

Application Hosting Environment Generic Application Tutorial

TABLE OF CONTENTS

1:	Generic Application.....	2
2:	Hosting a Generic Application within the AHE	2
2.1	Install the Generic Application on the Grid Resources	2
2.2	Add the Generic Application in each Grid Resource/RM's RMIInfo	2
2.3	Create Generic Application JSDL Templates for each Grid Resource/RM.....	4
2.4	Re-populate the Application Server Registry	4
2.5	Run the Generic Application Instance/Simulation	5

1: Generic Application

Here we consider a sort application implemented in a perl script, `ahe_sort.pl`, which sorts a list of words read in from an input file. The sort application represents any generic application that you wish to host within the AHE and the perl script itself represents the application executable.

2: Hosting a Generic Application within the AHE

In this tutorial, we assume that –

- 1) the user is interested in hosting a generic application, e.g `sort(ahe_sort.pl)`, within the AHE in order to run it on the TeraGrid sites at SDSC and NCSA and core National Grid Service (NGS) nodes at Leeds, Oxford, RAL and Manchester.
- 2) The AHE server has been installed and configured as per the AHE Server Installation Guide (Section 2) on the server machine in `$AHE_LOCATION`.
- 3) The AHE client has been installed as per the AHE Client User Guide (Section 1) on the client machine in `$AHECLIENT_HOME`.

2.1 Install the Generic Application on the Grid Resources

In the case of the sample sort application, copy `ahe_sort.pl` to an appropriate directory in your account on the Grid Resource. For any other application, copy the executable to an appropriate account on the Grid Resource. Note the path to your executable on the grid resource, e.g.

- 2.1.1 TeraGrid NCSA site – `/home/ac/xyz/bin/ahe_sort.pl`
- 2.1.2 TeraGrid SDSC site - `/users/xyz/bin/ahe_sort.pl`
- 2.1.3 Leeds node – `/home/data01_d/ngs0xyz /bin/ahe_sort.pl`
- 2.1.4 Oxford node - `/home/ngs0xyz//bin/ahe_sort.pl`
- 2.1.5 RAL node - `/home/ngs0xyz/bin/ahe_sort.pl`
- 2.1.6 Manchester node - `/home/ngs0xyz/bin/ahe_sort.pl`

The application executable should have execute permissions according to whom it is intended to run it. For example, to allow everyone to read and execute it, the permission on the `ahe_sort.pl` file should be set to 0755 using the `chmod` command.

2.2 Add the Generic Application in each Grid Resource/RM's RMLInfo

- 2.2.1 On the AHE server, in `$AHE_LOCATION/config/RMInfo/RMLList.xml` for each `<RM>` element corresponding to the Grid Resources, add the `<app>` sub-element.

```
<ahe:app xmlns:ahe="http://www.rahwl.org/v1.0">  
  <ahe:name>sort</ahe:name>  
  <ahe:JSDLTemplate>config/JSDLTemplates/sort.$RMName.jsdl</ahe:JSDLTemplate>  
</ahe:app>
```

Each `<RM>` element can have multiple `<app>` sub-elements for each of the applications installed on the RM

- 2.2.2 Sections of the `RMLList.xml` file for the Grid Resources with the sort application installed would look like:

```
<ahe:RMLList xmlns:ahe="http://www.rahwl.org/ApplicationHostingEnvironment/v1.0">
```

```

<!-- - - - - Entry for the NCSA node ----->
  <ahe:RM>
    <ahe:commonName>NCSA</ahe:commonName>
    <ahe:app>
      <ahe:name>sort </ahe:name>
      <ahe:JSDLTemplate>config/JSDLTemplates/sort.ncsa.jsdl</ahe:JSDLTemplates>
    <ahe:app>
      ...other RM properties
  </ahe:RM>

<!-- - - - - Entry for the SDSC node ----->
  <ahe:RM>
    <ahe:commonName>SDSC</ahe:commonName>
    <ahe:app>
      <ahe:name>sort</ahe:name>
      <ahe:JSDLTemplate>config/JSDLTemplates/sort.sdsc.jsdl</ahe:JSDLTemplates>
    <ahe:app>
      ...other RM properties
  </ahe:RM>

<!-- - - - - Entry for the Leeds node ----->
  <ahe:RM>
    <ahe:commonName>leeds</ahe:commonName>
    <ahe:app>
      <ahe:name>sort </ahe:name>
      <ahe:JSDLTemplate>config/JSDLTemplates/sort.leeds.jsdl</ahe:JSDLTemplates>
    <ahe:app>
      ...other RM properties
  </ahe:RM>

<!-- - - - - Entry for the Oxford node ----->
  <ahe:RM>
    <ahe:commonName>oesc</ahe:commonName>
    <ahe:app>
      <ahe:name>sort</ahe:name>
      <ahe:JSDLTemplate>config/JSDLTemplates/>sort.oesc.jsdl</ahe:JSDLTemplates>
    <ahe:app>
      ... other RM properties
  </ahe:RM>

<!-- - - - - Entry for the RAL node ----->
  <ahe:RM>
    <ahe:commonName>rl</ahe:commonName>
    <ahe:app>
      <ahe:name>sort</ahe:name>
      <ahe:JSDLTemplate>config/JSDLTemplates/>sort.rl.jsdl</ahe:JSDLTemplates>
    <ahe:app>
      ... other RM properties
  </ahe:RM>

<!-- - - - - Entry for the Manchester node ----->
  <ahe:RM>
    <ahe:commonName>man</ahe:commonName>
    <ahe:app>
      <ahe:name>sort</ahe:name>
      <ahe:JSDLTemplate>config/JSDLTemplates/>sort.man.jsdl</ahe:JSDLTemplates>

```

```
<ahe:app>
  ... other RM properties
</ahe:RM>
```

```
</ahe:RMList>
```

2.3 Create Generic Application JSDL Templates for each Grid Resource/RM

2.3.1 On the AHE server, for each Grid Resource, create a sort JSDL Template. All the JSDL Templates should be placed in \$AHE_LOCATION/config/JSDLTemplates.

