SGI® OpenFOAM™ Quick Start Guide

007–5817–001

Record of Revision

Version	Description
001	January 2012 Original Printing.

Contents

About This Guide

This publication documents how to install SGI® OpenFOAM™ computational fluid dynamics software on SGI hardware platforms.

Related Publications and Additional Information

This section describes documentation you may find useful, as follows:

- *SGI Performance Suite 1.3 Start Here*

  Describes SGI Performance Suite 1.3 release including release features, software installation, and product overview. It includes a list of active SGI software and hardware manuals.

- *Message Passing Toolkit (MPT) User's Guide*

  Describes industry-standard message passing protocol optimized for SGI computers.

- *MPInside Reference Guide*

  Documents the SGI MPInside MPI profiling tool.

Online caveats, release note updates, and other useful information is available on the SGI OpenFOAM pages on Supportfolio Online at the following location: https://support.sgi.com/

Note: You must be logged onto Supportfolio to access these pages.

Obtaining Publications

You can obtain SGI documentation in the following ways:

- See the SGI Technical Publications Library at: http://docs.sgi.com. Various formats are available. This library contains the most recent and most comprehensive set of online books, release notes, man pages, and other information.
- You can also view man pages by typing `man` *title* on a command line.

Conventions

The following conventions are used throughout this document:

Convention	Meaning
`command`	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
`manpage`(*x*)	Man page section identifiers appear in parentheses after man page names.
variable	Italic typeface denotes variable entries and words or concepts being defined.
`user input`	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. (Output is shown in nonbold, fixed-space font.)
[ ]	Brackets enclose optional portions of a command or directive line.
...	Ellipses indicate that a preceding element can be repeated.

Reader Comments

If you have comments about the technical accuracy, content, or organization of this publication, contact SGI. Be sure to include the title and document number of the publication with your comments. (Online, the document number is located in the front matter of the publication. In printed publications, the document number is located at the bottom of each page.)

You can contact SGI in any of the following ways:

- Send e-mail to the following address:

  techpubs@sgi.com

- Contact your customer service representative and ask that an incident be filed in the SGI incident tracking system.

- Send mail to the following address:

SGI
Technical Publications
46600 Landing Parkway
Fremont, CA 94538

SGI values your comments and will respond to them promptly.

Chapter 1

Introduction

This chapter describes the SGI® OpenFoam™ computational fluid dynamics software.

OpenFOAM is a C++ library, used primarily to create executables, known as applications. The applications fall into two categories:

- Solver applications that are each designed to solve a specific problem in continuum mechanics
- Utilities that are designed to perform tasks that involve data manipulation

The OpenFOAM distribution contains numerous solver applications and utilities covering a wide range of problems, as described in chapter 3 of the *OpenFOAM User Guide* available at http://www.openfoam.com/docs/.

For information on how to use OpenFOAM software, see the *OpenFOAM USER Guide*.

For information on installing the SGI OpenFOAM 2.1.0 package on your SGI system, see Chapter 2, "OpenFOAM Getting Started" on page 3.

Chapter 2

OpenFOAM Getting Started

This chapter describes how to install and use the SGI® OpenFOAM™ 2.1.0 package on SGI hardware systems.

Create an OpenFOAM Environment Settings Directory

Before using the SGI OpenFOAM 2.1.0 package, create an `$HOME/.OpenFOAM/2.1.0` directory for options setting files, such as, `prefs.c` or `prefs.sh` so that you can specify any special settings for your environment. This is optional. In these options setting files, you can set variables, such as, `foamCompiler`, `WM_COMPILER`, and `WM_MPLIB` and so on.

Software Requirements

The SGI OpenFOAM 2.1.0 package has been especially setup to run with the default `gcc` compiler on the operating system and SGI MPI. This installation has been compiled with SGI MPT 2.05 header files. You need install SGI MPI 1.3 which contains `sgi-mpt-2.05` and `scotch 5.1.11`. For detailed installation instructions, see the "MPT Installation" section in Chapter 2. Administering MPT in the *Message Passing Toolkit (MPT) User's Guide* available on SGI Technical Publications Library at: http://docs.sgi.com.

