server. So for example, if you want to move the file `/tmp/foo` from a file system accessible to the host on which you are running your client to a file name `/tmp/bar` on a host named `remote.machine.my.edu` running a GridFTP server, you would use this command:

```
globus-url-copy -vb -tcp-bs 2097152 -p 4 file:///tmp/foo gsiftp://remote.machine.my.edu/tmp
```



Note

In theory, `remote.machine.my.edu` could be the same host as the one on which you are running your client, but that is normally only done in testing situations.

1.2. Getting files

A `get`, i.e. moving a file from a server to your file system, would just reverse the source and destination URLs:

Tip

Remember `file:` always refers to your file system.

```
globus-url-copy -vb -tcp-bs 2097152 -p 4 gsiftp://remote.machine.my.edu/tmp/bar file:///tm
```

1.3. Third party transfers

Finally, if you want to move a file between two GridFTP servers (a *third party transfer*), both URLs would use `gsiftp:` as the protocol:

```
globus-url-copy -vb -tcp-bs 2097152 -p 4 gsiftp://other.machine.my.edu/tmp/foo gsiftp://re
```

1.4. For more information

If you want more information and details on URLs and the [command line options](#), the [Key Concepts](#) gives basic definitions and an overview of the GridFTP protocol as well as our implementation of it.

2. Accessing data from other data interfaces

2.1. Accessing data in a non-POSIX file data source that has a POSIX interface

If you want to access data in a non-POSIX file data source that has a POSIX interface, the standard server will do just fine. Just make sure it is really POSIX-like (out of order writes, contiguous byte writes, etc).

2.2. GridFTP and DSIs

The following information is helpful if you want to use GridFTP to access data in DSIs (such as HPSS and SRB), and non-POSIX data sources.

Architecturally, the Globus GridFTP *server* can be divided into 3 modules:

- the GridFTP protocol module,
- the (optional) data transform module, and
- the Data Storage Interface (DSI).

In the GT 4.2.1 implementation, the data transform module and the DSI have been merged, although we plan to have separate, chainable, data transform modules in the future.

Note

This architecture does NOT apply to the WU-FTPD implementation (GT3.2.1 and lower).

2.2.1. GridFTP Protocol Module

The GridFTP protocol module is the module that reads and writes to the network and implements the GridFTP protocol. This module should not need to be modified since to do so would make the server non-protocol compliant, and unable to communicate with other servers.

2.2.2. Data Transform Functionality

The data transform functionality is invoked by using the ERET (extended retrieve) and ESTO (extended store) commands. It is seldom used and bears careful consideration before it is implemented, but in the right circumstances can be very useful. In theory, any computation could be invoked this way, but it was primarily intended for cases where some simple pre-processing (such as a partial get or sub-sampling) can greatly reduce the network load. The disadvantage to this is that you remove any real option for planning, brokering, etc., and any significant computation could adversely affect the data transfer performance. Note that the *client* must also support the ESTO/ERET functionality as well.

2.2.3. Data Storage Interface (DSI) / Data Transform module

The Data Storage Interface (DSI) / Data Transform module knows how to read and write to the "local" storage system and can optionally transform the data. We put local in quotes because in a complicated storage system, the storage may not be directly attached, but for performance reasons, it should be relatively close (for instance on the same LAN).

The interface consists of functions to be implemented such as send (get), receive (put), command (simple commands that simply succeed or fail like mkdir), etc..

Once these functions have been implemented for a specific storage system, a client should not need to know or care what is actually providing the data. The server can either be configured specifically with a specific DSI, i.e., it knows how to interact with a single class of storage system, or one particularly useful function for the ESTO/ERET functionality mentioned above is to load and configure a DSI on the fly.

See [Appendix A, Developing DSIs for GridFTP](#) for more information.

2.3. Latest information about HPSS

Last Update: August 2005

Working with Los Alamos National Laboratory and the High Performance Storage System (HPSS) collaboration (<http://www.hpss-collaboration.org>), we have written a Data Storage Interface (DSI) for read/write access to HPSS. This DSI would allow an existing application that uses a GridFTP compliant client to utilize an HPSS data resources.

This DSI is currently in testing. Due to changes in the HPSS security mechanisms, it requires HPSS 6.2 or later, which is due to be released in Q4 2005. Distribution for the DSI has not been worked out yet, but it will *probably* be available from both Globus and the HPSS collaboration. While this code will be open source, it requires underlying HPSS libraries which are NOT open source (proprietary).



Note

This is a purely server side change, the client does not know what DSI is running, so only a site that is already running HPSS and wants to allow GridFTP access needs to worry about access to these proprietary libraries.

2.4. Latest information about SRB

Last Update: August 2005

Working with the SRB team at the San Diego Supercomputing Center, we have written a Data Storage Interface (DSI) for read/write access to data in the Storage Resource Broker (SRB) (<http://www.npaci.edu/DICE/SRB>). This DSI will enable GridFTP compliant clients to read and write data to an SRB server, similar in functionality to the `sput/sget` commands.

This DSI is currently in testing and is not yet publicly available, but will be available from both the SRB web site ([here](#)) and the Globus web site ([here](#)). It will also be included in the next stable release of the toolkit. We are working on performance tests, but early results indicate that for wide area network (WAN) transfers, the performance is comparable.

