LAMMPS on CMake

I couldn't stand the pain anymore

Christoph Junghans
Applied Computer Science
(CCS-7)
Aug, 2nd 2017
CMake for LAMMPS

- **Current make-based buildsystem of LAMMPS is an impediment**
  - Non-standard: Hard for new users/developers and distribution maintainers
  - Lots of duplication: Hard to maintain
  - Code gets copied around: very confusing

- **Adding a secondary build system:**
  [https://github.com/lammps/lammps/pull/573](https://github.com/lammps/lammps/pull/573)
  - Give users/developers some time to switch: Backwards support (C++ code untouched)
  - Shrink build system from 20K lines to 1k (no actual code refactor yet): easier to maintain
  (Compare: $ find . -iname "Makefile.*" -or -iname "Make.*" -or -iname "Install.*" | xargs cat | wc -l
  vs.
  $ find cmake -type f | xargs cat | wc -l)
  - More standard: easier to use and contribute (existing tutorials/documentation for CMake)

- **Overall: increase productivity through modernization**
  - Further modernization after code refactor
  - Better support of ECP platform (Cray support CMake)
Demo

CMake 3.8.2 - /Users/junghans/computing/gentoo/lammps/build

Where is the source code: /Users/junghans/computing/gentoo/lammps/cmake
Where to build the binaries: /Users/junghans/computing/gentoo/lammps/build

Search: [ ] Grouped [ ] Advanced [ ] Add Entry [ ] Remove Entry

Name | Value
--- | ---
BUILD_SHARED_LIBS | False
CMERGE_BUILD_TYPE | Release
CMERGE_INSTALL_PREFIX | /usr/local
CMERGE_OSS_ARCHITECTURES | 
CMERGE_OSS_DEPENDENCY | 
CMERGE_OSS_SYSROOT | /Applications/Xcode.app/Contents/Developer/
ENABLE_ALL | False
ENABLE_ASPHERE | False
ENABLE_BOND | False
ENABLE_CAISS | False
ENABLE_COLUMN | False
ENABLE_COMPRESS | False
ENABLE_CORESHELL | False
ENABLE_DIPOLAR | False
ENABLE_DGPUS | False
ENABLE_GRANULAR | False

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Generating style_reader.h...
Generating style_region.h...
Building package: CLASS
Building package: DIPOLAR
Configuring done
Generating done

Terminal output:
```
```
Notes from the Demo

- CMake can do out-of-source builds: allows to nuke build directory
- Using CMake:
  ```
  $ mkdir build; cd build; cmake /path/to/lammps/cmake;
  make -j5
  ```
- Enabling some package:
  ```
  $ cmake /path/to/lammps/cmake -DENABLE_BODY=ON
  (or edit CMakeCache.txt by hand)
  ```
- Curses-based interface (Don’t forget to press ‘g’ when you are done)
  ```
  $ ccmake cmake /path/to/lammps/cmake
  (graphical alternative cmake-gui)
  ```
- Generate Ninja Build instead of Make:
  ```
  $ cmake /path/to/lammps/cmake -G Ninja; ninja -j5
  ```
- Build with external libraries:
  ```
  $ cmake /path/to/lammps/cmake -DENABLE_KIM
  -DKIM_INCLUDE_DIR=/path/of/KIM_API.h
  -D KIM_LIBRARY=/path/to/kim-api-v1
  ```