

GT 4.2.0 Release Notes: GridWay

Table of Contents

1. Component Overview	1
2. Feature summary	1
3. Summary of Changes in GridWay	2
4. Bug Fixes	3
5. Known Problems	3
6. Technology dependencies	4
7. Tested platforms	4
8. Backward compatibility summary	5
9. Associated Standards	5
10. For More Information	6

<titleabbrev>4.2.0 Release Notes</titleabbrev>

1. Component Overview

The GridWay Metascheduler enables large-scale, reliable and efficient sharing of computing resources (clusters, computing farms, servers, supercomputers...), managed by different LRM (Local Resource Management) systems, such as PBS, SGE, LSF, Condor..., within a single organization (enterprise grid) or scattered across several administrative domains (partner or supply-chain grid).

2. Feature summary

Advanced Scheduling Capabilities	<p>GridWay implements several state-of-the-art Grid-aware scheduling policies, comprising <i>job prioritization</i> policies (fixed priority, urgency, share, deadline and waiting-time) and <i>resource prioritization</i> policies (fixed priority, usage, failure and rank).</p> <p>These policies are combined with:</p> <ul style="list-style-type: none"> • <i>Adaptive Scheduling</i>, to periodically adapt the schedule considering applications' demands and Grid resource characteristics. • <i>Adaptive Execution</i> to migrate running applications in terms of resource availability, capacity or cost, and new application requirements or preferences.
Transparent Grid Access	<p>GridWay interfaces infrastructures with different middleware stacks. With GridWay, users can access heterogeneous resources in a transparent way. For example, it can access resources configured with both GRAM pre web services and GRAM web services. It also permits the use of different grids with different software stacks, for example, Teragrid with Globus and EGEE with gLite.</p>
Flexible Deployment Capabilities	<p>GridWay supports multiple-user operation mode, and does not require additional middleware installation (apart from standard Globus services). Globus installation is not required in each end-user system.</p>

	GridWay allows different Grid deployment strategies, like Enterprise Grids, Partner Grids or Utility Grids.
Different Application Profiles	GridWay executes different Grid application profiles: <ul style="list-style-type: none">• Array (Bulk) jobs, for parameter sweep applications• DAG Workflows• Single-site MPI applications
Fault Detection and Recovery	GridWay is able to detect several problems that can occur when executing a remote job. It also implements mechanisms that make the execution more reliable. It can detect a remote system crash, a job failure (via the job exit code) or even a network disconnection (using the polling mechanism) and migrate the problematic job to another resource. GridWay also performs periodic saves of its state in order to recover from local failure.
Reporting and Accounting	GridWay provides detailed statistics of Grid usage. In this way, the Grid administrator can properly plan usage policies and forecast workload. In addition, these statistics can be used by the scheduler to predict (per user) Grid resource response time.
Standard Compliance	GridWay is an open-source project, flexible and completely based on standards to leverage its usability and interoperability. For example, users can describe their jobs using JSDL. Similarly, programmers can build grid enabled applications using the DRMAA standard.
User Interface	GridWay provides end-users with a familiar environment similar to that found on classical LRM systems. So GridWay CLI eases the adoption of Grid technologies.

3. Summary of Changes in GridWay

GridWay version shipped with GT4.2.0 is based on GridWay 5.4, a new stable version with increased functionality, including support to access the NorduGrid infrastructure, new bindings for DRMAA and support for DAGMAN workflows.

New to 4.2.0:

Integration with major Grid Infrastructures	Following the trend from previous releases, GridWay is capable of integrate with another major infrastructure: NorduGrid. Drivers for Execution and Transfer are included in this release.
New DRMAA bindings for scripting languages	Previous GridWay releases come with support for C and Java bindings implementing the DRMAA standard. Scripting languages are interesting from the point of view of rapid application prototyping among others. From dsa-research.org we believe this to be an interesting aspect, and that is why GridWay includes bindings to access the DRMAA standard from scripting languages such as Perl, Ruby and Python.
Support for DAGMAN workflows	There is an increasing interest from the scientific community to be able to run complex workflows on a Grid environment. An almost de-facto standard to

represent DAG workflows is Condor DAGMAN. In this development release GridWay includes a new workflow launcher (gwdag) that handles this kind of workflows.

