Installing and running COMSOL 4.3a on a Linux cluster

Introduction

- This quick guide explains how to install and operate COMSOL Multiphysics 4.3a on a Linux cluster. It is a complement to the *COMSOL Installation and Operations Guide** which contains the complete and detailed installation and operations instructions.
- This guide assumes that
  - you are a fully qualified system administrator of the cluster.
  - the cluster is tested fully operational, this guide does not provide instructions how to manage the cluster itself
  - you are using the latest available release of COMSOL, available from http://www.comsol.com/support/download
  - the latest COMSOL software update has been applied, available from http://www.comsol.com/support/updates

*Available at www.comsol.com/iog

COMSOL

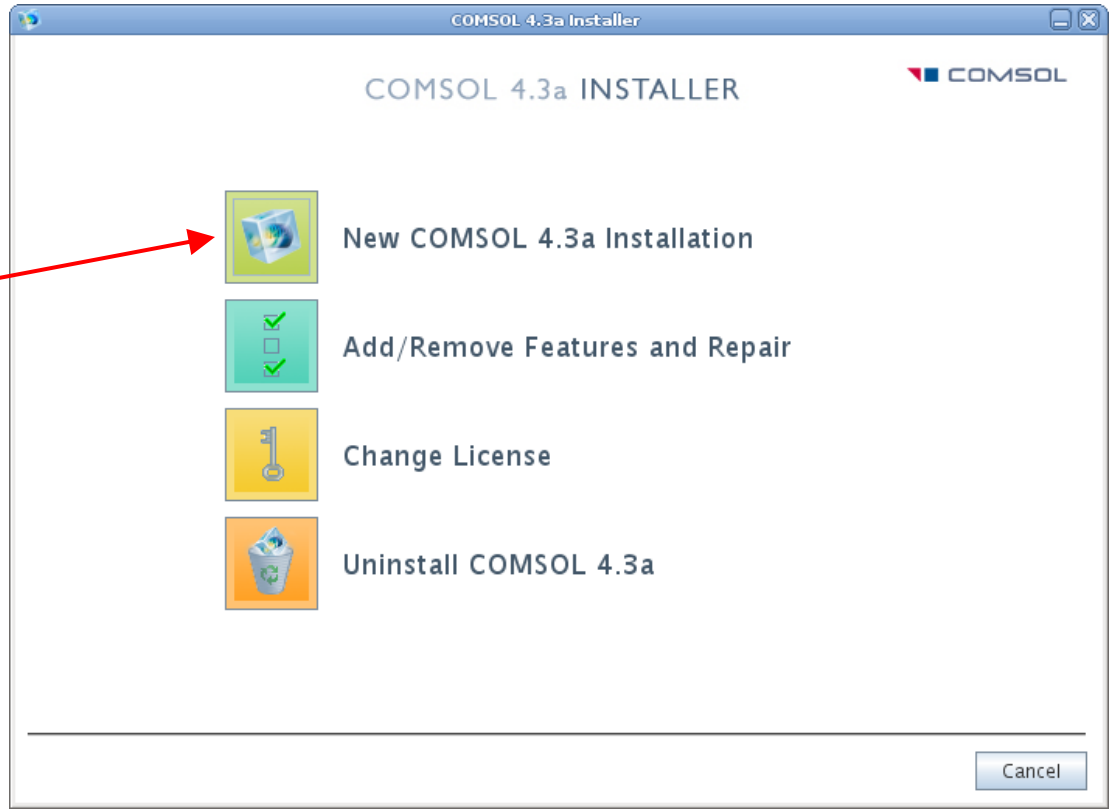
Installing COMSOL on the cluster

- Before you begin
  - Make sure you have received a FNL (Floating Network License) license file from your COMSOL representative. This is the only license type that enables cluster computing components in the software.
  - Make sure you have the COMSOL 4.3a Installation and Operations Guide available. Downloadable at www.comsol.com/iog.
  - Make sure you use an officially supported Linux distribution: www.comsol.com/products/requirements/linux/
- Installation of the COMSOL software is usually only needed on the headnode. The installation is shared by all compute nodes in the cluster environment, typically over an NFS file system or similar.
- Insert the COMSOL DVD and run the setup script found in the root of the DVD:

```
sh setup
```

