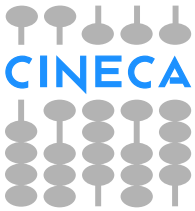

CINECA

Consorzio Interuniversitario

GridFTP 5 at CINECA: gsiftp://gftp-plx.cineca.it:2812

Author:
Giacomo Mariani
g.mariani@cineca.it

September 25, 2012

For any question feel free to send an email to: superc@cineca.it

Contents

1 Preface

GridFTP is a very effective tool for data trasnfer: it enhances the standard ftp service making it more reliable and faster. It was developed by the globus alliance as part of an open-source toolkit for data management. It is a client-server application: on Cineca HPC platforms a server is available. If you want to transfer files between one of these platforms and your local workstation or server, you need to install a "client" on it. The GridFTP extends the traditional FTP [1] . Its main improvements are:

- Parallel data transfer;

- Striped data transfer;
- GSI and Kerberos support;
- Third - Party control of data transfer;
- Automatic negotiation of TCP buffer/window sizes.

This means that the GridFTP offers an alternative to both the `classical-FTP` and the `OpenSSH-FTP`: gridFTP is faster, more reliable and more secure.

2 Quick-start

One instance of gridFTP is installed at CINECA. It leds you transfer files from and to all file-systems you have access to:

gsiftp://gftp-plx.cineca.it:2812 uses the certificates authentication. Lots of clients, including graphical ones, are available for both UNIX and Windows machines.

If you are in a UNIX environment, after creating a `globus` user and downloading the globus toolkit package, install the gridftp client:

```
root# mkdir /usr/local/globus-5.0.2/
root# chown globus:globus /usr/local/globus-5.0.2/
root# su - globus
globus$ cd /usr/local/globus-5.0.2/
globus$ tar -xjvf gt5.0.2-all-source-installer.tar.bz2
globus$ cd gt5.0.2-all-source-installer
globus$ export GLOBUS_LOCATION=/usr/local/globus-5.0.2
globus$ ./configure --prefix=$GLOBUS_LOCATION
globus$ make globus-data-management-client 2>&1 | tee build.log
globus$ make install
```

In order to use gridftp you have to load the necessary environment (depending on your shell):

```
user$ export GLOBUS_LOCATION=/usr/local/globus-5.0.2
user$ . $GLOBUS_LOCATION/etc/globus-user-env.sh

user$ setenv GLOBUS_LOCATION /usr/local/globus-5.0.2
user$ source $GLOBUS_LOCATION/etc/globus-user-env.csh
```

In order to use your certificate you have to initialise (and destroy) a proxy for authentication porpoises:

```
user$ grid-proxy-init
user$ globus-url-copy gsiftp://grid.cineca.it/remote_path/to/yourfile \
                file:///home/user/local_path/to/
user$ grid-proxy-destroy
```

3 Official-client installation from packages

The Globus Toolkit is packaged for most Linux distribution. You can install it following the guide at http://www.ige-project.eu/releases/downloads.

4 Official-client installation from source

The installation of the gridftp client is very similar to the installation of the whole Globus Toolkit 5.0.2 and proceed in the same way. The first suggested step is the creation of a user who will be the globus administrator [9] . For the sake of fantasy I called the one in the procedure that follows `globus`. Than a directory for the installation must be created and made owned by `globus`:

```
# mkdir /usr/local/globus-5.0.2/
# chown globus:globus /usr/local/globus-5.0.2/
# su - globus
```

The next step consists in downloading (in the `globus` home) the source code from the globus site: http://www.globus.org/toolkit/survey/index.php?download=gt5.0.2-all-source-installer.tar.bz2 . The source code should be de-archived:

```
$ tar -xjvf gt5.0.2-all-source-installer.tar.bz2
```

the environment set, the source configured, compiled and installed:

```
$ cd gt5.0.2-all-source-installer
$ export GLOBUS_LOCATION=/usr/local/globus-5.0.2
$ ./configure --prefix=$GLOBUS_LOCATION
$ make globus-data-management-client 2>&1 | tee build.log
$ make install
```

For more specific configuration and installation options you can take a look at output of the `configure` command by the following:

```
$ ./configure -help
```

It can be noted that the `-help` option is available for each GT5 command. These steps could be avoided if, for your operating system, a pre-compiled package exists [1] . The official documentation has a section dedicated to the installation of the whole Globus Toolkit 5.0.2 on various platform [9] . Please, for any question, refer to that.

