



Contribution ID: 94

Type: Oral

An Approach to Grid Interoperability using Ganga

Tuesday 3 March 2009 11:50 (25 minutes)

We present our work on Grid interoperability using Ganga. A Gridway backend module was developed to provide access to globus-based grid resources as well. We have also developed an intergrid backend that allows users to submit jobs that have access to both glite-based resources and globus-based resources. In order to demonstrate the usefulness of the new intergrid backend plugin module, we have integrated the WISDOM autodock application into Ganga as a new built-in application plugin module.

Conclusions and Future Work

Thanks to the easy-to-use interface that Ganga provides, our approach to Grid interoperability using Ganga allows users to readily harness the full power of multiple grids to carry out a large-scale deployment of ligand-protein docking across EGEE and PRAGMA grid infrastructures. We expect to collaborate with the Ganga development team to integrate the plug-in modules that we have prototyped into the future release of Ganga.

Keywords

Ganga, Grid Interoperability, Gridway, Autodock

Impact

We developed a plug-in module for the autodock application that allows users to deploy and run the WISDOM autodock code on both globus-based and glite-based grid infrastructures without having to know anything about grid interoperability issues relating to the two different grid infrastructures. We also developed a globus backend plug-in module called a gridway backend designed to enable job submission to globus resources as well. We have also integrated an intergrid plug-in module into the Ganga. What the intergrid backend does is to determine where jobs will be submitted either to globus or glite nodes depending on the given load information. Last but not least, we have also developed a LigandSplitter module that supports the bulk submission of the same number of ligand-protein docking jobs as the number of ligand files described in the ligand directory.

URL for further information

<http://ganga.web.cern.ch/ganga/>

Detailed analysis

We had to deal with some implementation issues in deploying the WISDOM docking application via our new intergrid backend on both EGEE and globus infrastructures. In terms of security, the voms proxy was used to access both infrastructures transparently. One specific problem that we have faced was that the autodock shell script developed in the WISDOM is to use the globus-url-copy to get all the files including executable, ligands and protein files necessary to run the autodock application on the grid. The script did not work on globus resources because no job delegation is permitted. To address this issue, we have developed the gridway plugin that takes charge of the staging-in of all the files necessary to launch the autodock script on the grid. Unlike the WISDOM original script, what our new script does is just to run the autodock executable code.

The built-in LCG backend module and Ganga wrapper script, on the other hand, is supposed to take care of the staging of all the files.

Authors: Dr HWANG, Soonwook (KISTI); Mr LEE, Yoonki (Seoul National University)

Co-authors: Mr KIM, Eunsung (Seoul National University); Prof. YEOM, Heon Y. (Seoul National University)

Presenter: Dr HWANG, Soonwook (KISTI)

Session Classification: Grid programming

Track Classification: Grid Services exploiting and extending gLite middleware