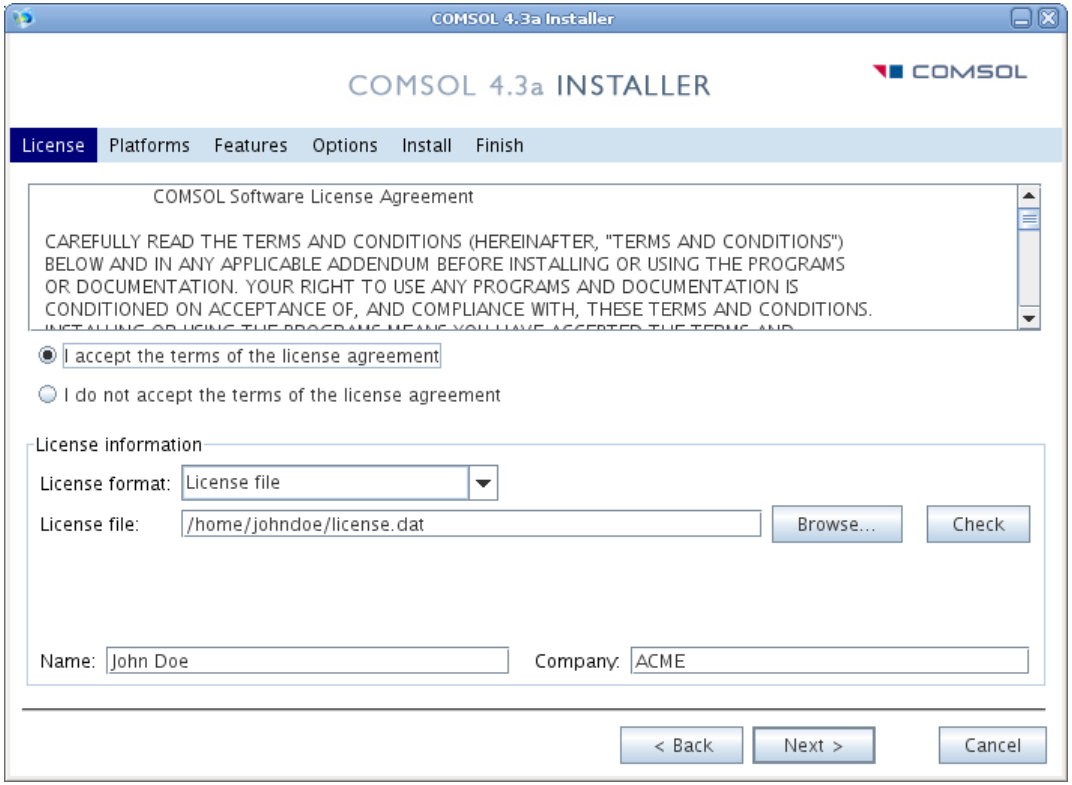
Installing COMSOL on the cluster, cont

- Click "New COMSOL 4.3a Installation" in the installer window.



