Initial Results

Samuel Grayson
grayson5

March 10, 2022

1 Installation

Enzo depends on MPI, HDF5, HYPRE, PAPI, and Grackle.

First, I searched for the packages on the User Guide for the campus cluster. This explains how to acquire an MPI implementation: `module load mvapich2/2.3-gcc-7.2.0`.

Second, I searched `module avail`. I found PAPI: `module load papi`.

Third, I resorted to downloading and building from source. Grackle already had a build file for the UIUC campus, but I needed to modify it as such:

```
diff --git a/src/clib/Make.mach.uiuc-campus-gnu b/src/clib/Make.mach.uiuc-campus-gnu
index ee17f2d..cf176fc 100644
--- a/src/clib/Make.mach.uiuc-campus-gnu
+++ b/src/clib/Make.mach.uiuc-campus-gnu
@@ -18,7 +18,7 @@ MACH_FILE  = Make.mach.uiuc-campus-gnu
 # Install paths (local variables)
 #----------------------------------------------------------------------

-LOCAL_HDF5_INSTALL = /projects/ncsa/grav/softwares/miniconda2
+LOCAL_HDF5_INSTALL = /home/grayson5/hdf5-1.12.1/hdf5
 LOCAL_FC_INSTALL   = /usr/local/gcc-4.7.1/lib64

 #----------------------------------------------------------------------
@@ -89,6 +89,6 @@ MACH_LIBS         = $(LOCAL_LIBS_HDF5) $(LOCAL_LIBS_MACH)

-MACH_INSTALL_PREFIX = $(HOME)/local
+MACH_INSTALL_PREFIX = $(HOME)/grackle
 MACH_INSTALL_LIB_DIR =
 MACH_INSTALL_INCLUDE_DIR =
```

HDF5 I built with `CC=gcc ./configure && make`. I don't need to `make install` because I don't have root permission on the login node. Therefore, I will build the software in my user directory and point Enzo to that location.

I was unable to build HYPRE. The configure script fails with because it is unable to find symbols

```
gcc -o conftest -g -O2   conftest.c   -L/usr/local/mpi/mvapich2/2.3/gcc/7.2.0/lib
 ↪  -L/usr/local/gcc/7.2.0/lib64/../lib64 -L/usr/local/gcc/7.2.0/lib/../lib64
 ↪  -L/usr/local/gcc/7.2.0/lib/gcc/x86_64-pc-linux-gnu/7.2.0
 ↪  -L/usr/local/gcc/7.2.0/lib/gcc/x86_64-pc-linux-gnu/7.2.0/../../../../lib64 -L/lib/../lib64
 ↪  -L/usr/lib/../lib64 -L/usr/local/gcc/7.2.0/lib64 -L/usr/local/gcc/7.2.0/lib
 ↪  -L/usr/local/gcc/7.2.0/lib/gcc/x86_64-pc-linux-gnu/7.2.0/../../.. -lmpifort -lmpi -lgfortran
 ↪  -lm -lquadmath >&5
/usr/.../libmpi.so: undefined reference to `ibv_modify_xrc_rcv_qp@IBVERBS_1.1'
/usr/.../libmpi.so: undefined reference to `ibv_unreg_xrc_rcv_qp@IBVERBS_1.1'
/usr/.../libmpi.so: undefined reference to `ibv_open_xrc_domain@IBVERBS_1.1'
/usr/.../libmpi.so: undefined reference to `ibv_create_xrc_srq@IBVERBS_1.1'
/usr/.../libmpi.so: undefined reference to `ibv_close_xrc_domain@IBVERBS_1.1'
/usr/.../libmpi.so: undefined reference to `ibv_reg_xrc_rcv_qp@IBVERBS_1.1'
/usr/.../libmpi.so: undefined reference to `ibv_create_xrc_rcv_qp@IBVERBS_1.1'
collect2: error: ld returned 1 exit status
```

It seems these symbols come from `libibverbs`, which is a library for interacting with InfiniBand technology. This is specific to the particular hardware manufacturer (e.g. IBM version, Mellanox version). I don't know how the Campus Cluster is put together. The Campus Cluster User Guide mentions InfiniBand, but it doesn't explain how to use it.

I tried installing the Mellanox version of libibverbs, which requires libnl. I installed `libnl` with `./configure --prefix=$PWD && make -j && make install -j`. However, I still couldn't build `libibverbs` because the `./configure.sh` script coulnd't detect `libnl`, despite my setting environment variables. `LDFLAGS=-L$HOME/libnl-3.2.25/lib CFLAGS=-I$HOME/libnl-3.2.25/include ./configure` did not help.

