GOVERNMENT OF INDIA
INDIRA GANDHI CENTRE FOR ATOMIC RESEARCH
COMPUTER DIVISION
KALPAKKAM – 603102

SYSTEM	HIGH PERFORMANCE COMPUTING SYSTEMS
TITLE	**DAMASK: Installation, Configuration and Integration with Job Scheduler in Ivy Cluster**

A	Original Issue	03/02/2020	R. Jehadeesan
No	Revisions	Date	Approved

IGCAR	**EIG**	**RCCG**	**CD**	**CSS**	**HPCS**	**18**	**REV** **A**

Electronics Instrumentation Group

	NAME	SIGN	DATE
PREPARED	**Suja Ramachandran**	Suja	27.1.2020
REVIEWED	**M L Jayalal**		28/01/2020
APPROVED	**R Jehadeesan**	R. Jehadeesan	03/02/2020

Distribution:
Head, CSS;
Head, CD;
AD, RCCG;
Director, EIG;

SUMMARY

This document covers the installation and configuration of DAMASK-2.0.3 and its dependencies in the Ivy cluster. DAMASK is a multi-physics software tool for conducting advanced microstructure property simulations. The pre and post processing tools and spectral solver of DAMASK are installed. A script is developed to integrate the solver to the job scheduler, slurm.

DAMASK: Installation, Configuration and Integration with Job Scheduler in Ivy Cluster

Introduction

The Ivy cluster is designed to meet the large-scale numerical and data-intensive computing requirements of IGCAR. The cluster comprises of two numbers of master node / head node, 400 compute nodes and one management node. Each cluster node is powered by 64-bit Intel Xeon dual twelve-core processor with Ivy Bridge architecture of clock speed 2.7 GHz. The cluster has 9600 processor cores with 52 Terabytes of distributed memory. The Ivy cluster delivers a Peak (theoretical) performance of 207 TeraFLOPS and maximal sustained performance of around 180 TeraFLOPS with industry-standard HPL benchmark.

Düsseldorf Advanced Material Simulation Kit (DAMASK) is a flexible, modular and hierarchically structured multi-physics software tool for conducting advanced microstructure property simulations. It is an open source freeware package and models material point behaviour for the solution of elastoplastic boundary value problems along with damage and thermal physics. Its main purpose is the simulation of crystal plasticity within a finite-strain continuum mechanical framework. The material models included in DAMASK can be interfaced to three different solvers: The commercial FEM solvers, MSC.Marc and Abaqus, and the built-in spectral solver.

This document covers the installation and configuration of DAMASK-2.0.3 and its dependencies in the Ivy cluster. The pre and post processing tools and spectral solver of DAMASK are installed. The spectral solver is configured to run parallel within a single node using thread parallelism. A script is developed to integrate the solver to the job scheduler, slurm.

2. External Software Packages and Dependencies

DAMASK requires a set of external packages and libraries which are to be pre-installed before DAMASK installation. The external software packages required by DAMASK are listed below:

2.1. PETSc

The built-in spectral solver of DAMASK makes use of the Portable, Extensible Toolkit for Scientific Computation (PETSc). PETSc is a suite of data structures and routines for the scalable (parallel) solution of scientific applications modelled by partial differential equations.

2.2. CMake

CMake is an open-source, cross-platform family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files, and generate native ‘makefiles’ and workspaces that can be used in any compiler environment. Installing DAMASK and PETSc requires pre-installation of cmake-3.10 or later.

2.3. BLAS (Basic Linear Algebra Subprograms) library

BLAS routines provide standard building blocks for performing basic vector and matrix operations. The specification prescribes a set of low-level routines for performing common linear algebra operations such as vector addition, scalar multiplication, dot products, linear combinations, and matrix multiplication.

2.4. LAPACK (Linear Algebra Package)

LAPACK is a standard software library for numerical linear algebra. It provides routines for solving systems of linear equations and linear least squares, eigen value problems, and singular value decomposition. It also includes routines to implement the associated matrix factorizations such as LU, QR, Cholesky and Schur decomposition.

2.5. FFTW (Fastest Fourier Transform in the West)

FFTW is a software library for computing the Discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data.

2.6. SuiteSparse

SuiteSparse is a suite of sparse matrix algorithms, including UMFPACK, CHOLMOD, SPQR, CSparse and CXSparse, ordering methods (AMD, CAMD, COLAMD, and CCOLAMD) and SSMULT and SFMULT. SuiteSparse is required to provide sparse matrix factorizations and solution of linear equations for sparse systems.

