



Grid(Lab) Resource Management System (GRMS) (white paper draft)

The GridLab Resource Management System (GRMS) [1] is an open source meta-scheduling system, developed under one of the leading grid projects in Europe, called GridLab (IST-2001-32133). GRMS allows developers and virtual organization (VO) administrators to build and deploy resource management systems for large scale distributed computing infrastructures, at level of an enterprise, VO or globally. It gives more abstract view of low level and complex Grid technologies. Based on dynamic resource selection, mapping and advanced scheduling methodology, combined with feedback control architecture, it deals with dynamic Grid environment and resource management challenges, e.g. load-balancing among clusters, work-load systems, remote job control or file staging support. Therefore, the main goal of the GRMS is to manage the whole process of remote job submission and control to various batch queuing systems, clusters systems or resources.

GRMS has been designed as an independent set of components for resource management processes that can take advantage of various low-level Core Services as well as various middleware services, e.g. GridLab Authorization Service, Replica Management Service, File Movement Service [3] or Adaptive Services. All these services working together provide a consistent, adaptive and robust grid middleware layer which fits dynamically to many different distributed computing infrastructures by improving security mechanisms or data management. GRMS is deployed now on the GridLab development testbed, and manages lots of machines from Europe and US.

Technically speaking, GRMS is a persistent service within the Tomcat/Axis environment and it supports Grid Security Infrastructure by providing the GSI-enabled web service interfaces for all clients, e.g. portals or applications, and thus can be integrated with any other middleware grid environment. Basic components of the GRMS have been developed entirely in Java and thus could be installed on various operating systems and resources. By using GSI-enabled Web Service interfaces the GRMS is able to communicate with portals (an example GRMS portal has been developed based on open-source GridSphere portal framework [5] and is available together with v1.8.0). Thus, the access for both end users and administrators can be done via graphic web interfaces or even from mobile devices.

One of the main assumptions for the GRMS is to perform remote job control and management in the way that satisfies both, Users (Job Owners) and Administrators (Resource Owners) and meet their requirements and preferences. Therefore, internal GRMS components allow developers and administrators build and define various resource management strategies, such as FIFO/LIFO, Round-Robin or matchmaking. Since the v1.8.0 all information and actions performed by on resources the GRMS are stored efficiently in a database for accounting purposes and an administrative support (e.g. for building fair-sharing or policy based resource management strategies).

Table 1. Main features of some existing meta-scheduling systems on top of Globus Core Services.

Features	GRMS v1.8.0	GRMS v2.0.0a	Condor-G	Platform CSF
Open Source	yes	Yes	yes	yes (*)
Support for Globus 2.X	yes	Yes	yes	no
Support for Globus 3.X	no	no (**)	no	yes
WSRF support	no	?	?	?
GSI-enabled Web Service interface	yes	Yes	no	yes
Integration with Authorization Services	yes	Yes	no	yes
Interfaces for Web Service application	yes	Yes	no	no

Command-line interface	yes	Yes	yes	yes
Web/mobile interface	yes	Yes	no	no
MPI support	yes	Yes	no	no
Java application support	yes	Yes	yes	yes
Work-flow experiments	no	yes (***)	yes (***)	no
User-level checkpointing	yes	Yes	no	no
Basic Scheduling Strategies	yes	Yes	yes	yes
Advanced Scheduling Strategies	no	Yes	no	yes
Reservation Support	no	Yes	no	yes
Accounting Database	yes	Yes	flat file	?
Access To outside reliable file transfer services	yes	Yes	no	no
Access to outside replica services	yes	Yes	no	no
Access to Outside Information Services (Dynamic Resource Discovery)	yes	Yes	no	yes

(*) There are some restrictions to the Platform CSF Open Source License

(**) Alfa release of GRMS components for Globus 3.X are planned

(***) GRMS v2.0.0 will support DAG based work-flows (similar to DAGMan from Condor) as well as PetriNet based work-flows which allow to construct more dynamic and complex computation experiments.

Table 1 above summarizes features of leading meta-scheduling systems available in March 2004, namely GRMSv1.8.0, GRMSv2.0.0 (alfa release), Condor-G and Platform CSF. It is worth to notice that a stable version of the GRMSv2.0.0 will include a bunch of new resource management strategies, complex work-flow support as well as basic reservation techniques. Also bug fixes and GRMS's performance improvements are planned.

First releases of GRMS (the last stable release is v1.8.0) are based on a relatively stable Globus 2.4 release and use Globus Core Services deployed on resources, in particular GRAM, GridFTP and GRIS/GIIS services [4]. Therefore, one needs at least all these Core Services being installed and configured correctly on all resources and front ends that GRMS is supposed to manage. GRMS v1.8.0 is under quality software review process now and all sources will be available for public in May 2004. A final stable release of GRMS v2.0.0 software together with technical guides and docs are due in November 2004 (see Table 1 for more information). Due to the fact that GRMS components are independent from underlying low level solutions we have started also some tests with Globus 3.0/3.2 solutions. However, since WSRF has been announced, we are keen in the future on moving with the GRMS towards specifications/solutions that have a commercial support from big IT companies and computer vendors.

FOR MORE DETAILS PLEASE CONTACT:

Jarek Nabrzyski, GridLab Project Manager: naber@man.poznan.pl

References:

[1] www.gridlab.org/WorkPackages/wp-9/

[2] www.gridlab.org

[3] www.gridlab.org/WorkPackages/wp-8/

[4] www.globus.org

[5] www.gridlab.org/WorkPackages/wp-4/

[6] www-106.ibm.com/developerworks/library/ws-resource/