



Contribution ID: 143

Type: Poster

Complex Systems Software Management

Monday, 11 April 2011 09:00 (8 hours)

Overview

The Complex Systems (CS) Virtual Organization (VO) started its operation in August 2009 providing European researchers in the field of Complexity Science with access to computing, storage and software resources from the European Grid. Initially, only one medium size cluster supported the VO but more resources from the European wide Grid have gradually supported the VO and its users. Currently the VO is supported on 11 European Grid clusters situated in France and Greece that provide users of the VO with a total of more than 3500 CPU cores and approximately 17TB of storage resources.

Impact

Installed software tools and libraries include mono, blas, numpy, scipy, matplotlib, igraph and more. Essentially any software package that may be related to the research activities of the VO members and has been packaged in RPM format can be installed on the supported computational resources.

Description of the work

In March 2010 the Complex Systems VO developed and deployed a robust software management service for installing and updating the software requested by the community of users on the available computing resources and was coupled with the Complex Systems VO Nagios monitoring service for checking the up-to-date of the software database.

The service is based on the Redhat Package Manager (RPM) framework and it provides a straightforward mechanism of installation, updating and removal mechanism for software on Redhat Linux binary compatible resources.

To manage RPM packages over the shared filesystem dedicated for VO software, we had to bypass several known limitations. Two obvious problems are the usage of the network filesystem provided, which is prohibited by the RPM framework, as well as the creation of a central RPM database for checking if all the packages were installed correctly and downloading the RPM packages.

As far as the network file system is concerned we implemented a bypass hook within our software management workflow that allows us to perform management operations on the temporary local filesystem and restoring afterwards these changes onto the shared software filesystem. In the second case, we have gone forward with the creation of an RPM directory on the storage elements as a reference point for the RPM database.

Using such bypass mechanisms we have achieved a robust and novel workflow that allows the software management unit to cleanly support user requests for installation and updating of scientific VO related software tools.

Conclusions

The set of workflow tools we have deployed allow us to effectively and in time respond and handle any software related requests from VO users.

Primary authors: KANELLOPOULOS, Christos (AUTH/GRNET); KOROSOGLOU, Paschalis (AUTH)

Co-author: MOUNTRAKIS, Lampros (AUTH)

Presenters: KANELLOPOULOS, Christos (AUTH/GRNET); KOROSOGLOU, Paschalis (AUTH)

Session Classification: Posters

Track Classification: Poster