



IST-2001-32133

GridLab - A Grid Application Toolkit and Testbed

Rules for software installed on GridLab testbed

Author(s):	Miroslav Ruda, Luděk Matyska, Zdeněk Salvat
Document Filename:	software
Work package:	Testbed
Partner(s):	Masaryk University
Lead Partner:	Masaryk University
Config ID:	GridLab-5_3-NOT-0003-1_1
Document classification:	Internal

Abstract: This document defines the rules that must be followed by any software package to be regularly deployed and maintained on the GridLab testbed.



Last amendment date: 2004/09/29 & time: 12:57:25



Contents

1 Preamble	2
2 Software definition parts	2
2.1 Documentation	2
2.2 Source code	3
2.3 Source code availability	3
2.4 Versioning	3
2.5 Installation	3
2.6 Environmental variables	4
2.7 Startup, cron, daemon rules	4
2.8 Tests	5
3 Variables defined in /etc/gridlab.conf	5
Relevant Documents	7

1 Preamble

The purpose of this document is to define common rules that must be fulfilled by all GridLab developed software that will be installed on the GridLab testbed. The document covers basic requirements on software packaging, software building procedure, software documentation, software startup scripts etc. It doesn't cover aspects discussed in documents "Quality Assurance Plan" and "Coding Guidelines" [1, 2], like protocol and API design, common source code style, source code documentation etc. *Rules for deploying and testing various versions of testbed?*

Because several documents are required for each package, we suggest to have a short web page for each package delivered to testbed. This page should include all required documents, links to source code tarballs etc.

General GridLab configuration file `/etc/gridlab.conf` is proposed in this document. This file should be installed on each machine in GridLab testbed and should be used to set the basic environmental variables necessary to locate GridLab installation tree, Globus installation and other installed software. To support several instalations of GridLab software, location of `/etc/gridlab.conf` can be overridden by `GRIDLAB_CONF` environment variable.

Complete description of all defined variables is still missing, but several variables are already discussed in this document. For up-to-date list of currently defined variables see section 3. Syntax of this configuration file must be sh-compatible, this file will be "sourced" by startup scripts as sh scripts:

```
#!/bin/sh
# read gridlab.conf
if [ "x$GRIDLAB_CONF" = "x" ]; then
    GRIDLAB_CONF=/etc/gridlab.conf
fi
. $GRIDLAB_CONF
#
$GRIDLAB_LOCATION/testbed/bin/start -f $GLOBUS_LOCATION/etc/
```

GridLab project has central CVS repository located on `cvs.gridlab.org`. However, for groups located in one site, it may be impractical to use this remote repository for their daily use. Therefore we require that access to local CVS repository is provided and that will be used to mirror the software into the central CVS or stable versions of software will be copied to central CVS repository. *Details of mirroring will be negotiated later.*

2 Software definition parts

2.1 Documentation

- Documentation including license, short description of provided functionality, download information, software and hardware requirements for build and use of software, installation and administration manual is required.
- Installation manual must cover building procedure and options, description of required startup commands, description of test procedure etc.
- Administration manual should cover description for daily administration tasks, requirements to system (temporary space (with location), network ports used for communication etc.)

- All third party software required for compilation or running must be specified (including required Globus packages and flavors). Specification must include at least
 - precise URL where software can be obtained
 - license of this software
 - required options for building
 - suggested options for building
 - required version.

2.2 Source code

Source code should be written with portability in mind. Before software is accepted by testbed, software should be tested by authors on at least two different operating systems available on GridLab testbed (Linux and Irix are the most frequent, another good candidate is True64 on HPQ machines).

All GridLab programmers can get account on GridLab testbed to test their software on these platforms.

2.3 Source code availability

- Tarball including complete source code must be provided by software authors.
- CVS access coordinates (including `CVSROOT` and tag corresponding for each release) must be specified. The use of central CVS repository is strongly preferred.

2.4 Versioning

- Versions of software should be maintained in three-level format (major.minor.patch), where
 - change on major version number means that version is not backward compatible with previous versions
 - minor version number changes when new feature is added, with backward compatibility with previous version
 - patch version number can change when no changes to API were done. It is intended for bug fixes etc.

2.5 Installation

- Not only source code itself, but also building procedure should be prepared for portability. Authors should check that all required software, tools and libraries are ported to all platforms in GridLab testbed. In specific cases not all platforms may be supported, but this must be clearly stated in the software description, together with a clear and concise reason (and should be considered as an exception).
- Source code should be prepared for standard build procedure

```
configure -with-gridlab && make && make install
```

Autoconf use is not mandatory, but some equivalent tool should be used during build to allow easier installation on all platforms.

