Preparing LAMMPS for exascale: the Kokkos approach

Paul Crozier, Christian Trott, Steve Plimpton

March 2014

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

As we head towards exascale, next-generation hardware will be heterogeneous and many-core, and MPI everywhere will be inadequate for good performance. Many legacy codes will therefore need to be reformulated to expose more parallelism and it will become increasingly desirable for software to be portable and efficient across a variety of heterogeneous hardware configurations.

Our LAMMPS (http://lammps.sandia.gov) molecular simulation software package currently uses spatial decomposition and MPI to achieve parallel scalability. Recently-added GPU and OMP packages enable users to go beyond MPI-only and speed up simulations using GPU or many-core hardware. However, these packages of rewritten kernels have limitations in terms of portability, maintainability, and efficiency, especially with uncertainty in possible exascale computer designs of the future.

Kokkos, a C++ template library (http://trilinos.sandia.gov/packages/kokkos/), has been developed at Sandia to provide scientific programmers the tools necessary for portable manycore performance. Kokkos attempts to achieve this by providing a single portable API without exposing the programmer to device-specific programming models such as CUDA, pthreads and OpenMP. It supplies both parallel dispatching methods and a data abstraction layer, separating algorithms and their data access from actual memory layout. Layout is chosen at compiler time, optimized to the targeted device architecture. The goal of Kokkos is to thereby achieve near optimal execution and data access patterns for a single version of a code kernel on a variety of manycore devices. Put another way, the goal is to separate the physics code from the hardware details. Even so, the programmer still needs to rewrite the compute-intensive kernels of their application code, exposing as much fine-grained parallelism as possible in their algorithms.

Work is underway at Sandia (by computer scientists) to improve and expand the functionality of the Kokkos library in the context of various applications, as well as to create a Kokkos-capable version of LAMMPS (by application experts). Prototype Kokkos-enabled versions of miniMD (a molecular dynamics miniapp, available at Mantevo.org) and LAMMPS have shown promise in terms of portability and performance. Our plan is to create and distribute a new optional package with LAMMPS that will include Kokkos versions of many of LAMMPS's most compute- intensive kernels. Our hope is to minimize disruption to the current LAMMPS community while providing a viable path for LAMMPS to perform and scale well on a variety of future heterogeneous platforms.

We note, that because LAMMPS supports a wide variety of models, Kokkos-izing all of them will involve a significant rewrite effort and also require restructuring of some core parts of the code. Kokkos is relatively new and not yet widely used, so it's possible that better programming models will come along, that will force yet another rewrite of compute-intensive kernels. Another risk is that restructuring will lead to greater complexity in the LAMMPS source code, negatively impacting software quality and maintainability. Even so, our current judgment is that this path is a good balance between future-proofing LAMMPS with regards to next-generation hardware, while not requiring a complete re-write of the code.

Further information:

1. LAMMPS homepage: http://lammps.sandia.gov/
2. Kokkos information: http://trilinos.sandia.gov/packages/docs/r8.0/packages/kokkos/doc/html/index.html
3. "Kokkos: Enabling performance portability across manycore architectures", Carter Edwards and Christian Trott, https://www.xsede.org/documents/271087/586927/Edwards-2013-XSCALE13-Kokkos.pdf
4. "A next generation LAMMPS: preparing for the many-core future with Kokkos", Christian Trott, http://lammps.sandia.gov/workshops/Aug13/Trott/LAMMPS-Kokkos3.pdf
5. "Towards Performance-Portable Applications through Kokkos: A Case Study with LAMMPS", Christian Trott, Carter Edwards, and Simon Hammond, http://on-demand.gputechconf.com/supercomputing/2013/presentation/SC3103_Towards-Performance-Portable-Applications-Kokkos.pdf