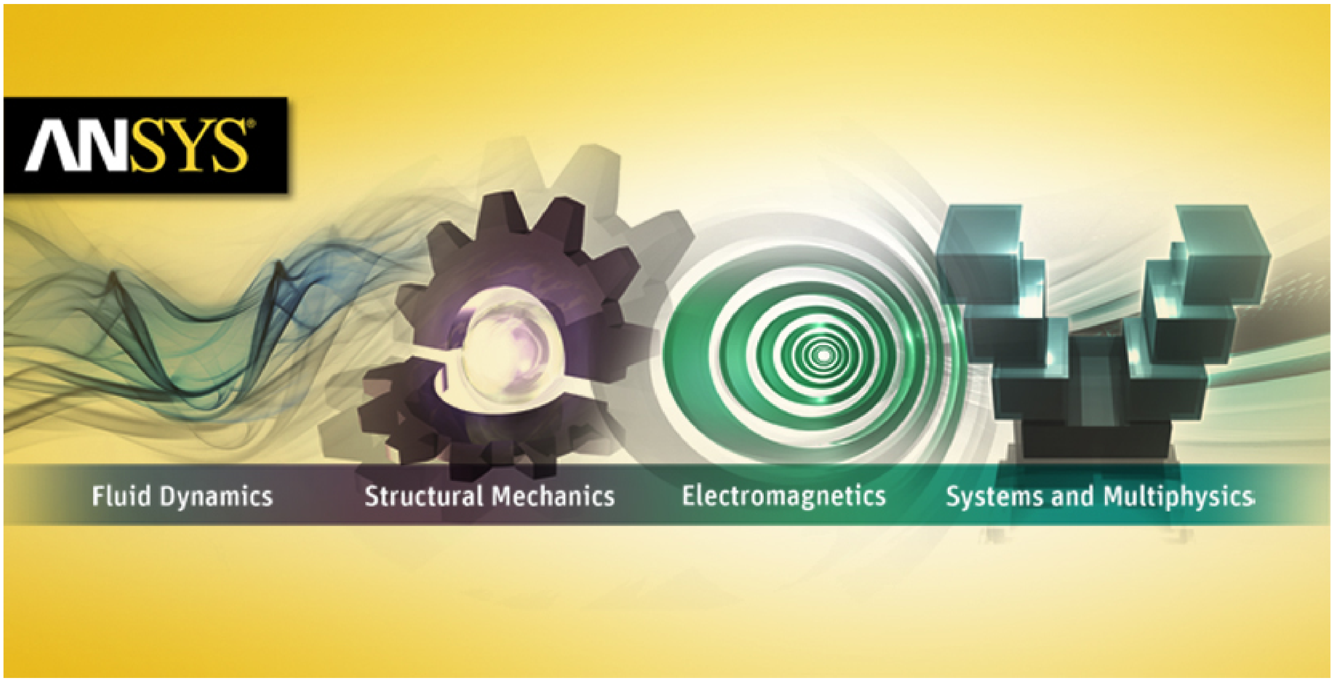

Running ANSYS FLUENT Under PBS Professional

ANSYS, Inc.
Southpointe
275 Technology Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
http://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

Release 14.0
November 2011

Copyright and Trademark Information

Disclaimer Notice

THIS ANSYS SOFTWARE PRODUCT AND PROGRAM DOCUMENTATION INCLUDE TRADE SECRETS AND ARE CONFIDENTIAL AND PROPRIETARY PRODUCTS OF ANSYS, INC., ITS SUBSIDIARIES, OR LICENSORS. The software products and documentation are furnished by ANSYS, Inc., its subsidiaries, or affiliates under a software license agreement that contains provisions concerning non-disclosure, copying, length and nature of use, compliance with exporting laws, warranties, disclaimers, limitations of liability, and remedies, and other provisions. The software products and documentation may be used, disclosed, transferred, or copied only in accordance with the terms and conditions of that software license agreement.

ANSYS, Inc. is certified to ISO 9001:2008.

U.S. Government Rights

For U.S. Government users, except as specifically granted by the ANSYS, Inc. software license agreement, the use, duplication, or disclosure by the United States Government is subject to restrictions stated in the ANSYS, Inc. software license agreement and FAR 12.212 (for non-DOD licenses).

Third-Party Software

See the legal information in the product help files for the complete Legal Notice for ANSYS proprietary software and third-party software. If you are unable to access the Legal Notice, please contact ANSYS, Inc.

