January 2009

Ligthweight Evaluation of EnginFrame

J. Schmidt, S. Kaushal, M.Griffiths, M.Hewitt, A.Turner
White Rose Universities of Leeds, Sheffield and York

1. Introduction

The purpose of this lightweight evaluation was to gain knowledge of the EnginFrame portal so we could assess its suitability for deployment on the White Rose Grid systems.

In our evaluation we used a downloaded demo version (EnginFrame for SGE.5.0.0-rc2 version -Oct20 @ 9:42 last modified) with a temporary licence for 1 month.

2. Overview of the software product

EnginFrame is a web portal that enables users to access computing resources. It allows them to submit their applications/jobs to a local or remote resource (this has not been confirmed on our test systems). Users login to the portal and are able to view the status of compute resources on a compute grid.

EnginFrame needs to be tailored for specific tasks and resources. The portal is configured by administrators to enable access to grid enabled applications and resources. For each application/job it requires local development of the relevant service definition files in XML. The service definition files use XML to describe the link between the command lines and the portal interface. For example, the XML may describe what user input is required. Input files can be defined as local or remote (i.e. on the remote storage area). Output files may be downloaded directly from the portal to user's PC or moved automatically to another remote area.

As this was a lightweight evaluation we used the demo version (with limited capability) of this portal and did not build any additional service definition files.

The portal uses a standard web browser, thus should be easily accessible to users. EnginFrame exports its services as portlets compliant to JSR168 standard so the portal should work in any browser that supports java script.

3. Software provider

NICE srl, an Italian company at 33, via Serra, 14020 Camerano Casasco (AT), http://www.nice-italy.com

4. Licence agreement and support

This is proprietary software which requires a licence and yearly maintenance subscription. The cost of licence has not been disclosed to us. The company offers support.

5. Platform availability

The vendor states that the following platforms are supported:

- Linux 2.4.x - Red Hat (Xeon – dual processor)
- Linux 2.4.x - Red Hat, SUSE, Mandriva (IA32)
- Linux 2.6.x - Fedora, SUSE, Mandriva (IA32)
- Linux 2.4.x - Red Hat, SUSE, Mandriva (AMD64, X64)
- Linux 2.6.x – Red Hat, SUSE, Mandriva (AMD64, X64)

- Linux 2.4.x - Debian (IA64)
- Solaris 8 (UltraSPARC)
- AIX 5.1 (PowerPC)
- HP-UX 11.23 (Itanium2)
- Windows Server 2003 R2 (AMD64, X64)

The portal depends on existence of a scheduler (e.g. SGE which we used) or grid middleware (e.g. gLite). It can also submit FORK manager type jobs (i.e. to OS).

6. Installation and configuration

At York EnginFrame was installed on a quad core x86_64 (64bit Intel) machine; Leeds installed EF on a laptop (Centrino Duo - Intel duo x86_32) running Linux 2.6.x Fedora.

We had to install:

a) a specific version of Sun Java JDK (i.e. JDK 5.0 update 16)
b) SunGrid Engine (SGE) version 6.2
c) plugin for the browser (e.g. Firefox – add-ons)

Once these products were installed it was easy to install the EnginFrame (EF) demo version which came with a graphical interface for installation. EnginFrame uses Apache Tomcat as embedded web server and servlet container. We used the default options to install EnginFrame.
The following ports were used:

a) 8080 for HTTP connections
b) 8005 for internal Tomcat for shutdown requests
c) 9999 for the RMI (Remote Method Invocation) registry listening for requests
d) 9998 for RMI requests from the EnginFrame server 9998

To configure EF you need an account (e.g. efsysadmin) to run the portal and the root account runs the agent.

Installation requires the administrator to run a jar file which can be run either in command line mode or through a graphical user interface wizard.

Sheffield configured their demonstration services in a slightly different way. The Access Control List (ACL) flag was removed for each service from the demonstration service description XML files.

7. Functionality

The demo version had very limited functionality due to only a few services actually being implemented in XML. Some features either were not implemented (e.g. ACL database feature /opt/enginframe/conf/authorization.xconf) or did not work correctly (e.g. creation of new plugins).

Removing the Access Control List completely (as in Sheffield's case) enabled support of multiple (non-admin) accounts.

The demo portal did not include any options for sending jobs beyond its local batch system (i.e. SGE on the same system). Therefore we could not test sending jobs from Leeds to the other two sites or even from one system at Leeds to another.

