

Tutorial sull'uso della grid
CNAF, 4-5 aprile 2002

The DataGrid Testbed is composed by some machine types called *Grid Elements*:

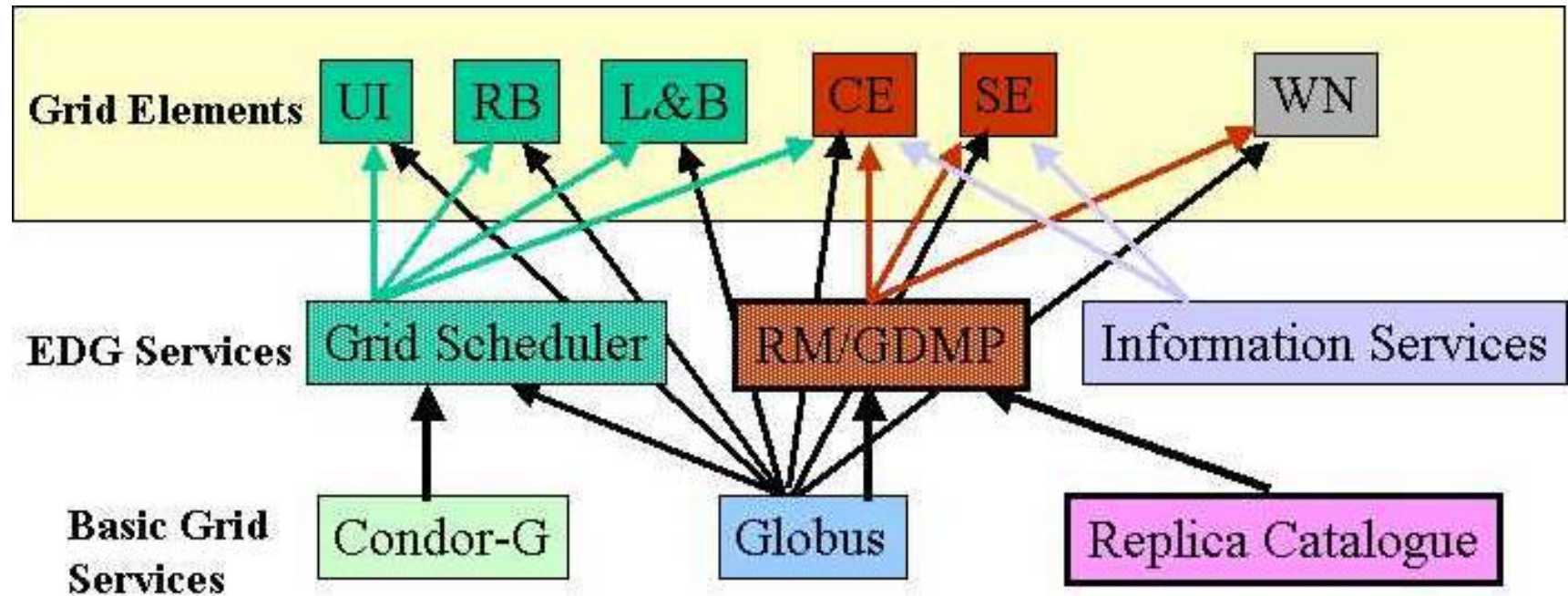
- **User Interface (UI)**
is the node where users submit jobs to the grid.
- **Computing Element (CE)**
is the gateway to the grid farm nodes. The Resource Broker dispatches a job to the CE which then sends it to a farm's specific WN by means of a local scheduler.
DataGrid supported local schedulers are PBS and LSF.
- **Worker Node (WN)**
one node of a computing farm.
- **Storage Element (SE)**
is the node that provides access to data and services for storing, locating and replicating data. It publishes information about data availability.

A farm is composed by a Computing Element and a variable number of Worker Nodes.

Grid Elements for '*special services*':

- **Resource Broker (RB)**
the node responsible to match job requirements and grid resources and to schedule the job submission, via JSS, to the “best matching” Computing Element.
- **Information Index (II)**
the access point to the Information System (database) containing the information about the grid elements
- **Logging & Bookkeeping Element (LB)**
the node responsible for keeping track of the history of the job execution; it provides also monitoring services.
- **Replica Catalog (RC)**
the node containing the LDAP database used by the Replica Manager (RM); it contains information about data and their localization
- **Virtual Organization server (VO)**
the node containing the LDAP database where the certificate's subject of the users are stored

Grid elements and services in the European DataGrid (EDG)



Software repository

The software is stored in a **CVS (Concurrent Versions System)** hosted at CC-IN2P3. It contains source code and binaries (rpm) of the EDG releases (Globus, DataGrid middleware and external software).

