Compiling and Running Aspect on TACC Stampede

Eric Heien <emheien@ucdavis.edu>

August 27, 2013

Setting up Aspect on TACC Stampede is much easier than on previous systems since most of the necessary libraries are already correctly installed and configured. Following all the instructions in this document should allow you to start running Aspect within 30 minutes.

If you have trouble with any part of this, don't hesitate to contact Eric in his office (EPS 2215) or by email (emheien@ucdavis.edu). For further information on Stampede you can also refer to the user guide at
`http://www.tacc.utexas.edu/user-services/user-guides/stampede-user-guide`

1 Account

If you do not already have an XSEDE account, sign up for an account at
`http://portal.xsede.org/` and send the account username to Eric.

2 Setup

To start, edit your login script to automatically load the required modules. The login script should be $HOME/.profile_user and you should add the following lines:

```
module load trilinos/10.12.2 phdf5 cmake
module swap intel intel/13.0.079
```

This will ensure the necessary modules are loaded when you log in. IMPORTANT NOTE: We have found a bug where deal.II does not return the correct results when compiled in optimized mode using the Intel compiler version 13.0.1 or higher. *BE SURE* to use Intel compiler 13.0.0. To ensure the setup is correct log out, log in again, and run:

```
login1$ module list
```

The result should include all the modules above as well as some default modules:

```
Currently Loaded Modules:
  1) TACC          4) cluster          7) intel/13.0.079    10) trilinos/10.12.2
  2) TACC-paths    5) cluster-paths    8) mvapich2/1.9a2
  3) Linux         6) cmake/2.8.9      9) phdf5/1.8.9
```

3 Download

Next you will download the deal.II library and Aspect source. Since the home directory has tight disk usage limits, move to your work directory using

```
login1$ cd $WORK
```

or the abbreviated command

```
login1$ cdw
```

Check out the latest deal.II and Aspect code using Subversion commands (this will take a while):

```
login1$ svn co https://svn.dealii.org/trunk/deal.II
login1$ svn co https://svn.aspect.dealii.org/trunk/aspect
```

Ignore complaints about certificates and just accept them permanently:

```
Error validating server certificate for 'https://svn.dealii.org:443':
 - The certificate is not issued by a trusted authority. Use the
   fingerprint to validate the certificate manually!
Certificate information:
 - Hostname: svn.dealii.org
 - Valid: from Sat, 22 Dec 2012 23:03:50 GMT until Mon, 22 Dec 2014 23:03:50 GMT
 - Issuer: http://www.CAcert.org, CAcert Inc.
 - Fingerprint: f3:0d:83:f1:f2:33:91:e0:02:68:18:12:7c:3f:9a:65:5b:49:f3:f5
(R)eject, accept (t)emporarily or accept (p)ermanently? p
```

4 p4est Library

To run efficiently in parallel, deal.II also requires p4est. Instructions for setting it up are at `http://www.dealii.org/developer/external-libs/p4est.html` or you can use the following:

```
login1$ mkdir deal.II/p4est
login1$ cd deal.II/p4est
login1$ wget http://www.dealii.org/developer/external-libs/p4est-setup.sh
login1$ wget http://burstedde.ins.uni-bonn.de/release/p4est-0.3.4.1.tar.gz
login1$ chmod u+x p4est-setup.sh
login1$ ./p4est-setup.sh p4est-0.3.4.1.tar.gz $WORK/local/
```

Successful setup should look like:

```
CFLAGS_FAST: -O2
CFLAGS_DEBUG: -O0 -g

This script tries to unpack, configure and build the p4est library.
Build FAST: /work/01766/emheien/deal.II/p4est/p4est-build/FAST
```

```
........

Build DEBUG version in /work/01766/emheien/deal.II/p4est/p4est-build/DEBUG
configure: WARNING: Static source code checker splint not found
configure: WARNING: Static source code checker splint not found
DEBUG version installed in /work/01766/emheien/local//DEBUG
```