When might you want to use this functionality:

- You have existing tools that use GridFTP clients and you want to access data that is in SRB
- You have distributed data sets that have some of the data in SRB and some of the data available from GridFTP servers.

3. Pipelining

Pipelining allows the client to have many outstanding, unacknowledged transfer commands at once. Instead of being forced to wait for the "Finished response" message, the client is free to send transfer commands at any time.

Pipelining is enabled by using the `-pp` option:

```
globus-url-copy -pp
```

4. GridFTP Where There Is FTP (GWTFTP)

GridFTP Where There Is FTP (GWTFTP) is an intermediate program that acts as a proxy between existing FTP clients and GridFTP servers. Users can connect to GWFTP with their favorite standard FTP client, and GWFTP will then connect to a GridFTP server on the client's behalf. To clients, GWFTP looks much like an FTP proxy server. When wishing to contact a GridFTP server, FTP clients instead contact GWTFTP.

Clients tell GWFTP their ultimate destination via the FTP **USER <username>** command. Instead of entering their username, client users send the following:

```
USER <GWTFTP username>::<GridFTP server URL>
```

This command tells GWTFTP the GridFTP endpoint with which the client wants to communicate. For example:

```
USER bresnaha::gsiftp://wiggum.mcs.anl.gov:2811/
```



Note

Requires [GSIC security](#).

5. Multicasting

To transfer a single file to many destinations in a multicast/broadcast, use the new `-mc` option.



Note

To use this option, the admin must enable multicasting. Click [here](#) for more information.

```
globus-url-copy -vb -tcp-bs 2097152 -p 4 -mc filename source_url
```

The *filename* must contain a line-separated list of destination urls. For example:

```
gsiftp://localhost:5000/home/user/tst1
gsiftp://localhost:5000/home/user/tst3
gsiftp://localhost:5000/home/user/tst4
```

For more flexibility, you can also specify a single destination url on the command line in addition to the urls in the file. Examples are:

```
globus-url-copy -MC multicast.file gsiftp://localhost/home/user/src_file
```

or

```
globus-url-copy -MC multicast.file gsiftp://localhost/home/user/src_file gsiftp://localhost
```

5.1. Advanced multicasting options

Along with specifying the list of destination urls in a file, a set of options for each url can be specified. This is done by appending a `?` to the resource string in the url followed by semicolon-separated key value pairs. For example:

```
gsiftp://dst1.domain.com:5000/home/user/tst1?cc=1;tcpbs=10M;P=4
```

This indicates that the receiving host `dst1.domain.com` will use 4 parallel stream, a tcp buffer size of 10 MB, and will select 1 host when forwarding on data blocks. This url is specified in the `-mc` file as described above.

The following is a list of key=value options and their meanings:

- | | |
|--|---|
| <code>P=<i>integer</i></code> | The number of parallel streams this node will use when forwarding. |
| <code>cc=<i>integer</i></code> | The number of urls to which this node will forward data. |
| <code>tcpbs=<i>formatted integer</i></code> | The TCP buffer size this node will use when forwarding. |
| <code>urls=<i>string list</i></code> | The list of urls that must be children of this node when the spanning tree is complete. |
| <code>local_write=<i>boolean: y/n</i></code> | Determines if this data will be written to a local disk, or just forwarded on to the next hop. This is explained more in the Network Overlay section. |
| <code>subject=<i>string</i></code> | The DN name to expect from the servers this node is connecting to. |

5.2. Network Overlay

In addition to allowing multicast, this function also allows for creating user-defined network routes.

If the `local_write` option is set to `n`, then no data will be written to the local disk, the data will only be forwarded on.

If the `local_write` option is set to `n` and is used with the `cc=1` option, the data will be forwarded on to exactly 1 location.

This allows the user to create a network overlay of data hops using each GridFTP server as a router to the ultimate destination.

Chapter 2. GridFTP Client Tool

Name

globus-url-copy -- Multi-protocol data movement

globus-url-copy

Tool description

globus-url-copy is a scriptable command line tool that can do multi-protocol data movement. It supports `gsiftp://` (GridFTP), `ftp://`, `http://`, `https://`, and `file:///` protocol specifiers in the URL. For GridFTP, `globus-url-copy` supports all implemented functionality. Versions from GT 3.2 and later support file globbing and directory moves.

- [Before you begin](#)
- [Command syntax](#)
- [Command line options](#)
 - [Informational options](#)
 - [Utility options](#)
 - [Reliability options](#)
 - [Performance options](#)
 - [Security-related options](#)
- [Default usage](#)
- [MODES in GridFTP](#)
- [If you run a GridFTP server by hand](#)
- [How do I choose a value for the TCP buffer size \(`-tcp-bs`\) option?](#)
- [How do I choose a value for the parallelism \(`-p`\) option?](#)
- [Limitations](#)
- [Interactive clients for GridFTP](#)

Before you begin

Important

To use `gsiftp://` and `https://` protocols, you must have a [certificate](#) to use `globus-url-copy`. However, you may use `ftp://` or `http://` protocols without a certificate.

1. First, as with all things Grid, you *must* have a valid proxy certificate to run `globus-url-copy` in certain protocols (`gsiftp://` and `https://`, as noted above). If you are using `ftp://` or `http://` protocols, security is *not* mandatory and you may skip the rest of this table.

