Building Trilinos/Albany on Mac OS X

Peter Bosler
Sandia National Laboratories
Center for Computing Research

June 17, 2015

The following procedures are guidelines for building Trilinos/Albany on Mac OS X and have been tested on the following systems:

1. 15" MacBook Pro, 2.8 GHz Intel Core i7, Mac OS X 10.9.5, XCode Version 6.2
2. 6-core Mac Pro, 3.5 GHz Intel Xeon E5, Mac OS X 10.9.5, XCode Version 6.2

1 Prerequisites

1. XCode must be installed. (Get XCode from Apple's App Store).
   (a) XCode command line tools must be installed. After XCode has installed, in a terminal window type: `sudo xcodebuild -license`

   This will install Apple's latest version of the CLang compilers from LLVM, and the LLVM debugger `lldb`. **Note:** This does not include a Fortran compiler.
2. The following packages are very helpful and can be obtained fairly easy using MacPorts (`www.macports.org`). Alternatively, these may be downloaded separately from their respective vendors and custom-built to your specifications. MacPorts installs got to `/opt/local` by default.
   - zlib
   - GCC Compilers (This does include a Fortran compiler, `gfortran`)
   - GCCSelect (This allows easy switching between Clang and GCC compilers)
   - CMake
   - Doxygen
   - GraphViz
   - GLM
   - Parmetis

1.1 OpenMPI

You'll need to configure a build of OpenMPI for each compiler suite you intend to use with Trilinos. To begin, download the latest stable release (currently 1.8.3) from `www.open-mpi.org`, unpack the tarball and from the OpenMPI root directory, do the following:

1. For Apple/CLang:

- Select the CLang compilers:

  ```
  sudo port select --set gcc none
  ```

- Configure:

  ```
  sudo mkdir /opt/openmpi-1.8.3/llvm-5.1
  ```

  ```
  ./configure --prefix=/opt/openmpi-1.8.3/llvm-5.1 CC=gcc CXX=g++ 2>&1 \
  | tee configClangOut.txt
  ```

- Build:
  ```
  make -j 6 2&1 | tee buildClangOut.txt
  ```
- Install:
  ```
  sudo make install
  ```

2. For GCC Compilers

   - Select the GCC compilers:

     ```
     sudo port select --set gcc mp-gcc49
     ```

   - Configure. From the OpenMPI root directory, type:

     ```
     sudo mkdir /opt/openmpi-1.8.3/gcc-4.9
     ```

     ```
     ./configure --prefix=/opt/openmpi-1.8.3/gcc-4.9 CC=gcc CXX=g++ FC=gfortran 2>&1 \
     | tee configOut.txt
     ```

   - Build:
     ```
     make -j 6 2>&1 | tee buildGCCOut.txt
     ```
   - Install:
     ```
     sudo make install
     ```

1.2 Boost

Download the latest Boost release from `www.boost.org` (currently 1.56). These instructions assume a GCC build; to build with CLang, change the relevant lines to your CLang/MPI installation paths. From the Boost root directory, do the following:

- Configure:

  ```
  sudo mkdir /opt/boost-1.56
  ```

  ```
  ./bootstrap.sh --prefix=/opt/boost-1.56 2>&1 | tee configBoostOut.txt
  ```

- Enable Parallel libraries: Edit the `project-config.jam` file to add your MPI path.

  ```
  if ! darwin in [ feature.values <toolset> ]
  {
      using darwin ;
      using mpi : /opt/openmpi-1.8.3/gcc-4.9/bin/mpicxx ;
  }
  ```

- Build:

  ```
  ./b2 -j 6 2>&1 | tee boostBuildOut.txt
  ```

- Install:

  ```
  sudo ./b2 install
  ```

1.3 HDF5

HDF5 is a trickier part of this install. You might be tempted to download it from MacPorts, but that port typically doesn't have the parallel setup that Trilinos and NetCDF require. Also, some Trilinos packages are incompatible with the latest version of HDF5 (currently 1.8.13). And complicating things further, HDF5 version 1.8.11 is incompatible with Mac OS X. Fortunately, HDF5 version 1.8.12 seems to work with both OS X and Trilinos, so download that version (this may require some creative Googling to find), unpack the tarball, and switch to the HDF5 root directory.