4. Bug Fixes

- [Bug 5579](#):¹ CLI option to not detach gwd from terminal
- [Bug 5592](#):² gwd blocks when misconfigured sudo
- [Bug 5641](#):³ Sudoers configuration not allowing sudo to be run from a program
- [Bug 5663](#):⁴ Autotools configuration for database is broken
- [Bug 5677](#):⁵ Thread blocking occurs when recovering the scheduler
- [Bug 5724](#):⁶ Environment for Non-Wrapper based jobs
- [Bug 5734](#):⁷ Documentation doesn't reference the new DRMAA 1.0 binding
- [Bug 5737](#):⁸ The new threaded im mad for mds4 (gw_im_mad_mds4_thr) doesn't discover self contained hosts
- [Bug 5850](#):⁹ Bug in gwhost -f
- [Bug 5898](#):¹⁰ JSDL-Support needs Java 1.5+
- [Bug 5944](#):¹¹ Ganglia is needed in resources to schedule jobs to Fork job managers
- [Bug 6062](#):¹² Wrong parsing of NEW_REQS and NEW_RANK from stdout.wrapper
- [Bug 6081](#):¹³ gwdagman - bad overall workflow execution time

5. Known Problems

The following problems and limitations are known to exist for GridWay at the time of the 4.2.0 release:

5.1. Limitations

- [list limitations]

¹ http://bugzilla.globus.org/globus/show_bug.cgi?id=5579

² http://bugzilla.globus.org/globus/show_bug.cgi?id=5592

³ http://bugzilla.globus.org/globus/show_bug.cgi?id=5641

⁴ http://bugzilla.globus.org/globus/show_bug.cgi?id=5663

⁵ http://bugzilla.globus.org/globus/show_bug.cgi?id=5677

⁶ http://bugzilla.globus.org/globus/show_bug.cgi?id=5724

⁷ http://bugzilla.globus.org/globus/show_bug.cgi?id=5734

⁸ http://bugzilla.globus.org/globus/show_bug.cgi?id=5737

⁹ http://bugzilla.globus.org/globus/show_bug.cgi?id=5850

¹⁰ http://bugzilla.globus.org/globus/show_bug.cgi?id=5898

¹¹ http://bugzilla.globus.org/globus/show_bug.cgi?id=5944

¹² http://bugzilla.globus.org/globus/show_bug.cgi?id=6062

¹³ http://bugzilla.globus.org/globus/show_bug.cgi?id=6081

5.2. Outstanding bugs

- [Bug 5308](#):¹⁴ gwd doesn't recover AIDs and TIDs
- [Bug 5626](#):¹⁵ GridWay should offer the ability to start from a specified job ID
- [Bug 5718](#):¹⁶ Transfer Errors
- [Bug 5719](#):¹⁷ Timeouts for EM and TM Drivers
- [Bug 5722](#):¹⁸ Autotool for the new doc
- [Bug 5727](#):¹⁹ Improve DB checks in autoconf
- [Bug 5735](#):²⁰ Update development guide with the binding 1.0 for DRMAA JAVA
- [Bug 6003](#):²¹ socket remains open when connection to gwd failed

6. Technology dependencies

GridWay uses different Globus services to perform the tasks of information gathering, job execution and data transfer.