5 Basic official-client usage

The core command for using GridFTP (i.e. for moving files) is the following:

```
globus-url-copy [options] <sourceURL> <destURL>
```

where a lot of options and protocols/authentication-methods (intrinsically expressed in the URL) are available. There is a second command, which, being in a developing stage, is not powerful: `globus-url-sync`. It is a command line tool which provides a list of files to be transfered, in order to synchronize two directories. It currently supports `gsiftp://` and

[1] http://www.globus.org/toolkit/downloads/

`sshftp://` protocol specifiers in the URL [2] . Indeed, the GridFTP server offers three main methods of authentication:

1. certificates;
2. SSH;
3. anonymous.

The instances at CINECA only support the first two, for which a short description follow. The ones at DEISA only the first one.

5.1 SSHFTP (GridFTP-over-SSH)

The username-password authentication method follows the openssh-ftp one. It means that, after a secure connection is established (*SSL*), the credentials, i.e. the username and the password, are sent to the server. This approach is obviously quite slow. Moreover the authentication is required at each transaction and the data channel is not authenticated [6] .

A big improvement is achievable using the *RSA Key* SSH authentication. This approach, for which an extended documentation is available on the interned, requires the user to generate a couple of keys (similar to the ones used in the certificates). The two keys are generated with the following command:

```
$ ssh-keygen -t rsa
```

The private one, called *id_rsa*, must be readable and writable by the user only:

```
user@client:$chmod 400 $HOME/.ssh/id_rsa
```

It is than necessary to make a copy of the public one, *id_rsa.pub*, in the `.ssh` directory in the user home-directory on the server and to call it `authorized_keys`, with the right permissions:

```
user@server:$chmod 600 $HOME/.ssh/authorized_keys
```

When these two files are available the authentication takes place automatically making all transfer operations easier and more safe. In order to use the username-password approach the *URL* of the remote hosts have to start with the prefix `sshftp` and the port should be the 22 (that is the default one and could be omitted) as shown in section 5.3 .

[2] The program globus-url-sync compares two endpoints, using GridFTP, and prints a list of GSI file transfers that should be performed using globus-url-copy. The current implementation of globus-url-sync supports very basic features for directory synchronization. It includes comparators for existence checks, file size checks, modification time-stamp checks, but not checksum comparison.

5.2 Certificate-based authentication

The Globus Toolkit v5.0.2 (hereafter GT5), of which GridFTP is a package, uses, for its security purpose, the *GSI* (Grid Security Infrastructure) package [2, 7] . This is based on X.509 certificates. A certificate is an electronic document released by a *CA* (Certificate Authority) that is, as the word states, a publish-recognised entity. The list of all the CA that GSI recognizes is usually contained in some directory, for example: `/etc/grid-security/certificates/` [3] . Each certificate contains the *public key* of a user (a host or a server are *users* too). This key is signed with the *private key* of the CA and, with its *public key*, it is possible to verify its authenticity [4] . The user can now demonstrate its identity using its *private key* [5] . The key is usually encrypted. To avoid the password insertion required by the decryption each time the authentication take place a proxy (*MyProxy* [10]) is used.

As stated, the GridFTP combined with GSI offers a certificate-based authentication. This means that each user must have two files: a certificate signed by a CA [6] (`usercert.pem`) and a private key (`userkey.pem`). These two files are usually placed in `$HOME/.globus/` [7] . The permissions on this two files required by GSI are very specific: they must be owned by the user, `usercert.pem` should be not more than 644 and `userkey.pem` not more than 400. As stated the `userkey.pem` is usually encrypted and would require a password each time is used. To avoid this problem, and the security risks that would arise, a temporary proxy is used. It is created once in a while (usually it lasts 12 hours) typing the following command:

```
$ grid-proxy-init
```

and inserting the password. It is temporary so its impairment wouldn't be that dangerous.

A third information is needed: the list of the recognized CA (see note 3 and 6) . If this information is missing in the client installation you have to provide it filling `$HOME/.globus/certificates` (see Appendix A) .