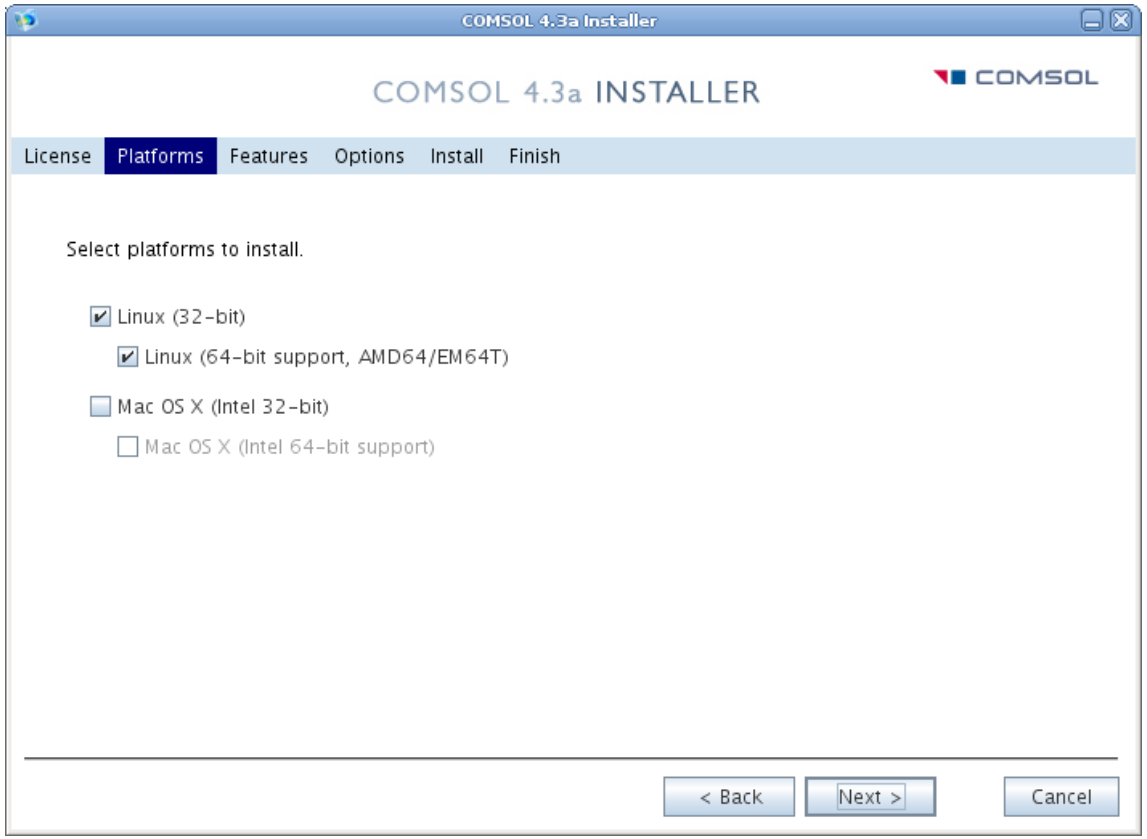
Installing COMSOL Multiphysics, cont.

- After accepting the license agreement, provide the required license information.



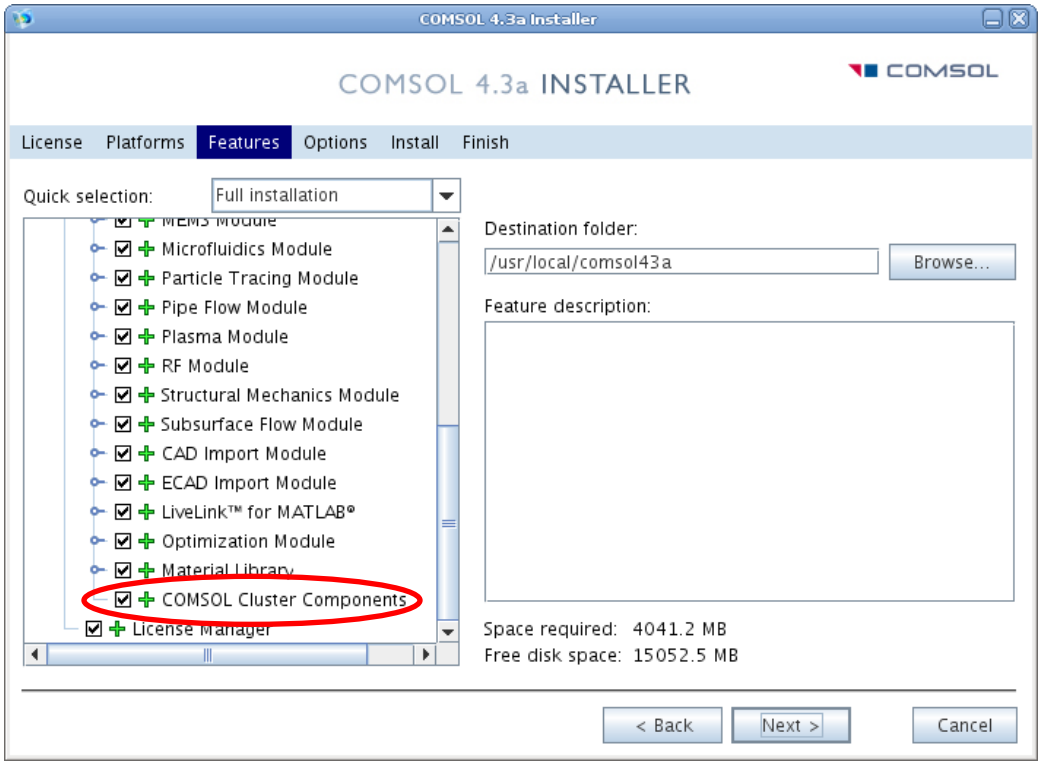
Installing COMSOL Multiphysics, cont.