It appears that this demo portal will be able to send jobs only to a single resource. It may be possible to have a broker or metascheduler as this resource.

The portal needs service definition files to be written by a sysadmin in XML for each application. These files enable the potential user to customize the portal's functionality.

The demo shows examples of how you can gather user options and input parameters describing user's application/job and how they are specified as environment variables.

8. Security

A number of options are available for user authentication including HTTP through apache Tomcat and LDAP authentication. The portal uses either HTTP, LDAP or Unix authentication. The authentication data (username & password) may be stored within portal (HTTP) or supplied by the user and verified by the OS or LDAP server.

To use the HTTP authentication mechanism the portal administrator is required to add individual users to the portal through the administration page. An attempt to configure the LDAP authentication module indicated that the EnginFrame LDAP authentication module was unable to provide a correctly formatted LDAP query to the LDAP server and authentication via this route was, at this time, not possible.

None of the ACL features were available to us in the demo. It is expected that the sysadmin might be able to define an Access Control List (in XML) in order to restrict access to services. Furthermore EngineFrame will expose only services that the user has been granted permission by the sysadmin to access.

It is necessary to open a firewall port for the Enginframe web application server and the Enginframe agent application.

9. Development facilities

The demo portal offers the ability to write customised service definition files and plugins, which will expand its functionality. However, this was not possible in our version due to portal failures (i.e. with the error message: environment variable EF_MODE was not specified).

10. Integration with other WRG software

This was not tested. However, it is expected that the portal will integrate with other software available on WRG systems such as globus, other job managers (LSF, PBS) and middleware (e.g. gLite).

11. Ease of use

The demo portal was very easy to use. Users are provided with a simple point and click capability. Once an application is setup it appears that the developed production portal will offer a simple user interface.

12. Special features

An important and useful feature was the ability to carry out a system self-check (i.e. checks on EF portal & its components). Host and full job information are graphically represented in the port. Extended remote file browsing is included but this feature did not appear to work for SGE. It is possible that it might require a plugin to be written.
The demo portal included a set of options for data management such as the ability to jointly manage a set of data files, output streaming (i.e. ability to inspect the output file being produced by a job), or uploading files into spooler.
Gridepedia (http://wwww.gridipedia.eu/engineframeportal.html) states that EF automatically creates a WDSL interface for every service defined in the portal. This may be used to connect applications together or used within workflows.

13. Documentation for system administrators
There is a full set of documentation for administrators, including *Quick Start Guide*, *Release Notes* and *Administration Guide*. However, these documents were not available to us from the company (NICE) web site; and were not supplied with the downloaded software.

The demo portal included tutorials for system administrators. Simple installation instructions were received from the EF support team.

14. Documentation for users
The demo portal includes built-in short clarifications of some terms and portal capabilities. We are not aware if other user documentation exists. Documentation for users will be dependent on customisation of the portal and specific application embedded into it. The tutorials in the demo portal are not available to users.

15. Usage by communities (in general, and locally)
An EnginFrame based portal is used by the EGEE project. This portal (GENIUS) is built on top of the gLite midlleware. It allows users to submit jobs to a resource broker.
The company seems to have a number of portal development projects with the engineering sector.

16. Suitability for the WRG
The demo portal does not offer immediately (without development effort) the functionality required by WRG. While it does integrate with SGE it will require significant efforts to develop and then support the portal exploiting XML based services.

Bioinformatics researchers at Sheffield have expressed an interest in accessing grid compute resources through a web front end. One of the difficulties with their request is the number of applications they want to make available. Thus increasing the amount of time required to implement particular solutions.

17. Conclusions and recommendations
EngineFrame offers an excellent infrastructure for providing web accessible applications hosted on grid compute resources. It appears that a large range of interfaces can be developed to enable coupling of the applications to a range of different management systems. It is uncertain how to build the service that would be used to host applications available on all of the White Rose Grid nodes simultaneously. Possible integrations may require the use of a metascheduling capability.

The EnginFrame portal performed well, within the restricted capability of the demo version with the exception of creating plugins. The licence cost and adaptation of this portal to our users' requirements, which needs significant staff effort, make this product less desirable for the WRG.

18. References

http://www.enginframe.com/
http://www.nice-italy.com/main/index2.php
http://www.gridipedia.eu/engineframeportal.html
http://www.altreia.com/EnginFrame%20Broschure%204%202003.pdf
http://www.teragridforum.org/mediawiki/index.php?title=EnginFrame
http://egee2.eu-egee.org/tutorialimages/tutorial/