Installing FEniCS in remote machines(Cedar/Graham/Niagara)

Sudhi Sharma P V

January 14, 2019

This document is made closely following the wiki page https://docs.computecanada.ca/wiki/FEniCS. Its always better to install in the same folder as compute-canada has suggested or else we may have to change paths at various places and this causes trouble. I have tried and got some errors and didn't want to go further. So, all the installations are done as specified in the document.

1 DOLFIN

CEDAR and GRAHAM have the same base software stack and filesystem and thus doesn't require any changes in case of installation other than some exceptions. Niagara has some differences in loading their modules and the filesystems. But one can easily load the same modules as in Cedar and Graham using following two commands and follow same steps of installation.
module load CCEnv
module load StdEnv
Another point to be noted for Niagara is the filesystem allocation for users. Niagara $HOME command and cedar and graham $HOME command is different and one may need to take care of them in their installation commands.

1 module load hdf5-mpi/1.8.18 boost eigen python/3.5 scipy-stack/2017b petsc/3.7.5 fftw-mpi/3.3.6

2 mkdir fenics && cd fenics (Provide folder name according to version number for easy recognition)

3 wget https://bitbucket.org/fenics-project/fiat-2017.1.0.tar.gz

4 tar xvfz fiat-2017.1.0.tar.gz

5 mv fiat-2017.1.0 fiat

6 Repeat the steps 3 4 and 5 as above for all dependancies namely,
wget https://bitbucket.org/fenics-project/instant-2017.1.0.tar.gz
wget https://bitbucket.org/fenics-project/dijitso-2017.1.0.tar.gz
wget https://bitbucket.org/fenics-project/ufl-2017.1.0.tar.gz
wget https://bitbucket.org/fenics-project/ffc-2017.1.0.tar.gz
One point to be noted here is that 'instant' has been made deprecated by fenics developers and has the latest version 2017.2.0 so download and install this version for all latest versions of dolfin.

7 Finally do the same for DOLFIN,
wget https://bitbucket.org/fenics-project/dolfin/downloads/dolfin-2017.1.0.tar.gz
Don’t forget to change the names of folders to original name without version number as in dolfin-2017.1.0 to dolfin

8 pyvenv ∼/fenics

9 source ∼/fenics/bin/activate

10 cd fiat && pip3 install . && cd - Repeat the same step as in 10 for all dependencies namely
cd instant && pip3 install . && cd -
cd dijitso && pip3 install . && cd -
cd ufl && pip3 install . && cd -
cd ffc && pip3 install . && cd -

11 pip3 install ply

12 pip3 install meshio

13 pip3 install lxml

14 pip3 install mpi4py

15 cd dolfin

16 mkdir build && cd build

17 In CEDAR the following command worked while in GRAHAM if one faces an error regarding ’RPATH’, an additional argument to skip install RPATH can be set along with cmake command as in first line below

```
-DCMAKE_SKIP_INSTALL_RPATH=ON

cmake .. -DDOLFIN_SKIP_BUILD_TESTS=true -DEIGEN3_INCLUDE_DIR=$EBROOTEIGEN/include

-DCMAKE_INSTALL_PREFIX=$HOME/software/dolfin -DCMAKE_SKIP_RPATH=ON -DRT_LIBRARY=

$EBROOTNIXPKGS/lib64/librt.so -DHDF5_C_LIBRARY_dl=$EBROOTNIXPKGS/lib64/libdl.so

-DHDF5_C_LIBRARY_m=$EBROOTNIXPKGS/lib64/libm.so -DHDF5_C_LIBRARY_pthread=

$EBROOTNIXPKGS/lib64/libpthread.so -DHDF5_C_LIBRARY_z=$EBROOTNIXPKGS/lib/libz.so
```

```
15  nice make -j 8 install && cd -
```

```
15  sed -i  s'^export LD_LIBRARY_PATH=/lib^#export LD_LIBRARY_PATH=/lib^'

    ~/software/dolfin/share/dolfin/dolfin.conf
```

16 Now for running fenics, activate the bin after loading the same modules as in step 1 if in different session

17 source ∼/software/dolfin/share/dolfin/dolfin.conf

18 source ∼/fenics/bin/activate

2 MSHR

mshr is the mesh generation component of FEniCS. It generates simplicial DOLFIN meshes in 2D and 3D from geometries described by Constructive Solid Geometry (CSG) or from surface files, utilizing CGAL and Tetgen as mesh generation backends.

1 module load hdf5-mpi/1.8.18 boost eigen python/3.5 scipy-stack/2017b petsc/3.7.5 fftw-mpi/3.3.6

2 Here, this command has worked for me in cedar and niagara but in graham i had to change the folder name slighty by removing the last folder in path 'cmake'.
So, check whether you are having this folder cmake and then only provide this full folder path.

```
 export CMAKE_PREFIX_PATH=/home/$USER/software/dolfin/share/dolfin/cmake/:

$CMAKE_PREFIX_PATH
```

3 git clone https://bitbucket.org/fenics-project/mshr.git

4 cd mshr && mkdir build && cd build

5
```
cmake .. -DCMAKE_INSTALL_PREFIX=$HOME/software/mshr

-DCMAKE_BUILD_TYPE=Release -DCMAKE_SKIP_RPATH=ON -DCMAKE_SYSTEM_PREFIX_PATH=$NIXUSE

$NIXUSER_PROFILE:$CMAKE_PREFIX_PATH -DEIGEN3_INCLUDE_DIR=$EBROOTEIGEN/include
```

6 Now for running mshr do,
module load hdf5-mpi/1.8.18 boost eigen python/3.5 scipy-stack/2017b petsc/3.7.5 fftw-mpi/3.3.6

7 source ∼/software/dolfin/share/dolfin/dolfin.conf

8 source ∼/fenics/bin/activate

9
```
 export PYTHONPATH=$HOME/software/mshr/lib/python3.5/site-packages:$PYTHONPATH
```

10
```
 export LD_LIBRARY_PATH=$HOME/software/mshr/lib:$LD_LIBRARY_PATH
```

3 v2018.1

Installing version 2018.1.0 requires PYBIND11 for python since fenics has removed python interface support in its default installation. Follow the instructions in the below URL for installing pybind and then linking it with dolfin.
https://bitbucket.org/fenics-project/dolfin/src/master/python/

4 Running Demos

Running the demos given inside the demo folder of fenics installation is important to verify that the installation is correct. Poisson and elasticity demos might have interactive plot commands inside them and its good to uncomment them to not get errors regarding this in remote machines. Also, for elasticity demo the output mode is given as ”compressed” which makes it in to binary format and impossible for the paraview to read it. So note it to remove that option from this demo. Another point is to remember to clean instant and dijitso using commands
instant-clean
dijitso clean
this is done when you are running the same problem again and again and once you got an error which might be saved inside these caches there might be chances that its pulling the same erroneous answer from this. You can also see that in the consecutive runs program is not calling FFC, and Instant.