- Review the platform selection.



Installing COMSOL Multiphysics, cont.

- Choose the required features and the destination folder. Cluster Components must be included (default choice).

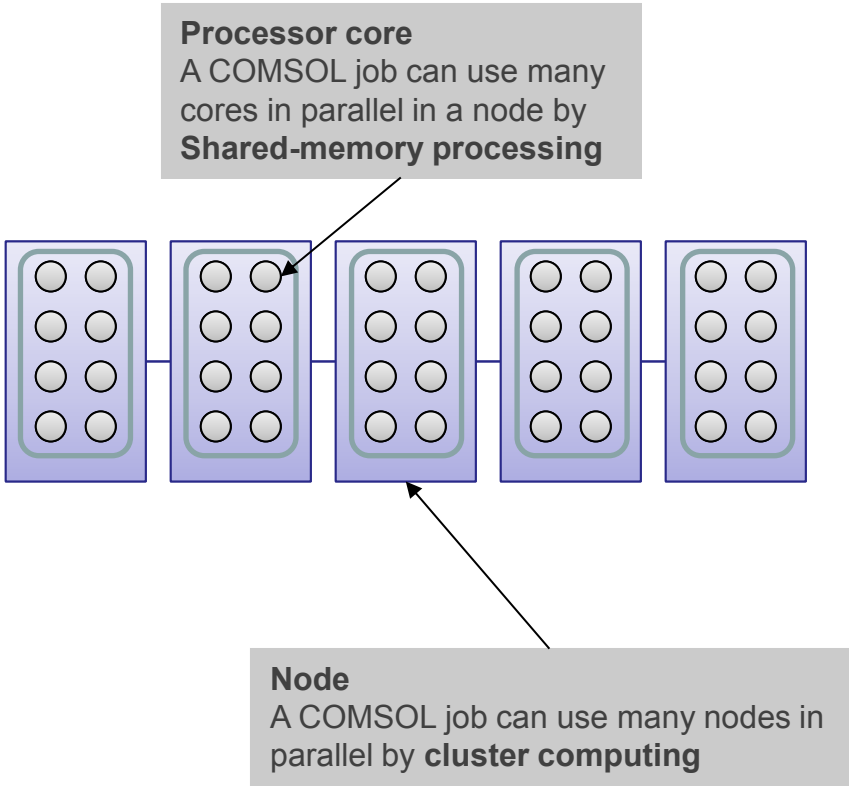


Installing COMSOL Multiphysics, cont.

- Complete the additional forms, click Install.

Running COMSOL in parallel

- COMSOL can run a job on many cores
  in parallel (Shared-memory processing or multithreading)
- COMSOL can run a job run on many physical nodes (cluster computing)
- Both parallel operations above can be used in combination to maximize parallelization benefits
- Cluster computing requires a floating network license



Four ways to run a cluster job, overview

- You can choose from different ways to submit a cluster job, depending on your personal preference:
  1. Submit cluster-enabled batch job from the Linux command prompt or scheduler script
  2. Branch off cluster-enabled batch jobs from a COMSOL Desktop GUI process started on headnode
  3. Start a cluster-enabled COMSOL desktop GUI on the headnode and work interactively with cluster jobs
  4. Start the COMSOL Desktop GUI as a client on a local PC or Macintosh and connect to a cluster-enabled COMSOL server on the Linux headnode and work interactively