If you do not have a certificate, you must [obtain one](#).

If you are doing this for testing in your own environment, the [SimpleCA](#) provided with the Globus Toolkit should suffice.

If not, you must contact the Virtual Organization (VO) with which you are associated to find out whom to ask for a certificate.

One common source is the [DOE Science Grid CA](#)¹, although you must confirm whether or not the resources you wish to access will accept their certificates.

Instructions for proper installation of the certificate should be provided from the source of the certificate.

Please note when your certificates expire; they will need to be renewed or you may lose access to your resources.

- Now that you have a certificate, you must generate a temporary proxy. Do this by running:

```
grid-proxy-init
```

Further documentation for **grid-proxy-init** can be found [here](#).

- You are now ready to use **globus-url-copy**! See the following sections for syntax and command line options and other considerations.

Command syntax

The basic syntax for **globus-url-copy** is:

```
globus-url-copy [optional command line switches] Source_URL Destination_URL
```

where:

[optional command line switches]	See Command line options below for a list of available options.
<i>Source_URL</i>	Specifies the original URL of the file(s) to be copied. If this is a directory, all files within that directory will be copied.
<i>Destination_URL</i>	Specifies the URL where you want to copy the files. If you want to copy multiple files, this must be a directory.



Note

Any url specifying a directory must end with `/`.

URL prefixes

As of GT 3.2, we support the following URL prefixes:

- file://** (on a local machine only)
- ftp://**
- gsiftp://**
- http://**

¹ <http://www.doe grids.org/pages/cert-request.htm>

- `https://`

By default, **globus-url-copy** expects the same kind of host certificates that **globusrun** expects from gatekeepers.



Note

We do *not* provide an interactive client similar to the generic FTP client provided with Linux. See the [Interactive Clients](#) section below for information on an interactive client developed by NCSA/NMI/TeraGrid.

URL formats

URLs can be any valid URL as defined by RFC 1738 that have a [protocol](#) we support. In general, they have the following format: **`protocol://host:port/path`**.



Note

If the path ends with a trailing / (i.e. `/path/to/directory/`) it will be considered to be a directory and all files in that directory will be moved. If you want a recursive directory move, you need to add the `-r/-recurse` switch described below.

Table 2.1. URL formats

<code>gsiftp://myhost.mydomain.com:2812/data/foo.dat</code>	Fully specified.
<code>http://myhost.mydomain.com/mywebpage/default.html</code>	Port is not specified; therefore, GridFTP uses protocol default (in this case, 80).
<code>file:///foo.dat</code>	Host is not specified; therefore, GridFTP uses your local host. Port is not specified; therefore, GridFTP uses protocol default (in this case, 80).
<code>file:/foo.dat</code>	This is also valid but is not recommended because, while many servers (including ours) accept this format, it is <i>not</i> RFC conformant and is not recommended.



Important

For GridFTP (`gsiftp://`) and FTP (`ftp://`), it is legal to specify a user name and password in the the URL as follows:

```
gsiftp://myname:[mypassword]@myhost.mydomain.com/foo.dat
```

If you are using GSI security, then you may specify the username (but you may *not* include the `:` or the password) and the grid-mapfile will be searched to see if that is a valid account mapping for your distinguished name (DN). If it is found, the *server* will setuid to that account. If not, it will fail. It will NOT fail back to your default account.

If you are using anonymous FTP, the username *must* be one of the usernames listed as a valid anonymous name and the password can be anything.

If you are using password authentication, you must specify both your username and password. **THIS IS HIGHLY DISCOURAGED, AS YOU ARE SENDING YOUR PASSWORD IN THE CLEAR ON THE NETWORK.** This is worse than no security; it is a false illusion of security.

Command line options

Informational Options

-help -usage	Prints help.
-version	Prints the version of this program.
-versions	Prints the versions of all modules that this program uses.
-q -quiet	Suppresses all output for successful operation.
-vb -verbose	During the transfer, displays: <ul style="list-style-type: none">• number of bytes transferred,• performance since the last update (currently every 5 seconds), and• average performance for the whole transfer.
-dbg -debugftp	Debugs FTP connections and prints the entire control channel protocol exchange to STDERR. Very useful for debugging. Please provide this any time you are requesting assistance with a globus-url-copy problem.
-list <url>	This option will display a directory listing for the given url.

Utility Ease of Use Options

-a -ascii	Converts the file to/from ASCII format to/from local file format.
-b -binary	Does not apply any conversion to the files. This option is turned on by default.
-f <i>filename</i>	Reads a list of URL pairs from a filename. Each line should contain: <i>sourceURL destURL</i> Enclose URLs with spaces in double quotes ("). Blank lines and lines beginning with the hash sign (#) will be ignored.
-r -recurse	Copies files in subdirectories.
-notpt -no-third-party-transfers	Turns third-party transfers off (on by default). Site firewall and/or software configuration may prevent a connection between the two servers (a <i>third party transfer</i>). If this is the case, globus-url-copy will "relay" the data. It will do a GET from the source and a PUT to the destination. This obviously causes a performance penalty but will allow you to complete a transfer you otherwise could not do.

