

LAMMPS on CMake

I couldn't stand the pain anymore



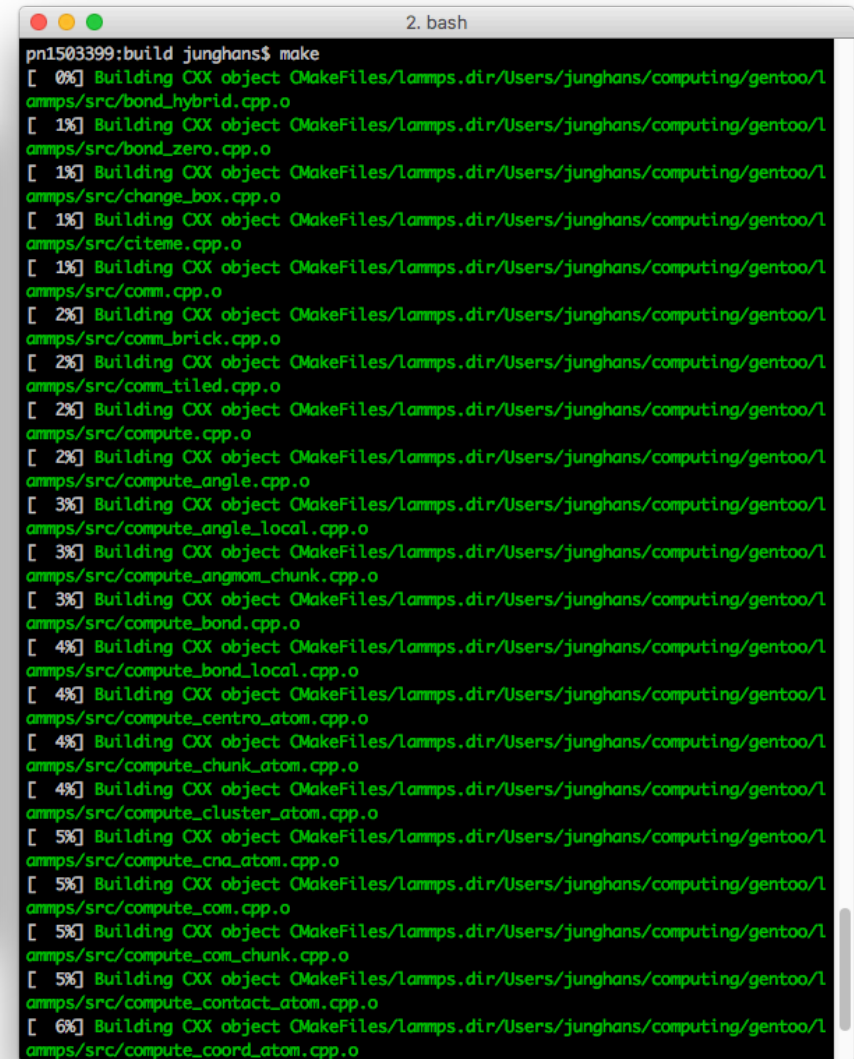
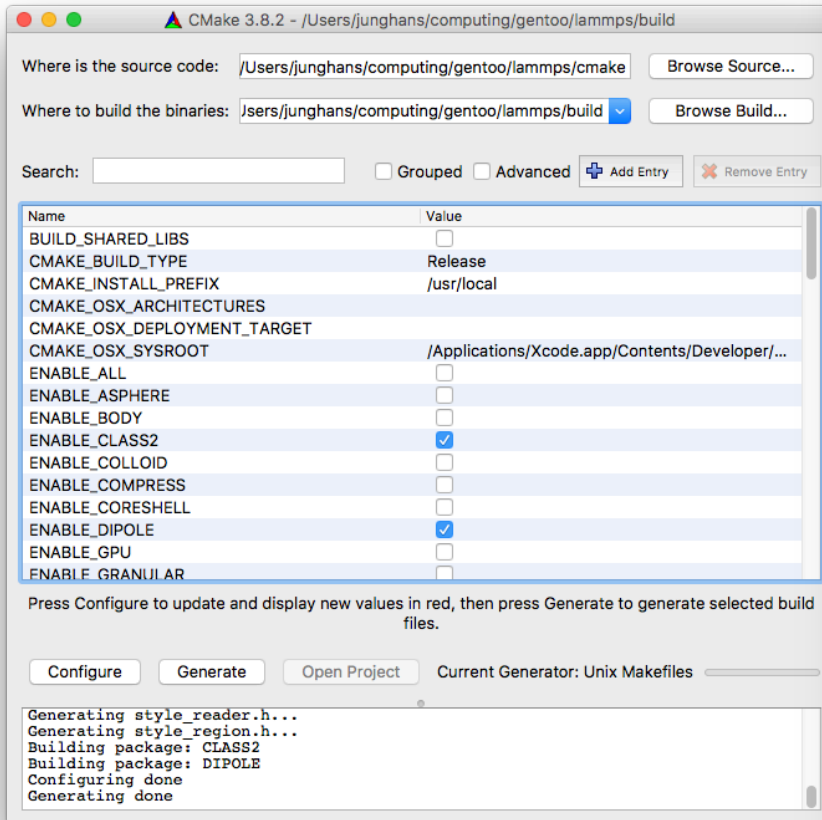
Christoph Junghans
Applied Computer Science
(CCS-7)

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>
 - Give users/developers some time to switch: Backwards support (C++ code untouched)
 - Shrunk 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



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`