

IBAMR install on KillDevil Cluster
July 18, 2016

Setup

```
ssh "onyen"@killdevil.unc.edu
>Type Password
You should be in your home directory on killdevil now.
```

KillDevil uses modules for its environment. In order to install IBAMR properly in our home directory, the following modules need to be added!

If these modules are not in your environment, the installation and any future projects will not run properly

IMPORTANT Make sure the modules below are added/ removed for the entire install and when you run the code.

module initclear removes all currently initialized modules in your environment

remove ibamr/3803 w/ module clear (remove all) or module rm (1 item). Just in case it is in your current environment.

Add the following modules to your initial environment!

```
Module initadd openmpi_gcc/4.8.1
Module initadd python/2.7.6
Module initadd git
Module initadd subversion
Module initadd gnuplot/4.4.0
```

"Changed my initial module load out. Before intel mph was conflicting and still saved. Had to remove all init modules and replace them with new ones above" **IMPORTANT**

Close out of your ssh and reopen it to initialize environment with correct modules
Make directory where all our ibamr libraries will be placed

```
cd $HOME
mkdir sfw
cd sfw
mkdir linux
```

Boost

```
cd $HOME/sfw/linux
mkdir boost
cd boost/
```

No need to download boost again. Just copy the previously installed file from your home computer onto killdevil

"On Home Computer"

```
scp $HOME/sfw/linux/boost/boost_1_63_0.tar.bz2 tdombro@killdevil.unc.edu:~/sfw/linux/boost/
```

"End On Home Computer"

I just followed the below instructions from a previous tutorial and there were no problems. I suggest you do the same

```
tar xvjf boost_1_63_0.tar.bz2
mv boost_1_63_0 1.63.0
cd 1.63.0/
export BOOST_ROOT=$HOME/sfw/linux/boost/1.63.0/
./bootstrap.sh --prefix=$HOME/sfw/linux/boost/1.63.0
./b2 install
```

In -s \$PWD/boost \$PWD/include/boost

Boost Installation on killdevil should be complete

HDF5

```
cd $HOME/sfw/linux
```

No need to download HDF5 again. Just copy the previously installed file from your home computer onto killdevil

“On Home Computer”

```
scp $HOME/sfw/linux/hdf5-1.10.1.tar tdombro@killdevil.unc.edu:~/sfw/linux/
```

“End On Home Computer”

Same as installing on home computer. Just change compilers

```
tar xvf hdf5-1.10.1.tar
cd hdf5-1.10.1
./configure \
CC=gcc \
CXX=g++ \
FC=gfortran \
F77=gfortran \
--enable-build-mode=production \
--prefix=$HOME/sfw/linux/hdf5/1.10.1
make
make check
make install
```

HDF5 installation on killdevil should be complete

Silo

```
cd $HOME/sfw/linux
```

No need to download Silo again. Just copy the previously installed file from your home computer onto killdevil

“On Home Computer”

```
scp $HOME/sfw/linux/silo-4.10.2.tar.gz tdombro@killdevil.unc.edu:~/sfw/linux/
```

“End On Home Computer”

Same as installing on home computer. Just change compilers

```
tar xvfz silo-4.10.2.tar.gz
cd silo-4.10.2
./configure \
CC=gcc \
CXX=g++ \
FC=gfortran \
```

```
F77=gfortran \  
--prefix=$HOME/sfw/linux/silo/4.10.2 \  
--disable-silex
```

```
make  
make check //Error but did not affect ibamr installation  
make install
```

Silo installation on killdevil should be complete

OpenMPI

Included in modules on killdevil
openmpi_gcc/4.8.1

OpenMPI installation on killdevil should be complete

PETSc

```
cd $HOME/sfw  
mkdir petsc  
cd petsc
```

No need to download Silo again. Just copy the previously installed file from your home computer onto killdevil

```
"On Home Computer"  
scp $HOME/sfw/petsc/pets-3.7.tar.gz tdombro@killdevil.unc.edu:~/sfw/petsc/  
"End on Home Computer"
```

```
tar xvfz petsc-3.7.tar.gz  
mv petsc-3.7.6 3.7.6  
cd 3.7.6/
```

We only need to install an optimized version of the following libraries since killdevil will be used mainly for builds projects that already work properly on our local system.
Optimized version to improve computation time.

