

ASPECT INSTALLATION - UBUNTU 14.04 v3.0

Phil Heron, Durham University

Oct 2017

email: philip.j.heron@durham.ac.uk

This is the setup guide for installing ASPECT on the Hamilton Cluster. There are some prerequisites for this endeavour, one of them being a basic knowledge of linux commands and VI editor commands.

Below are a few links to these as a refresher or to get you up to speed.

Linux commands: <https://diyhacking.com/linux-commands-for-beginners/>

VI Editor commands: <https://www.cs.colostate.edu/helpdocs/vi.html>

For this guide, lines that start with > indicate a terminal window. Anything after the > is what is needed to type in the terminal window.

e.g.,

>

> pwd

/ddn/home/

In order to install ASPECT you need to install the numerical code deal.ii. To do this, we need to use the installer package, Candi.

Therefore, we need to install deal.ii (and it's components trilinos and p4est) via Candi first.

STEP 1 - setting up modules

```
sudo apt-get install build-essential wget automake \  
    autoconf gfortran \  
    openmpi-bin openmpi-common \  
    libopenmpi-dev cmake subversion \  
    git libblas-dev liblapack-dev \  
    libblas3gf liblapack3gf \  
    libsuitesparse-dev libtool \  
    libboost-all-dev splint tcl \  
    tcl-dev environment-modules qt4-dev-tools
```

STEP 2 - downloading Candi

To download Candi we should set up a new folder for the installation.

> pwd

/home/

```
> mkdir install
> cd install
```

To download Candi, we can use a git command which will download the whole package into your folder.

```
> git clone https://github.com/dealii/candi
```

```
> cd candi
```

```
> ls
```

```
AUTHORS LICENSE README.md candi.cfg candi.sh deal.II-toolchain
```

STEP 3 - editing Candi setup

As candi can download a great number of packages that deal.ii can work with, we only need to download the ones that are directly relevant to ASPECT. As a result, we can comment out the packages that we do not need (which are open cascade, PETSC, SLEPC).

In the candi folder:

```
> ls
```

```
AUTHORS LICENSE README.md candi.cfg candi.sh deal.II-toolchain
```

```
> vi candi.cfg
```

line 19 needs to be edited to:

```
INSTALL_PATH=/home/install/candi (or wherever the candi folder is located)
```

We need to edit the following from line 48, by putting a # in the first column (therefore commenting out that line).

The packages open cascade, parmetis, superlu_dist, petsc, and slepc need to be commented out:

```
#PACKAGES="${PACKAGES} once:opencascade"
#PACKAGES="${PACKAGES} once:parmetis"
#PACKAGES="${PACKAGES} once:superlu_dist"
PACKAGES="${PACKAGES} once:hdf5"
PACKAGES="${PACKAGES} once:p4est"
PACKAGES="${PACKAGES} once:trilinos"
#PACKAGES="${PACKAGES} once:petsc"
#PACKAGES="${PACKAGES} once:slepc"
```

```
PACKAGES="${PACKAGES} dealii"
```

From here we need to exit the `candi.cfg` file and edit the `candi.sh` file.

```
> vi candi.sh
```

We need to edit the path to the build directories, to point to the `ddn/data` folder:

```
PREFIX=/home/install/dealii-candi *or wherever you would like the  
code to build*
```

STEP 4 - running Candi

From here we can run `candi`.

```
> ./candi.sh -j<N>
```

(where `<N>` is the number of processors you have available to run the configuration and installation of `deal.ii` and its dependencies). I would recommend running `-j8`.

Therefore it would be:

```
> ./candi.sh -j8
```

You are asked to review the set up of `candi` once the installation begins. Press enter for these parts unless you notice an error. The program will make sure you have the relevant external libraries installed and then make sure the compilers are correct: `c++` etc.

When using `-j8` it took 40 minutes.

Build finished in 2386 seconds.

Summary of timings:

```
dealii-prepare: 0 s  
hdf5: 161 s  
p4est: 101 s  
trilinos: 988 s  
dealii: 1112 s
```

Head to the `dealii` build directory to test:

```
> cd /home/install//deal.ii-candi/tmp/build/deal.II-v8.5.1
```

```
> make test
```

This should run 10 tests, they all should pass

```
      Start  1: step.debug
1/10 Test  #1: step.debug ..... Passed  30.08
sec
      Start  2: step.release
2/10 Test  #2: step.release ..... Passed  27.75
sec
      Start  3: affinity.debug
3/10 Test  #3: affinity.debug ..... Passed  20.79
sec
      Start  4: mpi.debug
4/10 Test  #4: mpi.debug ..... Passed  23.10
sec
      Start  5: tbb.debug
5/10 Test  #5: tbb.debug ..... Passed  19.57
sec
      Start  6: p4est.debug
6/10 Test  #6: p4est.debug ..... Passed  24.46
sec
      Start  7: step-trilinos.debug
7/10 Test  #7: step-trilinos.debug ..... Passed  27.25
sec
      Start  8: lapack.debug
8/10 Test  #8: lapack.debug ..... Passed  19.17
sec
      Start  9: umfpack.debug
9/10 Test  #9: umfpack.debug ..... Passed  25.24
sec
      Start 10: gsl.debug
10/10 Test #10: gsl.debug ..... Passed  20.01
sec
```

```
100% tests passed, 0 tests failed out of 10
```

```
Total Test time (real) = 237.45 sec
```

```
Built target test
```

STEP 5 - building ASPECT

Wherever you would like ASPECT to be downloaded, go into the directory and acquire the development package of ASPECT by:

```
> git clone --recursive https://github.com/geodynamics/aspect.git
```

Once downloaded, we can begin to build and then configure. For the build we need to let ASPECT know where deal.ii is, and from there deal.ii is configured and linked to trilinos and p4est.

```
> cd aspect
```

```
> ls
```

```
AUTHORS          CTestConfig.cmake  VERSION          cookbooks  docker
tests
CMakeLists.txt   LICENSE            benchmarks      data       include
CONTRIBUTING.md README.md          cmake           doc        source
```

```
> cmake -DDEAL_II_DIR=/home/install/deal.ii-candi/tmp/build/
deal.II-v8.5.1 .
```

Once the cmake is complete, you need to make the executable:

```
> make -j<N>
```

Again, I would recommend using a number of processors for this make, as it can take a while.

I used j8 and it took 5 minutes.

STEP 6 - testing ASPECT

Once complete, run a make test to check to see if the executable is working correct:

```
> cd tests
```

```
> make test
```

Running tests...

```
Test project /home/aspect/tests
```

```
Start 1: quick_mpi
```

```
1/1 Test #1: quick_mpi ..... Passed    14.66 sec
```

```
100% tests passed, 0 tests failed out of 1
```

```
Total Test time (real) = 14.73 sec
```

For the whole setup, you will need the module loaded as specified at the start. If you stop this process half way through and being again, you will need to reload the modules.

STEP 7 - making the release version of ASPECT (and not debug)

```
> cd ../  
> pwd  
/home/aspect  
> make release
```