It could be accessed through a Web browser as well as through cvs and wget commands.

Naming convention

Packages are created from module tagged releases. The name of created packages should be of the following form:

Package	name
Generic tarball sources	<code>edg-myModule-x.y.z.src.tar.gz</code>
Linux i386 tarball binaries	<code>edg-myModule-x.y.z.i386.tar.gz</code>
Sun-Sparc tarball binaries	<code>edg-myModule-x.y.z.sol7-sparc.tar.gz</code>
Generic source RPM	<code>edg-myModule-x.y.z-n.src.rpm</code>
Linux i386 RPM binaries	<code>edg-myModule-x.y.z-n.i386.rpm</code>
Sun-Sparc RPM binaries	<code>edg-myModule-x.y.z-n.sol7-sparc.rpm</code>

The `edg-` prefix is there to recall the inclusion of this module in the DataGrid software [\[3\]](#).

EDG rpms follow the standard Linux Standard Base (LSB):

anypath_relocable/	What's in it
bin	Commands binaries
etc	Configuration file
include	Header files of libraries
lib	Libraries
share	Architecture independant files
share/doc	Documentation
sbin	Configuration binaries
tmp	Might be a symlink
var	Variable data (might be a symlink)

The EDG_ROOT shell variable (top level of this hierarchy) should point to /opt/edg

In the CVS repository the releases are identified (and “blessed”) by means of a numerical tag.

E.g.

the tag

`v1_0_2`

identifies EDG release 1.0.2.

In this way it’s really easy to retrieve a specific software release.

The command:

```
cv$ co -r v1_1_0
```

retrieves the exact source code that was used to generate the 1.1.0 release.

CVS Root: [European Data Grid]

File	
Auth/	
Workload/	← WP1
biomed/	
edg-docs/	
edg-example/	
edg-release/	
edg-se-query/	
edg-storage-element/	
edg-tests/	
edg-utils/	
fabric_mgt/	← WP4
gdmp/	← WP2
gma/	← WP3
grm/	← WP3
ldapservices/	← WP3
network/	
spitfire/	← WP2

DataGRID CVSweb maintained by <sysunix@cc.in2p3.fr>

User's Authentication

Each grid user must be authenticated by means of an X509 certificate issued by a Certification Authority (CA).

Almost all the DataGrid partners have their own CA:

CA	CA Certificate	CA CRL
CERN	c35c1972.0	c35c1972.r0
Czech Republic - CESNET	ed99a497.0	ed99a497.r0
France - CNRS	cf4ba8c8.0	cf4ba8c8.r0
France - CNRS-Projets	34a509c3.0	34a509c3.r0
France - CNRS Datagrid-fr	6b4ddd18.0	6b4ddd18.r0
Ireland - Grid-Ireland	1e43b9cc.0	1e43b9cc.r0
Italy - INFN	df312a4e.0 <i>Php script</i>	df312a4e.r0 <i>DER format</i>
Netherlands - NIKHEF	16da7552.0	16da7552.r0
Nordic countries - NorduGrid	1f0e8352.0	1f0e8352.r0
Portugal - LIP	41380387.0	41380387.r0
Russia - Russian DataGRID	d64ccb53.0	d64ccb53.r0
Spanish - DATAGRID-ES	90e2484f.0	90e2484f.r0
United Kingdom - GridPP	0ed6468a.0	0ed6468a.r0

User's Authorization

To be able to use the Grid resources users must be authorized.

The authorization process can be split in two steps:

- ✍ sign-on on the User Interface through the `grid-proxy-init` command. Through the certificate-proxy mechanism a single sign-on is enough
- ✍ authorization to use the resource on the Resource Broker, Computing Elements and Storage Elements

To be authenticated through the Globus Security Infrastructure users have to place their certificate in PEM format ([usercert.pem](#) and [userkey.pem](#)) in the `.globus` subdirectory of their home area.

