

UNIVERSITY OF WESTMINSTER

**WestminsterResearch**<http://www.wmin.ac.uk/westminsterresearch>**Service-oriented production grids and user support.****Gabor Terstyanszky<sup>1</sup>****Tamas Kiss<sup>1</sup>****Thierry Delaitre<sup>1</sup>****Stephen Winter<sup>1</sup>****Peter Kacsuk<sup>2</sup>****Gabor Kecskemeti<sup>2</sup>**<sup>1</sup> School of Informatics, University of Westminster<sup>2</sup> MTA SZTAKI, Hungary

Copyright © [2006] IEEE. Reprinted from the Proceedings of the 7th IEEE/ACM International Conference on Grid Computing, Barcelona, Spain, September 28-29, 2006, pp. 323-324.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Westminster's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to [pubs-permissions@ieee.org](mailto:pubs-permissions@ieee.org). By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

---

The WestminsterResearch online digital archive at the University of Westminster aims to make the research output of the University available to a wider audience. Copyright and Moral Rights remain with the authors and/or copyright owners. Users are permitted to download and/or print one copy for non-commercial private study or research. Further distribution and any use of material from within this archive for profit-making enterprises or for commercial gain is strictly forbidden.

---

Whilst further distribution of specific materials from within this archive is forbidden, you may freely distribute the URL of the University of Westminster Eprints (<http://www.wmin.ac.uk/westminsterresearch>).

In case of abuse or copyright appearing without permission e-mail [wattsn@wmin.ac.uk](mailto:wattsn@wmin.ac.uk).

# Service-Oriented Production Grids and User Support

Gabor Terstyanszky<sup>1</sup>, Tamas Kiss<sup>1</sup>, Thierry Delaitre<sup>1</sup>, Stephen Winter<sup>1</sup>, Peter Kacsuk<sup>2</sup>, Gabor Kecskemeti<sup>2</sup>

<sup>1</sup>Centre for Parallel Computing, University of Westminster,  
115 New Cavendish Street, W1W 6UV, London United Kingdom

<sup>2</sup>MTA SZTAKI, Hungary  
<sup>1</sup>terstyg@wmin.ac.uk

**Abstract** – Currently several production Grids offer their resources for academic communities. These Grids are resource-oriented Grids with minimal user support. The existing user support incorporates Grid portals without workflow editing and execution capabilities, brokering with no QoS and SLA management, security solutions without privacy and trust management, etc. They do not provide any kind of support for running legacy code applications on Grids. Production Grids started the migration from resource-oriented Grids to service-oriented ones. The migration defines additional requirements towards the user support. These requirements include solving interoperability among Grids, automatic service deployment, dynamic user management, legacy code support, QoS and SLA-based brokering, etc. This paper discusses some aspects of the user support needed for service-oriented production Grids.

## I. INTRODUCTION

In recent years, as Grid computing has been able to show its potential benefits, both academic and non-academic (business, industry, etc.) communities have been adopting Grid computing. One of the reasons to use Grid computing has been the creation of a scalable infrastructure. Users require access to resources available within the community to execute vast computations or to handle a huge amount of data. Due to needs and technical capabilities, the Grid research addressed the creation of scalable Grid environments with large resource pools. This research has produced a set of Grid components, middleware and tools that rely on widely accepted standards. They allow creation of Virtual Organisations (VOs) involving large communities. These VOs incorporate and manage resources distributed among several administrative domains to allow users to share these resources.

