

ANSYS Fluent

ANSYS Fluent software contains the broad physical modeling capabilities needed to model flow, turbulence, heat transfer, and reactions for industrial applications ranging from air flow over an aircraft wing to combustion in a furnace, from bubble columns to oil platforms, from blood flow to semiconductor manufacturing, and from clean room design to wastewater treatment plants. Special models that give the software the ability to model in-cylinder combustion, aeroacoustics, turbomachinery, and multiphase systems have served to broaden its reach.

>1. Common way to run Fluent over pbs file

To run ANSYS Fluent in batch mode you can utilize/modify the default fluent.pbs script and execute it via the qsub command.

```
#!/bin/bash
#PBS -S /bin/bash
#PBS -l nodes=2:ppn=16
#PBS -q qprod
#PBS -N $USER-Fluent-Project
#PBS -A XX-YY-ZZ

#! Mail to user when job terminate or abort
#PBS -m ae

#!change the working directory (default is home directory)
#cd <working directory> (working directory must exists)
WORK_DIR="/scratch/$USER/work"
cd $WORK_DIR

echo Running on host `hostname`
echo Time is `date`
echo Directory is `pwd`
echo This jobs runs on the following processors:
echo `cat $PBS_NODEFILE`

#### Load ansys module so that we find the cfx5solve command
module load ansys

# Use following line to specify MPI for message-passing instead
NCORES=`wc -l $PBS_NODEFILE |awk '{print $1}'`

/ansys_inc/v145/fluent/bin/fluent 3d -t$NCORES -cnf=$PBS_NODEFILE -g -i fluent.jou
```

Header of the pbs file (above) is common and description can be find on this site. SVS FEM recommends to utilize sources by keywords: nodes, ppn. These keywords allows to address directly the number of nodes (computers) and cores (ppn) which will be utilized in the job. Also the rest of code assumes such structure of allocated resources.

Working directory has to be created before sending pbs job into the queue. Input file should be in working directory or full path to input file has to be specified. Input file has to be defined by common Fluent journal file which is attached to the Fluent solver via parameter -i fluent.jou

Journal file with definition of the input geometry and boundary conditions and defined process of solution has e.g. the following structure:

```
/file/read-case aircraft_2m.cas.gz
/solve/init
init
/solve/iterate
10
/file/write-case-dat aircraft_2m-solution
/exit yes
```

The appropriate dimension of the problem has to be set by parameter (2d/3d).

>2. Fast way to run Fluent from command line

```
fluent solver_version [FLUENT_options] -i journal_file -pbs
```

This syntax will start the ANSYS FLUENT job under PBS Professional using the qsub command in a batch manner. When resources are available, PBS Professional will start the job and return a job ID, usually in the form of *job_ID.hostname*. This job ID can then be used to query, control, or stop the job using standard PBS Professional commands, such as qstat or qdel. The job will be run out of the current working directory, and all output will be written to the file fluent.o> *job_ID*.

3. Running Fluent via user's config file

The sample script uses a configuration file called pbs_fluent.conf if no command line arguments are present. This configuration file should be present in the directory from which the jobs are submitted (which is also the directory in which the jobs are executed). The following is an example of what the content of pbs_fluent.conf can be:

```
input="example_small.flin"
case="Small-1.65m.cas"
```

```
fluent_args="3d -pmyrinet"
outfile="fluent_test.out"
mpp="true"
```

The following is an explanation of the parameters:

input is the name of the input file.

case is the name of the .cas file that the input file will utilize.

fluent_args are extra ANSYS FLUENT arguments. As shown in the previous example, you can specify the interconnect by using the -p interconnect command. The available interconnects include ethernet (the default), myrinet, class="monospace"> infiniband, vendor, altix>, and crayx. The MPI is selected automatically, based on the specified interconnect.

outfile is the name of the file to which the standard output will be sent.

mpp="true" will tell the job script to execute the job across multiple processors.

