OLCF-6 LAMMPS Benchmark

Authors: Nick Hagerty and Dilip Asthagiri

This repository describes the Materials Modeling benchmark using LAMMPS. The OLCF-6 benchmark run rules should be reviewed before running this benchmark.

This benchmark derives from the Materials by Design Workflow NERSC-10 benchmark (https://www.nersc.gov/systems/nersc-10/benchmarks/). However, there are major differences in the type of systems considered. Also, there are only two categories of performance optimization, "ported" and "optimized." This is in contrast to the three categories in the Materials by Design Workflow.

Note:

- The OLCF-6 Benchmark Run Rules apply to this benchmark except where explicitly noted within this README.
- The run rules herein define the "ported" category of performance optimization. Responses to the OLCF-6 RFP should include performance estimates for this category; results for "optimized" categories are optional.
- RFP responses should estimate the performance for a problem size greater than or equal to the provided reference problem on the target architecture.
- Concurrency adjustments (i.e. weak- or strong-scaling) may be needed to match the reference time.

Materials Simulation Overview

Molecular dynamics (MD) simulations help one understands the properties of matter from the knowledge of the inter-molecular interactions between the atoms/molecules/particles comprising the system. MD has revolutionized our understanding of materials in domains as diverse as biology, chemistry, physics, and material science. Modeling materials requires being able to describe both short-range and long-range interactions, each of which stresses different aspects of the computational approach, and thus different aspects of the computing architecture itself. To this end, the Materials Modeling benchmark uses two sub-problems, one that stresses the balance of short- and long-range interactions, and another that stresses short-range interactions involving changes that occur at a fast time-scale.

(i) Liquid water is the solvent matrix for all biological processes, and to understand how biomolecules function, it is essential to understand the properties of water itself. A proper description of the thermodynamics of water depends on the correct description of the balance of long-range and short-range forces. This is in contrast to other so-called van der Waals fluids that can be well described by simply accounting for short-range interactions. Thus, the first of the two benchmarks involves the modeling of liquid water at room temperature.

(ii) Biology also relies on chemical transformations, and chemical transformations involve bond making/breaking. The most rigorous description of chemical transformations require electronic structure calculations. However, much progress can be made by using an empirical description of reactivity. One such approach is the so-called ReaxFF approach. In ReaxFF, one identifies the distribution of atoms around a given atom, assesses the bond order, iteratively equilibrates the charge distribution, and uses this information to drive the next step in the dynamics. Clearly, the ReaxFF approach stresses the local (short-range) interactions. Thus, the second of the two benchmarks involves the ReaxFF modeling of Pentaerythritol tetranitrate (PETN), an explosive material and a good test case for modeling reactive transformations at short time scales.

Code access

The required LAMMPS version used for this benchmark, commit #08498637aa (closely follows February 7, 2024 release), is provided with this benchmark distribution. This commit contains required fixes and optimizations required to run the SPC/e benchmark at scale. For reference, it was obtained using the commands below. Optimized runs may use custom modifications of LAMMPS.

```
git clone https://github.com/lammps/lammps.git lammps_src
cd lammps_src/
git checkout 08498637aa
```

Kokkos version 4.2.0 is distributed with and used by this LAMMPS version. Optimized results may use later versions of Kokkos or custom Kokkos backends optimized for the target architecture, as well as code modifications that are expressed in languages or abstractions other than Kokkos.

LAMMPS configuration used in Frontier benchmark

This benchmark was constructed using the LAMMPS commit `08498637aa`, which closely follows February 7, 2024 release, with bug fixes and performance improvements. See https://docs.lammps.org/Build_make.html (https://docs.lammps.org/Build_make.html) for more information about building LAMMPS with the Make-based build system.

Please see the README provided in build/README for information about build system, system requirements, and required packages.

Running the benchmark

Input files and batch scripts for the various systems are provided in the benchmarks directory. Each problem has its own subdirectory within the benchmarks directory. There the appropriate run scripts are included.

Modeling liquid water --- subproblem (i)

Each problem size is constructed by replicating a base system of 32,768 water molecules (98,304 atoms) in the x, y, and z directions via the LAMMPS `replicate` command. For the purpose of this benchmark, velocities are included in the base system's data file. The simulation is under NVT (constant number of particles, volume, and temperature) conditions.

Responses to the OLCF-6 request for proposal (RFP) should provide results (measured or projected) for a problem size greater than or equal to the "reference" size. Figure-of-Merit (FOM) values from OLCF Frontier system were gathered using the "reference" problem size. Other problem sizes have been provided as a convenience, to facilitate profiling at different scales (e.g. socket, node, blade, or rack), and extrapolation to larger sizes. The collection of problems from size `Nano` to `Large` form a weak scaling series where each successively larger problem simulates eight times as many atoms as the previous one. The `Reference` problem occupying all of Frontier (with 9261 nodes) is a factor of about 2.6 larger than the `Large` problem. Computational requirements are expected to scale linearly with the number of atoms.

The following table lists the approximate system sizes of each of these jobs. The simulation log files generated on Frontier provide additional details such as memory and number of CPUs/GPUs used (see also the Results section below).

SPC/E water simulations

Index	Size	Replication factor	# atoms	# Frontier nodes
0	Nano	8 x 8 x 8	50.33 M	1
1	Tiny	16 x 16 x 16	402.65 M	8 (2^3)
2	Small	32 x 32 x 32	3.22 B	64 (4^3)
3	Medium	64 x 64 x 64	25.77 B	512 (8^3)
4	Large	128 x 128 x 128	206.16 B	4096 (16^3)
5	Reference	168 x 168 x 168	466.12 B	9261 (21^3)

ReaxFF modeling of PETN --- subproblem (ii)

Each problem size is constructed by replicating a base system of 58 atoms in the x, y, and z directions via the LAMMPS `replicate` command. These simulations are performed under NVE (constant number of particles, volume, and energy) conditions. The initial velocities are assigned using the same seed for the random number generator.

