Building and installing HElib

The HElib build, install, and regression tests suite have been built and tested on Ubuntu 18.04, Ubuntu 20.04, Fedora 32, Fedora 33, CentOS 7.8, CentOS 8.2, macOS Mojave >=10.14.6, and macOS Catalina >=10.15.7.

There are two different ways to build and install HElib. The first one will automatically download and build the GMP and NTL dependencies and pack the libraries in a relocatable folder. The second way, instead, requires the dependencies to be installed by you and available in the system.

Please read these instructions in full to better choose the type of build that is better for you.

General prerequisites

- GNU make >= 3.82
- pthreads
- git >= 1.8.3 (required to build and run the HElib test suite)

Linux environment:

- g++ >= 7.3.1
- cmake >= 3.10.2

macOS environment:

- Apple clang >= 11.0.0 (available with the latest Xcode for the tested versions of macOS)
- Xcode Command Line Tools (can be installed with the command `xcode-select --install` in a teminal)
- cmake >= 3.17.3 (available from CMake or MacPorts Project and Homebrew as packages)

For development:

- clang-format >= 9.0.0 (available with your linux distribution and for macOS from MacPorts Project and Homebrew as packages)

Option 1: package build (recommended for most users)

This option bundles HElib and its dependencies (NTL and GMP) in one directory which can then be moved around freely on the system. NTL and GMP will be automatically fetched and compiled. It can be installed globally (i.e. under `/usr/local`), which is the default option if no `CMAKE_INSTALL_PREFIX` is specified, but this should only be done with caution as existing versions of NTL, GMP, or HElib will be overwritten. These additional two prerequisites are required in this case:

- m4 >= 1.4.16
- patchelf >= 0.9 (if building on Linux)

Please note that if changing from library build to package build, it is safer to use a clean build directory.

Instructions

1. Create a build directory, typically as a sibling of `src`:

```
cd HElib
mkdir build
cd build
```

2. Run the cmake configuration step, specifying that you want a package build (via -DPACKAGE_BUILD=ON) and saying where you would like the installation to be. To install in `/home/alice/helib_install`, for example:

```
cmake -DPACKAGE_BUILD=ON -DCMAKE_INSTALL_PREFIX=/home/alice/helib_install ..
```

Extra options can be specified here, such as enabling HElib tests with `-DENABLE_TEST=ON`. See later section entitled "HElib build options" for details.

3. Compile, with an optional number of threads specified (16 in this example). The output of this will be in the relocatable folder `helib_pack`:

```
make -j16
```

4. (optional) If step 2 was performed with `-DENABLE_TEST=ON`, HElib tests can be run as follows:

```
ctest
```

Detailed HElib-specific test logs can be found in `Testing/Temporary/LastTest.log`.

5. (optional) Run the install step, to copy the folder `helib_pack` to `${CMAKE_INSTALL_PREFIX}` (in this example `/home/alice/helib_install`):

```
make install
```

of course, if the `CMAKE_INSTALL_PREFIX` was kept as the default `/usr/local` or some other system-wide path, step 5 may require `sudo` privileges.

Option 2: library build (advanced)

This option involves building HElib on its own, linking against pre-existing dependencies (NTL and GMP) on the system. In this way, the HElib library can be moved around, but its dependencies (NTL and GMP) cannot, as they are absolute paths. For this option, you must build GMP >=6.0.0 and NTL >=11.4.3 yourself. For details on how to do this, please see the section on building dependencies later. It is assumed throughout this installation option that the environment variables `$GMPDIR` and `$NTLDIR` are set to point to the installation directories of GMP and NTL respectively.

Please note that if changing from package build to library build, it is safer to use a clean build directory.

1. Create a build directory, typically as a sibling of `src`:

```
cd HElib
mkdir build
cd build
```

2. Run the cmake configuration step, specifying where to find NTL and GMP. If not specified, system-wide locations such as `/usr/local/lib` will be searched. To install in `/home/alice/helib_install`, for example:

```
cmake -DGMP_DIR="${GMPDIR}" -DNTL_DIR="${NTLDIR}" -DCMAKE_INSTALL_PREFIX=/home/alice/helib_install ..
```

Extra options can be specified here, such as enabling HElib tests with `-DENABLE_TEST=ON`. See later section entitled "HElib build options" for details.

3. Compile, with an optional number of threads specified (16 in this example):

```
make -j16
```

4. (optional) If step 2 was performed with `-DENABLE_TEST=ON`, tests can be run as follows:

```
ctest
```

Detailed HElib test logs can be found in `Testing/Temporary/LastTest.log`.

