# A local Globus install at NIKHEF

David Groep

29th August 2000

$Id: globus-local.tex,v 1.4 2000/08/29 14:26:23 davidg Exp davidg $

**Abstract**

What is *Globus*? What are its basic principles of operation, how can it be installed, and what local changes were necessary? What is a proper scheme for the authorization of end-users? What level of authority is needed to install and to run the daemons?



## 1 What is Globus

Globus [1] is a toolkit for distributed meta-computing based on the 'Grid' [2] concept. It consists of two parts: a set of daemons providing the infrastructure (remote job execution, cached data access, directory services, secure communications), and a wide variety of subsystems to create 'grid-enabled' applications (APIs for transparent access to secondary storage, platform-independent secure communications using familiar file/socket abstractions as well as a NEXUS-based system).

The Grid and the Globus toolkit are continuously evolving and change rapidly over time. This document focuses on release 1.1.3. In future releases one may expect amongst others new subsystems for resource discovery and brokering, and interfaces to commodity technology (Java/Jini, CORBA, DCOM).

# 2 The infrastructure

This document gives some (draft) information on the local Globus setup. It does *not* describe:

- How to use globus as an end-user – see www.globus.org for details on using the grid programs. A good start: the local `GLOBUS_INSTALL_PATH` should be set to `/global/ices/toolset/globus`. Start from there by sourcing the `globus-user-setup.sh` file.

  Certification requests should for the time being be mailed to `davidg`.
- How in properly run Globus for the VLab – see the document by Frank van der Linden [3] for some preliminary thoughts. The administrative environment suggested in this document is compatible with the proposed VLab environment.
- How to install Globus on another site – but you can peruse the information in this document together with the Administration guide for a jumpstart.
- How to deploy Globus on other hosts at NIKHEF – ask me `:-)`.

## 2.1 Directory infrastructure (GIIS)

The Globus Institute Information Service (GIIS) is used to locate resources that are part of the Grid. It contains:

- a host listing
- architectural characteristics and benchmarks per host
- contact information per host and per 'job manager'
- dynamic system information per host (load, free nodes)

A GIIS runs on a single machine and acts as a central repository for resource information at the site. To this end, it runs a LDAP server with a 'shell' back end. The shell back end consists of of scripts to retreive information from per-resource GRIS (GRid Information Service) services. It relays this information to the requesting LDAP client. The GIIS will temporatily cache this information on a per-host basis. The GRIS specifies the time-to-live for each piece of information.

The GRIS service runs on every resource and listens on port 2135. This GRIS will spawn a shell scripts (`grid-info-*`) to collect system information: architecture, current load, network connectivity, etc.

The GIIS and GRIS will log information to `var/grid-info-system.log`. It will not write to syslog.

*PS*: The instantaneous load on the GIIS (and GRISs) may become quite high, most of the processor time being spent in `system` and `iowait` cycles.

### Security

The access to the GIIS and GRIS is via the LDAPv2 protocol. The communication is not encrypted and anonymous, but read-only. Although there are no known exploits available for the GIIS/GRIS LDAP service, it is hard to guarantee that malicious requests will not results in a breach of security. This is largely related to the use of the shell back end by the LDAP server: it might be possible that malicious queries get propagated to a shell script. No security tests have been performed.

In a future release, the shell back end will be replaces by a more conventional ldbm back end with soft referrals from the GIIS to the GRIS. This will bring the LDAP server in line with more conventional installs elsewhere. It will likely increase the security of the system.

## 2.2 GRAM gatekeeper

The GRAM gatekeeper distributes jobs submitted to the resource by end-users. There is one gatekeeper per resource. This gatekeeper supports multiple 'back-ends' or job-managers; the available job managers are listed in the `etc/globus-services` file and published in the GRIS.

Typically at least a *fork* job-manager is available. This will directly execute the job on the machine running the gatekeeper. Other possible job-managers include Condor, LoadLeveler, and NQE. At NIKHEF, only 'fork' job managers are currently installed.

The GRAM gatekeeper will log to `var/globus-gatekeeper.log` and will also write user contact information to syslog with facility `daemon`. All syslog entries start with the literal '`GRAM gatekeeper`' and include the current gatekeeper PID.

### Access and authorization

Access to the GRAM gatekeeper is secured via SSL (Secure Socket Layer [5]) and also conforms to the X.509 standard for certificates. The security infrastructure is similar to secure HTTP, *e.g.*, used for secure payments via the Web.