- `usercert.pem` contains the user's public key (must be world readable)
- `userkey.pem` contains the user's private key (must be readable only by the user)

User's Authorization

Note that most browsers use certificates with PKCS12 (.p12 extension) format.

It's possible to convert certificates from pkcs12 to pem formats (and vice versa) using the appropriate openssl commands.

For instance

The commands

```
openssl pkcs12 -nocerts -in certificate.p12 -out userkey.pem
```

extracts the user's private key in pem format
while

```
openssl pkcs12 -clcerts -nokey -in certificate.p12 -out usercert.pem
```

extracts the user's public key.

N.B.

The certificate import password is required;
a new password (PEM pass phrase) is generated

User's Authorization

The second step (the real authorization) is done by means of a special authorization file ([gridmapfile](#)) on the grid elements (the Globus Security Infrastructure is used).

For each grid user the gridmapfile maps the certificate's subject to a local account (worker account) that will be used by the operating system to run the job.

On the various grid elements the mapping is different.

User's Authorization

On the Resource Broker the certificate's subject of each grid user is mapped to the local account used by the RB (e.g. wp1)

In order to simplify the tracking of who-did-what with respect to the job execution (these data should be passed to the Accounting System) it was decided that on the Computing and Storage Elements and the certificate's subject of each user must be mapped to a different local account.

The local accounts can be static or dynamic (created on-the-fly or pools) accounts.

Gridmapfile example

```
#####
## InfnGrid Testbed1 users (experiment-independent people)
#####
"/C=IT/O=INFN/L=Torino/CN=Andrea Guarise/Email=Andrea.Guarise@to.infn.it" user01
"/C=IT/O=INFN/L=Milano/CN=Francesco Prelz/Email=francesco.prelz@mi.infn.it" user03
"/C=IT/O=INFN/L=CNAF/CN=Francesco Giacomini/Email=Francesco.Giacomini@cnafe.infn.it" user04
"/C=IT/O=INFN/L=Padova/CN=Massimo Sgaravatto/Email=massimo.sgaravatto@pd.infn.it" user05
"/C=IT/O=INFN/L=Catania/CN=Carlo Rocca/Email=Carlo.Rocca@ct.infn.it" user06
#####
## ALICE Testbed1 users
#####
O=dutchgrid/O=users/O=nikhef/CN=Michiel Botje" user32
"/C=FR/O=CNRS/OU=SUBATECH/CN=Yves Schutz/Email=schutz@in2p3.fr" user33
"/O=Grid/O=CERN/OU=cern.ch/CN=Federico Carminati" user36
"/O=Grid/O=CERN/OU=cern.ch/CN=Predrag Buncic" user37
"/C=IT/O=INFN/L=Catania/CN=Roberto Barbera/Email=roberto.barbera@ct.infn.it" user38
"/C=IT/O=INFN/OU=Personal Certificate/L=Torino/CN=Chiara \
    Oppedisano/Email=Chiara.Oppedisano@to.infn.it" user40
"/C=IT/O=INFN/OU=Personal Certificate/L=Bari/CN=Domenico Di Bari \
    /Email=domenico.dibari@ba.infn.it" user41
"/C=IT/O=INFN/L=Torino/CN=Piergiorgio Cerello/Email=Piergiorgio.Cerello@to.infn.it" user42
```

Gridmapfile creation

In order to ease the gridmapfile creation, the certificate's subjects have been stored in appropriate repositories, managed by the Virtual Organization (experiments, Institutions, collaborations) the users belong to.

The VO certificates server is an LDAP server whose tree structure allows the grouping of the certificate's subjects of the users. Each VO LDAP server is managed by a VO manager who can add, modify, delete and group entries.

Some tools (e.g. [mkgridmap](#)) have been developed by the Authorization Group to get the certificate's subject for a specific users' group from the VO server and create the gridmapfile file.