Reliability Options

-rst -restart	Restarts failed FTP operations.
-----------------	---------------------------------

- `-rst-retries <retries>` Specifies the maximum number of times to retry the operation before giving up on the transfer.
Use 0 for infinite.
The default value is 5.
- `-rst-interval <seconds>` Specifies the interval in seconds to wait after a failure before retrying the transfer.
Use 0 for an exponential backoff.
The default value is 0.
- `-rst-timeout <seconds>` Specifies the maximum time after a failure to keep retrying.
Use 0 for no timeout.
The default value is 0.

Performance Options

- `-tcp-bs <size> | -tcp-buffer-size <size>` Specifies the size (in bytes) of the TCP buffer to be used by the underlying ftp data channels.

 **Important**

This is critical to good performance over the WAN.

[How do I pick a value?](#)

- `-p <parallelism> | -parallel <parallelism>` Specifies the number of parallel data connections that should be used.

 **Note**

This is one of the most commonly used options.

[How do I pick a value?](#)

- `-bs <block size> | -block-size <block size>` Specifies the size (in bytes) of the buffer to be used by the underlying transfer methods.

- `-pp` **(New starting with GT 4.1.3)** Allows pipelining. GridFTP is a command response protocol. A client sends one command and then waits for a "Finished response" before sending another. Adding this overhead on a per-file basis for a large data set partitioned into many small files makes the performance suffer. Pipelining allows the client to have many outstanding, unacknowledged transfer commands at once. Instead of being forced to wait for the "Finished response" message, the client is free to send transfer commands at any time.

- `-mc filename source_url` **(New starting with GT 4.2.1)** Transfers a single file to many destinations. File-name is a line-separated list of destination urls. For more information on this option, click [here](#).

Multicasting must be [enabled for use](#) on the server side.

Security Related Options

`-s <subject> | -subject <subject>` Specifies a subject to match with both the source and destination servers.



Note

Used when the server does not have access to the host certificate (usually when you are running the server as a user). See [the section called “If you run a GridFTP server by hand...”](#).

`-ss <subject> | -source-subject <subject>` Specifies a subject to match with the source server.



Note

Used when the server does not have access to the host certificate (usually when you are running the server as a user). See [the section called “If you run a GridFTP server by hand...”](#).

`-ds <subject> | -dest-subject <subject>` Specifies a subject to match with the destination server.



Note

Used when the server does not have access to the host certificate (usually when you are running the server as a user). See [the section called “If you run a GridFTP server by hand...”](#).

`-nodcau | -no-data-channel-authentication` Turns off data channel authentication for FTP transfers (the default is to authenticate the data channel).



Warning

We *do not* recommend this option, as it is a security risk.

`-dcsafe | -data-channel-safe` Sets data channel protection mode to SAFE.

Otherwise known as *integrity* or *checksumming*.

Guarantees that the data channel has not been altered, though a malicious party may have observed the data.



Warning

Rarely used as there is a substantial performance penalty.

`-dcpriv | -data-channel-private` Sets data channel protection mode to PRIVATE.

The data channel is encrypted and checksummed.

Guarantees that the data channel has not been altered and, if observed, it won't be understandable.

 **Warning**

VERY rarely used due to the VERY substantial performance penalty.

Default globus-url-copy usage

A **globus-url-copy** invocation using the **gsift** protocol with no options (i.e., using all the defaults) will perform a transfer with the following characteristics:

- binary
- stream mode (which implies no parallelism)
- host default TCP buffer size
- encrypted and checksummed control channel
- an authenticated data channel

MODES in GridFTP

GridFTP (as well as normal FTP) defines multiple wire protocols, or MODES, for the data channel.

Most normal FTP servers only implement *stream mode* (MODE S), i.e. the bytes flow in order over a single TCP connection. GridFTP defaults to this mode so that it is compatible with normal FTP servers.

However, GridFTP has another MODE, called Extended Block Mode, or *MODE E*. This mode sends the data over the data channel in blocks. Each block consists of 8 bits of flags, a 64 bit integer indicating the offset from the start of the transfer, and a 64 bit integer indicating the length of the block in bytes, followed by a payload of length bytes. Because the offset and length are provided, out of order arrival is acceptable, i.e. the 10th block could arrive before the 9th because you know explicitly where it belongs. This allows us to use multiple TCP channels. If you use the `-p | -parallelism` option, **globus-url-copy** automatically puts the servers into MODE E.



Note

Putting `-p 1` is not the same as no `-p` at all. Both will use a single stream, but the default will use stream mode and `-p 1` will use MODE E.

If you run a GridFTP server by hand...

If you run a GridFTP server by hand, you will need to explicitly specify the subject name to expect. The subject option provides **globus-url-copy** with a way to validate the remote servers with which it is communicating. Not only must the server trust **globus-url-copy**, but **globus-url-copy** must trust that it is talking to the correct server. The validation is done by comparing host DNs or subjects.

If the GridFTP server in question is running under a host certificate then the client assumes a subject name based on the server's canonical DNS name. However, if it was started under a user certificate, as is the case when a server is started by hand, then the expected subject name must be explicitly stated. This is done with the `-ss`, `-sd`, and `-s` options.

`-ss` Sets the `sourceURL` subject.

`-ds` Sets the `destURL` subject.

