Contribution ID: **33** Type: **Oral contribution**

MOTEUR: a data intensive service-based workflow engine enactor

Thursday, 2 March 2006 15:30 (30 minutes)

** Managing data-intensive application workflows

Many data analysis procedures implemented on grids are not only based on a single processing algorithm but rather assembled from a set of basic tools dedicated to process the data, model it, extract quantitative information, analyze results, etc. Given that interoperable algorithms packed in software components with a standardized interface enabling data exchanges are provided, it is possible to build complex workflows to represent such procedures for data analysis. High level tools for expressing and handling the computation flow are therefore expected to ease computerized medical experiments development.

Workflow processing is a thoroughly researched area. Grid enabled application often need to process large datasets made of e.g. hundreds or thousand of data to be processed according to a same workflow pattern. We are therefore proposing a workflow enactment engine which:
- Makes the description of the application workflow simple from the application developer point of view.
- Enables the execution of legacy code.
- Optimizes the performances of data-intensive applications by exploiting the potential parallelism of the grid infrastructure.

** MOTEUR: an optimized service-based workflow engine

MOTEUR stands for hoMe-made OpTimisEd scUfl enactoR. MOTEUR is written in Java and available under CeCILL Public License (a GPL-compatible open source license) at http://www.i3s.unice.fr/~glatard.
The workflow description language adopted is the Simple Concept Unified Flow Language (Scufl) used by the Taverna and that is currently becoming a standard in the e-Science community.

Figure 1 shows the MOTEUR web interface representing a workflow that is being executed. Each service is represented by a color box and data links are represented by curves. The services are color coded depending on their current status: gray services have never been executed; green services are running; blue services have finished the execution of all input data available; and yellow services are not currently running but waiting for input data to become available.

MOTEUR is interfaced to the job submission interfaces of both the EGEE infrastructure and the Grid5000 experimental grid. In addition, lightweight jobs execution can be orchestrated on local resources. MOTEUR is able to submit different computing tasks on

different infrastructures during a single workflow execution. MOTEUR is implementing an interface to both Web Services and GridRPC application services.

By opposition to the task-based approach implemented in DAGMan, MOTEUR is service-based. The services paradigm has been widely adopted by middleware developers for the high level of flexibility that it offers. Application services are similarly well suited for composing complex applications from basic processing algorithms. In addition, the independent description of application services and the data to be processed make this paradigm very efficient for processing large data sets. However, this approach is less common for application code as it requires all codes to be instrumented with the common service interface.

To ease the use of legacy code, a generic wrapper application service has been developed. This grid submission service is exposing a standard web interface and is controlling the submission of any executable code. It releases the user from the need to write a specific service interface and recompile its application code. Only a small executable invocation description file is required to enable the command line composition by the generic wrapper.

To enact different data-intensive applications, MOTEUR implements two data composition patterns. The data sets transmitted to a service can be composed pairwise (each input of the first input data set is processed with each input of the second one). This correspond to the case where the two input data sets are semantically connected. The data sets can also be fully composed (all inputs of the first set are processed with all inputs of the second one). The use of these two composition strategies significantly enlarges the expressiveness of the workflow language. It is a powerful tool for expressing complex data-intensive processing applications in a very compact format.

Finally MOTEUR enables 3 different levels of parallelism for optimizing workflow application code execution:
- workflow parallelism inherent to the workflow topology;
- data parallelism: different input data can be processed independently in parallel;
- services parallelism: different services processing different data are independent and can be executed in parallel.
To our knowledge, MOTEUR is the first service-based workflow enactor implementing all these optimizations.

** Performance analysis on an image registration assessment application

Medical image registration algorithms are playing a key role in a very large number of medical image analysis procedures. They are fundamental processings often needed prior to any subsequent analysis. The Bronze Standard application (http://egee-na4.ct.infn.it/biomed/BronzeStandard.html) is a statistical procedure aiming at assessing the precision and accuracy of different registration algorithms. The complex application workflow is illustrated in figure 1. This data-intensive application requires the processing of as much input image pairs as possible to extract relevant statistics.

The Bronze Standard application has been enacted on the EGEE infrastructure through the MOTEUR workflow execution engine. A 126 image pairs data base, courtesy of Dr Pierre-Yves Bondiau (cancer treatment center “Antoine Lacassagne”, Nice, France), was used for the computations. In total, the workflow execution resulted in 756 job submissions. The different levels of optimization implemented in MOTEUR permitted a speed-up higher than 9.1 when compared to a naive execution of the workflow.

Such data intensive applications are common in the medical image analysis community and there is an increasing need for compute

infrastructure capable of efficiently processing large image databases. MOTEUR is a generic workflow engine that was designed to efficiently process data intensive workflows. It is freely available for download under a GPL-like license.

Summary

In this abstract we introduce MOTEUR, a service-based workflow engine optimized for dealing with data intensive applications. MOTEUR eases the enactment of applications with complex data flow patterns on the EGEE production infrastrucutre. It is a generic workflow engine, based on current standards and freely available, that can be used to instrument legacy application code at low cost. Performances are demonstrated on a real medical image analysis application.

Primary authors: MONTAGNAT, Johan (CNRS); GLATARD, Tristan (CNRS); PENNEC, Xavier (INRIA)

Presenter: GLATARD, Tristan (CNRS)

Session Classification: 2a: Workload management and Workflows

Track Classification: Workload management and Workflows