N.B. The local administrator has complete control over the grid-mapfile

Virtual Organization Authorization servers

At the moment 8 VO LDAP servers exist

ALICE	grid-vo.nikhef.nl	o=alice,dc=eu-datagrid,dc=org
ATLAS	grid-vo.nikhef.nl	o=atlas,dc=eu-datagrid,dc=org
CMS	grid-vo.nikhef.nl	o=cms,dc=eu-datagrid,dc=org
LHCb	grid-vo.nikhef.nl	o=lhcb,dc=eu-datagrid,dc=org
WP6	marianne.in2p3.fr	o=testbed,dc=eu-datagrid,dc=org
InfnGrid	grid-vo.cnaf.infn.it	ou=testbed1,o=infn,dc=it
EarthOb	grid-vo.nikhef.nl	o=earthob,dc=eu-datagrid,dc=org
Biomedical	grid-vo.nikhef.nl	o=biomedical,dc=eu-datagrid,dc=org

DataGrid Testbed1 usage

The following slides will show how to:

- submit a job
- verify the status of a job
- retrieve the output
- check which resources are available for the job
- submit a job to a specific resource
- cancel a job

Simple job submission example

Prerequisite: user authentication on the UI.

```
[gaido@grid006 ~]$ grid-proxy-init
```

```
Your identity: /C=IT/O=INFN/L=Torino/CN=Luciano Gaido/Email=luciano.gaido@to.infn.it
```

```
Enter GRID pass phrase for this identity:
```

```
Creating proxy ..... Done
```

```
Your proxy is valid until Tue Feb 12 22:33:42 2002
```

Simple job submission example

A simple “Hello” job

```
[gaido@grid006 ~]$ more hello.jdl
```

```
#####
```

```
#
```

```
#-----Job Description File :
```

```
#   print "Hello World"
```

```
#
```

```
#####
```

```
Executable   = "/bin/echo";
```

```
Arguments    = "This job has been executed on CE: `/bin/hostname`: and submitted from UI: \  
grid006.to.infn.it";
```

```
StdOutput    = "message.txt";
```

```
StdError     = "stderr.log";
```

```
OutputSandbox = "message.txt";
```

Useful documentation

Job Description Language HowTo:

http://server11.infn.it/workload-grid/docs/DataGrid-01-TEN-0102-0_2.pdf

JDL Attributes:

http://server11.infn.it/workload-grid/docs/DataGrid-01-NOT-0101-0_6-Note.pdf

Job Submission User Interface Man Pages:

http://server11.infn.it/workload-grid/docs/DataGrid-01-TEN-0101-0_2.pdf

Job submission

```
[gaido@grid006 ~]$ dg-job-submit hello.jdl
```

```
Connecting to host grid004f.cnaf.infn.it, port 7771
```

```
Lookup:InputSandbox...No
```

```
Logging to host grid004f.cnaf.infn.it, port 15830
```

```
*****
```

JOB SUBMIT OUTCOME

```
The job has been successfully submitted to the Resource Broker.
```

```
Use dg-job-status command to check job current status. Your job identifier (dg_jobId) is:
```

```
- https://grid004f.cnaf.infn.it:7846/192.135.19.46/095228232744527?grid004f.cnaf.infn.it:7771
```

```
*****
```

Checking the status of a job

```
[gaido@grid006 ~]$ dg-job-status
```

```
"https://grid004f.cnaf.infn.it:7846/192.135.19.46/100042232905186?grid004f.cnaf.infn.it:7771"
```

```
Retrieving Information from server.
```

```
Please wait: this operation could take some seconds.
```

```
*****
```

```
BOOKKEEPING INFORMATION:
```

```
Printing status info for the Job :
```

```
https://grid004f.cnaf.infn.it:7846/192.135.19.46/100042232905186?grid004f.cnaf.infn.it:7771
```


```
Some bookkeeping information has not reached the LB server yet.
```

```
Missing information should come from GlobusJobmanager
```

```
---
```

```
dg_JobId          =
```

```
https://grid004f.cnaf.infn.it:7846/192.135.19.46/100042232905186?grid004f.cnaf.infn.it:7771
```

```
Status           = Done 
```

```
Last Update Time (UTC) = Tue Feb 12 10:03:47 2002
```

```
Job Destination    = dell10.cnaf.infn.it:2119/jobmanager-pbs/workq 
```

```
Status Reason      = terminated 
```

```
Job Owner          = /C=IT/O=INFN/L=Torino/CN=Luciano Gaido/Email=luciano.gaido@to.infn.it
```

```
Status Enter Time (UTC) = Tue Feb 12 10:00:56 2002
```

```
*****
```

Retrieving the output

```
[gaido@grid006 ~]$ dg-job-get-output  
"https://grid004f.cnaf.infn.it:7846/192.135.19.46/100042232905186?grid004f.cnaf.infn.it:7771"
```