2.7. Chaco

Chaco is a package of algorithms for decomposing a calculation into tasks that are assigned o different processors. It finds an optimal decomposition of a complex application and ensures that optimal performance is obtained.

2.8. Hypre

Hypre library of linear solvers offers a comprehensive suite of scalable solvers for large-scale scientific simulation, featuring parallel multigrid methods for both structured and unstructured grid problems.

2.9. MUMPS - MUltifrontal Massively Parallel sparse direct Solver.

MUMPS library offers solution of large linear systems with symmetric positive definite matrices, general symmetric matrices and general unsymmetric matrices. The features of MUMPS include matrix input in assembled format or elemental format, sequential or parallel analysis, detection of null pivots, computation of a Schur complement matrix, computation of determinant, forward elimination during factorization etc...

2.10. ParMETIS

METIS is a set of serial programs for partitioning graphs, partitioning finite element meshes, and producing fill reducing orderings for sparse matrices. ParMETIS is an MPI-based parallel library that extends the functionality provided by METIS and includes routines that are especially suited for parallel AMR computations and large scale numerical simulations.

2.11. ScaLAPACK

ScaLAPACK is a library of high-performance linear algebra routines for parallel distributed memory machines. ScaLAPACK solves dense and banded linear systems, least squares problems, eigenvalue problems, and singular value problems.

2.12. PNetCDF

NetCDF (Network Common Data Form) is a set of software libraries and machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data. PnetCDF is a high-performance parallel I/O library for accessing files in format compatibility with NetCDF. PnetCDF also provides a new set of non-blocking APIs which allow users to post multiple read and write requests, and later aggregate the requests into a large MPI-IO request, to achieve a better performance.

2.13. HDF5 (Hierarchical Data Format)

Hierarchical Data Format (HDF) is a set of file formats (HDF4, HDF5) designed to store and organize large amounts of data. HDF5 is a data model, library, and file format for storing and managing data. It supports an unlimited variety of data types, and is designed for flexible and efficient I/O and for high volume and complex data.

3. Installation of PETSc with dependencies

PETSc-3.10.5 is downloaded to */apps/petsc-install* directory in Ivy cluster. A directory /apps/petsc-3 is created for installing PETSc. Before installing PETSc, cmake-3.16.0 has to be installed and configured.

cmake-3.16.0 is downloaded to /apps partition of Ivy cluster. A directory */apps/CMAKE-3.16.0-rc3* is created for installing cmake. Before installing cmake, gcc-6.4 (which is already installed in Ivy cluster) is set as the default gcc by modifying PATH and LD_LIBRARY_PATH environment variables. The following are the steps for installing cmake:

```
export LD_LIBRARY_PATH=/apps/gcc-6.4.0/lib:/apps/gcc-6.4.0/lib64:$LD_LIBRARY_PATH
export PATH=/apps/gcc-6.4.0/bin:$PATH

cd /apps/
tar -xvzf cmake-3.16.0-rc3.tar.gz

cd cmake-3.16.0-rc3
mkdir /apps/CMAKE-3.16.0-rc3
./configure --prefix=/apps/CMAKE-3.16.0-rc3/
gmake
make install
```

Before configuring PETSc, cmake-3.16 and gcc-6.4 libraries are added to the path by modifying PATH and LD_LIBRARY_PATH environment variables. The other dependencies mentioned in Section 2 are downloaded and kept in a sub-directory so that it can be installed while configuring PETSc. PETSc is compiled using Intel MPI libraries. Intel Math Kernel Libraries (MKL) libraries are also required. So scripts are executed for setting up the proper environment for Intel MPI and MKL libraries. Following are the commands executed before PETSc installation:

---

export LD_LIBRARY_PATH=/apps/gcc-6.4.0/lib:/apps/gcc-6.4.0/lib64:$LD_LIBRARY_PATH

echo PATH=/apps/cmake-3.16.0-rc3/bin:$PATH
export PATH=/apps/cmake-3.16.0-rc3/bin:$PATH

source /apps/intel/compilers_and_libraries_2016.1.150/linux/bin/compilervars.sh intel64
source /apps/intel/compilers_and_libraries_2016.1.150/linux/mkl/bin/mklvars.sh intel64
source /apps/intel/compilers_and_libraries_2016.1.150/linux/mpi/intel64/bin/mpivars.sh

---

Before installing PETSc, FFTW is installed and configured. FFTW-3.3.8 is downloaded to /apps partition of Ivy cluster. A directory */apps/FFTW-3.3.8/* is created for installing FFTW. Following are the commands for installing and configuring FFTW:

---

tar –xvzf fftw-3.3.8.tar.gz
cd fftw-3.3.8

./configure --prefix=/apps/FFTW-3.3.8/ CC=icc CFLAGS="-g" --enable-mpi --enable-sse2 --enable-avx --enable-shared --disable-fortran
make
make install

export LD_LIBRARY_PATH=/apps/FFTW-3.3.8/lib/:$LD_LIBRARY_PATH
export PATH=/apps/FFTW-3.3.8/bin/:$PATH

PNetCDF is also installed in /apps prior to PETSc installation. Parallel-netcdf-1.9.0 is downloaded to /apps partition of Ivy cluster. A directory /apps/PNetCDF is created for installing PNetCDF. Following are the commands for installing and configuring PNetCDF:

---

tar –xvzf parallel-netcdf-1.9.0.tar.gz
cd parallel-netcdf

./configure --prefix=/apps/PNetCDF/ CFLAGS="-fPIC"
make -j8
make install

After these steps, PETSc is configured. All the other software dependencies including BLAS, LAPACK, Hypre, ScaLAPACK, Suitesparse, MUMPS, ParMETIS, CHACO and HDF5 are downloaded as tar files in the directory, */apps/petsc-install/petsc-depends/.* These tar file names are given as arguments to the configure command. The locations of cmake, FFTW and PNetCDF are also given as arguments to the configure command.

The commands for configuring PETSc and installing PETSc and its dependencies are:

```
cd /apps/petsc-install
tar –xvzf petsc-3.10.5.tar.gz
cd petsc-3.10.5

./configure --prefix=/apps/petsc-3/
 --with-mpi-dir=/apps/intel/compilers_and_libraries_2016.1.150/linux/mpi/intel64/
--with-cmake-dir=/apps/cmake-3.16.0-rc3/
--with-pnetcdf-dir=/apps/PNetCDF/
--download-netcdf=/apps/petsc-install/petsc-depends/netcdf-4.5.0.tar.gz
 --with-fftw-dir=/apps/FFTW-3.3.8/
--download-fblaslapack=/apps/petsc-install/petsc-depends/fblaslapack-3.4.2.tar.gz
--download-zlib=/apps/petsc-install/petsc-depends/zlib-1.2.11.tar.gz
 --download-hdf5=/apps/petsc-install/petsc-depends/hdf5-1.8.18.tar.bz2
--download-chaco=/apps/petsc-install/petsc-depends/Chaco-2.2-p2.tar.gz
--download-exodusii=/apps/petsc-install/petsc-depends/seacas-master.zip
--download-hypre=/apps/petsc-install/petsc-depends/hypre-2.15.1.tar.gz
--download-metis=/apps/petsc-install/petsc-depends/petsc-pkg-metis-49e61501c498.tar.gz
--download-ml=/apps/petsc-install/petsc-depends/petsc-pkg-ml-e5040d11aa07.tar.gz
--download-scalapack=/apps/petsc-install/petsc-depends/scalapack-2.0.2.tgz
--download-mumps=/apps/petsc-install/petsc-depends/petsc-pkg-mumps-5fe5b9e56f78.tar.gz
 --download-parmetis=/apps/petsc-install/petsc-depends/petsc-pkg-parmetis-73dab469aa36.tar.gz
--download-suitesparse=/apps/petsc-install/petsc-depends/SuiteSparse-4.4.3.tar.gz
--download-triangle=/apps/petsc-install/petsc-depends/Triangle.tar.gz
--with-c2html=0 --with-x=0 --with-ssl=0 --with-debugging=0 --with-cxx-dialect=C++11
COPTFLAGS="-O3 -xHost" CXXOPTFLAGS="-O3 -xHost" FOPTFLAGS="-O3 -xHost"
PETSC_ARCH=linux-intel PETSC_DIR=`pwd`

make PETSC_DIR=/apps/petsc-install/petsc-3.10.5 PETSC_ARCH=linux-intel all
make PETSC_DIR=/apps/petsc-install/petsc-3.10.5 PETSC_ARCH=linux-intel install
```

4. Installation of DAMASK

DAMASK-2.0.3 is downloaded to /apps directory. Before installing DAMASK, the environment variables, PETSC_DIR and PETSC_ARCH are set to reflect the installation directory of PETSc. Intel Fortran and C compilers are set as the default compilers. Following are the commands to be executed for DAMASK installation:

```
tar -xvf damask-2.0.3.tar.xz
cd damask-2.0.3

export FC=ifort
export CC=icc

export PETSC_DIR=/apps/petsc-3
export PETSC_ARCH=linux-intel

export PATH=/apps/anaconda3/bin:$PATH
```

