Introduction to the Application Hosting Environment

Stefan Zasada

Contents

- Motivation for the AHE
- Concepts & functionality
- Meeting the AHE design constraints
- Architecture of the AHE
- AHE client interaction
- Deploying the AHE

Motivation for the AHE

Problems with current middleware solutions:

- Difficult for an end user to configure and/or install
- Dependent on lots of supporting software also being installed
- Require modified versions of common libraries
- Require non-standard ports to be opened on firewall
- Large footprint – memory/disk space

The Application Hosting Environment

- Based on the idea of applications as web services
- Lightweight hosting environment for running unmodified applications on grid resources (NGS, TeraGrid) and on local resources (departmental clusters)
- Community model: expert user installs and configures an application and uses the AHE to share it with others
- Simple clients with very limited dependencies
- No intrusion onto target grid resources

Virtualizing Applications

- Application Instance/Simulation is central entity; represented by a stateful WS-Resource. State properties include:
  - simulation owner
  - target grid resource
  - job ID
  - simulation input files and urls
  - simulation output files and urls
  - job status
- Application exposed as web service

AHE Functionality

- Launch simulations on multiple grid resources
- Single interface to monitor and manipulate all simulations launched on the various grid resource
- Run simulations without manually having to stage files and GSISSH in
- Retrieve files to local machine when simulation is done
- Can use a combination of different clients – PDA, desktop GUI, command line