OpenFOAM Setup Scripts

You can use one of the following scripts ([t]csh and a [k]sh/bash) to help setup you SGI OpenFOAM environment:

When using the C shell:

```
#!/bin/csh
# setup.cs script:
# Source setup.csh if working with a [t]csh shell.
# Sourcing this script will setup your OpenFOAM environment
# for your shell session. You can source this script upon
# login or from scripts  such as PBSpro scripts.
# This script takes one argument, namely the precision
# which may be either SP (SinglePrecision) or
# DP (Double Precision).
#
if ($#argv != 1) then
   echo ’Usage  : source ./setup.csh SP|DP’
   echo ’Example: source ./setup.csh DP’
   exit
endif
#
# set FOAM_INST_DIR to the path for OpenFOAM installation
#
setenv FOAM_INST_DIR /opt/sgi/OpenFOAM/
#
# set MPI_ROOT to the directory where mpt-2.05 installed
#
setenv MPI_ROOT /opt/sgi/mpt/mpt-2.05
setenv foamDotFile ${FOAM_INST_DIR}/OpenFOAM-2.1.0/etc/cshrc
set path=(. $path)
if ( -f $foamDotFile ) source $foamDotFile           \
                              foamCompiler=system    \
                              WM_COMPILER=Gcc        \
                              WM_PRECISION_OPTION=$1 \
                              WM_MPLIB=SGIMPI
setenv SCOTCH_DIR=/opt/sgi/scotch/5.1.11
setenv LD_LIBRARY_PATH "${SCOTCH_DIR}/lib:\
                        /usr/lib64/paraview-3.12:\
                        ${LD_LIBRARY_PATH}"
setenv ParaView_DIR /usr
setenv PATH ${ParaView_DIR}/bin:${PATH}
setenv PV_PLUGIN_PATH ${FOAM_LIBBIN}/paraview-3.12
# set the prompt
set prompt="sof2.1.0_${WM_PRECISION_OPTION}% "
```

When using the BASH shell:

```
#!/bin/bash
# setup.sh script:
# Source setup.sh if working with sh, ksh or bash shell.
# Sourcing this script will setup your OpenFOAM environment
# for your shell session. You can source this script upon
# login or from scripts such as PBSpro scripts.
# This script takes one argument, namely the precision
# which may be either SP (SinglePrecision) or DP
#(Double Precision).
#
if [ $# != 1 ] ; then
   echo ’Usage  : . ./setup.sh SP|DP’
   echo ’Example: . ./setup.sh SP’
   exit
fi
#
# set FOAM_INST_DIR to the path for OpenFOAM installation
#
export FOAM_INST_DIR=/opt/sgi/OpenFOAM
#
# set MPI_ROOT to the directory where mpt-2.05 installed
#
export MPI_ROOT=/opt/sgi/mpt/mpt-2.05
export foamDotFile=$FOAM_INST_DIR/OpenFOAM-2.1.0/etc/bashrc
export PATH=".:$PATH"
if [ -f $foamDotFile ]; then
   . $foamDotFile foamCompiler=system          \
             WM_COMPILER=Gcc              \
             WM_PRECISION_OPTION=$1       \
             WM_MPLIB=SGIMPI
fi
export SCOTCH_DIR=/opt/sgi/scotch/5.1.11
export LD_LIBRARY_PATH="${SCOTCH_DIR}/lib:\
                        /usr/lib64/paraview-3.12:  \
                        ${LD_LIBRARY_PATH}"
export ParaView_DIR=/usr
export PATH=${ParaView_DIR}/bin:${PATH}
export PV_PLUGIN_PATH=${FOAM_LIBBIN}/paraview-3.12
# set the prompt
PS1="sof2.1.0_${WM_PRECISION_OPTION}% "
```

Note that the default location for the SGI OpenFOAM installation is under the /opt/sgi/OpenFOAM directory. If you choose to install it in a different location, you need to change the FOAM_INST_DIR environment variable in the setup scripts (see "OpenFOAM Setup Scripts" on page 3). If any of the scotch and paraview utilities on your system are currently installed in different locations, edit the SCOTCH_DIR and ParaView_DIR environment variables accordingly.

Verifying Your OpenFOAM Installation

You can verify your OpenFOAM installation by running an example case of incompressible laminar flow in a cavity, as follows:

```
cd $FOAM_RUN/tutorials/incompressible/icoFoam/cavity
   blockMesh
   icoFoam
   paraFoam
```

For information on how to use OpenFOAM software, see the *OpenFOAM USER Guide* available at http://www.openfoam.com/docs/.

OpenFOAM Example Scripts

This section provides example scripts for running OpenFOAM software on various SGI hardware platforms.

