

# Application Hosting Environment LAMMPS Tutorial

## TABLE OF CONTENTS

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

## 1: LAMMPS

LAMMPS is a massively parallel molecular dynamics code with optimizations for long-range interactions. The application has been hosted within the AHE to study problems in materials science and soft condensed matter. More information about LAMMPS can be obtained at <http://www.cs.sandia.gov/~sjplimp/lammps.html>

## 2: Hosting LAMMPS within the AHE

In this tutorial, we assume that –

- 1) the user is interested in hosting LAMMPS within the AHE in order to run LAMMPS simulations on the TeraGrid sites at SDSC and NCSA and the 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 LAMMPS on the Grid Resources

Download and compile LAMMPS with MPI as per the instructions at <http://www.cs.sandia.gov/~sjplimp/download.html> on each of the Grid resources. Note the path to your lammps executable on the grid resource, e.g.

- 2.1.1 TeraGrid NCSA site – /home/ac/xyz/bin/lammps2-mpi
- 2.1.2 TeraGrid SDSC site - /users/xyz/bin/lammps2-mpi
- 2.1.3 Leeds node – /home/data01\_d/ngs0xyz /bin/lmp\_leeds
- 2.1.4 Oxford node - /home/ngs0xyz//bin/lmp\_oesc
- 2.1.5 RAL node - /home/ngs0xyz/bin/lmp\_rl
- 2.1.6 Manchester node - /home/ngs0xyz/bin/lmp\_man

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

### 2.2 Add LAMMPS Application in each Grid Resource/RM's RMLInfo

- 2.2.1 On the AHE server, in \$AHE\_LOCATION/config/RMLInfo/RMLList.xml for each <RM> element corresponding to the GridResources, add the <app> sub-element.

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

      <ahe:JSDLTemplate>config/JSDLTemplates/lammps.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>lammps</ahe:name>

      <ahe:JSDLTemplate>config/JSDLTemplates/lammps.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>lammps</ahe:name>

      <ahe:JSDLTemplate>config/JSDLTemplates/lammps.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>lammps</ahe:name>

      <ahe:JSDLTemplate>config/JSDLTemplates/lammps.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>lammps</ahe:name>
      <ahe:JSDLTemplate>config/JSDLTemplates/lammps.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>lammps</ahe:name>
      <ahe:JSDLTemplate>config/JSDLTemplates/lammps.man.jsdl</ahe:JSDLTemplates>
    </ahe:app>
    ... other RM properties
  </ahe:RM>
</ahe:RMList>

```

## 2.3 Create LAMMPS JSDL Templates for each Grid Resource/RM

- 2.3.1 On the AHE server, for each Grid Resource, create a LAMMPS 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 LAMMPS JSDL templates we have –

```

lammps.ncsa.jsdl
lammps.sdsc.jsdl
lammps.leeds.jsdl
lammps.oesc.jsdl
lammps.rl.jsdl
lammps.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 environment variables that need to be set for the application to be run.

- 2.3.3.1 lammps.ncsa.jsdl

```

...
<Executable>/home/ac/xyz/bin/lmp_ncsa</Executable>
...

```

- 2.3.3.2 lammps.leeds.jsdl

```

...
<Executable>/home/data01_d/ngs0xyz/lmp_leeds</Executable>
<Environment name="NGSMODULES">gm/2.0.8</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 LAMMPS Simulations

From the AHE client, launch the LAMMPS simulation.

- 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 lammppsim1 -app lammmps`
- 2.5.3 Start the Simulation  
`$AHECLIENT_HOME/ahe-start -s lammppsim1 -config /tmp/lammmps.inp  
-RM leeds -n 2`
- 2.5.4 Monitor the Simulation  
`$AHECLIENT_HOME/ahe-monitor -s lammppsim1`
- 2.5.5 Retrieve output files on completion  
`$AHECLIENT_HOME/ahe-getoutput -s lammppsim1 -l .`
- 2.5.6 Destroy the Simulation  
`$AHECLIENT_HOME/ahe-destroy -s lammppsim1`
- 2.5.7 List all your simulations launched via the AHE  
`$AHECLIENT_HOME/ahe-list`
- 2.5.8 Script an ensemble of simulations

Scripts can be written using the command-line clients above to launch ensembles of simulations. For example, one could launch 10 simulations with the lammmps input files named `lammmps_1.inp`, `lammmps_2.inp`...`lammmps_10.inp`, 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 = (8,8,4,8,2);  
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 "$ENV{AHECLIENT_HOME}/ahe-prepare -s lammppsim$i -app  
lammmps";
```

```
    system "$ENV{AHECLIENT_HOME}/ahe-start -s lammppsim$i  
        -config /tmp/lammmps_{$i}.inp  
        -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 lammppsim.$i`;
        if(!$status =~ / files staged out/){
            next;
        }
        system "${ENV{AHECLIENT_HOME}}/ahe-getoutput -s lammppsim$i -l
";
        last;
    }
}

```

#post-processing of output files goes here

```
exit(0);
```

## 2.5.9 Using the GUI to run a LAMMPS 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).