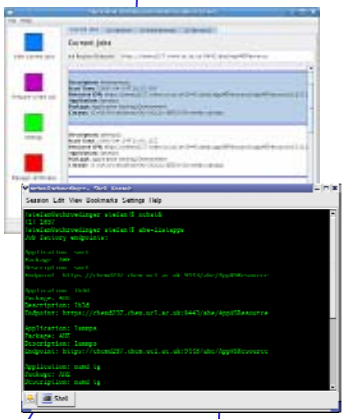
Accessing Resources

Local UCL resources

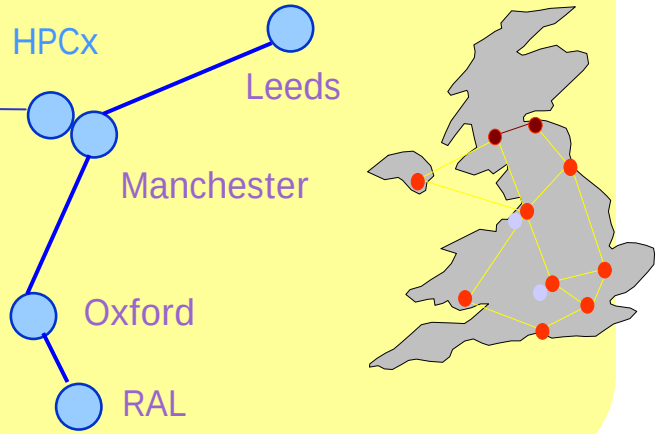
GridSAM / Globus

GridSAM / SGE

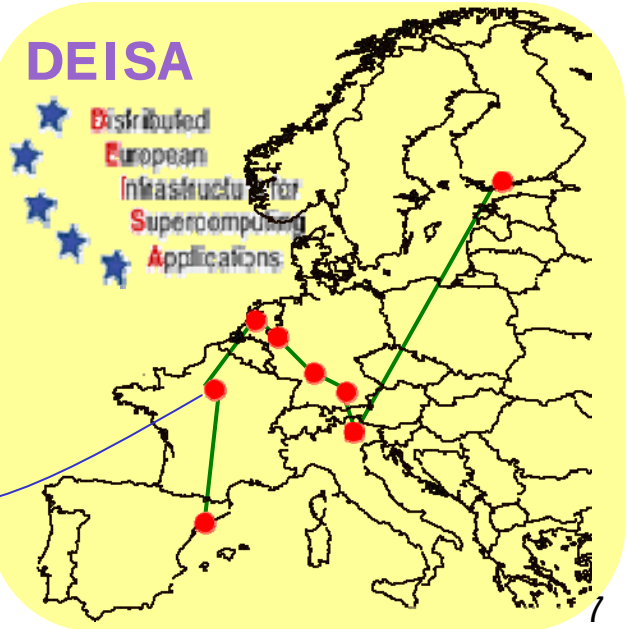
GridSAM / UNICORE



NGS



DEISA



RealityGrid

open middleware infrastructure institute uk
www.omii.ac.uk

EPSRC

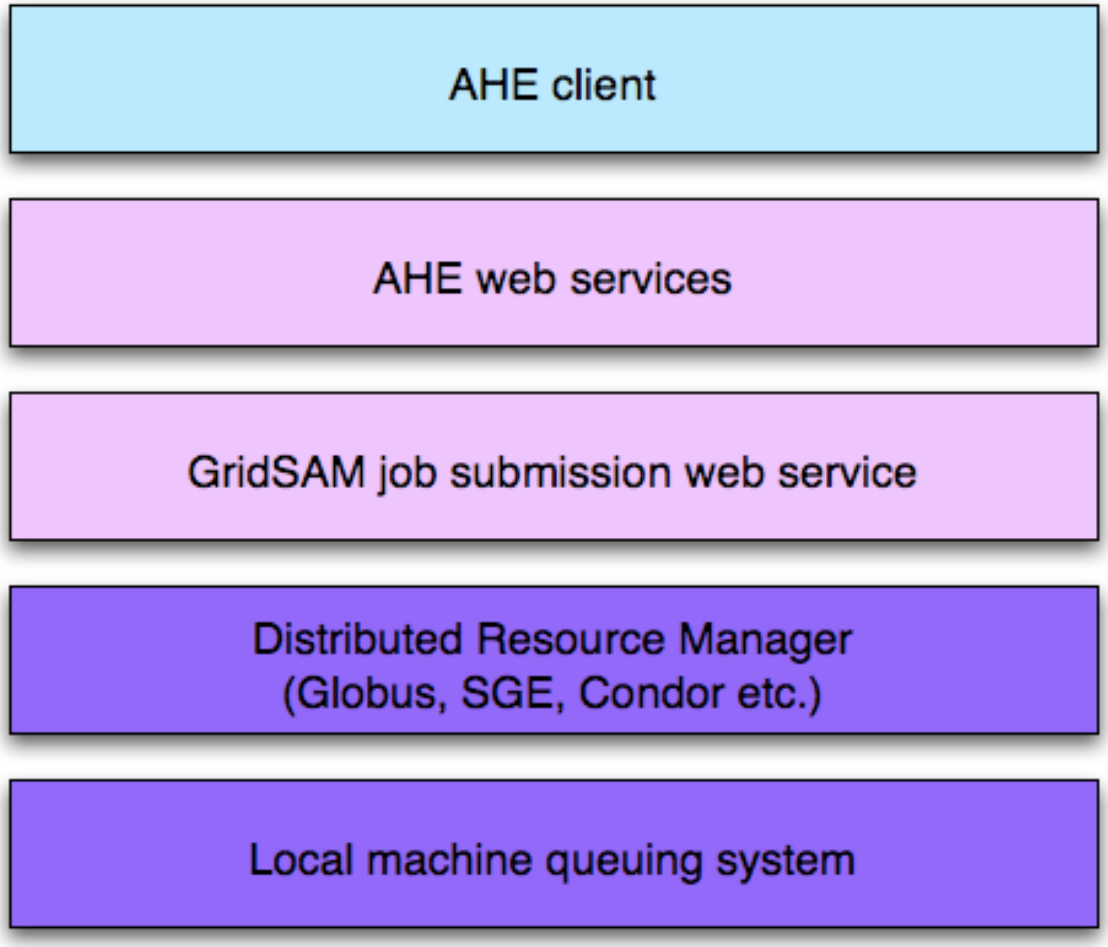
AHE Design Constraints

- Client does not have Globus installed locally
- Client is NAT'd and firewalled
- Client does not have to be a single machine
- Client needs to be able to upload and download files but doesn't have local installation of GridFTP
- Client doesn't maintain information on how to run the application
- Client doesn't care about changes to the backend resources