## II. PRODUCTION GRIDS

There are several production Grids, like the TeraGrid and the Open Science Grid in the US, or the EGEE Grid and the UK National Grid Service in Europe, that already provide reliable production quality access to computational and data resources for the academic community. They are used to process a large amount of data using specific computing environments communicating through dedicated high-speed networks. All these Grids were set up as resource-oriented Grids and all of these but EGEE are built on GT2. They consider moving towards a service-oriented architecture using either gLite or GT4. They also want to make Grid available not only for the academic communities but for business and industry. The migration from resource-oriented Grid to service-oriented ones raises among others the following issues:

**Interoperability among Grids.** It is expected that Grid services may be deployed onto different hosting environments, allocated to diverse service domains, follow multiple policy models and enforce different security mechanisms. Grids as heterogeneous environments raise interoperability issues in both Grid service deployment and invocation. Two scenarios illustrate the interoperability problems. In the first, a Grid service is copied from one Grid domain and deployed onto a new one. In the second, service requests are submitted to Grid service instances deployed in different Grid domains. To solve these interoperability problems four levels of interoperability should be addressed: data (or information)-, job-, resource management- and security-level interoperability. The research team is addressing the job- (or workflow) level interoperability [1].

**Legacy code support.** There is a vast legacy of applications solving scientific problems or supporting business critical functionalities which should be migrated onto the Grid with the least possible effort and cost.

**Dynamic user management.** To access a Grid resource, the user's global identity should be converted to a local identity. This conversion may be based on either a permanent one-to-one mapping or on temporary mapping. The existing grid-mapfile based solutions are not scalable because they require frequent manual interaction to map identities. As a result, they are not suitable for production Grids with a large number of users.

**Automatic deployment.** Grid services may have multiple instances running on multiple computing nodes. It may happen that the deployed services are not available to execute new service requests. To avoid this situation Grid services should be deployed automatically on new computing nodes and be made available on demand. The automatic deployment may be represented by the following scenario. The abstract workflow graph created by the user is passed to a resource broker together with quality of service (QoS) requirements. The broker contacts an information service and tries to map different components of the workflow to different resources and pre-deployed services. If user QoS requirements cannot be met with the deployed services, or if the required service is not deployed on any of the resources, the broker contacts the automatic deployer to deploy the code. As the sites can belong to different Grids with different middleware, policy and security standards, the deployer should also resolve these interoperability problems.

**Resource and service monitoring** Grid resources and services in production Grids should be automatically monitored and

tested. Production Grids run thorough tests on their available resources on a regular basis to offer a high quality of service. However, none of these solutions are integrated with GT4 at the moment, and there are no probes that can directly test GT4-based services.

### III. PRODUCTION GRIDS AND USER SUPPORT

Currently, the research team is addressing the legacy code support, dynamic user management, automatic deployment and monitoring as part of the user support for service-oriented Grids.

**Legacy Code Support.** The Grid Execution Management for Legacy Code Architecture (GEMLCA) [2] was developed by University of Westminster to enable legacy code programs written in any source language (Fortran, C, Java, etc.) to be easily deployed as a Grid Service without significant user effort. The current GEMLCA implementation is based on GT4. GEMLCA is also integrated with the P-GRADE Grid portal [3] providing a user-friendly Web interface to publish legacy codes as Grid services and to create, execute and visualise the execution of complex Grid workflows built from “gridified” legacy codes and Grid services. As GEMLCA is implemented as a GT4 Grid service it can be naturally integrated with GT4-based production Grids. Moreover, through the P-GRADE portal, even current GT2-based Grids can be extended with GT4 GEMLCA resources bridging different Grid generations and providing seamless integration for Grid users from the same user interface. This way GT4 GEMLCA services can be deployed for current GT2-based production Grids assisting the transition of these Grid systems towards service-oriented Grids.

**Dynamic user management.** To have a scalable solution production Grids require a dynamic and flexible user management. As part of the user management, they need a dynamic and fine-grained authorisation mechanism to control users’ access to resources. This authorisation mechanism could be implemented through dynamic account management and identity mapping. To provide a scalable user management GEMLCA was integrated with the Workspace Management Service using its identity mapping and dynamic account pooling features. The first feature maps Grid certificates into local accounts at run-time without manual intervention, while the second feature provides local accounts on demand from a pool of accounts.