A user has to present a valid user certificate, signed by a trusted Certificate Authority. A signing policy is specified in `share/certificates/ ca-singing -policy.conf`. To reduce exposure of the user cert, a 'proxy' with a limited lifetime (default: 12 hrs) is created and used for authenticating the user at all Grid resources. The user proxy is signed with the users private key. It is stored on the local filesystem. All communications between the GRAM gatekeeper and the submitting host is strongly encrypted (using 1024-bits keys).

A grid-map file (`etc/grid-mapfile`) exists to map end-users (using their Distinguished Name) to local user-IDs. As long as the signing CA is trusted, we can safely assume that the identity (DN) presented by the user is true, *i.e.*, he is indeed the person we thing he is. The map-file then translates the DN to a local uid. The job will be started by the gatekeeper under this uid. Note that the gatekeeper should be able to `setuid(2)` to this user! There is a choice on how to set up the grid-map file:

- It is a local choice to run either all Grid processes under one (1) uid and map all the DNs to this user. The various external users can no longer be distinguished locally and can read/write/influence each others data and processes.

  This is fundamentally insecure. But in this case the gatekeeper may run as a daemon under the uid of a generic Grid user. No root access is requires to any system.

  You still need two accounts (one for the Globus install and one to run the gatekeeper). These accounts should **never** be combined for obvious reasons.

- Map the DN to the equivalent local uid. Every user running jobs via the gatekeeper should also have a regular account on the system and this is not presented with access that he would normally not have. Accounting is also simplified and actions of users can be traced to individuals

  This requires the gatekeeper to run as uid 0 (on UNIX systems), either as a daemon or from `inetd(1M)`.

Of coarse, the grid-map file can be setup as a mixture of both. But in that case the gatekeeper should still run as root.

**Security**

Although currently no exploits against the GRAM gatekeeper are known (for version 1.1.3), access from 'external' sources should preferably be restricted to limit exposure of the gatekeeper. The GRAM gatekeeper listens on port 2119.

### 2.3 Gass server

The GASS (Global Access to Secondary Storage) is run on a per-user basis, either stand-alone (on a resource providing storage services) or as part of a job submission to propagate stdin/stdout. It also serves a purpose in prestaging executables to remote resources.

The GASS server uses secure http for authentication and data transfer, akin to the system used form GRAM job submission. However, regular connections to a GASS server using a web-browser will fail.

**Security**

The GASS server is as secure as the GRAM gatekeeper. Since the GASS server is run on demand only and uses a random TCP port for access, it is harder to detect and exploit. At the same time, however, it also makes it harder to block access to the GASS server from the router. The limited live cycle of a GASS server and the fact that it runs under a noon-privileged uid (the one executing the job), limit the exposure suffered by the network due to the GASS server.

## 3 Certification Authority

Two local certification authorities (CAs) are deployed to test the Globus install and to perform some initial testing. These CAs are:

**nikCA** This CA signed the host certificates for the micro grid and the two initial users (`davidg` and `meet`). This CA has the DN:
`O=Vlab, OU=NIKHEF, CN=nikCA Certificate Authority/ Email=davidg@nikhef.nl`
MD5 fingerprint: A5:E7:AD:60:9E:29:06:E1:40:BB:E6:D6:58:76:A8:07, hash: cee276c0.

This CA also signed the host certificate for a test-bed secure web server (the cert is for *.nikhef.nl). There is some tooling available to convert user certs signed by the CA to PKCS#12 type user certs for use by Netscape and MSIE.

**nikhefCA** This CA was initially used for testing certification policies (allowing access for different classes of users signed by different CAs). This CA has a 'better' DN and might be used as a basis for a more firm local CA. Its DN is:
`C=NL, O=NIKHEF, CN=NIKHEF CA Organization`
MD5 fingerprint: A3:F3:E8:16:12:F0:4B:5E:CA:94:38:06:E1:0E:B2:37, hash: 263d1de6.

Since these 'local' CAs will probably not inspire a lot of trust outside NIKHEF, it might be useful to apply for certification with another (new) CA, *e.g.*, a local CA for the WCW, a new CA to be operated by SURFNET (related to the PKI project) or maybe even a commercial CA like *Verisign* or *Twarthe*.

But since you can allow for multiple CAs to sign user certs, this is not an immediate issue (as long as all participants trust each other). Note that a specific user cert can be signed by one and only one CA. User key ring support is not currently part of the Globus toolkit (but is is foreseen for some later release).