[3] The location of the trusted certificates is looked, by default, in the following order:

1. value of the `X509_CERT_DIR` environment variable;
2. `$HOME/.globus/certificates`;
3. `/etc/grid-security/certificates`;
4. `$GLOBUS_LOCATION/share/certificates`.

[4] If you have to extract your signed public key (say usercert.pem) and private key (say userkey.pem) from a PKCS12 certivicate (say certificate.p12) you can proceed as follow:

```
$openssl pkcs12 -nokeys -in certificate.p12 -out usercert.pemi
$openssl pkcs12 -in certificate.p12 -out userkey.pem
$vim userkey.pem
```

where only the key-section is left.

[5] For informations about public and private keys you can see, for example, http://en.wikipedia.org/wiki/Public-key_cryptography.

[6] The CA recognized by CINECA are listed in Appendix A.

[7] A server needs two files too (the prefix `user` is substituted with `host`) in order to identify itself to the clients. Those files are usually placed in `/etc/grid-security/`.

5.3 Main options

The client-command is *globus-url-copy*. It needs a couple of setting to work. First you have to export the right variables:

```
$ export GLOBUS_LOCATION=/devel/products/globus/globus-5.0.2
$ . $GLOBUS_LOCATION/etc/globus-user-env.sh
```

if you use a *bash*-like shell or

```
$ setenv GLOBUS_LOCATION /devel/products/globus/globus-5.0.2
$ source $GLOBUS_LOCATION/etc/globus-user-env.csh
```

if you use a *csh*-like shell. Then you have to set-up your MyProxy:

```
$ grid-proxy-init
```

You should see something like this:

```
Your identity: /C=IT/O=INFN/OU=Personal Certificate/L=CINECA-SAP/CN=Giacomo Mariani
Enter GRID pass phrase for this identity:
Creating proxy ..................................... Done
Your proxy is valid until: Thu Sep 30 22:25:55 2010
```

It's worth nothing that your proxy has a due date. You can vary it with the `-valid` option, as described in the command help:

```
$ grid-proxy-init -help

Syntax: grid-proxy-init [-help][-pwstdin][-limited][-valid H:M] ...
[...]
-valid <h:m>              Proxy is valid for h hours and m
                          minutes (default:12:00)
[...]
```

You can now transfer files with the *globus-url-copy* command. The basic syntax is the following:

```
$ globus-url-copy -help

globus-url-copy [options] <sourceURL> <destURL>
globus-url-copy [options] -f <filename>

<sourceURL> may contain wildcard characters * ? and [ ] character ranges
in the filename only.
Any url specifying a directory must end with a forward slash '/'

If <sourceURL> is a directory, all files within that directory will
be copied.
<destURL> must be a directory if multiple files are being copied.
[...]
```

Here the *fileURL* if of the form `server-path`, where `server` may be `file://` for a local file or `protocol://[user@]host[:port]` for a remote one. `protocol` can be `sshftp` for username-password authentication or `gsiftp` for certificates authentication. It's worth to note that, for the gridFTP at CINECA the server should be **gsiftp://gftp-plx.cineca.it[:2812]** supporting certificates authentication. For example we can transfer the file `pig.cow` from our PC to a server called `slaughterhouse` with certificates authentication:

```
$ globus-url-copy file:///home/farmer/pig.cow \
   gsiftp://slaughter@slaughterhouse:2811/home/of/slaughter/
```

where the port 2811 is the standard one. We can then transfer the file from `slaughterhouse` to another server, called `butchery`, with username-password authentication:

```
$ globus-url-copy \
   gsiftp://slaughter@slaughterhouse:2811/home/of/slaughter/pig.cow \
   sshftp://butcher@butchery/home/of/butcher/
```

There are a few options that worths to know from the beginning:

-version prints the globus-url-copy version;

-vb during the transfer, displays: (1) number of bytes transferred (2) performance since the last update (every 5 seconds) (3) average performance for the whole transfer;

-tcp-bs<size> specifies the size (in bytes) of the TCP buffer to be used by the underlying GridFTP data channels;

-p<number> specifies the number of parallel streams to be used in the GridFTP transfer;

-stripe use this parameter to initiate a *striped* GridFTP transfer that uses more than one node at the source and destination [8] .