- Configure:

```
sudo mkdir /opt/hdf5-1.8.12

./configure --prefix=/opt/hdf5-1.8.12 --enable-parallel CC=mpicc CXX=mpicxx FC=mpifort \
CFLAGS=-I/opt/local/include LDFLAGS=-L/opt/local/lib --with-zlib=/opt/local 2>\&1 \
| tee configHDF5Out.txt
```

- Build:

```
make -j 6 2>&1 | tee buildHDF5Out.txt
```

- Install:

```
sudo make install
```

1.4 NetCDF

Download the current NetCDF release from `www.unidata.ucar.edu/software/netcdf/` (currently 4.3.2). In the NetCDF source tree, locate the file `netcdf.h` and find the `#define` statements listed below; if they do not have the same numeric values listed here, change them.

```
#define NC_MAX_DIMS 65536
#define NC_MAX_ATTRS 8192
#define NC_MAX_VARS 524288
#define NC_MAX_NAME 256
#define NC_MAX_VAR_DIMS 8
```

- Configure:

```
sudo mkdir /opt/netcdf-4.3.2

./configure --prefix=/opt/netcdf-4.3.2 CC=mpicc CXX=mpicxx FC=mpifort --enable-netcdf4 \
--enable-parallel-tests --disable-dap \
CFLAGS="-I/opt/local/include -I/opt/hdf5-1.8.12/include -O3" \
LDFLAGS="-L/opt/local/lib -L/opt/hdf5-1.8.12/lib" 2>\&1 | tee configNetCDFOut.txt
```

- Build:

```
make -j 6 2>&1 | tee buildNetCDFOut.txt
```

- Install:

```
sudo make install
```

1.5 SuperLU

Use Google to find the latest release of SuperLU (currently 4.3). In the SuperLU source tree, edit the `make.inc` file to match the following (except for the SuperLUroot line)

```
SuperLUroot = /Users/pabosle/superlu
BLASDEF  = -DUSE_VENDOR_BLAS
BLASLIB  = -L/usr/lib -lblas
ARCH         = ar
ARCHFLAGS    = cr
RANLIB       = ranlib
CC           = gcc
CFLAGS       = -DPRNTlevel=0 -O3
NOOPTS       =
FORTRAN      = gfortran
FFLAGS       = -O2
LOADER       = $(CC)
LOADOPTS     =
CDEFS        = -DAdd_
#MATLAB      =
```

- Build:

  ```
  make -j 6 2>&1 | tee buildSuperLUOut.txt
  ```

- Install:

  ```
  sudo mkdir /opt/superlu-4.3
  sudo mkdir /opt/superlu-4.3/include
  sudo mkdir /opt/superlu-4.3/lib
  cd SRC/include
  sudo cp *.h /opt/superlu-4.3/include/.
  cd ../lib
  sudo cp *.* /opt/superlu-4.3/lib/.
  ```

2 Trilinos

Download Trilinos.

Trilinos follows the standard configure, build, install recipe.
However, there are so many options available with Trilinos the configure step is best accomplished using a script, like the listing below, the file `configureTrilinosForAlbany.sh`, located in the build directory.