I tried building Enzo, to see how far I could get without having HYPRE. I based the Enzo build configuration for the Campus Cluster on the build configuration for Ubuntu with the following changes:

```
--- Make.mach.ubuntu        2022-03-06 11:34:14.949188056 -0600
+++ Make.mach.campuscluster        2022-03-10 00:22:53.304000188 -0600
-MACH_TEXT  = Use apt-get to install libhdf5-serial-dev gfortran openmpi-bin libopenmpi-dev
+MACH_TEXT  = Illinois Campus Cluster
 MACH_VALID = 1
-MACH_FILE  = Make.mach.ubuntu
+MACH_FILE  = Make.mach.campuscluster
+MACHINE_NOTES  = "Don't forget to run: module load mvapich2/2.3-gcc-7.2.0 papi"

 #----------------------------------------------------------------------
 # Install paths (local variables)
 #----------------------------------------------------------------------

-LOCAL_GRACKLE_INSTALL = $(HOME)/local
-LOCAL_HYPRE_INSTALL = $(HOME)/local
+LOCAL_HDF5_INSTALL    = $(HOME)/hdf5-1.12.1/hdf5
+LOCAL_GRACKLE_INSTALL = $(HOME)/grackle
+LOCAL_HYPRE_INSTALL   = $(HOME)/local
+LOCAL_PAPI_INSALL     = /usr/local/papi/5.6.0

 #----------------------------------------------------------------------
 # Compiler settings
@@ -77,27 +65,27 @@
 # Includes
 #----------------------------------------------------------------------

-LOCAL_INCLUDES_MPI    = # MPI includes
-LOCAL_INCLUDES_HDF5   = -I/usr/include/hdf5/serial # HDF5 includes
-LOCAL_INCLUDES_HYPRE  = -I$(LOCAL_HYPRE_INSTALL)/include
-LOCAL_INCLUDES_PAPI   = # PAPI includes
```

```
+LOCAL_INCLUDES_MPI     =
+LOCAL_INCLUDES_HDF5    = -I$(LOCAL_HDF5_INSTALL)/include
+LOCAL_INCLUDES_HYPRE   =
+LOCAL_INCLUDES_PAPI    = -I$(LOCAL_PAPI_INSTALL)/include
 LOCAL_INCLUDES_GRACKLE = -I$(LOCAL_GRACKLE_INSTALL)/include

 MACH_INCLUDES         = $(LOCAL_INCLUDES_HDF5)
 MACH_INCLUDES_MPI     = $(LOCAL_INCLUDES_MPI)
 MACH_INCLUDES_HYPRE   = $(LOCAL_INCLUDES_HYPRE)
 MACH_INCLUDES_PAPI    = $(LOCAL_INCLUDES_PAPI)
-MACH_INCLUDES_GRACKLE  = $(LOCAL_INCLUDES_GRACKLE)
+MACH_INCLUDES_GRACKLE = $(LOCAL_INCLUDES_GRACKLE)

 #-----------------------------------------------------------------------
 # Libraries
 #-----------------------------------------------------------------------

-LOCAL_LIBS_MPI    = # MPI libraries
-LOCAL_LIBS_HDF5   = -L/usr/lib/x86_64-linux-gnu/ -lhdf5_serial -lz
-LOCAL_LIBS_HYPRE  = -L$(LOCAL_HYPRE_INSTALL)/lib -lHYPRE
-LOCAL_LIBS_PAPI   = # PAPI libraries
-LOCAL_LIBS_MACH   = -lgfortran # Machine-dependent libraries
+LOCAL_LIBS_MPI     =
+LOCAL_LIBS_HDF5    = -L$(LOCAL_HDF5_INSTALL)/lib -lhdf5
+LOCAL_LIBS_HYPRE   =
+LOCAL_LIBS_PAPI    = -L$(LOCAL_PAPI_INSTALL)/lib -lpapi
+LOCAL_LIBS_MACH    = -lgfortran # Machine-dependent libraries
 LOCAL_LIBS_GRACKLE = -L$(LOCAL_GRACKLE_INSTALL)/lib -lgrackle

 MACH_LIBS         = $(LOCAL_LIBS_HDF5) $(LOCAL_LIBS_MACH)
```

. Enzo also depends on symbols from `libibverbs`.

```
[grayson5@golubh4 enzo]$ make machine-campuscluster
# lots of output

[grayson5@golubh4 enzo]$ make -j
# lots of output
Linking enzo executable. Type  cat out.compile  in case it fails.

[grayson5@golubh4 enzo]$ cat out.compile
/usr/.../libmpi.so: undefined reference to `ibv_reg_xrc_rcv_qp@IBVERBS_1.1'
/usr/.../libmpi.so: undefined reference to `ibv_close_xrc_domain@IBVERBS_1.1'
/usr/.../libmpi.so: undefined reference to `ibv_unreg_xrc_rcv_qp@IBVERBS_1.1'
/usr/.../libmpi.so: undefined reference to `ibv_open_xrc_domain@IBVERBS_1.1'
/usr/.../libmpi.so: undefined reference to `ibv_modify_xrc_rcv_qp@IBVERBS_1.1'
/usr/.../libmpi.so: undefined reference to `ibv_create_xrc_rcv_qp@IBVERBS_1.1'
/usr/.../libmpi.so: undefined reference to `ibv_create_xrc_srq@IBVERBS_1.1'
collect2: error: ld returned 1 exit status
```

1.1 Thoughts on resolution

I will email the `help@campuscluster.illinois.edu` to ask about how to install/use InfiniBand on the Campus Cluster.

2 Data

As such, I couldn't actually produce any data initially. However, I made progress in other ways: I found the code for the paper I want to replicate here. It is written for Arepo/Illustris, which I want to switch to Enzo in my replication study. As such, I have begun adapting that code for Enzo here.