OpenFOAM PBS Professional Script for SGI ICE and SGI Rackable Clusters Using SGI MPI

The following is an example script on how to run an OpenFOAM solver job using PBS Professional batch queue scheduler on an SGI ICE, SGI Altix XE, or SGI Rackable system.

```
#!/bin/csh
#
### PBS ###
#PBS -S /bin/csh -j oe -k o -r n
#PBS -l walltime=2:15:00
#
set echo
#
setenv WM_64
#
# submit this script to the PBSpro job queue using the command:
#
```

```
# qsub -l select=<number_compute_nodes>:ncpus=NCPU:mpiprocs=NTHD \
#      -v NPROCS=TOTAL_NTHD ./this_script
#
# where
#
#  NCPU       = Total number of cores per compute nodes including
#               hyperthrading if enabled.
#  NTHD       = Number of mpi threads to be spwaned on each compute node
#  TOTAL_NTHD = Total number of mpi threads for the running task. That
#               TOTAL_NTHD equals NTHD x <number_compute_nodes> in
#               qsub ’select’ arguments.
#
#
cd $PBS_O_WORKDIR
# Specify Single (SP) or Double (DP) precision, example here is SP
setenv OpenFOAM_PRECISION SP
# source the above setup.csh script to setup OpenFOAM environemnt:
source setup.csh SP >& run.log
# unsetenv FOAM_SIGFPE
#
# decompose geometry first
# copy the correct decomposeParDict file to decomposeParDict
#
#
   rm -rf proc*
   cp system/decomposeParDict.${NPROCS} system/decomposeParDict
    (dplace decomposePar -case . -parallel  ) >& decompose.out
#
#
# generating sgi mpi mpirun_hosts_list for the cluster:
set hosts_list=""
set mpirun_hosts_list=""
set host_token="0"
@ host_np = 0
foreach i (‘cat $PBS_NODEFILE | sort ‘)
 set hosts_list="$hosts_list $i"
end
foreach i ($hosts_list)
 if ($host_token == "0") then
# this is the first name in the list of hosts
set host_token="$i"
```

```
 endif
 if ($host_token == $i) then
# this is a repeated host list in the sequence
# thus increment the associated count host_np
  @ host_np = $host_np + 1
 else
# a new hostname has now appeared in the list
# thus update the hosts_list with the host_token
# and its associated process count
# and set host_token to the new hostname
# and reset the new host_token associated process count
# to 1
set mpirun_hosts_list="$mpirun_hosts_list,$host_token $host_np"
set host_token="$i"
@ host_np = 1
 endif
end
# update mpirun_hosts_list with the last entry
set mpirun_hosts_list="$mpirun_hosts_list,$host_token $host_np"
# remove the leading coma in mpirun_hosts_list
set mpirun_hosts_list=‘echo $mpirun_hosts_list| sed -e "s/,//"‘
#
setenv MPI_VERBOSE
setenv MPI_DSM_DISTRIBUTE
# Now run simpleFoam
(mpirun $mpirun_hosts_list \
        ‘which simpleFoam‘ \
         -case .           \
         -parallel         \
         < /dev/null ) >>& run.log
```

Using PBS Professional qsub Commands

This section contains some information about using the PBS Professional qsub commands.

To select a particular host(s) on a cluster using the host keyword, for example, to run a 24 core job on hosts r1i0n0 and r1i0n1 (12 threads on each), perform the following command:

```
# qsub -l select=1:host=r1i0n0:ncpus=24:mpiprocs=12+1:host=r1i0n1:ncpus=24:mpiprocs=12 ./script
```

To start an interactive PBS Professional session so as to work within on any node in the cluster, perform the following:

```
# qsub -I select=1:ncpus=24:mpiprocs=12 ./script
```

To start an interactive PBS Professional session on node `n032` only, perform the following:

```
# qsub -I select=1:host=n032:ncpus=24:mpiprocs=12 ./script
```

To start an ineractive PBS Professional task on four nodes so that the first node is `r1i0n15`, perform the following:

```
# qsub -I -l select=1:host=r1i0n15:ncpus=24:mpiprocs=12+3:ncpus=24:mpiprocs=12 ./script
```

To kill a `pbs` all ready submitted job, use the `qdel` command.

To look at the jobs statistics, use the `qstat -a` command.