- s If you use this option alone, it will set both urls to be the same. You can see an example of this usage under the [Troubleshooting](#) section.



Note

This is an *unusual* use of the client. Most times you need to specify both URLs.

How do I choose a value?

How do I choose a value for the TCP buffer size (-tcp-bs) option?

The value you should pick for the TCP buffer size (-tcp-bs) depends on how fast you want to go (your bandwidth) and how far you are moving the data (as measured by the Round Trip Time (RTT) or the time it takes a packet to get to the destination and back).

To calculate the value for -tcp-bs, use the following formula (this assumes that Mega means 1000² rather than 1024², which is typical for bandwidth):

$$-tcp-bs = \text{bandwidth in Megabits per second (Mbs)} * \text{RTT in milliseconds (ms)} * 1000 / 8$$

As an example, if you are using fast ethernet (100 Mbs) and the RTT was 50 ms it would be:

$$-tcp-bs = 100 * 50 * 1000 / 8 = 625,000 \text{ bytes.}$$

So, how do you come up with values for bandwidth and RTT? To determine RTT, use either ping or traceroute. They both list RTT values.



Note

You must be on one end of the transfer and ping the other end. This means that if you are doing a third party transfer you have to run the ping or traceroute between the two server hosts, not from your client.

The bandwidth is a little trickier. Any point in the network can be the bottleneck, so you either need to talk with your network engineers to find out what the bottleneck link is or just assume that your host is the bottleneck and use the speed of your network interface card (NIC).



Note

The value you pick for -tcp-bs limits the top speed you can achieve. You will NOT get bandwidth any higher than what you used in the calculation (assuming the RTT is actually what you specified; it varies a little with network conditions). So, if for some reason you want to limit the bandwidth you get, you can do that by judicious choice of -tcp-bs values.

So where does this formula come from? Because it uses the bandwidth and the RTT (also known as the latency or delay) it is called the *bandwidth delay product*. The very simple explanation is this: TCP is a reliable protocol. It must save a copy of everything it sends out over the network until the other end acknowledges that it has been received.

As a simple example, if I can put one byte per second onto the network, and it takes 10 seconds for that byte to get there, and 10 seconds for the acknowledgment to get back (RTT = 20 seconds), then I would need at least 20 bytes of storage. Then, hopefully, by the time I am ready to send byte 21, I have received an acknowledgement for byte 1 and I can free that space in my buffer. If you want a more detailed explanation, try the following links on TCP tuning:

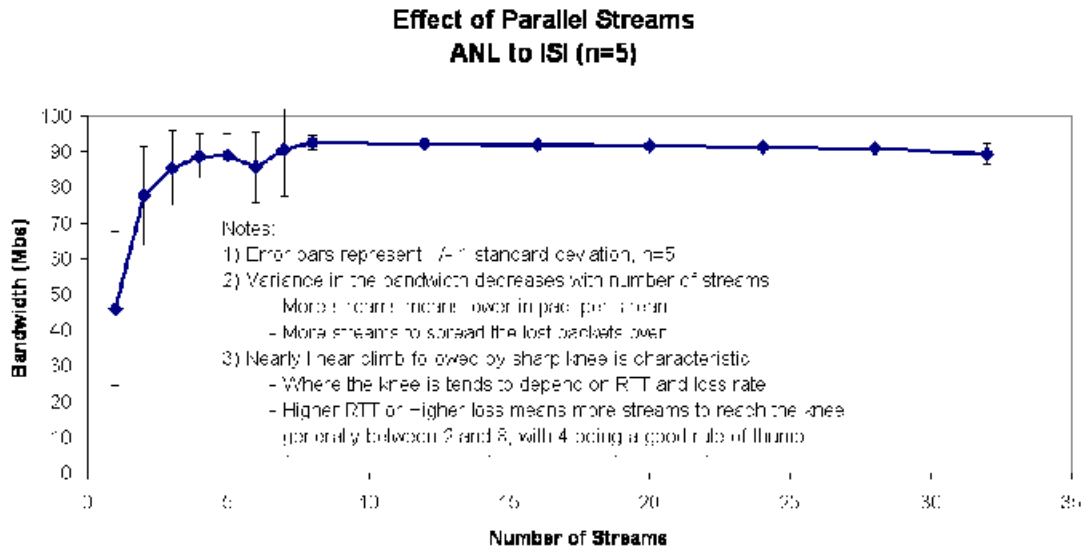
- http://www.psc.edu/networking/perf_tune.html

- <http://www.didc.lbl.gov/TCP-tuning/>
- <http://www.ncne.nlanr.net/research/tcp/>

How do I choose a value for the parallelism (-p) option?

For most instances, using 4 streams is a very good rule of thumb. Unfortunately, there is not a good formula for picking an exact answer. The shape of the graph shown here is very characteristic.

Figure 2.1. Effect of Parallel Streams in GridFTP



You get a strong, nearly linear, increase in bandwidth, then a sharp knee, after which additional streams have very little impact. Where this knee is depends on many things, but it is generally between 2 and 10 streams. Higher bandwidth, longer round trip times, and more congestion in the network (which you usually can only guess at based on how applications are behaving) will move the knee higher (more streams needed).