To run ANSYS Fluent in batch mode with user's config file you can utilize/modify the following script and execute it via the qsub command.

```
#!/bin/sh
#PBS -l nodes=2:ppn=4
#PBS -l qprod
#PBS -N $USE-Fluent-Project
#PBS -A XX-YY-ZZ

cd $PBS_O_WORKDIR

#We assume that if they didn't specify arguments then they should use the
#config file if [ "xx${input}${case}${mpp}${fluent_args}zz" = "xxzz" ]; then
if [ -f pbs_fluent.conf ]; then
    . pbs_fluent.conf
else
    printf "No command line arguments specified, "
    printf "and no configuration file found. Exiting n"
fi
fi

#Augment the ANSYS FLUENT command line arguments case "$mpp" in
true)
    #MPI job execution scenario
    num_nodes='cat $PBS_NODEFILE | sort -u | wc -l'
    cpus='expr $num_nodes * $NCPUS'
    #Default arguments for mpp jobs, these should be changed to suit your
```

```

#needs.
fluent_args="-t${cpus} $fluent_args -cnf=$PBS_NODEFILE"
;;
*)
#SMP case
#Default arguments for smp jobs, should be adjusted to suit your
#needs.
fluent_args="-t$NCPUS $fluent_args"
;;
esac
#Default arguments for all jobs
fluent_args="-ssh -g -i $input $fluent_args"

echo "----- Going to start a fluent job with the following settings:
Input: $input
Case: $case
Output: $outfile
Fluent arguments: $fluent_args"

#run the solver
/ansys_inc/v145/fluent/bin/fluent $fluent_args > $outfile

It runs the jobs out of the directory from which they are submitted
(PBS_O_WORKDIR).

```

4. Running Fluent in parallel

Fluent could be run in parallel only under Academic Research license. To do so this ANSYS Academic Research license must be placed before ANSYS CFD license in user preferences. To make this change `anslic_admin` utility should be run

```

/ansys_inc/shared_les/licensing/lic_admin/anslic_admin

```

ANSLIC_ADMIN Utility will be run

ANSYS Academic Research license should be moved up to the top of the list.

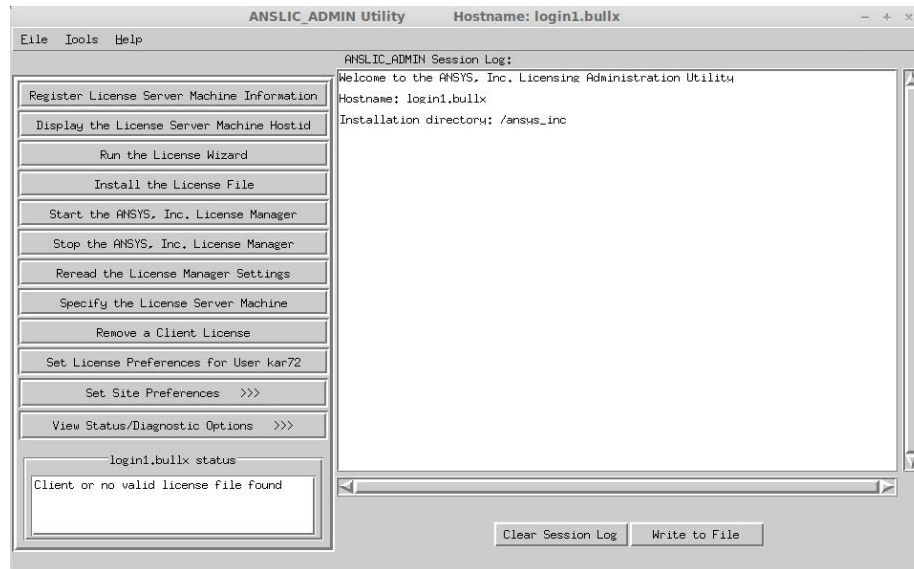


Figure 1:

