

INTEROPERABILITY AND INTEROPERATION OF GRID-INFRASTRUCTURES

D. Karpenko

*Bogolyubov Institute for Theoretical Physics of NAS of Ukraine, Metrologichna str. 14b, Kyiv, Ukraine,
Email: dkarpenko@bitp.kiev.ua*

ABSTRACT

The report contains information about the problems of interoperability and interoperation of grid-infrastructures and about different approaches to overcome these problems. Developed and developing approaches are considered. The world's experience in this area is evaluated, the most notable projects are considered, the architecture and peculiarities of grid and other software used in these projects are presented. The information about universal software, providing grid-interoperability (portals, resource brokers, meta-schedulers), their architecture, peculiarities, ways to use is provided. Some material is dedicated to NAS of Ukraine's work and research in solving the problems of interoperability and interoperation of grid-infrastructures.

Keywords: grid-infrastructure, interoperation, interoperability, grid-software, grid-computing

1 INTRODUCTION

Since 2007 National academy of sciences of Ukraine owns grid-infrastructure, based on NorduGrid ARC middleware. At the time of writing 13 scientific and educational ukrainian institutes provide their resources to this infrastructure. As the collaboration between NASU and european scientific organizations evolves, the need for replacing ARC software by gLite middleware, which is used in most european grid-projects, becomes obvious. And many institutes, that participate in NASU grid-infrastructure, own resources which are used for testing purposes or for participating in international scientific projects. Such resources are controlled by middleware, which is different from ARC, and thus they can't be used in the main grid-infrastructure. So, NASU faces the problems of replacing the middleware of the main grid-infrastructure and unifying different resources, controlled by different middleware, into united (from user's point of view) grid-environment.

2 THE WAYS OF PROVIDING GRID-INTEROPERABILITY AND GRID-INTEROPERATION

When it comes to unifying several grid-infrastructures, the basic terms here are «interoperability» and «interoperation». Interoperability is the grid-systems' ability to exchange information and use the information that was exchanged. Interoperation is the simultaneous using of several interoperable grid-systems. Interoperability is the first and mandatory step to interoperation. [1]

At the time of writing 4 approaches for providing interoperability are commonly used.

User-case scenario. The user, who is registered in all needed grid-infrastructures, manually sends a task to proper infrastructure, controls the task's flow, receives, processes and, maybe, combines its results.

Using in parallel. This is the modified version of user-case scenario. The difference is in the presence of a single entry point to several grid-infrastructures. This simplifies user's work, but nevertheless the user has to make all the needed actions manually. This approach, for example, is used in Bogolyubov institute for theoretical physics of NASU to work with the main NASU ARC [2] -based grid-infrastructure and AliEn [3] -based grid, which is the part of CERN ALICE project.

Gateway is the non-manual approach to interoperability. Gateway is usually a program module, which works on top of grid-middleware, can convert a task's description from this middleware's format to another middleware's format, send the converted task to another middleware, control its flow and receive results. Gateway uses the interface of underlying middleware, so, from user's point of view, it's possible to work with several middlewares using the system of command of just one middleware. Usually gateway doesn't have features for advanced planning or combining of results. Its work is just to put the received task to another middleware.

Adapters and translators. Unlike gateway, adapters and translators don't need the spesific middleware. Adapter is usually the program, which can communicate with different grid-middlewares, but doesn't require them to be present. Adapters have their own user interface and task description format and can control many aspects of task's execution

in grid-infrastructure. The task description is converted from one format to another by translators. Translator can be the part of adapter, or it can be built in grid-middleware (for example – VOBOX in AliEn middleware). Adapter can have features for advanced planning or combining several tasks in one logical workflow. Grid-portals are the good example of web-service adapters.

All mentioned approaches had been developed before the work for creating the unified grid-interoperability standard was started. Standardization is considered to be the best way for interoperability, but it's still in development state. The most notable organization in this field is OpenGrid Forum (OGF), that has developed Open Grid Service Architecture (OSGA) standard. The first version of this standard was released in 2004 and became the most popular standard among grid-software developers. The last version of this standard is 1.5 [4]

Several organization of grid-software developers work on interoperability independently. For example, in 2005 GIN [5] project has been started. This is the collaboration of developers from the biggest european grid-organization EGEE [6] and american grid-organization OSG [7]. The project participants work on approaches for exchanging information between currently used infrastructures. Adapters and translators have been chosen as the best solution for interoperability problems. The developers have achieved results not only in interoperability, but they also have developed interoperation between EGEE and OSG resources. Among other results of this project the interoperability AliEn - ARC, AliEn – LCG – gLite worth mentioning.

3 NASU WORK IN GRID-INTEROPERATION

Taking into account the experience of GIN project, it has been chosen in NASU to use adapters and translators for solving interoperability problem. The adapter with built-in resource broker, scheduler and convenient user-interface was needed.

The first step in solving interoperability problem was the interoperability between the main ARC-based grid-infrastructure and Globus [8] -based test sites of Bogolyubov institute for theoretical physics. GridWay [9] was chosen as an adapter that can help to provide such interoperability. This metascheduler can work only on top of Globus middleware, so it's not the pure adapter, but rather adapter with some Globus-based gateway features. GridWay has built-in translator for converting task's description from its own format to RSL language, which is used in Globus. GridWay can also send the task to several other middlewares, but when NASU started to work with GridWay it couldn't communicate with ARC middleware.

GridWay's architecture consists of the core and the loadable modules [10]. It's illustrated by the next figure.