In practice, between 4 and 8 streams are usually sufficient. If things look really bad, try 16 and see how much difference that makes over 8. However, anything above 16, other than for academic interest, is basically wasting resources.

Limitations

There are no limitations for `globus-url-copy` in GT 4.2.1.

Interactive clients for GridFTP

The Globus Project does *not* provide an interactive client for GridFTP. Any normal FTP client will work with a GridFTP server, but it cannot take advantage of the advanced features of GridFTP. The interactive clients listed below take advantage of the advanced features of GridFTP.

There is no endorsement implied by their presence here. We make no assertion as to the quality or appropriateness of these tools, we simply provide this for your convenience. We will *not* answer questions, accept bugs, or in any way shape or form be responsible for these tools, although they should have mechanisms of their own for such things.

UberFTP was developed at the NCSA under the auspices of NMI and TeraGrid:

- NCSA Uerftp only download: <http://dims.ncsa.uiuc.edu/set/uberftp/download.html>
- UberFTP User's Guide: <http://dims.ncsa.uiuc.edu/set/uberftp/userdoc.html>

Chapter 3. Graphical User Interface

Globus does not provide any interactive client for GridFTP, either GUI or text based. However, NCSA, as part of their TeraGrid activity, produces a text based interactive client called UberFTP, which you may want to check out. See [the section called “Interactive clients for GridFTP”](#) for more information.

Chapter 4. Security Considerations

1. Security Considerations

1.1. Ways to configure your server

As discussed in [Section 2, “Types of configurations”](#), there are three ways to configure your GridFTP server: the default configuration (like any normal FTP server), separate (split) process configuration and striped configuration. The latter two provide greater levels of security as described [here](#).

1.2. New authentication option

There is a new authentication option available for GridFTP in GT 4.2.1:

- **SSH Authentication** Globus GridFTP now supports SSH based authentication for the control channel. In order for this to work:
 - Configure server to support SSH authentication,
 - Configure client(globus-url-copy) to support SSH authentication,
 - Use sshftp:// urls in globus-url-copy

For more information, see [Section 4, “SSHFTP \(GridFTP-over-SSH\)”](#).

1.3. Firewall requirements

If the GridFTP *server* is behind a firewall:

1. Contact your network administrator to open up port 2811 (for GridFTP control channel connection) and a range of ports (for GridFTP data channel connections) for the incoming connections. If the firewall blocks the outgoing connections, open up a range of ports for outgoing connections as well.
2. Set the environment variable GLOBUS_TCP_PORT_RANGE:

```
export GLOBUS_TCP_PORT_RANGE=min,max
```

where min,max specify the port range that you have opened for the incoming connections on the firewall. This restricts the listening ports of the GridFTP server to this range. Recommended range is 1000 (e.g., 50000-51000) but it really depends on how much use you expect.

3. If you have a firewall blocking the outgoing connections and you have opened a range of ports, set the environment variable GLOBUS_TCP_SOURCE_RANGE:

```
export GLOBUS_TCP_SOURCE_RANGE=min,max
```

where min,max specify the port range that you have opened for the outgoing connections on the firewall. This restricts the outbound ports of the GridFTP server to this range. Recommended range is twice the range used for GLOBUS_TCP_PORT_RANGE, because if parallel TCP streams are used for transfers, the listening port would remain the same for each connection but the connecting port would be different for each connection.



Note

If the server is behind NAT, the `--data-interface <real ip/hostname>` option needs to be used on the server.

If the GridFTP *client* is behind a firewall:

1. Contact your network administrator to open up a range of ports (for GridFTP data channel connections) for the incoming connections. If the firewall blocks the outgoing connections, open up a range of ports for outgoing connections as well.

2. Set the environment variable `GLOBUS_TCP_PORT_RANGE`

```
export GLOBUS_TCP_PORT_RANGE=min,max
```

where min,max specify the port range that you have opened for the incoming connections on the firewall. This restricts the listening ports of the GridFTP client to this range. Recommended range is 1000 (e.g., 50000-51000) but it really depends on how much use you expect.

3. If you have a firewall blocking the outgoing connections and you have opened a range of ports, set the environment variable `GLOBUS_TCP_SOURCE_RANGE`:

```
export GLOBUS_TCP_PORT_RANGE=min,max
```

where min,max specify the port range that you have opened for the outgoing connections on the firewall. This restricts the outbound ports of the GridFTP client to this range. Recommended range is twice the range used for `GLOBUS_TCP_PORT_RANGE`, because if parallel TCP streams are used for transfers, the listening port would remain the same for each connection but the connecting port would be different for each connection.

Additional information on Globus Toolkit Firewall Requirements is available [here](http://www.globus.org/toolkit/security/firewalls/)¹.

¹ <http://www.globus.org/toolkit/security/firewalls/>

Chapter 5. Troubleshooting

If you are having problems using the GridFTP [server](#), try the steps listed below. If you have an error, try checking the server logs if you have access to them. By default, the server logs to stderr, unless it is running from inetd, or its execution mode is detached, in which case logging is disabled by default.

The command line options `-d`, `-log-level`, `-L` and `-logdir` can affect where logs will be written, as can the configuration file options `log_single` and `log_unique`. See the [globus-gridftp-server\(1\)](#) for more information on these and other configuration options.