- GridWay depends on the following GT components for job execution:
 - GRAM: GridWay can interface with both GRAM 2 (pre web services) and GRAM 4 (web services)
- GridWay depends on the following GT components for data staging:
 - RFT
 - GridFTP
- GridWay depends on the following GT component for information gathering:
 - MDS
- GridWay relies on the Globus basic security infrastructure for authentication and authorization. On top of that, GridWay can use other Globus services and components to complement this infrastructure:
 - Delegation Service
 - MyProxy

7. Tested platforms

Tested platforms for GridWay:

¹⁴ http://bugzilla.globus.org/globus/show_bug.cgi?id=5308

¹⁵ http://bugzilla.globus.org/globus/show_bug.cgi?id=5626

¹⁶ http://bugzilla.globus.org/globus/show_bug.cgi?id=5718

¹⁷ http://bugzilla.globus.org/globus/show_bug.cgi?id=5719

¹⁸ http://bugzilla.globus.org/globus/show_bug.cgi?id=5722

¹⁹ http://bugzilla.globus.org/globus/show_bug.cgi?id=5727

²⁰ http://bugzilla.globus.org/globus/show_bug.cgi?id=5735

²¹ http://bugzilla.globus.org/globus/show_bug.cgi?id=6003

- GridWay builds successfully for the following platforms:
 - Linux
 - Tru64
 - Mac OS X
 - Solaris
 - Aix

In addition, GridWay has been tested with the major Grid infrastructures. Click the following links to find more information on how to use GridWay with [EGEE](#)²², [TeraGrid](#)²³, [OSG](#)²⁴ and [Nordug](#)²⁵.

8. Backward compatibility summary

GT 4.2.0 includes a new stable release, GridWay 5.4.0, that ships with functionality additions focused in extending usability for the end users. The following information regards compatibility with the previous stable version included in GT 4.0.7:

API changes since 4.0.7:

- Added DRMAA scripting language bindings for Python, Ruby and Perl.

CLI changes since 4.0.7:

- Added gwdag, a tool to run Condor DAGMAN compatible workflows.

Configuration interface changes since 4.0.7:

- None

9. Associated Standards

GridWay is an open project, flexible and completely based on standards to leverage its usability and interoperability. GridWay supports several standards developed by the [Open Grid Forum](#)²⁶, namely:

Distributed Resource Management Application API Working Group (DRMAA-WG)

The [DRMAA working group](#)²⁷ has developed an API specification for the submission and control of jobs to one or more Distributed Resource Management (DRM) systems. The goal is to facilitate the direct interfacing of applications to today's DRM systems by application's builders, portal builders, and Independent Software Vendors (ISVs). The Distributed Resource Management Application API (DRMAA) provides a generalized API to distributed resource management systems (DRMSs) in order to facilitate integration of application programs.

The scope of DRMAA is limited to job submission, job monitoring and control, and retrieval of the finished job status. DRMAA provides application developers and distributed resource management builders with a programming model that

²² <http://www.gridway.org/documentation/stable/egeehowto>

²³ <http://www.gridway.org/documentation/stable/tghowto/>

²⁴ <http://www.gridway.org/documentation/stable/osghowto/>

²⁵ <http://www.gridway.org/documentation/stable/nghowtoo/>

²⁶ <http://www.ogf.org>

²⁷ <http://drmaa.org/wiki/>

enables the development of distributed applications tightly coupled to an underlying DRMS. DRMAA preserves flexibility and choice in system design.

GridWay provides support for the DRMAA API (JAVA and C bindings) to develop distributed applications. OGF document GFD.104: GridWay DRMAA 1.0 Implementation - Experience Report²⁸ provides details of the implementation.

Job Submission Description Language (JSDL-WG)

The goal of the JSDL working group²⁹ is to produce a language that describes the requirements of jobs for submission to Grids. JSDL 1.0, published as OGF recommendation GFD-R-P.056³⁰ is an XML-based language that focuses mainly on computational jobs. The JSDL-WG is working on extending this language to address a wider class of jobs, including Web service invocations.

Last GridWay release provides support for JSDL, the POSIX Application profile and HTC Profile schemas can be used to describe jobs.

10. For More Information

Click here³¹ for more information about this component.

²⁸ <http://www.ogf.org/documents/GFD.104.pdf>

²⁹ <http://forge.gridforum.org/sf/projects/jsdl-wg>

³⁰ <http://www.ggf.org/documents/GFD.56.pdf>

³¹ <http://www.gridway.org>