Four Ways to Run a Cluster job, details

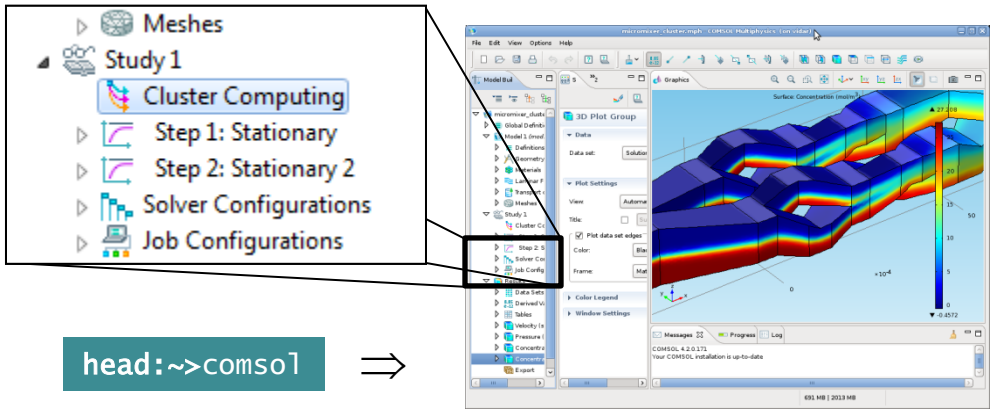
1. Submit batch job from the command line of the headnode
   - No GUI
   - Direct control through commands, can be used in shell scripts
   - Requires completed and saved model mph file
   - Recommended for use together with schedulers, like LSF or PBS.

See **comsol -h** for details

```
head:~>comsol -nn 8 batch -inputfile in.mph
```

Four Ways to Run a Cluster job, cont.

2. Branch off cluster-enabled batch jobs from a COMSOL Desktop GUI started on headnode
   - Allows GUI model work, and batch job submission from within the GUI
   - Limited command-line proficiency needed
   - Allows you to submit jobs from one COMSOL Desktop GUI process. With the new Cluster Sweep feature in version 4.3 it is possible to submit a single batch job from the COMSOL GUI and continue working in the GUI while the cluster job is computing in the background; this requires a license for one concurrent user.
   - Can be used together with custom schedulers like LSF, PBS, and OGS/GE (SGE).

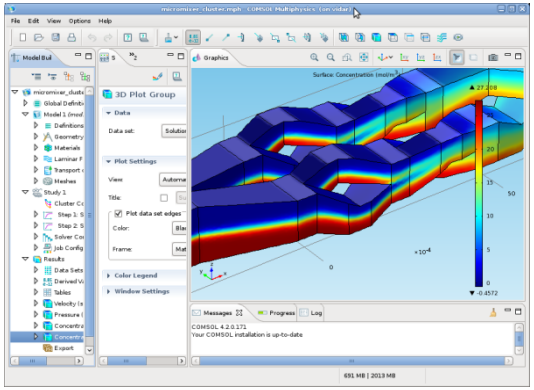


Four Ways to Run a Cluster job, cont.

3. Start a cluster-enabled COMSOL desktop GUI on headnode
   - Allows interactive modeling and cluster jobs by the click of a button from within the COMSOL Desktop GUI.
   - Desktop GUI will be mostly locked up while computing the solution
   - Limited command line proficiency needed

```
head:~>comsol -nn 8
```