You should also be familiar with the [security considerations](#).

For a list of common errors in GT, see [Error Codes](#).

1. Error Codes in GridFTP

Table 5.1. GridFTP Errors

Error Code	Definition	Possible Solutions
<pre>globus_ftp_client: the server responded with an error 530 530-glo- bus_xio: Authentication Error 530-OpenSSL Error: s3_srvr.c:2525: in lib- rary: SSL routines, function SSL3_GET_CLI- ENT_CERTIFICATE: no cer- tificate returned 530- globus_gsi_callback_mod- ule: Could not verify credential 530-glo- bus_gsi_callback_module: Can't get the local trusted CA certificate: Untrusted self-signed certificate in chain with hash d1b603c3 530 End.</pre>	<p>This error message indicates that the GridFTP server doesn't trust the certificate authority (CA) that issued your certificate.</p>	<p>You need to ask the GridFTP server administrator to install your CA certificate chain in the GridFTP server's trusted certificates directory.</p>
<pre>globus_ftp_control: gss_init_sec_context failed OpenSSL Error: s3_clnt.c:951: in lib- rary: SSL routines, function SSL3_GET_SERV- ER_CERTIFICATE: certific- ate verify failed glo- bus_gsi_callback_module: Could not verify creden- tial globus_gsi_call- back_module: Can't get the local trusted CA certificate: Untrusted self-signed certificate in chain with hash d1b603c3</pre>	<p>This error message indicates that your local system doesn't trust the certificate authority (CA) that issued the certificate on the resource you are connecting to.</p>	<p>You need to ask the resource administrator which CA issued their certificate and install the CA certificate in the local trusted certificates directory.</p>

2. Establish control channel connection

Verify that you can establish a control channel connection and that the server has started successfully by telnetting to the port on which the server is running:

```
% telnet localhost 2811
      Trying 127.0.0.1...
      Connected to localhost.
```

```
Escape character is '^]'.  
220 GridFTP Server mldev.mcs.anl.gov 2.0 (gcc32dbg, 1113865414-1) ready.
```

If you see anything other than a 220 banner such as the one above, the server has not started correctly.

Verify that there are no configuration files being unexpectedly loaded from `/etc/grid-security/gridftp.conf` or `$GLOBUS_LOCATION/etc/gridftp.conf`. If those files exist, and you did not intend for them to be used, rename them to `.save`, or specify `-c none` on the command line and try again.

If you can log into the machine where the server is, try running the server from the command line with only the `-s` option:

```
$GLOBUS_LOCATION/sbin/globus-gridftp-server -s
```

The server will print the port it is listening on:

```
Server listening at gridftp.mcs.anl.gov:57764
```

Now try and telnet to that port. If you still do not get the banner listed above, something is preventing the socket connection. Check firewalls, `tcp-wrapper`, etc.

If you now get a correct banner, add `-p 2811` (you will have to disable `(x)inetd` on port 2811 if you are using them or you will get port already in use):

```
$GLOBUS_LOCATION/sbin/globus-gridftp-server -s -p 2811
```

Now telnet to port 2811. If this does not work, something is blocking port 2811. Check firewalls, `tcp-wrapper`, etc.

If this works correctly then re-enable your normal server, but remove all options but `-i`, `-s`, or `-S`.

Now telnet to port 2811. If this does not work, something is wrong with your service configuration. Check `/etc/services` and `(x)inetd` config, have `(x)inetd` restarted, etc.

If this works, begin adding options back one at a time, verifying that you can telnet to the server after each option is added. Continue this till you find the problem or get all the options you want.

At this point, you can establish a control connection. Now try running `globus-url-copy`.

3. Try running `globus-url-copy`

Once you've verified that you can establish a control connection, try to make a transfer using `globus-url-copy`.

If you are doing a *client/server* transfer (one of your URLs has `file:` in it) then try:

```
globus-url-copy -vb -dbg gsiftp://host.server.running.on/dev/zero file:///dev/null
```

This will run until you control-c the transfer. If that works, reverse the direction:

```
globus-url-copy -vb -dbg file:///dev/zero gsiftp://host.server.running.on/dev/null
```

Again, this will run until you control-c the transfer.

If you are doing a *third party transfer*, run this command:

```
globus-url-copy -vb -dbg gsiftp://host.server1.on/dev/zero gsiftp://host.server2.on/dev/nu
```

Again, this will run until you control-c the transfer.

If the above transfers work, try your transfer again. If it fails, you likely have some sort of file permissions problem, typo in a file name, etc.

4. If your server starts...

If the server has started correctly, and your problem is with a security failure or gridmap lookup failure, verify that you have security configured properly [here](#).

If the server is running and your client successfully authenticates but has a problem at some other time during the session, please ask for help on gt-user@globus.org¹. When you send mail or submit bugs, please always include as much of the following information as possible:

- Specs on all hosts involved (OS, processor, RAM, etc).
- `globus-url-copy -version`
- `globus-url-copy -versions`
- Output from the telnet test above.
- The actual command line you ran with `-dbg` added. Don't worry if the output gets long.
- Check that you are getting a FQDN and `/etc/hosts` that is sane.
- The server configuration and setup (`/etc/services` entries, `(x)inetd` configs, etc.).
- Any relevant lines from the server logs (not the entire log please).