And, at the end of your session, if you want to be sure:

```
$ grid-proxy-destroy
```

6 Official-client tuning

The main parameters available for optimizing the performance of your transfer operations are those connected to striping, streams and buffer features.

[8] As multiple nodes contribute to the transfer, each using its own network interface, a larger amount of the network bandwidth can be consumed than with a single system. Thus, at least for big (i.e. $> 100MB$) files, striping can considerably improve performance.

6.1 Streams and TCP buffer

6.1.1 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 1000^2 rather than 1024^2, which is typical for bandwidth):

```
-tcp-bs = bandwidth in Megabits per second (Mbs) * 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 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 [9] .
The bandwidth is a little trickier. Any point in the network can be the bottleneck, so you either need to talk with your network administrators 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) [10] . 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 [11] .

6.1.2 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. You get a strong increase in bandwidth, then a sharp knee, after which

[9] 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.

[10] 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.

[11] 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/

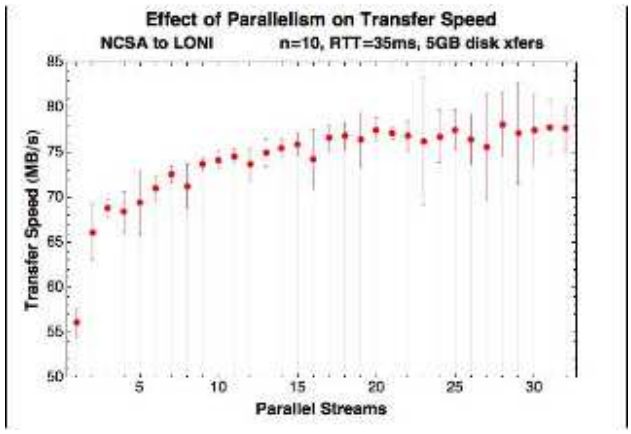

Figure 1: Effect of Parallel Streams in GridFTP[8].

additional streams have very little impact. Where this knee is depends on many things, but in general lies 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.

6.2 Test performances

CINECA provides a script
`https://hpc.cineca.it/sites/default/files/TestPerformances.tar` which is able to test the two parameters discussed above and produce a graphical representation of the network performances as their function. The script uses *gnuplot* to plot his results, *seq2* to range the values. Both of them should be available in your path to execute the script. Moreover you should be able to access to the CINECA server with one between GSI and SSH+RSA keys authentication method. In Fig. 2 the results of the script from within CINECA network is shown.

6.3 Strip

The striping functionality enables one to use a set of computers at both ends of a network to transfer data. At both the source and destination ends the computers need to have a shared file system in order to use multiple network paths.
This feature is especially useful in configurations where individual nodes at the source and destination clusters have significantly less network capacity when compared to the network capacity available between the clusters. An example would be clusters with the individual nodes connected by 1 Gbit/s Ethernet connections to a switch that is itself connected to the external network at 10 Gbit/s or faster. Usually you won't connect to CINECA through such a powerful network and so this option will result unavailable for you.

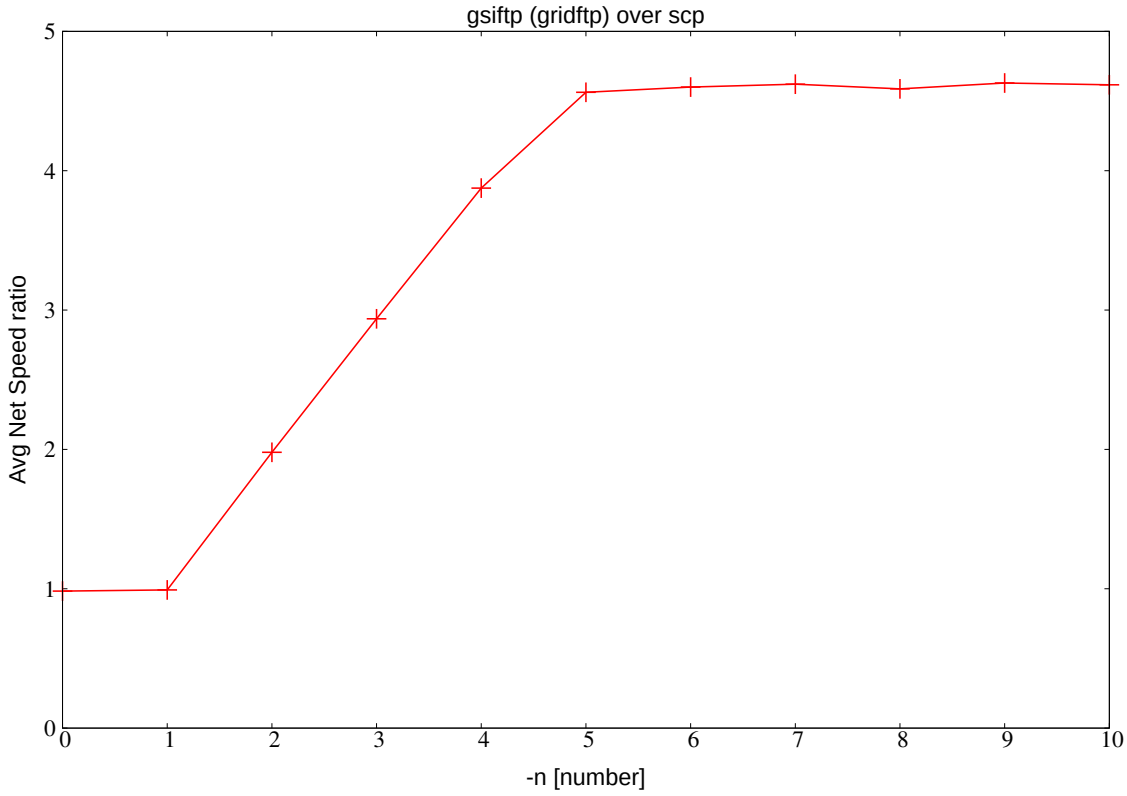

Figure 2: Effect of Parallel Streams variation in GridFTP performances between *INAF Catania Astrophysical Observatory* and CINECA compared with scp.

7 Troubleshooting

