Portals and Friendly interfaces for job submission and control

Rafael Nunes
Grid Computing
Department of Computer Science
Faculty of Science of the University of Porto

1 Introduction

With the increase of scientific research, the analysis and computation of high amounts of data made necessary the use of distributed computing infrastructures (Cloud, Grid, etc.). But frequently it's hard for non expert researchers to use this computing infrastructures. The portals deploy the access to the computing infrastructures reducing the difficulty of use through a web interface, allowing users to submit and manage jobs in a friendly and fast way. This way, researchers can focus on their research and leave the low-level actions on the grid for the portal and grid middleware. Many portals offer advanced features like workflows, authentication and authorization procedures, creating a environment for scientific research for the community, as P-Grade and DIET.
Portals can be classified by their functionality, technology or institution. At the functionality level, portals can be purposed to a specific application (OurGrid, JETS), oriented to workflows (P-GRADE, Triana, Taverna, JETS), with ready applications (Gisela Science Gateway), of certain domains (caGrid, mammoGrid) or for generic use. At the technology level they are divided in first generation grid portals, made in traditional web pages, and second generation made with portlets. And per institution we have for example de TACC and the UCLA portal.

Outline The remainder of this article is organized as follows. Section 2 describes portals based on P-Grade. The DARE standard for middlewares

is explained in Section 3. Finally, Section 4 talks about a middleware to support Many-Parallel-Task workflows.

2 P-GRADE

The P-GRADE (Parallel Grid Run-time and Application Development Environment) grid portal is a portal based in workflows, which enables the user to submit and manage jobs in a grid. The portal makes it easier for all users, even non expert ones, to use the grid infrastructure by hiding all the low-level details of the grid with high-level graphical interfaces, while still making possible the execution of jobs in distributed computing infrastructures. In P-GRADE users can create tasks graphically using a workflow editor, in which the user can define a set of tasks that can be interconnected just with the use of the mouse. The most used workflow editors are Triana and Taverna. The authorization is made by the user uploading its respective X509 certificates to the server.

3 DARE

A Science Gateway is a community-developed set of tools, applications, and data collections that are integrated via a portal. A typical gateway provides authentication, authorization, the ability to submit user jobs and data management. All this capabilities shouldn't be tied to specific resources, there should be a role for a middleware framework and another for the user interface.
DARE (Distributed Application Runtime Environment) is a standards-based middleware framework for science-gateways. The two main aspects for the creation of science-gateways are the development of simple interfaces that capture the common science scenarios and the integration of this interface with the high-performance distributed computational (HPDC) infrastructure in a scalable and flexible way. The role of DARE is of integrating with the HPDC infrastructure.
Most features of the DARE framework are supplied by SAGA (Simple API for Grid Applications) and Pilot-Job. The SAGA components provide the distributed management capabilities like job submission and the Pilot-Job provides the support for dynamic execution. This way, DARE allows devel-

opers and scientists to focus on their scientific applications instead of focusing on using HPDC infrastructures as well as making it easier to support application-level constructs.

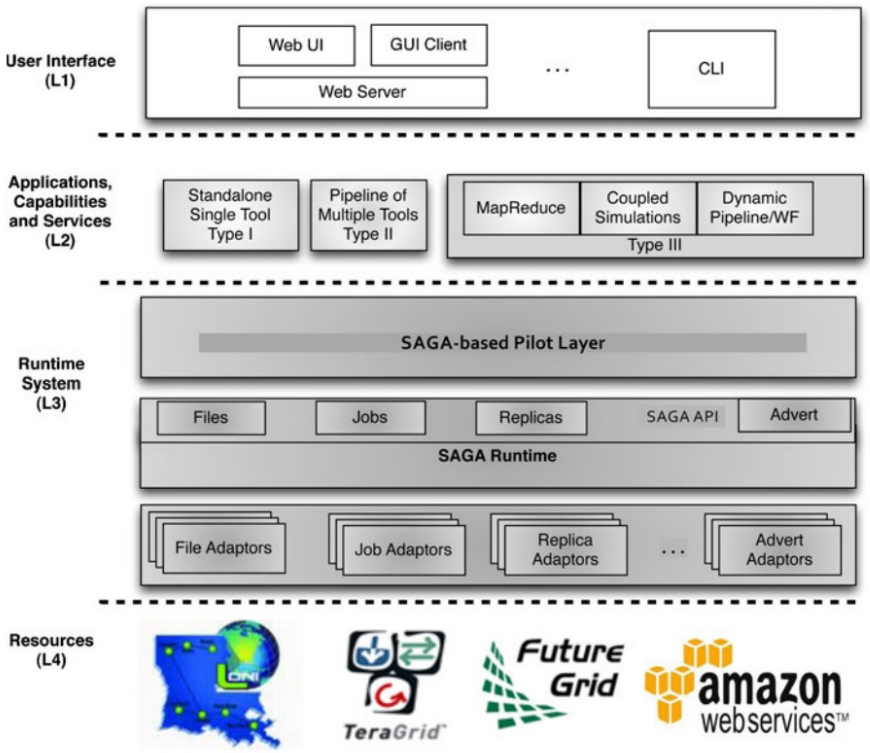

Figure 1: Architecture of a DARE-based gateway. The primary layers of DARE are in grey.

The Level L1 contains the interface. The logic to create jobs, pipelines and execution of non-linear task dependencies are at the application level L2. The L3 level contains the runtime system and the SAGA adaptors. And all the resources are present in L4. The development is therefore modular and layered, allowing to add for example resources adding only a SAGA adaptor to L3 and the resource to L4.

4 JETS

Many-task computing (MTC) became essential for the development of scalable scientific applications on distributed computational infrastructures. The MTC model consists of many individual tasks without intertask communication, that are executed individual system components, like processor cores. This tasks communicate only through the file system. MTC allows large-scale parallel systems with little to no explicit parallel programming.
JETS is a middleware component providing high performance support for many-parallel-task computing (MPTC). It requires a specialized operating system, ZeptoOS Linux, on the nodes capable of communicating with the user scheduler and the ability to remotely invoke MPI programs. JETS can be used as a stand-alone system or as a part of Swift, a highly concurrent programming model for deploying workflows to grids and clusters.

5 Conclusions

The three articles used are very recent (two of them from 2013 and the other from 2012) so they are very relevant to the state-of-art of this theme. The contents are well explained with images and an explanation of each of the concepts used. But these articles talk about only a small part of Portals, if we were to analyse in big depth Portals it would be needed a wider range of articles.

References

[1] *On the Easy Use of Scientific Computing Services for Large Scale Linear Algebra and Parallel Decision Making with the P-Grade Portal.* Hrachya Astsatryan, Vladimir Sahakyan, Yuri Shoukouryan, Michel Dayd, Aurelie Hurault, Ronan Guivarch, Harutyun Terzyan, Levon Hovhannisyan, Journal of Grid Computing, 2013

[2] *Distributed Application Runtime Environment (DARE): A Standards-based Middleware Framework for Science-Gateways.* Sharath Maddineni, Joohyun Kim, Yaakoub El-Khamra, Shantenu Jha, Journal of Grid Computing, 2013

[3] *JETS: Language and System Support for Many-Parallel-Task Workflows.* J. M. Wozniak, M. Wilde, D. S. Katz, Preprint ANL/MCS-P4005-1212, December 2012.