- When software provides runtime and development parts, it must be possible to install only selected parts. The installation procedures must be well documented.
- Division of installed files and directories to shareable, machine architecture-specific and host-specific ones should be documented.
- Software should use `/opt/gridlab/<software-dir>` as default location. However, software must be relocatable using `GRIDLAB_LOCATION` variable when started and using `-prefix` during building.
- Software must support installation of several versions. Proposed solution is to use directory `/opt/gridlab/<software-dir>-version` for installation and `/opt/gridlab/<software-dir>` as symbolic link to stable version.
- Subtree `/opt/gridlab/<software-dir>` should be organized in “standard way”—with subdirs `bin`, `sbin`, `doc`, `include`, `lib`, `shared`, etc

2.6 Environmental variables

- At runtime, package should use `$GRIDLAB_LOCATION` (defined in `/etc/gridlab.conf`) to locate installation tree.
- Startup scripts should use `/etc/gridlab.conf` to set variables like `$GLOBUS_LOCATION`, `$GRIDLAB_LOCATION`, `$GRID_SECURITY` etc.
By default, the value of `$GRIDLAB_LOCATION` is defined as `/opt/gridlab`. Similarly, `$GLOBUS_LOCATION` and `$GRID_SECURITY` should be used to locate Globus installation tree and `grid-security` directory.
- Packages are free to define their own variables in the form `GRIDLAB_software_variable`
- No file (including host certificates etc.) should be expected in “default” location— all filenames must be relative to `$GLOBUS_LOCATION`, `$GRIDLAB_LOCATION` etc. The only departure from this rule is location of the master configuration file `/etc/gridlab.conf`.

Should we set predefined variable names for logs, temporary files, configuration files, and other host-specific files? Should `$GRIDLAB_LOCATION_LOCAL` point to them?

2.7 Startup, cron, daemon rules

- Software startup scripts which can be used in System-V style init script (`/etc/init.d/` directory) must be provided. It should support at least `start`, `stop`, and `restart` commands. Dependencies on other services running should be documented.
- cron jobs can be used for periodical tasks. Crontab entries (provided by package authors) should have their command part in the form

`. /etc/gridlab.conf; $GRIDLAB_LOCATION/<software-dir>/.../command,`

use of more complex scripts is strongly discouraged.

- Permanent grid services should be installed as daemons. Startup and shutdown of these daemons must be controlled using startup scripts from `/etc/init.d` directories. Daemons can be controlled by configuration files located in `$GRIDLAB_LOCATION/<software-dir>/etc`

. Daemons should not be expected to run with root privileges. Similarly, services should use special service certificates instead of host certificates.

It is important to describe in the documentation the purpose of such permanent grid service and also give a qualified guess of the resources consumptions (not only CPU, but also memory, disk space—don't forget any log files generated—etc.).

- Commands (scripts) to force log rotation should be provided.

2.8 Tests

- Simple programs testing basic functionality must be provided. Documentation must describe procedures which can be used to verify installation and basic functionality.
- When service is expected to be monitored by testbed monitoring tools, simple test must be provided.

3 Variables defined in `/etc/gridlab.conf`

Location of `/etc/gridlab.conf` can be overridden by `GRIDLAB_CONF` environment variable. Variables defining installation root:

Variable	Default value	Description
<code>HOSTNAME</code>	–	FQDN
<code>HEADNODE_HOSTNAME</code>		FQDN of head node on clusters
<code>GLOBUS_LOCATION</code>	<code>/opt/globus</code>	Root of Globus installation tree
<code>GLOBUS_FLAVOR</code>	<code>gcc32dbg</code>	Globus flavor
<code>GLOBUS_FLAVOR_THREADS</code>	<code>gcc32dbgpthr</code>	Theaded Globus flavor
<code>GLOBUS_INCLUDE</code>		Include directory for non-theaded Globus flavor
<code>GLOBUS_INCLUDE_THREADS</code>		Include directory for theaded Globus flavor
<code>GRIDLAB_LOCATION</code>	<code>/opt/gridlab</code>	Root of GridLab installation tree
<code>GRID_SECURITY</code>	<code>/etc/grid-security</code>	<code>grid-security</code> directory, it contains host certificates, certificates of accepted CAs
<code>GRID_MAPFILE</code>	<code>\$GRID_SECURITY/grid-mapfile</code>	Location of <code>grid-mapfile</code>
<code>CERTIFICATION_AUTHORITY_DIR</code>	<code>\$GRID_SECURITY/certificates/</code>	Location of accepted CAs

Variables defining path to required or optional software. These variables are defined only if corresponding software is installed.