The following files contain localized information on subject and CA names:

| | |
|---|---|
| `etc/globus-gatekeeper.cert` | name in this cert (and matching key) is extracted at gatekeeper startup and used to re-write the jobmanager configuration file. |
| `share/certificates/ca-sign...` | determined which CA can sign which certificates. |
| `~/.globus/...` | user information. The `grid-cert-request` script takes the user cert DN from the cofiguration script at `etc/grid-security.conf`. |
| `etc/grid-security.conf` | Contains the baseDN used for both gatekeeper and user DNs. |

The name of the gatekeeper certificate is generated by the `grid-cert-request` program, using the hostname obtained from the master `etc/gatekeepers.conf` file. Its invication, as taken from

```
${bindir}/grid-cert-request -gatekeeper ${name} -force \
    -dir ${deploy_etc} \
    -cert ${cert_file} -key ${key_file} \
    -req ${req_file} > /dev/null 2>&1
```

# 4 Miscellaneous Information

## 4.1 Ports used by the Globus toolkit

| portno | proto | purpose |
|---|---|---|
| 2119 | TCP | GRAM gatekeeper |
| 2135 | TCP | GRIS per-resource information service |
| 30001* | TCP | GIIS directory (on bilbo only) |
| 52493 (`xcd0d`)* | TCP | system monitoring information in.gsd (non-Globus) |

(* – local choice)

## 4.2 Local install

The current installation directory of the Globus toolkit at NIKHEF is `/global/ices/toolset/globus`, with a per-host deploy directory, whose path is state din the `globushosts` file in the `.../localize.nikhef/` directory. This should be a local filesystem for the resource. Also here, a 'README' file describes the procedure to add hosts to the grid and how to propagate modifications to all deployed hosts.

## 4.3 Temporary characteristics

All files are currently owner by `davidg`, but ownership should be transferred to a 'grid admin' account that is also part of the `ices` group. This account is prevents to Globus install base and information services from being tampered with. This new account should under no circumstance be used to run the Gatekeeper!

The microgrid consists of: bilbo (GIIS), bombur, triplex, triomf (currently not active).

The following patches were applied to the Globus 1.1.3 system:

- build `globus-gass-cache` by hand for each architecture. A faulty configure.in 'forgets' this cache management tool. You can build by hand using the Makefiles from the DevelopersTutorial.

- The `etc/grid-security.conf` file contains a typo. Modify the line `GSI_CA_EMAIL_ADDR` by including the forgotten `CA_`.
- For production operation, apply the patch to `globusrun.c` suggested on the *discuss* mailing list (*subject: Re: Problem with globus-job-clean*).

On some hosts a simple perl-based daemon (in.gsd) will answer quesries about the current load and utilisation of the system. It will also report on the status of a potential Gatekeeper. The daemon listens on port 52493. A graphical user interface (gsm-0.1) will visualise the results. The daemon *is* liable to denial-of-service attacks (it can be killed remotely), but is otherwise secure: it cannot be lured into doing any more than printing the system load. It is not part of the Globus toolkit and used for monitoring the microgrid testbed only.

# 5 Local modifications

The globus toolkit install is well documented in the System Administration Guide. The following local choices and modifications are applied:

**Install paths** SSLeay in installed in `/global/ices/toolset/ssl/`*`arch`*`/`. The `.../ssl/lib` directory contains a configuration file that was formerly used by the nikCA but is being phased out.

OpenLDAP sorces were retreives from the Globus site and contain the required 'time-out patch'. They were built using the default options. A special version (for i386-linux) that includes the ldbm backend is available. This installation can be used to evaluate a regular LDAP direcroty for other purposes.

Globus in installed in `/global/ices/toolset/globus`. It contains a '`localoize.nikhef`' directory with some per-host setups.

**Modifications to the globus files** These files were modified:

`etc/grid-info.conf` – this file is modified by `globus_setup` and contains information on the GIIS. Modify this file in the installdir and every deploy dir to change, *e.g.*, the port the GIIS listens to (30001). Our info model is `MDS_SITE_INDEX`.

`etc/grid-info-hosts.conf` – lists the hosts in the microgrid. It was redistributed to all deploy directories after Globus was deployed on all four hosts.

`etc/globus-services.conf`
`etc/globus-services` – unchanged, it contains only a fork-style job manager. This is a per-resource file, generated from a system-wide file `etc/globus-services.conf`. Modify the latter file and rerun `globus-local-deploy` on the relevant machine.

`etc/globus-gatekeepers.conf` – the default was changed to 'daemon'. The original 'inetd' requires root privilege to install. It contains explicit entries for the four hosts, although that might be unneccesary.

`etc/grid-mapfile` – changed and distributed to all four hosts after deployment. It contains some sample entries for 'David Groep', 'Kors Bos', 'Victor Klos' and 'EMIN-meet shared account'. Currently, they all map to 'davidg', the user running the gatekeeper.

