

Abaqus Explicit 6.8 – Execution Guidelines for running applications in aggregated environment using ScaleMP’s vSMP Foundation

Overview

Abaqus Explicit is a multi-process application that uses MPI for inter-process communication. HP-MPI has been set as the default MPI for the Abaqus Explicit application. In addition Abaqus Explicit supports MPICH2 as well by using a dmp library.

While it is possible to run Abaqus Explicit on the aggregation platform with the HP-MPI implementation, using MPICH2 tuned for vSMP Foundation may yield a performance improvement of 5-15%.

Running Abaqus-Explicit with HP-MPI

HP-MPI has a built-in mechanism for assigning MPI processes to specific CPUs. Process placement is controlled by environment variables named **MPI_BIND_MAP** and **MPRUN_OPTIONS**. When these variables are not set, process placement will not be performed.

Environment variables – HP-MPI

If you are running with HPMPI, you should set the following environment variables prior to running Abaqus to yield the optimal performance:

```
export MPI_BIND_MAP=0,1,2,3,4,5,6,7 (For example)

export MPRUN_OPTIONS="-cpu_bind=map_cpu,v"

export HPMP_FRAGSIZE=131072

export MPI_SHMEMCNTL=16,24000000,4000000
```

MPI_BIND_MAP specifies a list of CPUs to which MPI ranks will be bound. You should replace the list above with a list of integers, zero to #cpus-1.

For more information on HP-MPI CPU affinity settings, refer to the HP-MPI user's guide available from ["http://docs.hp.com/en/B6060-96022/B6060-96022.pdf"](http://docs.hp.com/en/B6060-96022/B6060-96022.pdf).

Running Abaqus-Explicit with MPICH2 tuned for vSMP

To run Abaqus using MPICH2, the Intel MPI wrapper that came with Abaqus should be used.

Environment variables – MPICH2

```
export VSMP_PLACEMENT=PACKED

export VSMP_MEM_PIN=YES
```

Create Symbolic link for libmpi:

```
cd /opt/ScaleMP/mpich2/1.0.8/lib
ln -s libmpich.so libmpi.so.3.1
```

Above variables are already set in “abaqus_v6.env” file below. Uncomment out mp_mpi_implementation & mp_mpi_implementation in “abaqus_v6.env”.

Running Abaqus-Explicit with Ram-Drive

Since Abaqus jobs perform a lot of scratch IO as they run out-of-core, running with a Ram-Drive may yield shorter runtimes. In order to run with Ram-Drive, perform the following:

```
export LM_LICENSE_FILE=<port>@<license server>

sudo mkdir /ramfs

sudo mount /dev/ram -t ramfs /ramfs -o noatime

sudo chmod 777 /ramfs

source <path_to_intel_compiler>/iccvars.sh

cp abaqus_v6.env e1.inp /ramfs

cd /ramfs

abq681 job=exp-8 input=e1.inp cpus=8 interactive
```

Sample script for HP-MPI and MPICH2 tuned for vSMP

```
----- abaqus_v6.env -----
mp_mpi_implementation = "intel"
mp_mpirun_path={"intel": "/opt/ScaleMP/mpich2/1.0.8/bin/mpiexec"}

import os
os.environ["ABA_ALLOW_USER_MPI"]="1"
os.environ["ABA_MPI_NOPAM"] = "1"
os.environ["VSMP_PLACEMENT"]='PACKED'
os.environ["VSMP_MEM_PIN"]='YES'
os.environ["VSMP_VERBOSE"]='YES'

os.environ["HPMP_FRAGSIZE"]="131072"
os.environ["MPI_SHMEMCNTL"]="16,24000000,4000000"
os.environ["MPI_BIND_MAP"]="0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16"
os.environ["MPIRUN_OPTIONS"]="-cpu_bind=map_cpu,v"

def mp_mpiCommand():
    import os, uti

    print 'Preparing command for %s...' % str(mpi_implementation)
    newEnvironment = {}
    newCommand = None

    if mpi_implementation == 'intel':
        newCommand = mpirun_path
        newCommand = '%s %s -np %d %s %s' % (newCommand, mpirun_options, cpus, program,
arguments)
        print "newCommand = ", newCommand
    else:
        print "You are not using the SCALEMP MPI."
        newCommand = command
    return(newCommand,newEnvironment)
-----
```