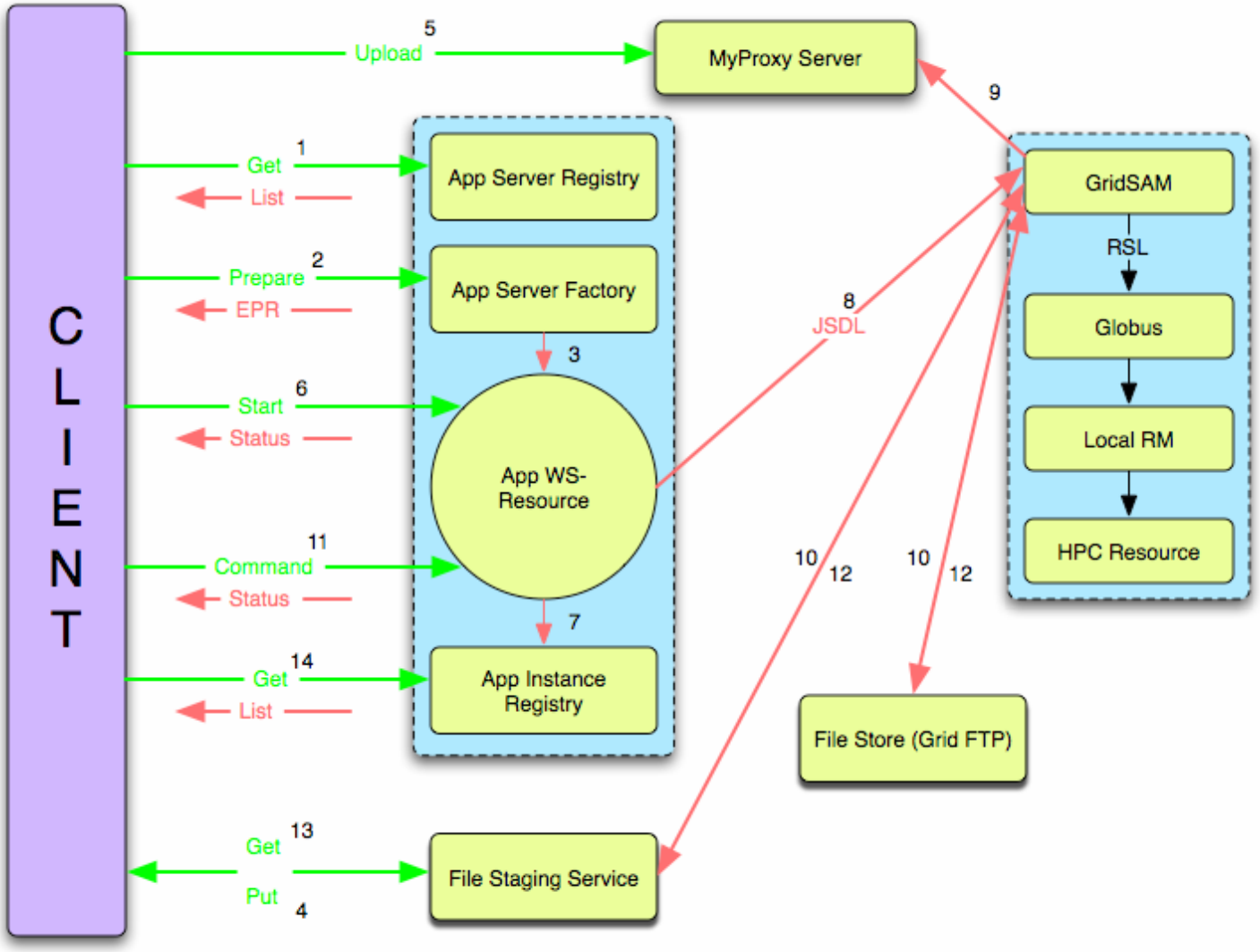
Meeting the Constraints

- AHE Client behind firewall => polls server to update job state etc.
- Uses intermediate filestaging area => GridFTP not installed
- All application specific information for running simulations on the grid resource is maintained on a central service => user can switch clients etc.
- Location of binary on grid resource configured on server => user doesn't need to know
- GridSAM provides interface to job queue