5. High latency for GridFTP server connections

If you run GridFTP servers via Xinetd and notice high latency for connections and/or transfers, check if `/etc/xinetd.conf` or the `gsiftp` service configuration inside `/etc/xinetd.d` is set to log USERID as follows:

```
log_on_success += USERID
log_on_failure += USERID
```

Such a configuration tells Xinetd to log the remote user using the method defined in RFC 1413, which causes an ident client to attempt to query the machine that the connection is coming from before the service will run. Even when this succeeds, the response can't be trusted, and more often than not it is rejected or simply dropped (which results in the longest delays) by the remote firewall.

Latency can be reduced by making sure Xinetd does *not* log the USERID.

¹ http://dev.globus.org/wiki/Mailing_Lists

Chapter 6. Usage statistics collection by the Globus Alliance

1. GridFTP-specific usage statistics

The following GridFTP-specific usage statistics are sent in a UDP packet at the end of each transfer, in addition to the standard header information described in the [Usage Stats](#)¹ section.

- Start time of the transfer
- End time of the transfer
- Version string of the server
- TCP buffer size used for the transfer
- Block size used for the transfer
- Total number of bytes transferred
- Number of parallel streams used for the transfer
- Number of stripes used for the transfer
- Type of transfer (STOR, RETR, LIST)
- FTP response code -- Success or failure of the transfer



Note

The client (`globus-url-copy`) does NOT send any data. It is the *servers* that send the usage statistics.

We have made a concerted effort to collect only data that is not too intrusive or private and yet still provides us with information that will help improve and gauge the usage of the GridFTP server. Nevertheless, if you wish to disable this feature for GridFTP only, use the `-disable-usage-stats` option of `globus-gridftp-server`. Note that you can disable transmission of usage statistics globally for all C components by setting "GLOBUS_USAGE_OPTOUT=1" in your environment.

Also, please see our [policy statement](#)² on the collection of usage statistics.

¹ ../../Usage_Stats.html

² ../../Usage_Stats.html

Glossary

C

client A process that sends commands and receives responses. Note that in GridFTP, the client may or may not take part in the actual movement of data.

E

extended block mode (MODE E) MODE E is a critical GridFTP components because it allows for out of order reception of data. This in turn, means we can send the data down multiple paths and do not need to worry if one of the paths is slower than the others and the data arrives out of order. This enables parallelism and striping within GridFTP. In MODE E, a series of “blocks” are sent over the data channel. Each block consists of:

- an 8 bit flag field,
- a 64 bit field indicating the offset in the transfer,
- and a 64 bit field indicating the length of the payload,
- followed by length bytes of payload.

Note that since the offset and length are included in the block, out of order reception is possible, as long as the receiving side can handle it, either via something like a seek on a file, or via some application level buffering and ordering logic that will wait for the out of order blocks.

S

server A process that receives commands and sends responses to those commands. Since it is a server or service, and it receives commands, it must be listening on a port somewhere to receive the commands. Both FTP and GridFTP have IANA registered ports. For FTP it is port 21, for GridFTP it is port 2811. This is normally handled via inetd or xinetd on Unix variants. However, it is also possible to implement a daemon that listens on the specified port. This is described more fully in in the Architecture section of the GridFTP Developer's Guide.

stream mode (MODE S) The only mode normally implemented for FTP is MODE S. This is simply sending each byte, one after another over the socket in order, with no application level framing of any kind. This is the default and is what a standard FTP server will use. This is also the default for GridFTP.

T

third party transfers In the simplest terms, a third party transfer moves a file between two GridFTP servers.

The following is a more detailed, programmatic description.

In a third party transfer, there are three entities involved. The client, who will only orchestrate, but not actually take place in the data transfer, and two servers one of which will be sending data to the other. This scenario is common in Grid applications where you may wish to stage data from a data store somewhere to a super-computer you have reserved. The commands are quite similar to the client/server transfer. However, now the client must establish two control channels, one to each server. He will then choose one to listen, and send it the PASV command. When it responds with the IP/port it is listening on, the client will send that IP/port as part of the PORT command to the other server. This will cause the second server to connect to the first server, rather than the client. To initiate the actual movement of the data, the client then sends the RETR “filename” command to the server that will read from disk and write to the network (the “sending” server) and will send the STOR “filename” command to the other server which will read from the network and write to the disk (the “receiving” server).

See Also [client/server transfer](#).

Index

A

- accessing data
 - HPSS, 3
 - non-POSIX data source, 2
 - non-POSIX file data source that has a POSIX interface, 2
 - SRB, 3

C

- commandline tool
 - globus-url-copy, 8

E

- errors, 22

G

- globus-url-copy, 8
- GUI information for GridFTP, 18

I

- interactive clients
 - UberFTP, 16

M

- moving files
 - basic procedure, 1
 - between two GridFTP servers (a third party transfer), 2
 - existing FTP, 4
 - from a server to your file system, 2
 - from your file system to the server, 1
 - many outstanding transfers at once (pipelining), 4
 - single file to many destinations, 4
 - advanced options, 5
 - user-defined network routes, 5

S

- security considerations for GridFTP, 19

T

- troubleshooting for GridFTP, 21

U

- usage statistics for GridFTP, 25