Retrieving OutputSandbox files...

```
*****
```

JOB GET OUTPUT OUTCOME

Output sandbox files for the job:

```
- https://grid004f.cnaf.infn.it:7846/192.135.19.46/100042232905186?grid004f.cnaf.infn.it:7771  
have been successfully retrieved and stored in the directory:  
/tmp/100042232905186
```

```
*****
```

```
[gaido@grid006 ~]$ cd /tmp/100042232905186
```

```
[gaido@grid006 100042232905186]$ ls -l
```

```
total 4  
-rw-r--r--  1 gaido  tbusers   97 Feb 12 11:03 message.txt
```

```
[gaido@grid006 100042232905186]$ cat message.txt
```

This job has been executed on CE: dell10.cnaf.infn.it: and submitted from UI: grid006.to.infn.it

Checking which resources are available for this job (aka match the user's requirements)

```
[gaido@grid006 ~]$ dg-job-list-match hello.jdl
```

```
Connecting to host grid004f.cnaf.infn.it, port 7771
```

```
*****
```

COMPUTING ELEMENT IDs LIST

The following CE(s) matching your job requirements have been found:

- dell10.cnaf.infn.it:2119/jobmanager-pbs-workq
- grid001.ct.infn.it:2119/jobmanager-pbs-workq
- grid001.pd.infn.it:2119/jobmanager-lsf-grid01pd
- grid001.pd.infn.it:2119/jobmanager-lsf-grid02pd
- grid002.to.infn.it:2119/jobmanager-pbs-workq

```
*****
```

Forcing the submission of the job to a specific resource

```
[gaido@grid006 ~]$ dg-job-submit hello.jdl \  
-resource grid001.pd.infn.it:2119/jobmanager-lsf-grid01pd
```

```
Connecting to host grid004f.cnaf.infn.it, port 7771  
Lookup:InputSandbox...No  
Logging to host grid004f.cnaf.infn.it, port 15830
```

JOB SUBMIT OUTCOME

The job has been successfully submitted to the Resource Broker.
Use dg-job-status command to check job current status. Your job identifier (dg_jobId) is:

- <https://grid004f.cnaf.infn.it:7846/192.135.19.46/103714233509253?grid004f.cnaf.infn.it:7771>

```
[gaido@grid006 ~]$ dg-job-status \  
" https://grid004f.cnaf.infn.it:7846/192.135.19.46/103714233509253?grid004f.cnaf.infn.it:7771"
```

Retrieving Information from server.

Please wait: this operation could take some seconds.

BOOKKEEPING INFORMATION:

Printing status info for the Job :

<https://grid004f.cnaf.infn.it:7846/192.135.19.46/103714233509253?grid004f.cnaf.infn.it:7771>

Some bookkeeping information has not reached the LB server yet.

Missing information should come from GlobusJobmanager

```
dg_JobId          =  
https://grid004f.cnaf.infn.it:7846/192.135.19.46/103714233509253?grid004f.cnaf.infn.it:7771  
Status           = Done  
Last Update Time (UTC) = Tue Feb 12 10:40:22 2002  
Job Destination   = grid001.pd.infn.it:2119/jobmanager-lsf/grid01pd  
Status Reason     = terminated  
Job Owner         = /C=IT/O=INFN/L=Torino/CN=Luciano  
Gaido/Email=luciano.gaido@to.infn.it  
Status Enter Time (UTC) = Tue Feb 12 10:40:17 2002
```



Cancelling a job

```
[gaido@grid006 ~]$ dg-job-submit hello.jdl
```

```
Connecting to host grid004f.cnaf.infn.it, port 7771
```

```
Lookup:InputSandbox...No
```

```
Logging to host grid004f.cnaf.infn.it, port 15830
```

```
*****
```

JOB SUBMIT OUTCOME

```
The job has been successfully submitted to the Resource Broker.
```

```
Use dg-job-status command to check job current status. Your job identifier (dg_jobId) is:
```

```
- https://grid004f.cnaf.infn.it:7846/192.135.19.46/162132237024641?grid004f.cnaf.infn.it:7771
```

```
*****
```