- Make a build directory and an install directory:

  ```
  mkdir $HOME/Trilinos-gcc49-albany
  mkdir $HOME/TrilinosInstalls/albany-gcc49
  ```

- Configure:

  Save your configure file in your build directory.

  ```
  cd $HOME/Trilinos-gcc49-albany
  ```

```
./configureTrilinosForAlbany.sh 2>&1 | tee configTrilinosOut.txt
```

- Build:

```
make -j 6 2>&1 | tee buildTrilinosOut.txt
```

- Install:

```
make install
```

Listing 1: configureTrilinosForAlbany.sh

```
#!/bin/bash

TRILINOS_SRC=/Users/pabosle/Trilinos

INSTALL_DIR=/Users/pabosle/TrilinosInstalls/albany-gcc49

MPI_ROOT=/opt/openmpi-1.8.3/gcc-4.9

HDF5_DIR=/opt/hdf5-1.8.12
BOOST_DIR=/opt/boost-1.56
NETCDF_DIR=/opt/netcdf-4.3.2
PARMETIS_DIR=/opt/parmetis
SUPERLU_DIR=/opt/superlu-4.3
GLM_DIR=/opt/glm

EXTRA_ARGS=$@

rm -rf CMakeFiles/ CMakeCache.txt

cmake \
-D CMAKE_INSTALL_PREFIX:PATH=$INSTALL_DIR \
-D CMAKE_BUILD_TYPE:STRING=NONE \
-D CMAKE_C_FLAGS:STRING='-O2 -g' \
-D CMAKE_CXX_FLAGS:STRING='-O2 -g' \
\
-D Trilinos_WARNINGS_AS_ERRORS_FLAGS:STRING=' ' \
-D Trilinos_ENABLE_ALL_PACKAGES:BOOL=OFF \
-D Trilinos_ENABLE_ALL_OPTIONAL_PACKAGES:BOOL=OFF \
-D Trilinos_ENABLE_TESTS:BOOL=OFF \
-D Trilinos_ENABLE_EXAMPLES:BOOL=OFF \
-D Trilinos_ENABLE_CXX11:BOOL=ON \
-D Kokkos_ENABLE_CXX11:BOOL=ON \
-D Teuchos_ENABLE_LONG_LONG_INT:BOOL=ON \
\
-D TPL_ENABLE_MPI:BOOL=ON \
-D MPI_BASE_DIR:PATH=$MPI_ROOT \
\
-D TPL_ENABLE_BoostLib:BOOL=ON \
-D TPL_ENABLE_BoostAlbLib:BOOL=ON \
-D Boost_INCLUDE_DIRS:FILEPATH=$BOOST_DIR/include \
```

```
-D Boost_LIBRARY_DIRS:FILEPATH=$BOOST_DIR/lib \
-D BoostLib_INCLUDE_DIRS:FILEPATH=$BOOST_DIR/include \
-D BoostLib_LIBRARY_DIRS:FILEPATH=$BOOST_DIR/lib \
-D BoostAlbLib_INCLUDE_DIRS:FILEPATH=$BOOST_DIR/include \
-D BoostAlbLib_LIBRARY_DIRS:FILEPATH=$BOOST_DIR/lib \
\
-D TPL_ENABLE_HDF5:BOOL=ON \
-D HDF5_INCLUDE_DIRS:PATH=$HDF5_DIR/include \
-D HDF5_LIBRARY_DIRS:PATH=$HDF5_DIR/lib \
\
-D TPL_ENABLE_Netcdf:BOOL=ON \
-D TPL_Netcdf_INCLUDE_DIRS=$NETCDF_DIR/include \
-D TPL_Netcdf_LIBRARY_DIRS=$NETCDF_DIR/lib \
\
-D TPL_ENABLE_ParMETIS:STRING=ON \
-D ParMETIS_INCLUDE_DIRS:PATH=$PARMETIS_DIR/include \
-D ParMETIS_LIBRARY_DIRS:PATH=$PARMETIS_DIR/lib \
\
-D TPL_ENABLE_SuperLU:BOOL=ON \
-D SuperLU_INCLUDE_DIRS:PATH=$SUPERLU_DIR/include \
-D SuperLU_LIBRARY_DIRS:PATH=$SUPERLU_DIR/lib \
\
-D TPL_ENABLE_Matio:BOOL=OFF \
-D Trilinos_ENABLE_Zoltan:BOOL=ON \
-D Trilinos_ENABLE_Zoltan2:BOOL=ON \
-D Zoltan2_ENABLE_Experimental:BOOL=ON\
-D Zoltan_ENABLE_ULONG_IDS:BOOL=ON \
-D ZOLTAN_BUILD_ZFDRIVE:BOOL=OFF \
\
-D Trilinos_ENABLE_OpenMP:BOOL=ON \
-D TPL_ENABLE_OpenMP:BOOL=OFF \
-D TPL_ENABLE_Pthread=OFF \
-D Trilinos_ENABLE_Kokkos:BOOL=ON \
-D Trilinos_ENABLE_KokkosClassic:BOOL=ON \
-D Trilinos_ENABLE_KokkosCore:BOOL=ON \
-D Trilinos_ENABLE_TeuchosKokkosCompat:BOOL=ON \
