AstroGrid-D

Deliverable D6.3

Architecture of generic grid-enabled Monitoring & Steering Methods in AstroGrid-D Applications[1]

Deliverable	D6.3
Authors	Thomas Radke (AEI)
Editors	Thomas Radke (AEI)
Date	22 December 2006
Document Version	1.0.0
Current Version	1.0.0
Previous Versions	

A: Status of this Document

Approved document for project deliverable D6.3.

B: Reference to project plan

This deliverable document refers to the task TA VI-III *"Architektur generischer gridfähiger Überwachungs- und Steuerungsmethoden"* and milestone M12 of work package WP-6 in the project plan.

[1] This work is part of the D-Grid initiative and is funded by the German Federal Ministry of Education and Research (BMBF).

C: Abstract

This document describes the general architectural design of generic grid-enabled monitoring & steering methods in AstroGrid-D applications.

D: Changes History

Version	Date	Name	Brief summary
0.1.0	20 November 2006	Thomas Radke	Working Draft Creation
0.1.1	30 November 2006	Thomas Radke	Sketched out Architectural Picture
0.1.2	13 December 2006	Thomas Radke	Finalised Architectural Picture and Design
0.1.3	15 December 2006	Thomas Radke	Added Design of Monitoring/Steering Methods for Cactus
1.0.0	22 December 2006	Thomas Radke	Document finally approved

E:

Contents

1 Introduction

In the previous deliverable documents of Working Group VI (D6.1: *"Existing Monitoring & Steering Functionality in AstroGrid-D Applications"* [1]; D6.2: *Requirements on grid-enabled Monitoring & Steering Methods in AstroGrid-D Applications* [2]), a thorough analysis was done to determine concrete requirements of AstroGrid applications on application monitoring and steering capabilities in a Grid context. Each AstroGrid use case (see [3] for the full overview) was surveyed specifically in respect to already existing monitoring/steering functionality and of necessary future additions to this functionality when such applications are finally to run in a Grid environment such as D-Grid.

Based on the derived list of detailed requirements on application monitoring & steering as described in [2], this deliverable document D6.3 now defines the general architectural picture for an implementation of the necessary grid-enabled monitoring/steering functionality for AstroGrid applications. This picture also illustrates the interaction of monitoring/steering methods with other software components and Grid services in AstroGrid which are currently being developed by other working groups.

As the next milestone in the work plan of Working Group VI, a prototype implementation of such monitoring/steering methods will be developed. This implementation will have limited monitoring & steering functionality for a selected subset of AstroGrid use cases and is intended to be deployed not on the full AstroGrid testbed yet but only on a small set of specific resources.

For one AstroGrid use case – Cactus – the requirements specified were very detailed and well known in advance, therefore a concrete implementation design of grid-enabled monitoting and steering methods for Grid simulations is also sketched out in this document to serve as an example for other use cases.

At the end of this implementation period, a deliverable document D6.4 will be presented which documents the developed software components in this prototype and provides a detailed test report for the selected AstroGrid applications.

2 Architecture

An overview of the general architecture for generic grid-enabled monitoring & steering methods in AstroGrid is sketched out in figure 2.

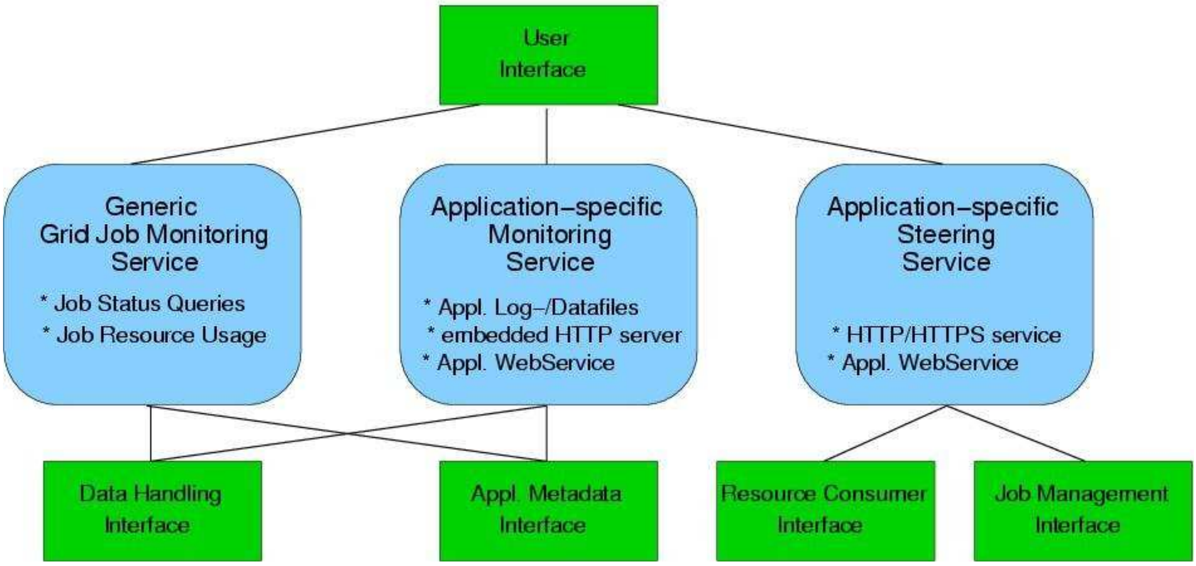

Figure 2: Architectural overview of application monitoring and steering components and their interfacing to other AstroGrid services (taken from [4])