5 deal.II Library

Now you can configure and compile deal.II. deal.II now uses CMake for configuration and compilation. Go to the deal.II directory and run the following commands:

```
login1$ cd $WORK/deal.II
login1$ mkdir build
login1$ cd build
login1$ CC=mpicc CXX=mpicxx cmake -DWITH_MPI=ON -DP4EST_DIR=$WORK/local/ -DTRILINOS_DIR=$TACC_TRILINOS_DIR \
-DHDF5_DIR=$TACC_HDF5_DIR -DCMAKE_INSTALL_PREFIX=$WORK/local/ ..
-- This is CMake 2.8.9
-- The C compiler identification is Intel 13.0.0.20120731
-- The CXX compiler identification is Intel 13.0.0.20120731
-- Check for working C compiler: /opt/apps/intel13/mvapich2/1.9/bin/mpicc
-- Check for working C compiler: /opt/apps/intel13/mvapich2/1.9/bin/mpicc -- works
........
#  Configured Features (DEAL_II_ALLOW_BUNDLED = ON, DEAL_II_ALLOW_AUTODETECTION = ON):
#      ( DEAL_II_WITH_64BIT_INDICES = OFF )
#      ( DEAL_II_WITH_ARPACK = OFF )
#        DEAL_II_WITH_BOOST set up with bundled packages
#        DEAL_II_WITH_FUNCTIONPARSER set up with bundled packages
#        DEAL_II_WITH_HDF5 set up with external dependencies
#        DEAL_II_WITH_LAPACK set up with external dependencies
#      ( DEAL_II_WITH_METIS = OFF )
#        DEAL_II_WITH_MPI set up with external dependencies
#      ( DEAL_II_WITH_MUMPS = OFF )
#      ( DEAL_II_WITH_NETCDF = OFF )
#        DEAL_II_WITH_P4EST set up with external dependencies
#      ( DEAL_II_WITH_PETSC = OFF )
#      ( DEAL_II_WITH_SLEPC = OFF )
#        DEAL_II_WITH_THREADS set up with bundled packages
#        DEAL_II_WITH_TRILINOS set up with external dependencies
#        DEAL_II_WITH_UMFPACK set up with bundled packages
#        DEAL_II_WITH_ZLIB set up with external dependencies
login1$ make -j 8 install
...........
Linking CXX shared library libdeal_II.g.so
...........
Linking CXX shared library libdeal_II.so
...........
-- Installing: /work/01766/emheien/local/include/deal.II/...
```

Be sure DEAL_II_WITH_HDF5, DEAL_II_WITH_MPI, DEAL_II_WITH_P4EST, and DEAL_II_WITH_TRILINOS are not OFF. If they are OFF then the configuration failed to find them and Aspect won't compile correctly. The Intel compiler may generate a number of warnings, but if you see the "Linking CXX library" step complete successfully then the compilation worked.

6 Aspect

Now you can compile Aspect. Aspect should be able to find deal.II automatically, but if not you can specify the deal.II install location. Configure Aspect using CMake in a similar way to deal.II:

```
login1$ cd aspect
login1$ mkdir build
login1$ cd build
login1$ cmake -DDEAL_II_DIR=$WORK/local ..
=====================================================
============ Configuring ASPECT ====================
=====================================================
...
-- Configuring done
-- Generating done
-- Build files have been written to: /work/01766/emheien/aspect/build
```

And run make:

```
login1$ make -j 8
Scanning dependencies of target aspect
login1$ make
[  0%] Building CXX object CMakeFiles/aspect.dir/source/...
[  0%] Building CXX object CMakeFiles/aspect.dir/source/...
..............
Linking CXX executable aspect
[100%] Built target aspect
```

Again there may be many warnings from the compiler, but if the aspect target is created then it was successful. By default, Aspect compiles the debug version of the code. You should use this version when changing the Aspect code. When doing scientific work you should change to the optimized version. To compile the optimized version use the `-DCMAKE_BUILD_TYPE=Release` flag in CMake:

```
login1$ cmake -DDEAL_II_DIR=$WORK/local -DCMAKE_BUILD_TYPE=Release ..
```

and run make again.

7 Running

When testing Aspect, it can be useful to run in interactive mode on the cluster. This way you can use multiple processors for fast testing, but still make changes to files and test different options. To start an interactive shell with multiple processors, use:

```
srun -p development -t <maximum time> -n <#cores> --pty /bin/bash -l
```

For example:

```
srun -p development -t 0:30:00 -n 4 --pty /bin/bash -l
```

Once logged in you can run Aspect in parallel using commands such as:

```
ibrun ./lib/aspect <parameter file>
```

The development queue only allows small short runs (<4 hours, <256 cores). To perform larger runs, you need to submit a job to the scheduler similar to other systems like ymir. See the user guide for details about how to do this.