-D Trilinos_ENABLE_TpetraKernels:BOOL=ON \
-D Trilinos_ENABLE_TeuchosKokkosComm:BOOL=ON \
-D Trilinos_ENABLE_KokkosContainers:BOOL=ON \
-D Trilinos_ENABLE_KokkosCompat:BOOL=ON \
-D Trilinos_ENABLE_KokkosTPL:BOOL=ON \
-D Trilinos_ENABLE_KokkosLinAlg:BOOL=ON \
-D Trilinos_ENABLE_KokkosAlgorithms:BOOL=ON \
-D Trilinos_ENABLE_KokkosMpiComm:BOOL=ON \
-D Trilinos_ENABLE_KokkosExample:BOOL=OFF \
-D Kokkos_ENABLE_EXAMPLES:BOOL=OFF \
-D Kokkos_ENABLE_TESTS:BOOL=OFF \
-D Kokkos_ENABLE_CUDA:BOOL=OFF \
-D Kokkos_ENABLE_OpenMP:BOOL=OFF \
-D Kokkos_ENABLE_Serial:BOOL=ON \
```

```
-D Kokkos_ENABLE_Pthread:BOOL=OFF \
-D Kokkos_ENABLE_Thrust=OFF \
\
-D Trilinos_ENABLE_Teuchos:BOOL=ON \
-D Teuchos_ENABLE_COMPLEX:BOOL=OFF \
\
-D Trilinos_ENABLE_Phalanx:BOOL=ON \
-D Phalanx_ENABLE_TEUCHOS_TIME_MONITOR:BOOL=ON \
-D Phalanx_ENABLE_COMPILETIME_ARRAY_CHECK:BOOL=ON \
-D Phalanx_ENABLE_EXAMPLES:BOOL=OFF \
-D Phalanx_KOKKOS_DEVICE_TYPE:STRING='SERIAL' \
-D Phalanx_INDEX_SIZE_TYPE='INT' \
-D KokkosClassic_DefaultNode:STRING='Kokkos::Compat::KokkosSerialMPWrapperNode' \
\
-D Trilinos_ENABLE_Stokhos:BOOL=ON \
-D Stokhos_ENABLE_TEUCHOS_TIME_MONITOR:BOOL=ON \
\
-D Trilinos_ENABLE_Stratimikos:BOOL=ON \
-D Stratimikos_ENABLE_TEUCHOS_TIME_MONITOR:BOOL=ON \
\
-D Trilinos_ENABLE_Tpetra:BOOL=ON \
-D Tpetra_ENABLE_Kokkos_Refactor:BOOL=ON \
\
-D Trilinos_ENABLE_Piro:BOOL=ON \
-D Trilinos_ENABLE_Shards:BOOL=ON \
-D Trilinos_ENABLE_Sacado:BOOL=ON \
-D Trilinos_ENABLE_Epetra:BOOL=ON \
-D Trilinos_ENABLE_EpetraExt:BOOL=ON \
-D Trilinos_ENABLE_Ifpack:BOOL=ON \
-D Trilinos_ENABLE_AztecOO:BOOL=ON \
-D Trilinos_ENABLE_Amesos:BOOL=ON \
-D Trilinos_ENABLE_Anasazi:BOOL=ON \
-D Trilinos_ENABLE_Belos:BOOL=ON \
-D Trilinos_ENABLE_ML:BOOL=ON \
-D Trilinos_ENABLE_Intrepid:BOOL=ON \
-D Trilinos_ENABLE_NOX:BOOL=ON \
-D Trilinos_ENABLE_Thyra:BOOL=ON \
-D Trilinos_ENABLE_Rythmos:BOOL=ON \
-D Trilinos_ENABLE_MOOCHO:BOOL=ON \
-D Trilinos_ENABLE_OptiPack:BOOL=ON \
-D Trilinos_ENABLE_GlobiPack:BOOL=ON \
-D Trilinos_ENABLE_Isorropia:BOOL=ON\
-D Trilinos_ENABLE_Galeri:BOOL=ON\
-D Trilinos_ENABLE_STKIO:BOOL=ON \
-D Trilinos_ENABLE_STKMesh:BOOL=ON \
-D Trilinos_ENABLE_SEACASExodus:BOOL=ON \
-D Trilinos_ENABLE_Teko:BOOL=ON \
-D Trilinos_ENABLE_MueLu:BOOL=ON \
-D Trilinos_ENABLE_Ifpack2:BOOL=ON \
-D Trilinos_ENABLE_Amesos2:BOOL=ON \
```

```
-D Trilinos_ENABLE_TrilinosCouplings:BOOL=ON \
-D Trilinos_ENABLE_ThyraTpetraAdapters:BOOL=ON \
-D Trilinos_ENABLE_Didasko:BOOL=ON \
-D Trilinos_ENABLE_ThreadPool:BOOL=ON \
-D Trilinos_ENABLE_SEACASIoss:BOOL=ON \
-D Trilinos_ENABLE_Pamgen:BOOL=ON \
-D Trilinos_ENABLE_STKIO:BOOL=ON \
\
-D Trilinos_ENABLE_DEBUG:BOOL=OFF \
-D Trilinos_ENABLE_FEI:BOOL=OFF\
-D Trilinos_ENABLE_Mesquite:BOOL=OFF\
-D Trilinos_ENABLE_TriKota:BOOL=OFF \
-D Trilinos_ENABLE_STKClassic:BOOL=OFF \
$EXTRA_ARGS \
$TRILINOS_SRC
```

3 Albany

Configuring Albany is much simpler than Trilinos, as Albany will use the Trilinos install to define most of its configuration parameters. To begin, clone Albany from `www.github.com`, and `cd` into the Albany root directory. Albany is also configured with a script, `configureAlbany.sh` listed below.

- Configure:

  ```
  mkdir build
  ```

  Save your configure file in the `build` directory.

  ```
  cd build
  ```

  ```
  ./configureAlbany.sh 2>&1 | tee configAlbanyOut.txt
  ```

- Build:

  ```
  make -j 6 2>&1 | tee buildAlbanyOut.txt
  ```

Listing 2: configureAlbany.sh

```
#!/bin/bash

export TRILINOS_INSTALL_DIR=$HOME/TrilinosInstalls/albany-gcc49
export ALBANY_ROOT=$HOME/Albany



cmake \
-DALBANY_TRILINOS_DIR:PATH=${TRILINOS_INSTALL_DIR} \
-DENABLE_64BIT_INT:BOOL=ON \
-DENABLE_AERAS:BOOL=ON \
-DENABLE_SCOREC:BOOL=OFF \
-DENABLE_LCM:BOOL=OFF \
-DENABLE_SEE:BOOL=OFF \
-DENABLE_CHECK_FPE:BOOL=ON \
-DENABLE_ALBANY_EPETRA_EXE:BOOL=OFF \
```

```
-DENABLE_LCM_SPECULATIVE:BOOL=OFF  \
-DENABLE_HYDRIDE:BOOL=OFF  \
-DENABLE_SG_MP:BOOL=OFF  \
-DENABLE_QCAD:BOOL=OFF  \
-DENABLE_MOR:BOOL=OFF  \
$ALBANY_ROOT
```