`etc/globus-jobmanager.conf` – this file seems modified but is actually a default. It is updates by `SXXglobus start` to reflects a possibly changed certificate subject of the gatekeeper (see sources).

`sbin/globus-startup-lib.sh` – this file contains the uid used by the SXX startup scripts to run commands that do *not* need root priviliges. It is set at local-deploy time to the user performing the deployment. The statement says: `GLOBUS_UID="gridadm"`.

`share/certificates` – the directosy is used for a hash-based certificate lookup from SSL. It contains the certificates for the two new CAs (nikCA and nikhefCA), whose hashes are 'cee276c0' and '263d1de6', respectively.

`share/certificates/ca-signing-policy.conf` – This file was extended to allow: (1) the nikCA authority to sign '/O=Vlab/O=Globus/' certificates and (2) the nikhefCA to sign '/C=NL/O=NIKHEF/' as well as '/O=Vlab/O=Globus/' certificates. This file has the regular EACL format used by OpenSSL (SSLeay).

Optionally, you can distrust the Globus CA to sign Globus certificates '/O=Grid/O=Globus/' and '/C=US/O=Globus/' if needed. It seems better not to allow acces (using the grid mapfile) to any '/O=Grid/O=Globus/' credentials.

**the nikCA and nikhefCA authority organizations** The relevant scripts are (almost) identical to the 'demoCA' shipped with SSLeay. The nikhefCA uses a local configuration file, the nikCA still uses the global `ssleay.cnf` in the toolset/ssl directory.

A 'signmail' script automated the process to signeing incoming certification requests, generated by either 'globus-local-deploy' (for Grid hosts) or 'grid-cert-request' (for Grid users).

**Options for globus_setup** The Globus toolkit has been set up with a a minimum number of local modifications. The setup uses the 'new' GIIS model (*i.e.*, the `globus_setup` script was called without the `-classic` option).

The MDS/GIIS setup options – MDS host is 'bilbo.nikhef.nl', MDS port is 30001 and the Organization DN is 'dc=nikhef, dc=nl, o=Grid'. This DN is compatible with the default Globus install (it uses the o=Grid as the base DN for the DIT).

The security (GSI) setup options – base DN for hosts is 'c=nl, o=nikhef', the base DN for users is (also) 'c=nl, o=nikhef'. If you modify these values directly in `etc/grid-security.conf`, run `grid-cert-request-config` from the tools directory afterwards.

To 'localize' a newly deployed globus system, and to propagate changes in the local setup to all relevant hosts, a 'localize' distribution script `Dist.sh` is available in the deploy directory `localize.nikhef`. Changes in global configuration should be propagated using this script. Mapfiles are stored in a central location `localize.nikhef/mapfiles/`. Add and remove users there and use `./Dist.sh` to propagate.

# 6 Administrative environment

There should be two accounts:

**gridadm** A Globus/Grid administation account. This pseudo-user owns the files related to the globus install and deployment. It is preferably part of the `ices` group, so it can use the installed base at `/global/ices`. It should not run *any* globus services.

**griduser** A (temporary) account for testing grid services. It should preferably be in its own group, not be able to write anywhere accept for its home directory (and /tmp), akin to user 'nobody'.

The gatekeeper should run as root, either from inetd or as a stand-alone daemon. On selected hosts, Globus should be started by default from the system startup scripts. The 'services' may include reference to the globus gatekeeper on port 2119. These actions require a certain amount of trust regarding the Globus admin person.

A number of rulesets should preperably be added to the hef-router configuration to secure the Globus deployment:

- deny access to tcp/2119 on trusted networks from anywhere outside trusted networks.
- deny access to tcp/2135 on trusted networks anywhere outside trusted networks.
- deny access to tcp/30001 on ⟨GIIS host⟩ (currently bilbo) from anywhere outside WCW. The GIIS is, at this time, extremely cpu-intensive and makes the host liable to denail-of-service attacs from outside. It can in principle halt the machine. On the other hand, the networks on the WCW are relatively well supervised and it is therefore not strictly necessary to block these as well. Keeping them open allows a better exchange of information among the participating VLab institutes.

These rules might later be relaxed slightly to allow access from selected WCW sites participating in the Virtual Lab or from participating DutchGrid institutes like KNMI/SARA. Persons submitting jobs to the Grid via Globus should have a local account.

# References

[1] `http://www.globus.org`

[2] *The Grid: Blueprint for a New Computing Infrastructure*, I. Foster and C. Kesselman eds., Morgan Kaufmann Publishers 1998.

[3] *Site-specific setup of Globus for the Virtual Laboratory*, Frank van der Linden, July 14, 2000.

[4] `http://www.openldap.org`

[5] `http://www.openssl.org`