Artefact and reproduction descriptions for LAMMPS CPU benchmark simulations

MD Simulation software (LAMMPS):
Large-scale Atomic/Molecular Massively Parallel Simulator - 8 Feb 2023
git commit: commit 730e5d2e64106f3e5357fd739b44c7eec19c7d2a (origin/develop, origin/HEAD)
Installed packages*:* MANYBODY, MEAM, MOLECULE, RIGID

Makefile:
Default Makefile.mpi (which used the compiler and linker flags “-g -O3 -std=c++11”)

Compiler:
Intel Classic C++ 20.21.6 / Intel(R) C++ g++ 10.3.1 mode with OpenMP not enabled
C++ standard: C++11

Communication library:
MPI v3.1: MVAPICH2 Version : 2.3.7
MVAPICH2 Release date : Wed March 02 22:00:00 EST 2022

Operating system:
Linux "Red Hat Enterprise Linux 8.9 (Ootpa)" 4.18.0-513.18.1.2toss.t4.x86_64 x86_64

Hardware (Quartz cluster):
https://www.top500.org/system/178971/
CPU: 2 x Xeon E5-2695v4 18C 2.1GHz per node
Memory: 128GB per node
Network: Intel Omni-Path
SLURM scheduler

Run command for the timing benchmark:

```
# Assuming the variable ‘nodes’ holds the number of nodes to
# be used for this run, ‘elem’ holds the element to simulate
# (Cu,Ta, or W), ’nxy’ the number of unit cells in the X- and
# Y-dimensions, ‘nz’ the number unit cells in the Z-dimension,
# and ‘bc’ is 0, 1, or 2 for open (shrink-wrapped), periodic,
# and periodic in Z only, respectively.
srun -N$nnodes -n$(($nnodes*36))  lmp_mpi_quartz -in in.bench \
  -var element $elem \
  -var nx $nxy -var ny $nxy -var nz $nz -var pbc $bc \
  -log bench.log.$nxy.$nz.$elem.$bc \
  > bench.run.$nxy.$nz.$elem.$bc
```

Full reproduction procedure:

1. Unpack the attached file benchmarks.tgz.
2. Go into the directory benchmarks.
3. Edit the variable ‘lmp’ in the begging of make_max_cfgs.sh and scale_bench.sh to point to the LAMMPS executable to use.
4. Edit the srun commands in make_max_cfgs.sh and scale_bench.sh to be the correct job running/job submission commands for your machine.
5. Adjust the ‘nnodes’ loop in scale-bench.sh to run over the desired node counts.
6. Run or source the make_max_cfgs.sh script to produce initial conditions. Using 8 nodes (as listed on the ‘srun’ line in the script, this process will run three jobs which each should complete in less than two minutes. The resulting configuration files can be found in {Cu,Ta,W}/cfgs.
7. Run or source the scale_bench.sh script to perform the benchmark simulations. It run one jobs per element per node count. On 8 nodes such a job takes up to 7 minutes. The runtime is inversely proportional to the node-count up to around 50 nodes.
8. The runtime for the simulations can be extracted by the command:

   ```
   grep Loop nbench.log.*
   ```