Following the classification of use case requirements in [2], grid-enabled monitoring and steering functionality for AstroGrid applications is separated into three different Grid services:

1. Generic Grid Job Monitoring Service
2. Application-specific Monitoring Service
3. Application-specific Steering Service

Each service in this picture should be seen not as a single functional entity by itself but rather as a collection of sub-services with similar functionality.

2.1 Generic Grid Job Monitoring Service

This service provides basic monitoring functionality for any type of job running on the Grid. Just like starting a Grid job, basic monitoring capabilities are provided by means of a Grid job management service (as being developed in WG-V). Monitoring data would include information about the job's current status (*queued, running, stopped, terminated* etc.) and about its resource usage (eg. CPU load, memory/disk usage, network I/O). Such information will either be stored (and periodically updated) directly as metadata in an AstroGrid information service (as developed in WG-II), or

alternatively this service is asked to return the contact address of the local job management system to query where the job is running on. In both cases, monitoring metadata is queried through a generic application metadata interface.
In addition to job run status and resource usage information, the generic Grid job monitoring service will also offer access to `stdout/stderr` logfiles that are generated by the job. This access is provided by means of a grid file management service (WG-III) which is contacted through a generic data (file) handling interface.

The presentation of generic job status monitoring information will be accomplished via appropriate user interfaces, eg. command line tools or an application-specific user web portal (as a monitoring portlet for the GridSphere portal framework, as developed in WG-VII).

Job status monitoring is done outside and independent of the actual Grid job – the application itself is not involved and doesn't need to be instrumented for that purpose.

2.2 Application-specific Monitoring

Besides general job status information, users also want to be able to obtain other monitoring data. Often such data is written to application-specific logfiles which are output by the application, hence users need interactive access to such logfiles. Although this functionality is very similar to providing `stdout/stderr` logfile access via the generic grid job monitoring service (as described above), an application-specific monitoring service is needed which *I)* has some internal knowledge about the name and location of logfiles for the specific type of application the job belongs to (this information will be queried from an information service (WG-II) through a generic application metadata interface), or *II)* offers a flexible user interface (WG-VII) for the scientist to browse a Grid job's output directory and its list of application-specific output files.
In both cases standard grid file management functions (developed in WG-III) are called through the generic data handling interface in order to fetch the application-specific logfiles.

While for this type of application-specific monitoring the application code doesn't need to be instrumented either, the users will need to provide additional information in the corresponding job description about the logfile naming schemes for this use case so that it can be queried automatically (case I), or they will want to browse a remote job's output directory and logfiles interactively (case II).

Other, even more application-specific monitoring services (as required by the Cactus use case UC5) will be implemented as a Web service (eg. an embedded HTTP webserver [1][2]) directly into the application code (see section 3).

2.3 Application-specific Steering

A number of AstroGrid use cases already implement a simple method of application-level steering by offering a user-triggered graceful termination of running jobs. This is achieved by placing a file of fixed (hence application-specific) name in the job's output directory. The job will periodically check for the existence of such a file and stop once it has been found.
To make this type of application-level steering grid-aware, users need to be able to create such trigger-termination files remotely: either manually via a remote file browser (using WG-III's grid file management services embedded in an appropriate user interface such as an application-specific

web portal developed by WG-VII), or automatically by adding appropriate logic into an application-specific steering service which queries the job's output directory from the grid job management service (via a generic job management interface) and then starts a simple grid job on its own (through the resource consumer interface) in order to create the trigger-termination file there.

Other, even more application-specific steering services (as required by the Cactus use case UC5) will be implemented as a Web service (eg. an embedded HTTP webserver [1][2]) directly into the application code (see section 3).

3 Design of Monitoring & Steering Methods for Cactus

Due to AEI's former involvement in other Grid activities and successful contributions to the Cactus code in order to make it grid-aware, the requirements on application-level monitoring and steering methods for the AstroGrid use case Cactus were well known in advance already and hence could be precisely described in our use case analysis assessment [3] during the start phase of our project.

Derived from this advanced knowledge was a design of grid-aware monitoring and steering methods which will be implemented in Cactus during the AstroGrid project. This design is sketched out in greater detail in the following section.

