

MSC Nastran – Execution Guidelines for running applications in aggregated environment using ScaleMP’s vSMP Foundation

Overview

MSC Nastran is a multi-process application that uses MPI for inter-process communication. Our recommendation is to use Intel MPI.

In addition Nastran usually requires high memory and I/O bandwidth; hence our recommendation is to use ram-drive (see below), and for higher core counts it may be beneficial to run in ‘spread’ mode.

The following execution guidelines are relevant for MSC Nastran 2008.

Running Nastran with Intel MPI

MSC Nastran comes with several MPI “flavors”. Our recommended one is Intel MPI, but any of the others should run fine as long as the application is limited to a single board.

The following arguments are required on the mdnast2008 command line to support multiprocessing with Intel MPI: **dmp=<number-of-mpi-ranks> intelmpi=yes proc=intel**

Environment variables – Intel MPI

You should set the following environment variables prior to running Nastran with Intel MPI to yield the optimal performance:

```
export I_MPI_DEVICE=shm

export I_MPI_PIN_MODE=lib

export I_MPI_PIN_PROCS=15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0

export I_MPI_PIN_PROCESSOR_LIST=15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0
```

The setting above specifies exactly where to place the processes for a multi-process run. If you wish to place the Nastran processes on a different set of CPUs, edit the values of the environment variables I_MPI_PIN_PROCS & I_MPI_PIN_PROCESSOR_LIST.

NOTE: It is also possible to leave the process affinity to NUMABIND. As this is an MPI process the following rule could apply:

```
name=nastran pattern=analysis.dmp verbose=1 rule=RULE-procgroup.so process_allocation=cpu
task_affinity=cpu
```

Running Nastran with Ram-Drive

Nastran is notorious for performing lots of I/O. Therefore it will perform better if it uses a RAMFS mount for its scratch file space.

In order to run with Ram-Drive, perform the following before running Nastran

```
sudo mkdir -p /ramfs
sudo mount -t ramfs ramfs /ramfs -o noatime
sudo chmod 777 /ramfs
```

To ensure that Nastran will use the Ram-Drive as its scratch space, edit the Nastran configuration file <nastran-install-dir>/conf/nast2008rc and add the following line at the end:

```
sdirectory=/ramfs/tmp
```

After the run terminates, it is recommended to delete any remaining files in the scratch directory.

Sample script for Nastran with Intel MPI and Ram-Drive

```
-----
# Setup RAM-Drive
sudo mkdir -p /ramfs
((`mount|grep ramfs|wc -l` == 0)) sudo mount -t ramfs ramfs /ramfs -o noatime
sudo chmod 777 /ramfs

# Prepare Environment
ulimit -s unlimited

mkdir -p results

HOSTS=`hostname`
export TMPDIR=/ramfs/tmp

mkdir -p $TMPDIR

export PATH=/opt/msc/nastran/bin:$PATH

export I_MPI_DEVICE=shm

export I_MPI_PIN_MODE=lib

export I_MPI_PIN_PROCS=15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0

export I_MPI_PIN_PROCESSOR_LIST=15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0

export LD_PRELOAD=/opt/ScaleMP/libvsmpplib/0.1/lib64/libvsmpplib.so

args="intelmpi=yes proc=intel"

#

# Ole model w/ ACMS

time mdnast2008 xx0cmd2 scr=yes bat=no app=no mem=1800mb out=xx0cmd2_1
```

```
cp -p *.log *.f04 *.f06 ../results
```

```
rm $TMPDIR/*
```

```
time mdnast2008 xx0cmd2 scr=yes bat=no app=no mem=1800mb out=xx0cmd2_2 dmp=2  
hosts=${HOSTS} $args
```

```
cp -p *.log *.f04 *.f06 ../results
```

```
rm $TMPDIR/*
```

```
time mdnast2008 xx0cmd2 scr=yes bat=no app=no mem=1800mb out=xx0cmd2_4 dmp=4  
hosts=${HOSTS} ${MODE} $args
```

```
cp -p *.log *.f04 *.f06 ../results
```

```
rm $TMPDIR/*
```

```
time mdnast2008 xx0cmd2 scr=yes bat=no app=no mem=1800mb out=xx0cmd2_8 dmp=8  
hosts=${HOSTS} ${MODE} $args
```

```
cp -p *.log *.f04 *.f06 ../results
```

```
rm $TMPDIR/*
```

```
#
```