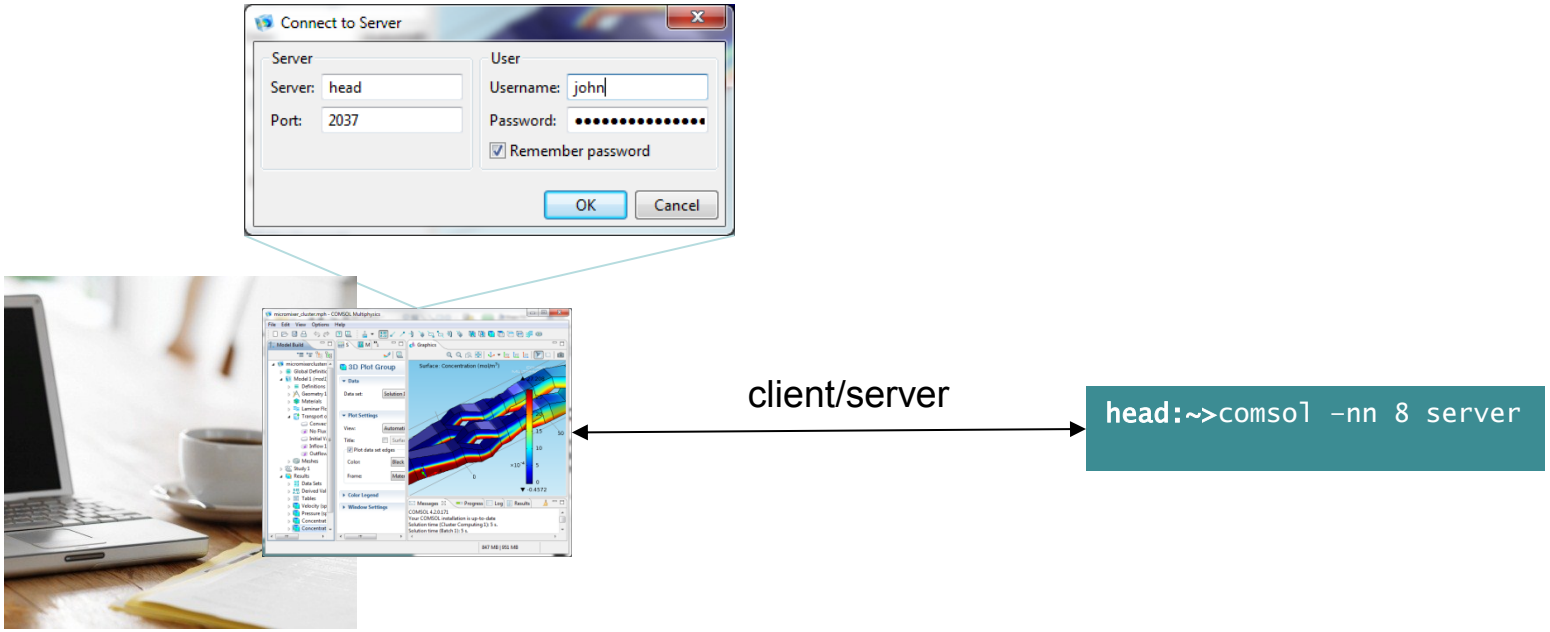
⇒



COMSOL

Four Ways to Run a Cluster job, cont.

4. Start the COMSOL Desktop GUI as a client on a local PC or Macintosh and connect to a cluster-enabled COMSOL server on the Linux headnode and work interactively
   - Same as alternative 3 above, except that the COMSOL desktop is on a work computer, separated from the cluster.



Example 1: Command line batch submission

a) Shorthand version. The file “hostfile” should contain the host names of the computer nodes that you intend to use.

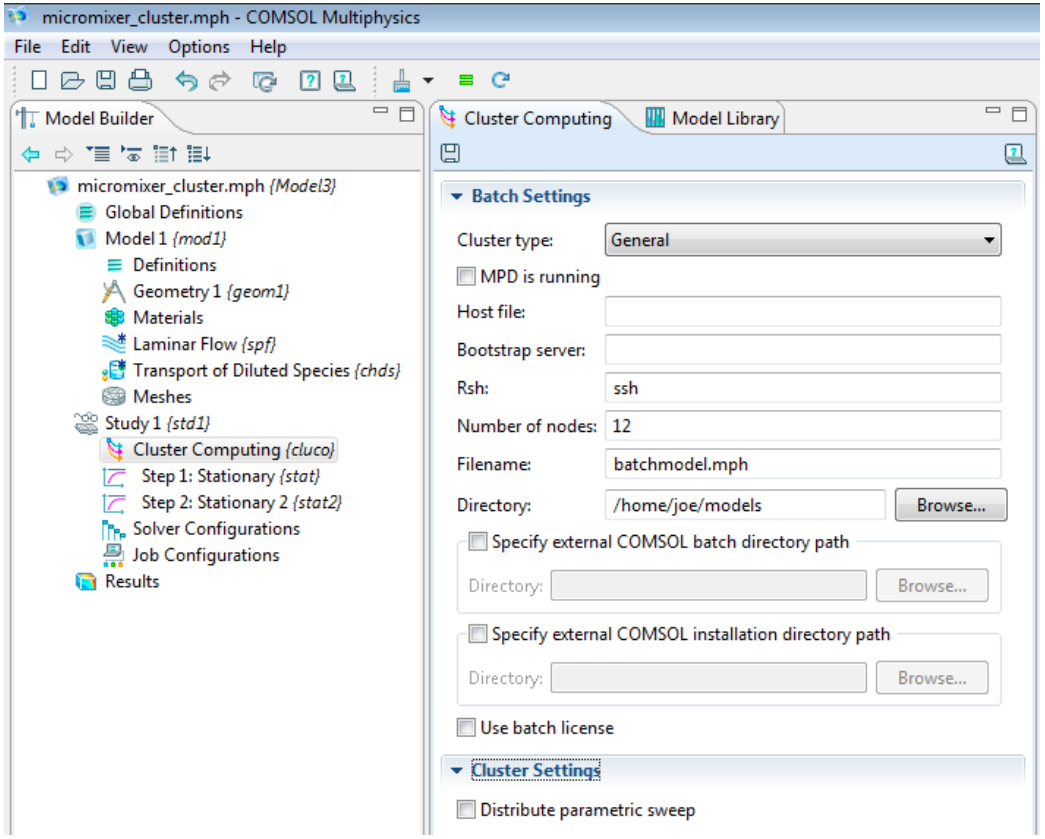
```
head:~> comsol -nn 8 batch -f hostfile -inputfile in.mph -outputfile out.mph -batchlog b.log
```