Variable	Description
F90_NATIVE	Pathname of the native F90 compiler
F90_KAI	Pathname of the Kai F90 compiler
F90_PGI	Pathname of the Portland Group F90 compiler
F90_INTEL	Pathname of the Intel F90 compiler
F90_GNU	Pathname of the GNU F90 compiler
CXX_NATIVE	Pathname of the native C++ compiler
CXX_KAI	Pathname of the Kai C++ compiler
CXX_PGI	Pathname of the Portland Group C++ compiler
CXX_INTEL	Pathname of the Intel C++ compiler
CXX_GCC	Pathname of the GNU C++ version 2.x compiler
CXX_GCC3	Pathname of the GNU C++ version 3.x compiler
C_NATIVE	Pathname of the native C compiler
C_KAI	Pathname of the Kai C compiler
C_PGI	Pathname of the Portland Group C compiler
C_INTEL	Pathname of the Intel C compiler
C_GCC	Pathname of the GNU C version 2.x compiler
C_GCC3	Pathname of the GNU C version 3.x compiler
JAVA_HOME	Path to the Java installation tree
GMAKE	Pathname of the GNU make
CVS	Pathname of the CVS
PERL	Pathname of the Perl (version ≥ 5.005)
HDF5_LOCATION	Root of HDF5 installation tree

Variables defining path to Gridlab software. These variables are defined only if corresponding software is installed.

Variable	Description
GRIDLAB_MONITORING_LOCATION	Root of Gridlab monitoring software (Mercury) installation tree
GRIDLAB_MDS_LOCATION	Root of Gridlab information service installation tree
GRIDLAB_IGRID_LOCATION	Root of iGrid installation tree
GRIDLAB_ADAPTIVE_LOCATION	Root of Gridlab adaptive software (Pythia) installation tree
CACTUS_LOCATION	Root of Cactus installation tree
CACTUS_DEV_LOCATION	Root of Cactus (development version) installation tree
GAT_LOCATION	Root of GAT installation tree
GAT_ADAPTOR_PATH	GAT Adaptor list
GAT_DEV_LOCATION	Root of GAT (development version) installation tree

Variables describing MPI installation are defined to be compatible with Cactus developers manual (<http://www.cactuscode.org/Guides/Stable/UsersGuide/UsersGuideHTML/node14.html>). The main difference is that all available MPI installations should be described. These variables are defined only if corresponding software is installed.

Variable	Description
MPLDEFAULT	Default MPI, used by globus jobmanagers when MPI job is started. Can be one of native, mpich, mpich-gm, lam, mpipro.
MPLNATIVE_DIR	Native MPI is available on this machine available in defined directory.
MPLNATIVE_LIBS	If defined, linking with native MPI will use this libraries instead of "-lmpi -lm"
MPICH_DIR	Path to MPICH (http://www-unix.mcs.anl.gov/mpi/mpich) installation tree. If defined, corresponding MPICH_ARCH and MPICH_DEVICE variables should be defined too.
MPICH_ARCH	Machine architecture
MPICH_DEVICE	The device used by MPICH
MPICH_GM_DIR	Path MPICH-GM (www.myri.com) installation tree. If defined, corresponding MYRINET_DIR variable should be defined too.
MYRINET_DIR	Directory in which Myrinet libraries are installed.
LAM_DIR	Path to LAM (http://www.lam-mpi.org/) installation tree.
MPIPRO_DIR	Path to MPIPro (http://www.mpi-softtech.com/) installation tree.
MPICHG2_DIR	Path to MPICH-G2 (http://www3.niu.edu/mpi/) installation tree. If defined, corresponding MPICHG2_FLAVOR variable should be defined too.
MPICHG2_FLAVOR	Globus flavor used by MPICH-G2.

Rules for decision which mpicc/mpif77 corresponds to default (preferred) MPI installation (used for example by globus jobmanager when MPI job is started) are defined this way:

- if MPICC/MPIF77 variables are set, these programs will be used to compile MPI programs
- if mpicc/mpif77 is available in PATH after /etc/gridlab.conf is sourced, it's used
- if MPLNATIVE_DIR is set, native C/Fortran compiler is used (MPLNATIVE_LIBS)
- if MPICH_DIR is set, \$MPICH_DIR/bin/mpicc is used

All variables defined in /etc/gridlab.conf should be also exported in this file. If programs or dynamic libraries required by Gridlab software are installed in non-standard directories, variables LD_LIBRARY_PATH and PATH should be set in /etc/gridlab.conf too.

In C calling conventions are well defined so it shouldn't be a problem to link together objects created by different compilers. This fact doesn't hold for C++ so authors of software that would link with Cactus should check whether their code is portable enough to be compiled using same compilers as Cactus.

Relevant Documents

[1] GridLab Quality Assurance Plan.

[2] GridLab C/C++/Java Coding Guidelines.

- [3] Datagrid configuration management and build conventions, <http://datagrid.in2p3.fr/d6.2/DataGrid-ConfMngmt-BuildConv.html>.
- [4] Datagrid Rules for Software, <http://marianne.in2p3.fr/datagrid/documentation/Developer-Guidelines.html>.
- [5] European DataGrid Developer s Guide, <http://edms.cern.ch/document/358824>.