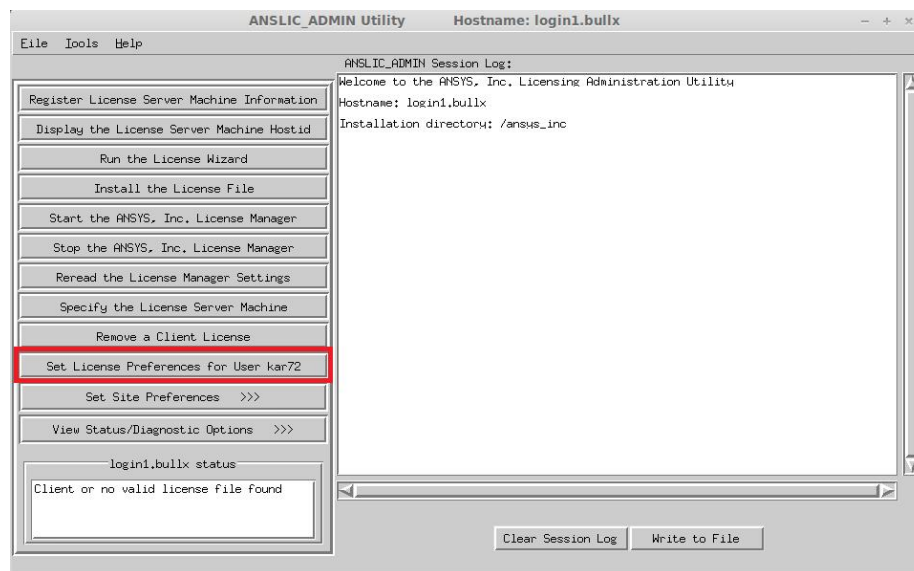


Figure 2:

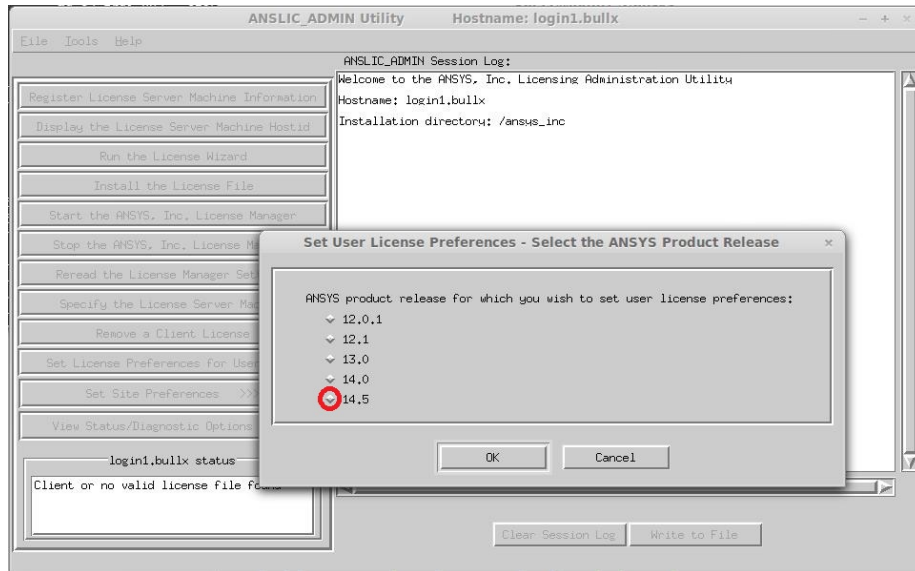


Figure 3:

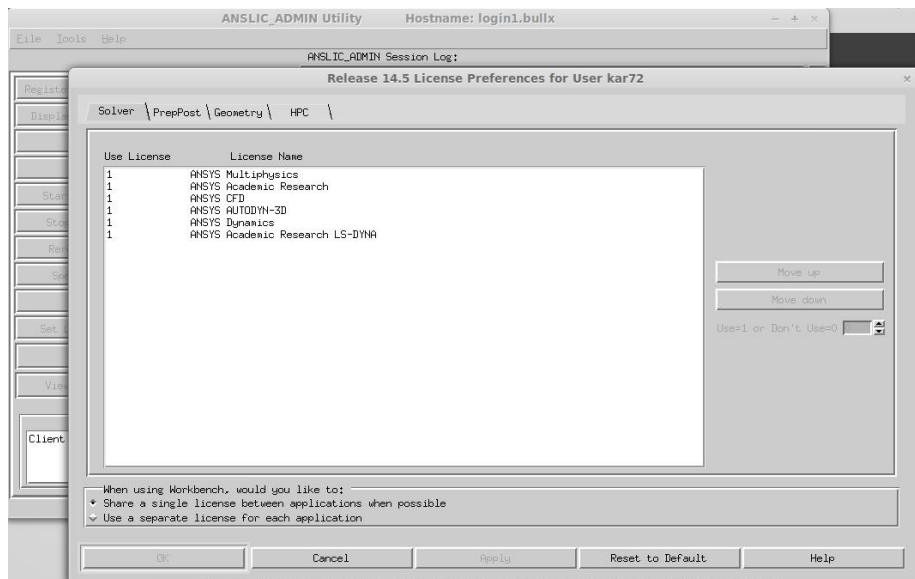


Figure 4: