Hyak Compiling HDF5 Serial

Below you will find a sample session demonstrating how to use the Intel compilers to build a serial version of the HDF5 data format library with szlib compression support. This session can serve as a basic guide for compiling most software packages which use the autoconf (configure) and make tools as well.

Change to your directory in your group's space on /gscratch.

```
$ cd /gscratch/mygroup/user
```

Download the package to your directory. You must do this from one of the login servers or a build node. Compute nodes do not have access to the Internet.

```
$ wget http://www.hdfgroup.org/ftp/HDF5/current/src/hdf5-1.8.7.tar.bz2
```

Start an interactive session on a build node. Return to your work directory, unpack the archive, and change to the software build directory.

```
$ srun -p build --time=2:00:00 --mem=100G --pty /bin/bash
$ cd /gscratch/mygroup/user
$ bunzip2 -c hdf5-1.8.5-patch1.tar.bz2 | tar -xf -
$ cd hdf5-1.8.5-patch1
```

Most packages should have a README, INSTALL, or other similarly named file in their root directory. You should always read these for information about how to build the software.

```
$ less README.txt
```

The README.txt file referred us to release_docs/INSTALL. Sometimes they will provide a web page with build information rather than including it in the software distribution as well.

```
$ less release_docs/INSTALL
```

Load the environment configuration scripts for the Intel compilers. You can view all the available modules using module avail (more details).

```
$ module load icc_<version>
```

You can specify additional options to pass to the compilers during the build step. The below options will optimize code as much as possible for the current host. This could cause the compiler to produce code which will not run on other systems in Hyak. The compilers have numerous optimization flags. Please see the man pages or check the documentation on the Intel web site for more information.

```
$ export CFLAGS=-xHost
$ export FCFLAGS=-xHost
$ export CXXFLAGS=-xHost
```

We've previously built a supporting library for hdf5 called szip. It's a compression library. Since we've installed it in a non-system path (/gscratch/mygroup /shared/szip-2.1), we must specify the location where the header files and libraries can be found.

```
$ export CPPFLAGS="-I/gscratch/mygroup/shared/szip-2.1/include"
$ export LDFLAGS="-L/gscratch/mygroup/shared/szip-2.1/lib"
```

We also need to add the lib directory to our shared library path.

```
$ export LD_LIBRARY_PATH=/gscratch/mygroup/shared/szip-2.1/lib:$LD_LIBRARY_PATH
```

Many times there are options available for building a software package that are not listed in the installation documentation. It's useful to review all the options before going ahead.

```
$ ./configure --help
```

Once you've reviewed all the options and decided which ones you want, you can go ahead and configure and build your software.

```
$ ./configure --prefix=/gscratch/mygroup/shared/hdf5-1.8.5-patch1_intel --with-szlib \
  --enable-cxx --enable-fortran --enable-gpfs --with-default-api-version=v16
$ make
```

Often software packages come with a test suite. Running these tests provides some assurance your software package is compiled correctly and will work as expected.

```
$ make check
```

Finally, install your software at the prefix path specified in the configure step.

```
$ make install
```