5. Run the install step, to copy the files to `${CMAKE_INSTALL_PREFIX}` (in this example `/home/alice/helib_install`):

```
make install
```

of course, if the `CMAKE_INSTALL_PREFIX` was kept as the default `/usr/local` or some other system-wide path, step 5 may require `sudo` privileges.

Building dependencies (for option 2)

GMP

Many distributions come with GMP pre-installed. If not, you can install GMP as follows.

1. Download GMP from http://www.gmplib.org -- make sure that you get GMP >=6.0.0 (current version is 6.2.0).
2. Decompress and cd into the gmp directory (e.g., `gmp-6.2.0`).
3. GMP is compiled in the standard unix way:

```
        ./configure
        make
        sudo make install
```

This will install GMP into `/usr/local` by default.

NOTE: For further options when building GMP, run `./configure --help` in step 3.

NTL

You can install NTL as follows:

1. Download NTL >=11.4.3 (current version is 11.4.3) from http://www.shoup.net/ntl/download.html
2. Decompress and cd into the directory, e.g., `ntl-11.4.3/src`
3. NTL is configured, built and installed in the standard Unix way (but remember to specify the following flags to `configure`):

```
        ./configure NTL_GMP_LIP=on SHARED=on  NTL_THREADS=on NTL_THREAD_BOOST=on
        make
        sudo make install
```

This should install NTL into `/usr/local`.

NOTE: For further options when building NTL, run `./configure --help` in step 3.

NOTE: if linking against a non-system GMP, pass `GMP_PREFIX=<path/to/gmp>` to the `./configure` step.

HElib build options

Generic options

- `BUILD_SHARED=ON/OFF` (default is `OFF`): Build as a shared library. Note that building HElib (regardless of `BUILD_SHARED`) will fail if NTL is not built as a shared library. The default for NTL is static library, to build NTL as a shared library use `./configure SHARED=on` in step 1.
- `CMAKE_BUILD_TYPE`: (default is `RelWithDebInfo`): Choose the type of build, options are: `Debug`, `RelWithDebInfo`, `Release`, `MinSizeRel`.
- `CMAKE_INSTALL_PREFIX`: Desired installation directory for HElib.
- `ENABLE_TEST=ON/OFF` (default is `OFF`): Enable building of tests. This will include an automatic download step for the google test framework stable release (googletest v1.10.0)
- `ENABLE_THREADS=ON/OFF` (default is `ON`): Enable threading support. This must be on if and only if NTL was built with `NTL_THREADS=ON`.
- `PEDANTIC_BUILD=ON/OFF` (default is `ON`): Use `-Wall -Wpedantic -Wextra -Werror` during build.
- `HELIB_DEBUG=ON/OFF` (default is `OFF`): Activate the debug module when building HElib (by defining the `HELIB_DEBUG` macro). When the debug module is active, this generates extra information used for debugging purposes.

`HELIB_DEBUG` will propagate to programs using HElib, when using cmake. When this is enabled, programs using HElib will generate a warning during configuration. This is to remind the user that use of the debug module can cause issues, such as `sigsegv`, if initialized incorrectly.

Parameters specific to option 1 (package build)

- `PACKAGE_DIR`: Location that a package build will be installed to. Defaults to `${CMAKE_INSTALL_PREFIX}/helib_pack`.
- `FETCH_GMP`: Whether or not to fetch and build GMP. Defaults to `ON`. If set to `OFF`, there should either exist a system-installed GMP library, or `GMP_DIR` should point to a valid GMP prefix.
- `GMP_DIR`: Prefix of the GMP library. Ignored if `FETCH_GMP=ON`.

Parameters specific to option 2 (library build)

- `GMP_DIR`: Prefix of the GMP library.
- `NTL_DIR`: Prefix of the NTL library.

Using HElib in a project

Standard method

After `make install` has been run in either option 1 or option 2, one can find the required shared library files to link against in `lib` and the header files in `include`. These can be used in the preferred way with your build system of choice.

Package build with cmake

Another, easier way is possible if you are using HElib in a cmake project.

1. Include the following line in your `CMakeLists.txt`:

```
find_package(helib)
```

2. Run your `cmake` step with `-Dhelib_DIR=<helib install prefix>/share/cmake/helib`.

Example

Full working examples of cmake-based projects which uses HElib can be found in the `examples` directory.