# PBS Pro and Ansys Examples

## Introduction

This document contains a number of different types of examples of using Ansys on the HPC, listed below.



## Single-node Ansys Job

```
#!/bin/bash -l
#
#### Set shell
#PBS -S /bin/bash
#
# set job name
#PBS -N AnsysTest
#
# set default resource requirements for job
# (4 processors on 1 node requesting 3 hours 30 minutes of runtime)
# - these can be overridden on the qsub command line
#PBS -l select=1:ncpus=4
#PBS -l walltime=03:30:00

# load ansys module so that we find the cfx5solve command
module load ansys/16.2

# Change to directory from which job was submitted
cd $PBS_O_WORKDIR

# Set number of processors to run on
# (list of node names is in file $PBS_NODEFILE)
nprocs=`wc -l $PBS_NODEFILE | awk '{ print $1 }'`

# Load ansys module so that we find the ansys121 command
module load ansys

# Run ansys commands in file plate in batch mode
#( the -m & -db flags increase the total amount of memory usable by ansys)
ansys162 -b -m 20000 -db 10000 -np $nprocs  -j plate < plate > plate.out
```

## Single-node CFX Job

```
#!/bin/bash -l
#
#### Set shell
#PBS -S /bin/bash
#
# set job name
#PBS -N CFXTest
#
# Set default resources requirements for job
# (4 processors on 1 node requesting 120 hours runtime using 4Gb of memory)
# - these can be overridden on the qsub command line
#PBS -l select=1:ncpus=4
#PBS -l mem=4gb
#PBS -l walltime=120:00:00
#
# request that regular output (stdout) and
# terminal output (stderr) go to the same file
#PBS -j oe
#
# mail options
#PBS -m abe
#PBS -M username@usq.edu.au
#
# set the queue to run job on
#PBS -q default

# load ansys module so that we find the cfx5solve command
module load ansys/16.2

# goto the directory from which you submitted the job
cd $PBS_O_WORKDIR

# set number of processors to run on
nprocs=`cat $PBS_NODEFILE | wc -l`

# start calculation
cfx5solve -batch -def cfxtest.def -fullname cfxtest -monitor cfxtest.res -part $nprocs -par-local -solver-
double
```

## Single-node Fluent Job

```
#!/bin/bash -l
#
# set shell
#PBS -S /bin/bash
#
# set job name
#PBS -N FluentTest
#
# set default resources requirements for job
# (1 processor on 1 node requesting 4 hours 30 minutes of runtime)
# - these can be overridden on the qsub command line
#PBS -l select=1:ncpus=1
#PBS -l walltime=04:30:00
#
# request that regular output (stdout) and
# terminal output (stderr) go to the same file
#PBS -j oe
#
# set mail options to send job notifications
#PBS -m abe
#PBS -M username@usq.edu.au
#
# set the queue to run job on
#PBS -q default

# set number of processors to run on
# (list of node names is in file $PBS_NODEFILE)
nprocs=`wc -l $PBS_NODEFILE | awk '{ print $1 }'`

# load ansys module so that we find the fluent command
module load ansys/16.2

# specifies the version of ANSYS FLUENT to run
version=3d

# specifies journal file to use
journal=elbow1_journal

# change to the directory from which you submitted the job
cd $PBS_O_WORKDIR

# start computation
fluent $version -t$nprocs -cnf=$PBS_NODEFILE -g -i $journal
```

## Multi-node CFX Job

```
#!/bin/bash
#
# set shell
#PBS -S /bin/bash
#
# set job name
#PBS -N CFXParallelTest
#
# torque job script to run CFX solver in parallel over multiple nodes
# set default resource requirements for job
# - these can be overridden on the qsub command line
# (this is for an 16 CPU job, requesting 6 hours)
#PBS -l select=2ncpus=8
#PBS -l walltime=6:00:00

# change to directory from which job was submitted
cd $PBS_O_WORKDIR

# set number of processors per host listing
# (set to 1 as $PBS_NODEFILE lists each node twice if :ppn=2)
procs_per_host=1

# Create host list
hl=""
for host in `cat $PBS_NODEFILE`
do
 if [ "$hl" = "" ]
 then hl="$host*$procs_per_host"
 else hl="${hl},$host*$procs_per_host"
 fi
done

# load ansys module so that we find the cfx5solve command
module load ansys/16.2

# run using PVM for message-passing between nodes by default
cfx5solve -double -def BluntBody.def -parallel -par-dist $hl

# use following line to specify MPI for message-passing instead
#cfx5solve -double -def BluntBody.def -parallel -par-dist $hl -parallel-mode mpi
```

## Parallel Fluent Job

```
#!/bin/bash
#
# set shell
#PBS -S /bin/bash
#
#### Job Name
#PBS -N FluentParallelTest
#
# run simple fluent job, works on a single node or multiple nodes
# set default resource requirements for job
# - these can be overridden on the qsub command line
# (this is for a 4 processor job requesting 1 hr 30 minutes)
#PBS -l select=1:ncpus=4
#PBS -l walltime=01:30:00
#
# request that regular output (stdout) and
# terminal output (stderr) go to the same file
#PBS -j oe
#
# set mail options to send job notifications
#PBS -m abe
#PBS -M r.user@usq.edu.au
#
# set the queue to run job on
#PBS -q default

# change to directory from which job was submitted
cd $PBS_O_WORKDIR

# set number of processors to run on
# (list of node names is in file $PBS_NODEFILE)
nprocs=`wc -l $PBS_NODEFILE | awk '{ print $1 }'`

# specifies the version of ANSYS FLUENT to run
version=3d

# load ansys module so that we find the fluent command
module load ansys/16.2

# run default version of Fluent in 3d mode in parallel over $nprocs processors
# fluent commands are in file Fluent_par.jou, output messages go to output_file
fluent $version -t$nprocs -cnf=$PBS_NODEFILE -g -i Fluent_par.jou > output_file
```

# Reference


1. Ansys Documentation