Published in the U.S.A.

Table of Contents

About This Document

This document provides general information about running ANSYS FLUENT under PBS Professional. Examples have also been included, where available.

Information in this document is presented in the following chapters:

This document is made available via the ANSYS, Inc. website for your convenience. Please contact Altair Engineering, Inc. (`http://www.altair.com/`) directly for support of their product.

Chapter 1: Introduction

Altair PBS Professional is an open workload management tool for local and distributed environments. You can use PBS Professional when running ANSYS FLUENT simulations, and thereby control the number of jobs running and dynamically monitor the licenses. This document provides general information about running ANSYS FLUENT under PBS Professional, and is made available via the ANSYS, Inc. website for your convenience. Please contact Altair Engineering, Inc. (`http://www.altair.com/`) directly for support of their product.

For more information, please see the following section:

1.1. Overview of Running ANSYS FLUENT Jobs with PBS Professional

For more information, please see the following sections:

1.1.1. Requirements

- Standard
  - PBS Professional 7.1–11.0
  - FLUENT 6.0–ANSYS FLUENT 14.0

> **Important**
>
> Running ANSYS FLUENT under PBS Professional is not supported on Windows.

- Optional
  - Checkpoint restart scripts must be obtained/written to use the checkpoint restart functionality

1.1.2. ANSYS FLUENT and PBS Professional Communication

- PBS Professional has the ability to send signals to the ANSYS FLUENT processes inside a PBS Professional job.
- Dynamic resource functionality provides a way to monitor the availability of software licenses. Please refer to the PBS Professional Administrators Guide for more information on creating custom resources.

Chapter 2: Configuring PBS Professional for ANSYS FLUENT

PBS Professional does not require any special configuration to run a normal ANSYS FLUENT job. Configuration is needed for more advanced functionality, such as controlling the number of jobs that can run with a static limit and dynamic license monitoring.

For more information, please see the following section:
2.1. Job Limiting

2.1. Job Limiting

PBS Professional offers several ways to limit the number of jobs that can run. Two of the more application-specific methods will be illustrated here. When no special limiting actions are taken, PBS Professional establishes limits based on available resources and the requests made by the job submission (i.e., memory, number of CPUs). Administrators should read Section 9 of the PBS Professional 7.1 Administrators Guide for information on creating custom resources and applying techniques mentioned in this section.

For more information, please see the following sections:
2.1.1. Static Limits
2.1.2. Dynamic License Monitoring

2.1.1. Static Limits

A static limit could be useful in a scenario where the cluster has dedicated licenses of a constant quantity. In this case, PBS Professional could simply have a maximum limit set and it would keep track of the jobs it runs itself. Static limits could be applied at the serverwide level, or could be applied in different queues individually. The limit can also be applied to individual nodes to control how many ANSYS FLUENT jobs (if any) a certain node can run at once. The examples that follow outline a couple of common uses for resources in limiting application execution.

More detailed information on the conventions used in the examples that follow can be found in the PBS Professional Administrators Guide and/or PBS Professional Release Notes.

2.1.1.1. Examples

- To create a resource named `fluent` that controls how many jobs can run at one time on the server, the following content should be added by the administrator:

File:	`$PBS_HOME/server_priv/resourcedef`
Content:	`fluent type=long flag=q`
File:	`$PBS_HOME/sched_priv/sched_config`
Content:	`resources: "` *existing_content* `fluent"`

  Note that `$PBS_HOME` should be defined in `/etc/pbs.conf`, and *existing_content* is comprised of the existing variable definitions (either default or customized). Then the `qmgr` interpreter should be launched and the following command entered:

```
qmgr> set server resources_available.fluent = X
```

- To create a resource named `fluent_bool` that controls whether or not node `nodename` can run ANSYS FLUENT jobs, the following content should be added by the administrator:

File:	`$PBS_HOME/sever_priv/resourcedef`
Content:	`fluent_bool type=Boolean flag=h`
File:	`$PBS_HOME/sched_priv/sched_config`
Content:	`resources: "`*existing_content* `fluent"`

  Note that $PBS_HOME should be defined in `/etc/pbs.conf`, and *existing_content* is comprised of the existing variable definitions (either default or customized). Then the `qmgr` interpreter should be launched and the following command entered:

```
qmgr> set node nodename resources_available.fluent_bool = True
```

2.1.2. Dynamic License Monitoring

Dynamic license monitoring is useful when a cluster may be sharing a license pool with noncluster users. In this case PBS Professional cannot manage licenses on its own by simply counting how many jobs it runs. If this is your situation, you can write a script to check the license server with lmstat. PBS Professional will take a value returned by a script and assign it to a dynamic resource at the beginning of a scheduling cycle. By doing this the scheduler can hold jobs in the queue until the resource has a high enough value to fulfill the license amount request.

