Taverna workflows across Grids, web services and the command line

Hajo N. Krabbenhöft, Daniel Bayer, Steffen Möller

University of Lübeck, Institute for Neuro- and Bioinformatics,
Ratzeburger Allee 160, 23538 Lübeck, Germany

Introduction

Technologies introduced for grid computing support the arbitrary sharing of resources. While the term 'Grid' is commonly associated with high performance computing, other aspects are access control, shared storage, and an increased variability in available data sources and applications that can be accessed via the command line.

Last year we have introduced an interface for Taverna to access resources of the Advanced Resource Collector (ARC) grid middleware (Krabbenhöft et al., Bioinformatics, 2008). It allowed the user of Taverna to upload the grid certificate and to retrieve from a database a series of use cases of common applications in bioinformatics that may be plugged into regular Taverna workflows. These are offered in Taverna like any other action, the user needs no knowledge of grid technology or technical details about the invocation of the respective program. All that is needed is the repository URL and a certificate to grant him computational resources.

Taverna has now evolved into version 2 and the ARC grid interface was adapted, accordingly. For testing purposes and as a fallback for offline operation, support for local invocation and a local distribution via SSH was implemented. Furthermore, the plug-in can now read use cases from local XML files or from custom repository URLs. This way, also internal applications can be offered as Taverna use cases and be intertwined with grid jobs.

Implementation

The ARC plug-in is written completely in Java and was tested on multiple Linux platforms, MacOS X and Windows. Uses cases are stored as an XML-based description for the assembly of

- UNIX command lines,
- Grid jobs,
- Taverna workflow elements

. Use cases are maintained collaboratively. The selection of compute sites is performed by a reference to runtime environments, which different subsets of the sites on a grid have installed.

One special addition to the regular Taverna introduced by the plug-in was the handling of data, which could also be addressed by reference. Since Taverna 2 offers referencing capabilities by default, the implementation was changed to use those with gsiftp URLs for grid storage elements.

Challenges

The biggest challenge is probably the handling of errors. In complex workflows, there should be a way to continue computations when a site is temporarily not available or a single computation has failed because of unforeseen hardware incompatibilities etc. With many redundant sites available on a grid, here grid computing differs from the direct accession of site-specific web services, error recovery mechanisms may be most beneficial. The plug-in extends Taverna's retry logic in a way that it will loop through different grid sites, thereby offering a lower failure rate by exploiting the inherent redundancy of computational grids.

ARC is currently evolving and offers web-services to control jobs and request status updates. Also, job migration and auto-resubmissions are coming. There is consequently the tendency wait for the middleware prior to implementing redundancies.

The availability of applications to be executed is another major concern. This will be helped by an automated deployment of runtime environments to grid sites that is currently under preparation. New software will then be available sooner and at a larger distribution across the grid, allowing for more complex workflows.

Availability

The Taverna plug-in to ARC is available on `http://grid.inb.uni-luebeck.de`. To join the NordGrid see `http://www.nordugrid.org`.