The file */apps/damask-2.0.3/CONFIG* is modified to include the path to FFTW and Abaqus version. The number of threads that DAMASK can use is set to 24, which is the number of cores in a single node of Ivy cluster.

```
vi /apps/ damask-2.0.3/CONFIG

# "set"-syntax needed only for tcsh (but works with bash and zsh)
# DAMASK_ROOT will be expanded

set DAMASK_BIN          = ${DAMASK_ROOT}/bin
set DAMASK_NUM_THREADS    = 24

set MSC_ROOT            =
set MARC_VERSION        =

set ABAQUS_VERSION      = 2019
set DAMASK_HDF5         = OFF
set FFTW                = /apps/FFTW-3.3.8
```

The script /apps/damask-2.0.3/env/DAMASK.sh is executed to set DAMASK environment variables.

```
source /apps/damask-2.0.3/env/DAMASK.sh
```

To install DAMSK post and pre-processors and spectral solver, the following commands are executed:

```
cd /apps/damask-2.0.3

make clean processing
make spectral
```

5. Usage of DAMASK and Integration with job scheduler

For using DAMASK, a user should execute the script /apps/damask-2.0.3/env/DAMASK.sh for setting DAMASK environment variables. The command and output will be as follows:

```
source  /apps/damask-2.0.3/env/DAMASK.sh

Düsseldorf Advanced Materials Simulation Kit --- DAMASK
Max-Planck-Institut für Eisenforschung GmbH, Düsseldorf
https://damask.mpie.de
Using environment with ...
DAMASK          /apps/damask-2.0.3
Spectral Solver    /apps/damask-2.0.3/bin/DAMASK_spectral
Post Processing    /apps/damask-2.0.3/bin/postResults

Multithreading     DAMASK_NUM_THREADS=24
```

```
heap  size      unlimited
stack size      unlimited
```

In Ivy cluster, the jobs are submitted through the job manager, Slurm. A script is developed for submitting the spectral solver to slurm. Since DAMASK is not working with Lustre filesystem in Ivy cluster, the script is configured such as it is running on a single compute node, by copying the input files to a local directory of the node. The output files are written in the local directory, and at the end, they are copied to the user's home directory by the script.

The sample script (slurm-damask.sh) is as follows:

```
#!/bin/bash

#SBATCH --ntasks-per-node 24
#SBATCH --error=job.%J.err
#SBATCH --output=job.%J.out
#SBATCH -p all.q

source /apps/intel/compilers_and_libraries_2016.1.150/linux/bin/compilervars.sh intel64
source /apps/intel/compilers_and_libraries_2016.1.150/linux/mkl/bin/mklvars.sh intel64
source /apps/intel/compilers_and_libraries_2016.1.150/linux/mpi/intel64/bin/mpivars.sh

export PETSC_DIR=/apps/petsc-3
export PETSC_ARCH=linux-intel
source /apps/damask-2.0.3/env/DAMASK.sh

export JOBNAME="${SLURM_JOB_NAME}"
export JOBID="${SLURM_JOBID}"

# Copy the working dir from Slurm and switch to that directory

export WORKDIR="${SLURM_SUBMIT_DIR}"
cd "${WORKDIR}"

INP_FILES="*.config *.geom *.load"
#Change the above line according to your input
  # make a scratch dir in the local disk space under /tmp
  # make job specific sub directory

JOB_DIR="$SLURM_JOBID"_damask
SCRATCH_DIR=/tmp/$JOB_DIR
mkdir ${SCRATCH_DIR}
cp $INP_FILES ${SCRATCH_DIR}
cd $SCRATCH_DIR
echo  $SCRATCH_DIR
DAMASK_spectral -l tensionX.load -g test.geom
#Change the above line according to your input

  mv ${SCRATCH_DIR} ${WORKDIR}
  rm -rf /tmp/$JOB_DIR
```

In the script, the lines,

INP_FILES==".config *.geom *.load"*

and

DAMASK_spectral -l tensionX.load -g test.geom

are to be modified by specifying the input files of DAMASK.

Once the script is ready, it has to be submitted using the command:

---

sbatch –n 1 slurm-damask.sh

---

6. Conclusions

Düsseldorf Advanced Material Simulation Kit (DAMASK) is successfully installed and configured in Ivy cluster along with its dependencies. A script is developed for submitting a DAMASK job to the cluster through the job manager, Slurm.