The problems that more often arise follow. First of all consider that `globus-url-copy` is very sensible to syntax and wild characters, so be carefull and use explicit paths if possible. If is not the case...

Error Code	CodeDefinition	Possible Solutions
globus_ftp_client: the server responded with an error 530 530-globus_xio: Authentication Error 530-OpenSSL Error: s3_srvr.c:2525: in library: SSL routines, function SSL3_GET_CLIENT_CERTIFICATE: no certificate returned 530-globus_gsi_callback_module: Could not verify credential 530-globus_gsi_callback_module: Can't get the local trusted CA certificate: Untrusted self-signed certificate in chain with hash d1b603c3 530 End.	This error message indicates that the GridFTP server doesn't trust the certificate authority (CA) that issued your certificate.	You need to ask the GridFTP server administrator to install your CA certificate chain in the GridFTP server's trusted certificates directory.

globus_ftp_control: gss_init_sec_context failed OpenSSL Error: s3_clnt.c:951: in library: SSL routines, function SSL3_GET_SERVER_CERTIFICATE: certificate verify failed globus_gsi_callback_module: Could not verify credential globus_gsi_callback_module: Can't get the local trusted CA certificate: Untrusted self-signed certificate in chain with hash d1b603c3	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.	You need to ask the resource administrator which CA issued their certificate and install the CA certificate in the local trusted certificates directory.
530-globus_xio: Authentication Error 530-globus_gsi_callback_module: Could not verify credential 530-globus_gsi_callback_module: Could not verify credential 530-globus_gsi_callback_module: Invalid CRL: The available CRL has expired 530 End.	This error message indicates one of the following: Certificate Revocation List (CRL) for the source or destination server CA at the client has expired or CRL for client CA has expired at source or destination server or CRL for source (destination) server CA has expired at destination (source) server. CRL is a file CA_hash.r0 in /etc/grid-security/certificates or $USER_HOME/.globus/certificates or $X509_CERT_DIR	The tool available at http://dist.eugridpma.info/distribution/util/fetch-crl/ can be run in a crontab to keep the CRLs up to date.

7.0.1 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 grid.cineca.it 2811
Trying 130.186.16.166...
Connected to grid.cineca.it.
Escape character is '^]'.
220-###################################################
220-####    Welcome on SP6 GridFTP service!        ####
220-####                                           ####
220-####    The usage of this service is           ####
220-####    for cineca users.                      ####
220-####                                           ####
220-####    Any misuse will be persecuted by law   ####
220-####    Art. 615-ter C.P.                      ####
220-####                                           ####
```

```
220-####    Globus version 5.0.2                    ####
220-####    GridFTP Server 3.23                     ####
220-########################################################
220-
220 End.
```

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:

```
$ globus-gridftp-server -s
  Server listening at fec02:57151
```

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.