**Automatic deployment [4].** An Automatic Deployment Service (ADS) is being developed at the University of Westminster. It is built on the Automatic Deployment Architecture (ADA). To migrate a service, ADS is contacted, which queries the information system to access site descriptions, and also generates the description of the service to be migrated. The classifier module tests the description of the service against the site descriptions, and generates a set of sites that are capable of hosting the service. Next, the dependency checker investigates the capabilities of the selected sites. Based on the information received from the dependency checker the comparator prepares some metrics (cost and time requirements of the deployment based on the descriptions), and selects the site with the lowest deployment cost. To make the new site compatible with the old one the dependency installer prepares several installation scripts and

environment configuration files and setup scripts. The deployer prepares a sandbox on the new site to separate the execution of the service from others. The deployer interfaces with the actual sandboxing technique to create a new sandbox, and then the installation scripts are executed in it. The deployer notifies the source site and negotiates the transfer of the service between the sites.

**Monitoring [5].** The GEMLCA was extended by the Monitoring Toolkit (GMT) to provide monitoring information based on probes checking the status of GEMLCA resources. The GMT is based on MDS4 (Monitoring and Discovery System). MDS4 is capable to collect, store and index information about resources, respond to queries concerning the stored information using the XPath language, and control the execution of testing and information retrieval tools. Using the GMT, system administrators are automatically alarmed when a test fails and can also request the execution of any test on-demand. The GMT also assists P-GRADE portal users when mapping the execution of workflow components to resources by offering only verified Grid resources when creating a new workflow or when rescuing a failed one.

### IV. CONCLUSIONS AND FURTHER WORK

Existing production Grid systems are based on second generation Grid technology and supporting only the academic community. However, transition to service-oriented middleware and extending the usage of these Grids towards businesses and industry are the next inevitable steps for the widespread take-up of Grid computing. This paper described how the current GEMLCA architecture has to be extended in order to support these next generation Grids. Besides the currently existing functionalities of GEMLCA that has already been offered at production level, we concentrated on the latest ongoing research and development work that extends the architecture with dynamic user management, automatic service deployment, resource monitoring and availability prediction. Most of these concepts are already implemented as prototypes and their full integration into the GEMLCA architecture is currently work in progress.

### REFERENCES

- [1] P. Kacsuk, T. Kiss, G. Sipos, Solving the Grid Interoperability Problem by P-GRADE Portal at Workflow Level, Submitted to the Grid-Enabling Legacy Applications and Supporting End Users Workshop, within the framework of the 15th IEEE International Symposium on High Performance Distributed Computing , HPDC’15, Paris, France, June 19-23, 2006
- [2] T. Delaittre, T. Kiss, A. Goyeneche, G. Terstyanszky, S.Winter, P. Kacsuk: GEMLCA: Running Legacy Code Applications as Grid Services, Journal of Grid Computing Vol. 3. No. 1-2. June 2005, Springer Science + Business Media B.V., Formerly Kluwer Academic Publishers B.V. ISSN: 1570-7873, pp 75-90
- [3] P. Kacsuk G. Sipos: Multi-Grid, Multi-User Workflows in the P-GRADE Grid Portal, Journal of Grid Computing, , Springer Science + Business Media Vol. 3 No. 3-4 Dec 2005, pp 221-238, ISSN: 1570-7873
- [4] K. Keahey, M. Ripeanu, K. Doering: Dynamic Creation and Management of Runtime Environment in the Grid, Workshop on Designing and Building Web Services, GGF9, Chicago, October 2000 Distributed Object Computing, University of Munich, 2001
- [5] I.C.Legrand, H.B.Newman, R.Voicu, C.Cirstoiu, C.Grigoras, M.Toarta, C. Dobre, MonALISA: An Agent based, Dynamic Service System to Monitor, Control and Optimize Grid based Applications, CHEP 2004, Interlaken, Switzerland, September 2004,