b) Allow the Intel MPI library to automatically detect the number of nodes that is scheduled for the job:

```
head:~> comsol -clustersimple batch -inputfile in.mph -outputfile out.mph -batchlog b.log
```

Example 2, Desktop GUI, branch off batch job for cluster computation

- See details in COMSOL Multiphysics User's Guide, section *Running COMSOL in Parallel*



Find more information

- For a fully documented Desktop GUI cluster example, see the model library examples COMSOL_Multiphysics/Tutorial_Models micromixer_cluster and thermal_actuator_jh_distributed
- For troubleshooting, see
  - COMSOL Installation and Operations Guide, the section *Troubleshooting Distributed COMSOL and MPI.*
  - knowledge base solution 1001 http://www.comsol.com/support/knowledgebase/1001/
- For command line options, see ”comsol –h” and COMSOL Installation and Operations Guide:
  - *Running COMSOL on Clusters.*
  - *COMSOL Cluster Commands.*
- COMSOL Multiphysics User’s Guide, sections:
  - *Parallel COMSOL.*
  - *Running COMSOL in Parallel.*

COMSOL

Appendix 1: Example of job submission from LSF scheduler

```
#!/bin/sh

# Rerun process if node goes down, but not if job crashes
# Cannot be used with interactive jobs.
#BSUB -r

# Job name
#BSUB -J comsoltest

# Number of processes.
#BSUB -n 20

# Redirect screen output to output.txt
#BSUB -o output.txt

rm -rf output.txt

# Launch the COMSOL batch job
comsol -clustersimple batch -inputfile in.mph -outputfile
out.mph
```

Appendix 2: Example of job submission from PBS scheduler

```
#!/bin/bash
# ##############################################################################
#
export nn=2
export np=8
export inputfile="simpleParametricModel.mph"
export outputfile="outfile.mph"
#
qsub -V -l nodes=${nn}:ppn=${np} <<'__EOF__'
#
#PBS -N COMSOL
#PBS -q dp48
#PBS -o $HOME/cluster/job_COMSOL_$$.log
#PBS -e $HOME/cluster/job_COMSOL_$$.err
#PBS -r n
#PBS -m a -M email@domain.com
#
echo "------------------------------------------------------------------------------"
echo "--- Starting job at: `date`"
echo
#
cd ${PBS_O_WORKDIR}
echo "--- Current working directory is: `pwd`"
#
np=$(wc -l < $PBS_NODEFILE)
echo "--- Running on ${np} processes (cores) on the following nodes:"
cat $PBS_NODEFILE
#
echo "--- parallel COMSOL RUN"
/share/comsol43a/bin/comsol -clustersimple batch -mpiarg -rmk -mpiarg pbs -inputfile $inputfile \
-outputfile $outputfile -batchlog batch_COMSOL__$$.log
echo
echo "--- Job finished at: `date`"
echo "------------------------------------------------------------------------------"
#
__EOF__
```

Appendix 3: Launching on Sun Grid Engine (SGE), and Oracle Grid Engine

- Configuring the Cluster Computing feature in the COMSOL GUI
  - Add a Cluster Computing feature to the model you want to run (see “Example 2” slide).
  - Set the Cluster type to OGS/GE.
  - Specify the Number of Slots, Queue name, etc.

- Command line example for COMSOL batch job

```
qsub -pe <SGE queue name> <number of slots to allocate> \
-o <path to standard out file> -e <path to standard error file> \
-b y <path to COMSOL root installation directory>/bin/comsol \
-nn '${NHOSTS}' batch -inputfile in.mph -outputfile out.mph
```