Although the proposed monitoring and steering functionality is taylored for the specific needs of Cactus, its design should be generic and flexible enough to be incorporated in other AstroGrid use case as well. For this purpose, the design draft has been presented at various AstroGrid project meetings ([5, 6]). Especially due to its very close connection to grid-enabled user interfaces such as a Cactus web portal, it was also driving the requirements specification of WG-VII ([7]).

3.1 Cactus Monitoring in the Portal

This refined use case for Cactus should allow physicists to better keep track of all the Cactus simulations they are running on various resources. For this purpose, basic information about each job is sent to and archived in an AstroGrid information service (WG-II), and can then be queried by physicists at any later time through a Cactus user portal.

A minimum set of information desribing a Cactus job would include the following static metadata:

- start time of the job
- name of the resource where the job was started
- login name of the job's owner
- executable name and version
- parameter filename and contents

In addition, dynamic metadata such as periodic updates of the simulation's progress (current iteration number and/or physical time, event notifications, or termination conditions) could be included optionally.

To design this type of simulation monitoring, the application code needs to be extended with following functionality:

1. an RDF schema to generically describe all possible types of simulation metadata
2. a module to gather static simulation metadata, translate it into RDF (using the defined schema), and send it off to an external information service
3. a generic API to be used by other code modules in Cactus to announce their own type of (dynamic) simulation metadata

Closely related to this metadata functionality on the application side is the user interface in the form of a Cactus user web-portal based on the GridSphere portal framework (developed in WG-VII). A Cactus metadata management portlet will provide functionality to query Cactus simulation data (based on the RDF schema defined for Cactus metadata), offer different views on the various types of job information, and present it in a flexible user-friendly way. These requirements on the Cactus user portal have been specified in [7].

3.2 Online Simulation Monitoring & Steering

Monitoring Cactus simulations in a portal is designed to be decoupled from the actual application – each job acts as an information producer, sending its simulation metadata to the information service for archiving; there is no feedback from a portal (acting as an information consumer) back to a given job (which usually has finished already).

For active Cactus simulations, a different design is necessary to provide online monitoring and also runtime steering functionality: physicists should be able to directly connect to a running job and *login* to the simulation (with proper user authentication/authorisation). They can then interactively monitor and (given the appropriate user permissions) also steer the simulation online.

This design is based on the already existing monitoring/steering functionality implemented in the standard Cactus application toolkit in the form of a web-server module built into the application code. The approach was described in detail in [1] and [3], and will be picked up in AstroGrid as the basic design of grid-enabled online application monitoring & steering methods for Cactus simulations.

As illustrated in [2], the existing design of the integrated Cactus web-server will need to be enhanced by

- adding GSI-based user authentication/authorisation
- using a secure network protocol such as HTTPS for the client-server connection
- providing a generic and robust method to contact simulations which are running behind a firewall on a cluster's internal node with no direct access to the internet

All of the above functionality required to make Cactus simulation monitoring and steering grid-aware will be based on standard Grid technologies and designed generically so that they can be reused in other AstroGrid use cases as well.

F: References / Bibliography

References

[1] Thomas Radke: Existing Monitoring & Steering Functionality in AstroGrid-D Applications. Comparison Study, Work Group VI deliverable document D6.1, AstroGrid project;
http://www.gac-grid.org/project-documents/deliverables/wp6/WG6_D6_1.pdf

[2] Thomas Radke: Requirements on grid-enabled Monitoring & Steering Methods in AstroGrid-D Applications. Requirements Specification, Work Group VI deliverable document D6.2, AstroGrid project;
http://www.gac-grid.org/project-documents/deliverables/wp6/WG6_D6_2.pdf

[3] AstroGrid-D Use Case inquiry. AstroGrid Intranet webpage;
http://mintaka.aip.de:8080/lenya/intranet/live/project-documents/usecases.html
http://mintaka.aip.de:8080/lenya/intranet/live/project-documents/UseCasesNew.html

[4] AstroGrid-D Architecture. AstroGrid project webpage;
http://www.gac-grid.org/project-documents/architecture.html

[5] Thomas Radke: Status WG6 and Informationservice for Cactus. Presentation at the 4th AstroGrid Project Meeting, 24./25. July 2006, ZAH Heidelberg;
http://www.gac-grid.org/project-overview/events-meetings/meetings/meetingzib-1/wg6-status-report.pdf

[6] Thomas Radke: Cactus Metadata Management. Presentation at the 5th AstroGrid Project Meeting, 14./15. November 2006, MPE Garching;
http://www.gac-grid.org/project-overview/events-meetings/meetings/meeting-MPE/cactus-metadata-management.pdf

[7] Oliver Wehrens: Requirement analysis for specific components and services of the Astro community for the GACG portal. Work Group VII deliverable document D7.2, AstroGrid project;
http://www.gac-grid.org/project-documents/deliverables/wp7/M2.pdf