Layered Architecture of the AHE



Service Architecture of the AHE



AHE Server Implementation

- WSRF::Lite => services developed in Perl
- WebDAV server
- GridSAM => Globus grid
  => Sun Grid Engine
  => Condor pool
  => Unicore
- MyProxy
- PostgreSQL database
- Apache/Tomcat container

AHE Server Deployment

The expert user must:

- Set up container to host services:
  - Apache/WSRF::Lite or modified Tomcat/WSRF::Lite
- Set up PostgreSQL database and WebDAV server
- If not already running set up GridSAM instance for grid resource
- Deploy and configure the AHE services in the container
- OMII stack installer will do all of this automatically

Once deployed, any number of applications can be hosted

Hosting a New Application

Expert user must:

- Install and configure application on all resources on which it is being shared
- Create a JSDL template for the application (easily cloned from exiting template)
- Add the application to the RMInfo.xml file
- Run a script to reread the configuration

Documentation covers whole process of deploying AHE & applications on NGS and TeraGrid

Client Implementation

- GUI & command line clients implemented in Java
- Client allows user to:
  - Discover appropriate resources
  - Launch application
  - Monitor running jobs
  - Query registry of running jobs
  - Stage files to and from resource
  - Terminate jobs
- GUI client implements application launching as a wizard

Client Extensibility

- Plugins can be added to process application input files to automatically discover the input and output files that need to be staged
- If no plugin is available then a default case will allow users to specify input and output files manually
- Plugins implement AHEConfParser interface and follow specific naming convention
- Plugin .class files dropped into plug-in directory and picked up by GUI/command line clients

AHE Client Deployment

- Deploying client is trivial for the end user:
  - User's machine must have Java installed
  - User downloads and untars client package
  - Imports X.509 certificate into Java keystore using provided script
  - Configures client with endpoints of AHE services supplied by expert user
- Ready to go!

Constructing workflows with the AHE

- By calling command line clients from Perl script complex workflows can be achieved
- Easily create chained or ensemble simulations
- E.g. HIV equilibration protocol implemented by:
  - ahe-prepare → prepare a new simulation for the first step
  - ahe-start → start the step
  - ahe-monitor → poll until step complete
  - ahe-getoutput → download output files
  - repeat for next step

Current Deployed Applications

- Currently hosting:
  - NAMD
  - LAMMPS
  - DL_POLY
  - LB3D
  - Gromacs
  - CHARMM
- Plan to host:
  - Trubal
  - POLCOMS

Summary

- The AHE provides a lightweight, easily deployable environment for running unmodified scientific applications on the grid and local resources
- The AHE server is designed to be deployed by an expert user who uses it to share applications installed on grid resources
- The client is easily installed by any end user, requiring no intervention by system/network administrators
- By calling the command line clients from scripts, complex scientific workflows can be implemented

Any Questions?

- Released in OMII 3.2.0

  http://www.omii.ac.uk/downloads/
- RealityGrid web site:

  http://www.realitygrid.org/AHE
- NeSCForge:

  http://forge.nesc.ac.uk/projects/ahe/
- Mailing list:

  http://www.mailinglists.ucl.ac.uk/mailman/listinfo/ahe-discuss

Exercise 1

- Installing and configuring the AHE client

- Tasks:
  - Install the AHE client on your system
  - Set up a keystore containing your grid certificate
  - Configure the client with settings for UCL's AHE server
  - Confirm that the client is installed and working

- Outcome
  - Installed and configured AHE client

Exercise 2

Launching an Application on the NGS using the AHE

Tasks:

- Launch the sort application with the AHE GUI client
- Launch the sort application with the AHE command line client
- Manually specify input and output files for an application
- Retrieve application output from NGS machine

- Outcome
  - Successfully run applications on NGS machines