Install an optimized version of PETSc:

```
export PETSC_DIR=$PWD  
export PETSC_ARCH=killdevil-opt //Note: Can change name to something else
```

Same as on Home computer. The flags have capital O, not zero 0. The compilers are changed to match that of killdevil

```
./config/configure.py \  
--CC=mpicc \  
--CXX=mpicxx \  
--FC=mpif90 \  
--COPTFLAGS="-O3" \  
--CXXOPTFLAGS="-O3" \  
--FOPTFLAGS="-O3" \  
--PETSC_ARCH=$PETSC_ARCH \  
--with-debugging=0 \  
--with-c++-support \  
--with-hypre=1 \  
--download-hypre=1 \  
--download-hypre=1
```

--with-x=0

The terminal commands specified below should be mentioned at the end of the configuration. Make sure you execute exactly what is specified by the configuration.

```
make PETSC_DIR=$HOME/sfw/petsc/3.7.6 PETSC_ARCH=killdevil-opt all
make PETSC_DIR=$HOME/sfw/petsc/3.7.6 PETSC_SRCH=killdevil-opt test
make PETSC_DIR=$HOME/sfw/petsc/3.7.6 PETSC_ARCH=killdevil-opt streams #I don't think this is
necessary
```

PETSc installation on killdevil should be complete

SAMRAI

*download SAMRAI-v2.4.4.tar.gz from <http://computation.llnl.gov/projects/samrai/software>

```
cd $HOME/sfw
mkdir samrai
cd samrai
mkdir 2.4.4
cd 2.4.4
```

No need to download SAMRAI again. Just copy the previously installed file from your home computer onto killdevil

“On Home Computer”

```
scp $HOME/sfw/samrai/2.4.4/SAMRAI-v2.4.4.tar.gz
$HOME/sfw/samrai/2.4.4/SAMRAI-v2.4.4-patch-121212.gz tdombro@killdevil.unc.edu:~/sfw/samrai/2.4.4/
“End On Home Computer”
```

```
tar xvfz SAMRAI-v2.4.4.tar.gz
mv SAMRAI SAMRAI-2.4.4
cd SAMRAI-2.4.4/
```

Patch SAMRAI with specified patch 121212 from ibamr GitHub.

```
./source/scripts/includes --link
gunzip -c ../SAMRAI-v2.4.4-patch-121212.gz | patch -p2
```

Should be a bunch of activity in the terminal if done correctly

Install an optimized version of SAMRAI:

```
cd $HOME/sfw/samrai/2.4.4
mkdir objs-opt
cd objs-opt/
```

Same as on Home computer. The flags have capital O, not zero 0. The compilers are changed to match that of killdevil

```
../SAMRAI-2.4.4/configure \
CFLAGS="-O3" \
CXXFLAGS="-O3" \
FFLAGS="-O3" \
--prefix=$HOME/sfw/samrai/2.4.4/killdevil-g++-opt \
--with-CC=gcc \
--with-CXX=g++ \
--with-F77=gfortran \
```

```
--with-MPICC=mpicc \
--with-hdf5=$HOME/sfw/linux/hdf5/1.10.1 \
  --without-hypr \
--with-silo=$HOME/sfw/linux/silo/4.10.2 \
--without-blaslapack \
--without-cubes \
--without-eleven \
--without-kinsol \
--without-petsc \
--without-sundials \
--without-x \
--with-doxygen \
--with-dot \
--disable-debug \
--enable-opt \
--enable-implicit-template-instantiation \
--disable-deprecated
```

```
make
make install
```

SAMRAI installation on killdevil should be complete

libMesh

```
cd $HOME/sfw/linux
mkdir libmesh
cd libmesh
mkdir 1.0.0
cd 1.0.0/
```