7.1 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/null
```

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.

7.2 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.org. 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).

8 Unsupported clients

A lot of gridFTP clients are provided by various people and organizations, thanks to the openness of GT [3] . What follow are short comments on the three more *user-friendly* ones. It's worth to note that they support only GSI authentication.

8.1 UberFTP installation and usage[11]

The Uber client is the most close to the definition of "official client". It is a CL (command line) one that improves the basic features of globus-url-copy and globus-url-sync in a way that recalls the standard ftp working and syntax. You can read about it at http://dims.ncsa.illinois.edu/set/uberftp/install.html. If you are not satisfied with globus-url-copy this is probably the best choice you can do.

8.2 GridFTP GUI (Java) installation and usage[5]

This client (see [5]) depends on the Java javawebstart [12] for its working. Its completely platform independent and has a very intuitive GUI. More informations can be found at http://www-unix.globus.org/cog/demo/.

8.3 GridFTP Client (Java) installation and usage[4]

This client (see [4]) depends on the Java Development Kit v1.5 or greater to work. Nonetheless it is not platform independent and you have to choose between a GNU/linux version, a Mac version and a Windows one. Again the GUI is a very intuitive one.

[12] http://java.sun.com/products/javawebstart/

A CA recognized at CINECA

The Globus tools will trust certificates issued by a CA if (and only if) they can find information about the CA in the trusted certificates directory. The updated list of all the CA recognized by GT5 at CINECA is visible and dowloadable at http://winnetou.sara.nl/deisa/certs/. Each CA's certificate in the certificates directory is usually made of two files:

cert_hash.0 the trusted CA Certificate;

cert_hash.signing_policy a configuration file defining the distinguished names of certificates signed by the CA.

The cert_hash that appears in the file names above is the hash of the CA certificate, which can be found by running the command:

```
$ GLOBUS_LOCATION/bin/openssl x509 -hash -noout < ca_certificate
```

This certificates are used to verify signature on certificates issued by the CA. The certificates are usually printed in the ldif format, for example:

INFN C=IT, O=INFN, CN=INFN CA;

TERENA C=NL, O=TERENA, CN=TERENA eScience Personal CA;

CERN DC=ch, DC=cern, CN=CERN Trusted Certification Authority.

References

[1] W. Allcock et al. "GridFTP: Protocol extensions to FTP for the Grid". In: *GWD-R (Recommendation)* (2001), p. 3.

[2] Sharon Brunett et al. "Application Experiences with the Globus Toolkit". In: 1998, pp. 81–89.

[3] *Globus Toolkit 5.0.2 Developer's Guide*. URL: http://www.globus.org/toolkit/docs/5.0/5.0.

[4] *GridFTP Client - Keeps An Eye On Your Large Data Transfers*. URL: http://bi.offis.de/gridi

[5] *GridFTP GUI*. URL: http://www-unix.globus.org/cog/demo/ogce/ftp.jnlp.

[6] *GT 5.0.2 GridFTP : System Administrator's Guide*. URL: http://www.globus.org/toolkit/doc

[7] *GT 5.0.2: GSI C Admin Guide*. URL: http://www.globus.org/toolkit/docs/5.0/5.0.2/secur

[8] *GT 5.0.2 User's Guide*. URL: http://www.globus.org/toolkit/docs/5.0/5.0.2/user/#gtuser

[9] *Installing GT 5.0.2*. URL: http://www.globus.org/toolkit/docs/5.0/5.0.2/admin/install/

[10] *MyProxy Credential Management Service*. URL: http://grid.ncsa.illinois.edu/myproxy/.

[11] *UberFTP Client Homepage*. URL: http://dims.ncsa.illinois.edu/set/uberftp/index.html.