2.1.2.1. Example

The following content would need to be entered by an administrator:

File:	`$PBS_HOME/server_priv/resourcedef`
Content:	`fluent_lic type=long`
File:	`$PBS_HOME/sched_priv/sched_config`
Content	`resources: "`*existing_content* `fluent_lic" server_dyn_res: fluent_lic !/path/to/command`

Note that $PBS_HOME should be defined in `/etc/pbs.conf`, and *existing_content* is comprised of the existing variable definitions (either default or customized).

Chapter 3: Running an ANSYS FLUENT Simulation under PBS Professional

For more information, please see the following sections:

3.1. Using the Integrated PBS Professional Capability

For more information, please see the following sections:

3.1.1. Overview

One option of using PBS Professional with ANSYS FLUENT is to use the PBS Professional launching capability integrated directly into ANSYS FLUENT. In this mode, ANSYS FLUENT is started from the command line with the additional `-pbs` argument. ANSYS FLUENT then takes responsibility for relaunching itself under PBS Professional. This has the following advantages:

- The command line usage is very similar to the non-RMS (resource management system) usage.
- You do not need to write a separate script.

The integrated PBS Professional capability is intended to simplify usage for the most common situations. If you desire more control over the PBS Professional `qsub` options for more complex situations or systems (or if you are using an older version of ANSYS FLUENT), you can always write or adapt a PBS Professional script that starts ANSYS FLUENT in the desired manner (see *Using Your Own Supplied Job Script* (p. 7) and *Using Altair's Sample Script* (p. 7) for details).

3.1.2. Usage

The integrated PBS Professional capability is activated by simply adding the `-pbs` option to the ANSYS FLUENT line command:

`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*.

3.1.3. Examples

- Submit a parallel, 4-process job using a journal file `fl5s3.jou`:

```
 > fluent 3d -t4 -i fl5s3.jou -pbs
Relaunching fluent under PBSPro
134.les29
```

In the previous example, note that `134.les29` is returned from `qsub`. `134` is the job ID, while `les29` is the hostname on which the job was started.

- Check the status of the job:

```
 > qstat -s

les29:
                                                         Req’d  Req’d   Elap
Job ID        Username Queue    Jobname   SessID NDS TSK Memory Time  S Time
------------- -------- -------- --------- ------ --- --- ------ ----- - -----
134.les29     user1    workq    fluent     11958   4   4    --    --  R 00:00
   Job run at Thu Jan 04 at 14:48

> qstat -f 134
Job Id: 134.les29
    Job_Name = fluent
    Job_Owner = user1@les29
<... additional status of jobID 134>
```

The first command in the previous example lists all of the jobs in the queue. The second command lists the detailed status about the given job.

After the job is complete, the job will no longer show up in the output of the `qstat` command. The results of the run will then be available in the file `fluent.o134`.

3.1.4. Limitations

The integrated PBS Professional capability in ANSYS FLUENT 14.0 has the following limitations:

- The PBS Professional commands (such as `qsub`) must be in the users path.
- The output is directed to the file `fluent.o`*job_ID*.
- For parallel jobs, ANSYS FLUENT processes are placed on available compute resources using the `qsub` options `-l select=`N. (Optionally, for better control over the placement of processes, you can set the environment variable SET_PBS_PPN to a value of `M`, where `M` is the number of processes per node. Setting this variable will place ANSYS FLUENT processes using the option `-l nodes=`*total_processes_requested*`/M:`ppn=`M`. The value used for `M` has to be same for all nodes. If you desire more sophisticated placement, you may write separate PBS Professional scripts as described in *Using Your Own Supplied Job Script* (p. 7).)
- RMS-specific checkpointing is not available. PBS Professional only supports checkpointing via mechanisms that are specific to the operating system on SGI and Cray systems. The integrated PBS Professional capability is based on saving the process state, and is not based on the standard application restart files (e.g., ANSYS FLUENT case and data files) on which the LSF and SGE checkpointing is based. Thus, if you need to checkpoint, you should checkpoint your jobs by periodically saving the ANSYS FLUENT data file via the journal file.