PETN ReaxFF simulations

Index	Size	Replication factor	# atoms	# Frontier nodes
0	Nano	72 x 72 x 72	21.65 M	1
1	Tiny	144 x 144 x 144	173.19 M	8 (2^3)
2	Small	288 x 288 x 288	1.39 B	64 (4^3)
3	Medium	576 x 576 x 576	11.08 B	512 (8^3)
4	Large	1152 x 1152 x 1152	88.67 B	4096 (16^3)
5	Reference	1512 x 1512 x 1512	200.49 B	9261 (21^3)

Parallel decomposition

LAMMPS uses a 3-D spatial domain decomposition to distribute atoms among MPI processes. The default decomposition divides the simulated space into rectangular bricks. The proportions of the bricks are automatically computed to minimize surface-to-volume ratio of the bricks. LAMMPS will run correctly with any number of MPI processes, but better performance is often obtained when the number of MPI processes is the product of three near-equal integers. If additional control of the domain decomposition is needed, the processors command may (optionally) be added to the input files. This command requires three integer parameters that correspond to the x,y,z dimensions of the process grid. Changes to processors may require updates to the job submission script so that the correct number of MPI processes is started. Refer to the LAMMPS documentation for more information about the processors command. The parameters of the LAMMPS input file may not be modified except for the addition of a processors command.

Job submission

The essential steps are:

1. add symlinks to the files that are common to all runs. These files are the LAMMPS configuration file `input.in` and the data/forcefield descriptions that are common to the SPC/E water simulations and ReaxFF simulations, respectively. These common files are in `benchmark-main`.
2. the size-specific simulation parameters are in the file `base_size_cmd.txt` in the root directory for each potential. (See also Table above.) This file contains the number of times to replicate the base system in the x, y, and z dimensions to form the per-node problem size, which is then replicated by the number of nodes (must be a perfect cube).
3. run the job. For CPU-only runs (for additional validation only), please refer to `run_cpu.slurm` and for the GPU accelerated runs please refer to `run_gpu.slurm`. A helper script named `submit_helper.sh` has been provided.

Results

For both the SPC/E water and ReaxFF simulations, the results obtained with GPU-acceleration set the baseline that must be met or exceeded. The figure of merit (FOM) is `atom-steps/sec`. All runs utilized 8 MPI ranks per node, each rank with 7 CPU cores and 1 GCD of an AMD MI250X.

Verification of correctness

A Python script named `validate.py` has been provided in the directory for each benchmark (ie, `benchmarks/spce` and `benchmarks/reaxff`). This script verifies scientific quantities of interest.

(i) For SPC/e water simulations, the potential energy per molecule, a quantity that is related to the heat of vaporization, must be -11.12 +/- 0.02 kcal/mol/molecule.

For example, to check the results in spce/0_nano/lammps_benchmark_gpu-1691584.out, from the `spce` directory, one can do

```
python validate.py --input 0_nano/lammps_benchmark_gpu-1691584.out
```

(ii) For the PETN ReaxFF simulations, the total energy per cubic Angstrom, the energy density, must be -9.082354 +/- 5e-4 kcal/mol/A^3, and the maximum difference in energy from step 0 to 100 is 5e-4 kcal/mol/A^3.

For additional validation, CPU-only job scripts and input files have been provided for each model at all provided node counts and problem sizes. A Python script named `compare_two_runs.py` has been provided that can be used to verify correctness between CPU-only runs and GPU-enabled runs. The CPU-only runs utilize the OpenMP backend. The default tolerance is set to 0.1% difference between CPU and GPU implementations.

Reference performance on Frontier

In the `validate.py` script, we compute the FOM using the metrics LAMMPS prints in the line starting with `Loop time ...`. The FOM computed this way can be slightly different from the value LAMMPS reports in the line starting with `Performance:` due to rounding error.

SPC/E water simulations

Index	Size	# atoms	# Frontier nodes	FOM (atom-steps/second/10^6)
0	Nano	50.33 M	1	27.79
1	Tiny	402.65 M	8	177.15
2	Small	3.22 B	64 (4^3)	1,152.40
3	Medium	25.77 B	512 (8^3)	6,538.57
4	Large	206.16 B	4096 (16^3)	63,552.65
5	Reference	466.12 B	9261 (21^3)	102,950.19

PETN ReaxFF simulations

Index	Size	# atoms	# Frontier nodes	FOM (atom-steps/sec/10^6)
0	Nano	21.65 M	1	23.91
1	Tiny	173.19 M	8 (2^3)	180.93
2	Small	1.39 B	64 (4^3)	1,439.09
3	Medium	11.08 B	512 (8^3)	11,481.62
4	Large	88.67 B	4096 (16^3)	68,534.34
5	Reference	200.49 B	9261 (21^3)	173,074.19

Reporting

Benchmark results should include projections of the FOM for a problem size greater than or equal to the "reference" system. The complete hardware configuration needed to achieve the estimated timings must also be provided. For example, if the target system includes more than one type of compute node, then report the number and type of nodes used to run the benchmark. For the electronic submission, include all the source and makefiles used to build on the target platform and input files and run scripts. Include all standard output files.