OpenFOAM PBS Professional Script for SGI UV Systems Using SGI MPI

The following is an example script on how to run an OpenFOAM solver job using PBS Professional batch queue scheduler on an SGI UV system.

```
#!/bin/csh
#
### PBS ###
#PBS -S /bin/csh -j oe -k o -r n
#PBS -l walltime=5:15:00
#
#
# submit this script to the PBSpro job queue using the command:
#
# qsub -l select=1:ncpus=NCPU:mpiprocs=NTHD \
#      -l place=excl:group=<grp>            \
#      -v NPROCS=NTHD ./this_script
#
# where
#
#  NCPU       = Total number of cores required. This may not necessarily
#               be the same as NTHD, it may be larger if more memory is
#               required or if the user selects to use a lesser number
#               of cores per socket for better bandwidth performance.
#               Note that hyperthrading is normally disabled on UV.
#               Normally NCPU will be the same as NTHD.
#  NTHD       = Number of mpi threads to be spwaned.
#  <grp>      = socket       if   8 NTHD job
#               board        if  16 NTHD job
#               boardpair    if  32 NTHD job
#               iruquadrant  if  64 NTHD job
#               iruhalf      if 128 NTHD job
#               iru          if 256 NTHD job
#               rack         if 512 NTHD job
#
set echo
#
setenv WM_64
#
cd $PBS_O_WORKDIR
# Specify Single (SP) or Double (DP) precision, example here is SP
setenv OpenFOAM_PRECISION SP
# source the above setup.csh script to setup OpenFOAM environemnt:
source setup.csh SP >& run.log
#
# unsetenv FOAM_SIGFPE
#
# decompose first
# copy the correct decomposeParDict file to decomposeParDict
#
#
  rm -rf proc*
  cp system/decomposeParDict.${NPROCS} system/decomposeParDict
   (dplace -c1 decomposePar -case . ) >& decompose.out
#
setenv MPI_DSM_DISTRIBUTE
setenv MPI_SHARED_NEIGHBORHOOD HOST
(mpirun -np ${NPROCS} ‘which simpleFoam‘ \
        -case                      \
        .                          \
        -parallel < /dev/null ) >>& ${NPROCS}.log
```

Standard OpenFOAM Script for SGI ICE and SGI Rackable Clusters Using SGI MPI

The following is a standard example scripts on how to run an OpenFOAM directly on an SGI cluster system without the use of batch job schedulers, such as, PBS Professional.

```
#!/bin/csh
#
set echo
#
setenv WM_64
#
# Example of 128 core job running on 8 compute nodes as 16 threads
# per node, where the nodes selected are:
# host1,host2,host3,host4,host5,host6,host7,host8
#
set no_cores_per_node=16
set NPROCS=128
#
# Specify Single (SP) or Double (DP) precision, example here is SP
setenv OpenFOAM_PRECISION SP
# source the above setup.csh script to setup OpenFOAM environemnt:
source setup.csh SP >& run.log
# unsetenv FOAM_SIGFPE
#
# decompose geometry first
# copy the correct decomposeParDict file to decomposeParDict
#
#
   rm -rf proc*
   cp system/decomposeParDict.${NPROCS} system/decomposeParDict
    (dplace decomposePar -case . -parallel  ) >& decompose.out
#
set host_list="host1,host2,host3,host4,host5,host6,host7,host8"
set mpirun_hosts_list="$host_list $no_cores_per_node"
#
#
(mpirun $mpirun_hosts_list \
        ‘which simpleFoam‘ \
         -case .           \
         -parallel         \
         < /dev/null ) >>& run.log
```

Standard OpenFOAM Script for SGI UV Systems Using SGI MPI

The following is a standard example scripts on how to run an OpenFOAM directly on the SGI UV system without the use of batch job schedulers, such as, PBS Professional.

```
#!/bin/csh
#
# Example of 128 core job running on an SGI UV
#
set NPROCS=128
#
set echo
#
setenv WM_64
#
# Specify Single (SP) or Double (DP) precision, example here is SP
setenv OpenFOAM_PRECISION SP
# source the above setup.csh script to setup OpenFOAM environemnt:
source setup.csh SP >& run.log
#
# unsetenv FOAM_SIGFPE
#
# decompose first
# copy the correct decomposeParDict file to decomposeParDict
#
#
  rm -rf proc*
  cp system/decomposeParDict.${NPROCS} system/decomposeParDict
   (dplace -c1 decomposePar -case . ) >& decompose.out
#
setenv MPI_DSM_DISTRIBUTE
# Users may also use the sgimpi env var MPI_DSM_CPULIST for
# a more specific process to core pinning, see man mpi
#
setenv MPI_SHARED_NEIGHBORHOOD HOST
 (mpirun -np ${NPROCS} ‘which simpleFoam‘ \
         -case                      \
         .                          \
        -parallel < /dev/null ) >>& ${NPROCS}.log
```

Index