No need to download libMesh again. Just copy the previously installed file from your home computer onto killdevil

```
“On Home Computer”
scp lib $HOME/sfw/linux/libmesh/1.0.0/libmesh-1.0.0.tar.gz
tdombro@killdevil.unc.edu:~/sfw/linux/libmesh/1.0.0/
“End On Home Computer”
```

```
tar xvfz libmesh-1.0.0.tar.gz
mv libmesh-1.0.0 LIBMESH
```

Install an optimized version of libMesh:

```
cd $HOME/sfw/linux/libmesh/1.0.0
mkdir objs-opt
cd objs-opt/
export BOOST_ROOT=$HOME/sfw/linux/boost/1.63.0
```

Same as on Home computer. The compilers are changed to match that of killdevil

```
./LIBMESH/configure \
--prefix=$HOME/sfw/linux/libmesh/1.0.0/1.0.0-opt \
—disable-maintainer-mode \
—with-boost=$HOME/sfw/linux/boost/1.63.0 \
--with-methods=opt \
PETSC_DIR=$HOME/sfw/petsc/3.7.6 \
```

```
PETSC_ARCH=killdevil-opt \
CC=mpicc \
CXX=mpicxx \
FC=mpif90 \
F77=mpif90 \
--enable-triangle \
--disable-cxx11 \
--disable-openmp \
--disable-perflog \
--disable-strict-lgpl \
--disable-threads \
--disable-cpthreads \
--disable-unique-ptr
```

```
make
make install
```

libMesh installation on killdevil should be complete

IBAMR

```
cd $HOME/sfw
mkdir ibamr
cd ibamr
```

Obtain ibamr through GitHub clone. Note: Need git module in your environment

```
git clone https://github.com/IBAMR/IBAMR.git
```

Install an optimized build of IBAMR:

```
cd $HOME/sfw/ibamr
mkdir ibamr-objs-opt
cd ibamr-objs-opt
export BOOST_ROOT=$HOME/sfw/linux/boost/1.63.0/
export PETSC_ARCH=killdevil-opt
export PETSC_DIR=$HOME/sfw/petsc/3.7.6
```

Same as on Home computer. The flags have capital O, not zero 0. The compilers are changed to match that of killdevil

```
../IBAMR/configure \
CC=mpicc \
CXX=mpicxx \
F77=mpif90 \
FC=mpif90 \
MPICC=mpicc \
MPICXX=mpicxx \
CFLAGS="-O3 -Wall" \
CXXFLAGS="-O3 -Wall -std=gnu++11" \
FFLAGS="-O3 -Wall" \
FCFLAGS="-O3 -Wall" \
CPPFLAGS="-DOMPI_SKIP_MPICXX" \
--with-hypre=$PETSC_DIR/$PETSC_ARCH \
--with-samrai=$HOME/sfw/samrai/2.4.4/killdevil-g++-opt \
--with-hdf5=$HOME/sfw/linux/hdf5/1.10.1 \
```

```
--with-silo=$HOME/sfw/linux/silo/4.10.2 \
--with-boost=$HOME/sfw/linux/boost/1.63.0 \
--enable-libmesh \
--with-libmesh=$HOME/sfw/linux/libmesh/1.0.0/1.0.0-opt \
--with-libmesh-method=opt
```

```
make lib
make examples
```

Run the IBAMR examples. For example:

```
cd $HOME/sfw/ibamr/ibamr-objs-opt/examples/IB/explicit/ex0
```

NOTE: DO NOT RUN IBAMR PROJECTS UNLESS ON A COMPUTE NODE! THEY WILL GET MAD
Either submit an interactive job or submit a job in queue

Interactive: `bsub -q debug -n 1 -lp /bin/bash`
`./main2d input2d`

Job Queue: `bsub -q hour -n 8 mpirun ./main2d input2d`

If either job above works properly, Congratulations! You now have ibamr installed on killdevil!
Now your jobs can be completed with multiple processors and take a lot less time. Time to have fun! :)