3.2. Using Your Own Supplied Job Script

PBS Professional can accept any custom shell script for running ANSYS FLUENT that you may already use. Note that this shell script may require modification to move files between machines in the cluster. The shell script should also endeavor to clean up after itself when execution completes/fails.

3.3. Using Altair's Sample Script

Altair provides a short sample script for running ANSYS FLUENT inside of PBS Professional. This script uses a configuration file to tell it several pieces of information that it needs (outlined in the section that follows). Optionally, these pieces of information can be specified on the command line. The `fluent_args` option does not need to be used for specifying the number of CPUs in the default script. The script will address this for the user, based on their PBS Professional resource requests.

For more information, please see the following sections:

3.3.1. Configuration File Example
3.3.2. Altair's Sample Script
3.3.3. Submitting Altair's Sample Script
3.3.4. Epilogue/Prologue

3.3.1. Configuration File Example

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`, `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.

3.3.2. Altair's Sample Script

Altair's Sample script is not intended to be a full solution for running ANSYS FLUENT jobs in PBS Professional, but rather a simple starting point. It runs the jobs out of the directory from which they are submitted (PBS_O_WORKDIR). Care should be taken to submit jobs from locations that are going to be available on any node (perhaps via NFS).

```
#!/bin/sh
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

#Set up the license information (Note: you need to substitute your own
#port and server in this command)
export LM_LICENSE_FILE="port@server"

#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/v140/fluent/bin/fluent $fluent_args  > $outfile
```

Note that for versions of ANSYS FLUENT prior to 12.0, the final line of the sample script should be changed to the following:

```
usr/apps/Fluent.Inc/bin/fluent $fluent_args  > $outfile
```

3.3.3. Submitting Altair’s Sample Script

- To submit the script with command line arguments:
  - qsub -l *resource_requests* -v input=*input_file*,case=*case_file*,fluent_args= *fluent_arguments*,outfile=*output_file*[,mpp="true"] fluent-job.sh
- To submit the script without command line arguments:
  - Edit fluent_pbs.conf to suit your needs
  - qsub -l *resource_requests* fluent-job.sh

Note that the resources necessary for the job (i.e., *resource_requests*) should be entered with the proper syntax. For more information about requesting resources, see the PBS Professional 7.1 Users Guide Section 4.3.

3.3.4. Epilogue/Prologue

PBS Professional provides the ability to script some actions immediately before the job starts and immediately after it ends (even if it is removed). The epilogue is a good place to put any best effort cleanup and stray process functionality that might be necessary to clean up after a job. For instance, one could include functionality to clean up the working/scratch area for a job if is deleted before its completion. ANSYS FLUENT provides a way to get rid of job-related processes in the form of a shell script in the work directory (`cleanup-fluent-`*host-pid*). It could be useful to have the epilogue call this script to ensure all errant processes are cleaned up after a job completes.

3.4. Monitoring/Manipulating Jobs

For more information, please see the following sections:

3.4.1. Monitoring the Progress of a Job
3.4.2. Removing a Job from the Queue

3.4.1. Monitoring the Progress of a Job

The `qstat` command is used to monitor the status of jobs in a PBS Professional queue. When no job ID is specified, the `qstat` command will display all jobs. If you use `qstat -an job_ID` you will receive information about a specific job. This information includes the location at which the job is executing, which will be listed under the `Job ID` column.

```
% qstat -an

 server:
                                                          Req’d  Req’d   Elap
 Job ID        Username Queue   Jobname    SessID NDS TSK Memory Time  S Time
 ------------- -------- ------- ---------- ------ --- --- ------ ----- - -----
 32.server     user     workq   fluent-job  15886  --   1    --    --  R 01:07
    server/0
```

There are several job states to be aware of when using `qstat`, which are listed under the S column. The two main states you will see are Q and R. Q indicates that the job is waiting in the queue to run. At this point the scheduler has not found a suitable node or nodes to run the job. Once the scheduler has found a suitable area to run the job and has sent the job to run, its state will be set to R. Full details on the different job statuses reported by `qstat` can be found in the PBS Professional 7.1 Users Guide Section 6.1.1.

3.4.2. Removing a Job from the Queue

The `qdel` *job_ID* command will delete the job indicated by *job_ID*.

Index