2.3.2 We follow the following convention in naming JSDL Templates – \$applicationName.\$RMName.jsdl. So for the sort application JSDL templates we have –

```
sort.ncsa.jsdl
sort.sdsc.jsdl
sort.leeds.jsdl
sort.oesc.jsdl
sort.rl.jsdl
sort.man.jsdl
```

2.3.3 The JSDL templates are included in the AHE server source distribution. The important sections are the specification of the location of the executable on the Grid Resource and any environment variables that need to be set for the application to be run.

2.3.3.1 sort.ncsa.jsdl

```
...
<Executable>/home/ac/xyz/bin/ahe_sort.pl</Executable>
```

2.3.3.2 sort.leeds.jsdl

```
...
<Executable>/home/data01_d/ngs0xyz/ahe_sort.pl</Executable>
<Environment name="NGSMODULES">clusteruser</Environment>
...
```

2.4 Re-populate the Application Server Registry

2.4.1 Destroy the old registry of Applications hosted within the AHE.

```
cd $AHE_LOCATION/scripts
./ahe_destroyAppServerRegistry.pl
```

2.4.2 Populate the registry of Applications hosted within the AHE.

```
cd $AHE_LOCATION/scripts
./ahe_createAppServerRegistry.pl
```

2.4.3 Print out the list of Applications hosted within the AHE

```
cd $AHE_LOCATION/scripts
./ahe_queryAppServerRegistry.pl
```

You should see the new application:
Printing Application Server Registry.....

...

```
Application Type: newApp
Application Factory EPR:
https://chemd.rahwl.ac.uk:8443/ahe/AppWSResource
```

...

2.5 Run the Generic Application Instance/Simulation

From the AHE client, launch the sort application.

2.5.1 List all applications available within the AHE
`$AHECLIENT_HOME/ahe-listapps`

2.5.2 Prepare the Application Instance/Simulation
`$AHECLIENT_HOME/ahe-prepare -s sortapp1 -app sort`

2.5.3 Start the Simulation
The `ahe-start` client needs to parse the generic application's configuration file, in this case, `config.txt`, to determine the names of the input and output files to be staged to the Grid Resource for the application to run. Detailed description for writing an application-specific configuration file parser is given in the AHE Client User Guide (Section 4). Once the parser plug-in has been integrated with the AHE client, the generic application can be started -
`$AHECLIENT_HOME/ahe-start -s sortapp1 -config /tmp/config.txt -RM leeds -n 1`

2.5.4 Monitor the Simulation
`$AHECLIENT_HOME/ahe-monitor -s sortapp1`

2.5.5 Retrieve output files on completion
`$AHECLIENT_HOME/ahe-getoutput -s sortapp1 -l .`

2.5.6 Destroy the Simulation
`$AHECLIENT_HOME/ahe-destroy -s sortapp1`

2.5.7 List all your simulations launched via the AHE
`$AHECLIENT_HOME/ahe-list`

2.5.8 Script an ensemble of application instances/simulations
Scripts can be written using the command-line clients above to launch ensembles of application instances. For example, one could launch 10 sort application instances with the input files named `config_1.txt`, `config_2.txt`...`config_10.txt`, by using the following perl script.

```
#!/usr/bin/perl -w
```

```
#pre-processing of input files goes here
```

```
my @RMArray = ("NCSA", "SDSC", "leeds", "oesc", "rl");  
my @CPUArray = (1,1,1,1,1);  
my $numSims = 2*@RMArray;
```

```
#assign the 10 simulations to the resources in a round-robin fashion  
for(my $i = 0;$i < $numSims;$i++){
```

```
    $idx = $i % scalar(@RMArray);
```

```
    system "$ENV{AHECLIENT_HOME}/ahe-prepare -s sortapp$i -app sort";  
    system "$ENV{AHECLIENT_HOME}/ahe-start -s sortapp$i  
        -config /tmp/config_$.txt  
        -RM $RMArray[$idx]
```

```

        -n $CPUArray[$idx]";
    }

#wait until all the simulation has finished and retrieve the output to local directory
for(my $i = 0;$i < $numSims;$i++){
    while(1){
        $status =~ `${ENV{AHECLIENT_HOME}}/ahe-monitor sortapp$i`;
        if(!$status =~ / files staged out/){
            next;
        }
        system "${ENV{AHECLIENT_HOME}}/ahe-getoutput -s sortapp$i -l .";
        last;
    }
}

#post-processing of output files goes here

exit(0);

```

2.5.9 Using the GUI to run a Generic Application Instance/Simulation

Each of the steps 2.5.1 to 2.5.7 can be performed using the AHE Graphical User Interface Client. Please refer to the instructions in the AHE Client User Guide (Section 2).