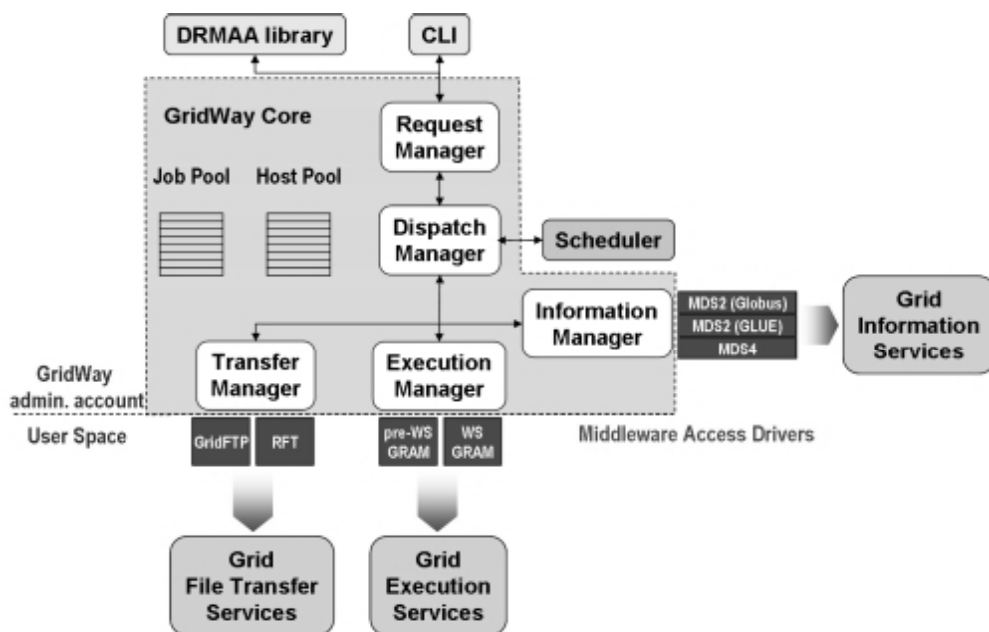


Figure 1. GridWay architecture

The core controls tasks and using of available resources. Among the loadable modules there are scheduler and middleware access drivers. The latter are used to communicate with the specific middleware.

Middleware access drivers for ARC-software needed to be created in order to provide the interoperability between Globus and ARC. GridWay has 3 MADs for each middleware: informational, which receives information about available resources; transport, which is needed to perform actions with remote files and directories and to transfer data to/from remote resources; execution, which sends tasks into grid-infrastructure, controls its flow and deletes task if it's needed. All 3 MADs for ARC have been developed. Every MAD performs the same functions as its Globus counterpart, but has some specific features because of ARC software peculiarities. For example, informational and transport system of ARC are basically the same as the corresponding systems in Globus, so standard Globus MADs just have been modified to work with ARC; but execution system of ARC has almost nothing in common with Globus execution system, so execution MAD required to write the code to be written almost from scratch.

Informational ARC MAD is written in Linux shell language; it uses `ldapsearch` shell command to find available resources by requesting information from LDAP-trees on informational servers. Another 2 MADs are written in C. Transport MAD uses `globus_ftp_client_*` functions from Globus API to create and delete remote directories and to transfer data to/from there. All these actions are allowed only if the user, on behalf of whom GridWay manipulates data on the remote resource, has the appropriate rights. Execution MAD uses `globus_ftp_control_*` functions to send the task into the infrastructure, renew the task's credentials or delete the task if it's needed. This MAD uses functions for working with GridFTP service, because ARC controls its tasks using GridFTP (by utilization of specific port and the number of virtual directories). This MAD is also used for task flow control by communicating with information system on the remote resource using `ldap_search` function.

Developed MADs don't require any additional software and will work on any resource where GridWay is already working. The main disadvantage of the MADs is the fact that they required minor changes in the core. So before MADs installation it is needed to reinstall GridWay with patched core. Another disadvantage of the MADs is their current inability to work with ARC services that use not standard network ports.

Test mode exploitation showed that BITP users can, using GridWay metascheduler, send their tasks into the main NASU grid-infrastructure and into Globus-based BITP grid.

The MADs might be used by GridWay developers in the next versions of GridWay.

The current version of MADs is available at <http://grid.bitp.kiev.ua/gridway>

4 CONCLUSION

After evaluating the world's experience in grid-interoperation area, NASU developers have modified GridWay metascheduler and made it compatible with NorduGrid ARC software, giving BITP users the ability to send their tasks into the main NASU grid-infrastructure and into Globus-based BITP grid. Now NASU continues to improve GridWay ARC MADs functionality and install GridWay metascheduler on other NASU resources.

5 PREFERENCES

- [1]. Laurence Field Markus W. Schulz "Grid Interoperability: The Interoperations Cookbook" 2007.
- [2]. <http://www.nordugrid.org>
- [3]. <http://alien.cern.ch>
- [4]. <http://www.ogf.org/documents/GFD.80.pdf>
- [5]. <http://egee-sa3.web.cern.ch/egee-sa3/gin-cg.html>
- [6]. <http://www.eu-egee.org/>
- [7]. <http://www.opensciencegrid.org/>
- [8]. <http://www.globus.org/>
- [9]. E. Huedo, R. S. Montero, I. M. Llorente. "A modular meta-scheduling architecture for interfacing with pre-WS and WS Grid resource management services". *Future Generation Computing Systems* 23 (2): 252-261, 2007
- [10]. E. Huedo, R. S. Montero, I. M. Llorente. "The GridWay Framework for Adaptive Scheduling and Execution on Grids". *Scalable Computing - Practice and Experience* 6 (3): 1-8, 2005