Canceling a job

```
[gaido@grid006 ~]$ dg-job-status
```

```
"https://grid004f.cnaf.infn.it:7846/192.135.19.46/162132237024641?grid004f.cnaf.infn.it:7771"
```

```
.....
```

```
Printing status info for the Job :
```

```
https://grid004f.cnaf.infn.it:7846/192.135.19.46/162132237024641?grid004f.cnaf.infn.it:7771
```

```
dg_JobId          =
```

```
https://grid004f.cnaf.infn.it:7846/192.135.19.46/162132237024641?grid004f.cnaf.infn.it:7771
```

```
Status           = Scheduled
```

```
Last Update Time (UTC) = Tue Feb 12 16:21:18 2002
```

```
Job Destination   = dell10.cnaf.infn.it:2119/jobmanager-pbs/workq
```

```
Status Reason     = initial
```

```
.....
```

```
[gaido@grid006 ~]$ dg-job-cancel \
```

```
https://grid004f.cnaf.infn.it:7846/192.135.19.46/162132237024641?grid004f.cnaf.infn.it:7771
```

```
Are you sure you want to remove specified job(s)? [y/n]n:y
```

```
Cancel request submitted to RB "grid004f.cnaf.infn.it". Waiting for job(s) cancellation results....
```

Cancelling a job

```
[gaido@grid006 ~]$ dg-job-status \  
"https://grid004f.cnaf.infn.it:7846/192.135.19.46/162132237024641?grid004f.cnaf.infn.it:7771"
```

Retrieving Information from server.
Please wait: this operation could take some seconds.

BOOKKEEPING INFORMATION:

Printing status info for the Job :

<https://grid004f.cnaf.infn.it:7846/192.135.19.46/162132237024641?grid004f.cnaf.infn.it:7771>

.....

dg_JobId = <https://grid004f.cnaf.infn.it:7846/192.135.19.46/162132237024641?grid004f.cnaf.infn.it:7771>

Status = Done (Cancelled) 

Last Update Time (UTC) = Tue Feb 12 16:24:40 2002

Job Destination = dell10.cnaf.infn.it:2119/jobmanager-pbs/workq

Status Reason = cancelled on ResourceBroker/grid004f.cnaf.infn.it 

Job Owner = /C=IT/O=INFN/L=Torino/CN=Luciano

Gaido/Email=luciano.gaido@to.infn.it

Status Enter Time (UTC) = Tue Feb 12 16:21:47 2002

Running a benchmark program

```
[gaido@grid006 test]$ more benchmark.jdl
```

```
#####
```

```
#
```

```
#-----Job Description File :
```

```
#
```

```
# command:
```

```
#   dg-job-submit benchmark.jdl
```

```
#   dg-get-job-output
```

```
#
```

```
#           authors: INFN-TO
```

```
#####
```

```
Executable = "benchmark.sh";
```

```
Requirements = Member (other.RunTimeEnvironment , "TORINO") && Member
```

```
(other.RunTimeEnvironm
```

```
ent , "INFN");
```

```
StdOutput = "benchmark.output";
```

```
StdError = "benchmark.err";
```

```
OutputSandbox = {"benchmark.output","benchmark.err"};
```

```
InputSandbox = {"/home/gaido/test/benchmark.sh","/home/gaido/test/benchmark.cpp"};
```

```
[gaido@grid006 test]$ more benchmark.sh
#!/bin/bash
c++ benchmark.cpp -o benchmark
time ./benchmark -v -num_iters 100
exit
```

`num_iters` is the number of cycles executed by the program

Benchmark.cpp is a program that executes benchmarks for a simple evaluation of computer's performance.

It's possible to evaluate independently:

- the CPU power
- the I/O performance

with different memory allocations.

Options.

Usage: ./benchmark [-v]

[-in <input file>]	(default: /dev/zero)
[-out <output file>]	(default: stdout)
[-n_in <# of input blocks>]	(default: 0)
[-dim_in <input block length>]	(default: 0)
[-n_out <# of output blocks>]	(default: 0)
[-dim_out <output block length>]	(default: 0)
[-ms <malloc size>]	(default: 1280)
[-num_iters <# of program cycles>]	(default: 100000)
[-cpu_iters <# cpu loops per cycle>]	(default: 1)

The default behaviour is to execute a loop with 100000*1 cpu iterations (~ 1 min wall time on a PIII 600), with 128 MB RAM allocated and no I/O.

A description of the program is available at:

http://www.to.infn.it/grid/fbp/doc/globus_bench_howto.ps

Scripts available at:

<http://www.to.infn.it/grid/testbed/examples/>