LAMMPS KOKKOS Installation Best Practices

HPC ADVISORY COUNCIL
NETWORK OF EXPERTISE

BEST PRACTICES

1. Introduction:

The following best practices document is provided as courtesy of the HPC Advisory Council.

2. Application Description:

LAMMPS is a classical molecular dynamics code, and an acronym for Large-scale Atomic/Molecular Massively Parallel Simulator. LAMMPS has potentials for solid-state materials (metals, semiconductors) and soft matter (biomolecules, polymers) and coarse-grained or mesoscopic systems.
It can be used to model atoms or, more generically, as a parallel particle simulator at the atomic, meso, or continuum scale. LAMMPS runs on single processors or in parallel using message-passing techniques and a spatial-decomposition of the simulation domain. LAMMPS is distributed as an open source code under the terms of the GPL. More information on LAMMPS can be found at the LAMMPS web site: http://LAMMPS.sandia.gov.

3. Version Information:

This guideline is based on the stable version of LAMMPS dated 15May15 (LAMMPS-15May15.tar.gz). The source code can be downloaded from this URL: http://LAMMPS.sandia.gov/tars/LAMMPS-15May15.tar.gz.

4. Prerequisites:

4.1 Hardware:

The instructions from this best practice have been tested on the HPC Advisory Council, Dell™ PowerEdge™ R730 32-node cluster.

- Dual Socket Intel® Xeon® 14-core CPUs E5-2697 V3 @ 2.60 GHz
- Mellanox ConnectX-4 EDR 100Gb/s InfiniBand adapters
- Mellanox Switch-IB SB7700 36-Port 100Gb/s EDR InfiniBand switches

4.2 Software:

a. OS: Red Hat Enterprise Linux 6.5

b. GNU Compiler for Linux; 4.8.2

c. Other:

- CUDA-6.5 or 7.0
- Cuda-aware MPI - MVAPICH2, OpenMPI, or Platfor MPI

5. Installation using HPC-X

```
module load gnu/4.8.2
module use /opt/hpcx-v1.3.336-gcc-MLNX_OFED_LINUX-3.0-1.0.1-redhat6.5-x86_64/modulefiles
module load hpcx
```

```
CVER=6.5
export PATH=/usr/local/cuda-$CVER/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda-$CVER/lib64:$I_MPI_ROOT/lib64:$LD_LIBRARY_PATH
```

5.1 Building GPU library

```
cd LAMMPS-15May15/lib/gpu
```

Set "CUDA_HOME = /usr/local/cuda-6.5" in Makefile.linux and update "-arch=sm_21" according to your GPU.

```
make -f Makefile.linux
```

It should generate libgpu.a and Makefile.LAMMPS.

If /usr/local/cuda is not a sym link to cuda-6.5, either fix the sym link or make a following change in Makefile.LAMMPS
/usr/local/cuda ==> /usr/local/cuda-6.5

5.2 Building CUDA library

```
cd LAMMPS-15May15/lib/cuda
```

If /usr/local/cuda is not a sym link to cuda-6.5, make a following change in Makefile.common.
/usr/local/cuda ==> /usr/local/cuda-6.5

```
make
```

It should generate libLAMMPScuda.a and Makefile.LAMMPS.

5.3 Building kokkos-cuda and kokkos-omp

```
cd LAMMPS-15May15/src
```

Update GPU arch in MAKE/OPTIONS/Makefile.kokkos_cuda if necessary.
Change LINK = nvcc to LINK = mpicxx if make fails on "-fopenmp".

a. kokkos-omp

```
make clean-all
make yes-kokkos
make yes-USER-CG-CMM
make kokkos_omp
```

b. kokkos-cuda

```
make clean-all
make yes-gpu
make yes-kokkos
```

```
    make yes-USER-CUDA
    make yes-USER-CG-CMM
    make kokkos_cuda
```

6. Running LAMMPS with CPU or with GPU

KOKKOS_CUDA with GPU

```
mpirun -np <NPROC> lmp_kokkos_cuda -i <input> -sf kk
-k on t <# threads> g <# gpus> -v kokkos <1..7>
```

KOKKOS_OMP with CPU

```
mpirun -np <NPROC> lmp_kokkos_omp -i <input> -sf kk
-k on t <# threads> -v kokkos <1..7>
```

7. Sample input file

```
---
# 3d Lennard-Jones melt

variable        kokkos index 0

if “${kokkos} == 1” then &
  “package      kokkos neigh half/thread comm/forward
device comm/exchange host” &
  “newton off”
if “${kokkos} == 2” then &
  “package      kokkos neigh full comm/forward device
comm/exchange device” &
  “newton off”
if “${kokkos} == 3” then &
  “package      kokkos neigh full comm/forward host
comm/exchange host” &
  “newton off”
if “${kokkos} == 4” then &
  “package      kokkos neigh full comm/forward host
comm/exchange no” &
  “newton off”
if “${kokkos} == 5” then &
  “package      kokkos neigh half/thread comm/forward
no comm/exchange no” &
  “newton off”
if “${kokkos} == 6” then &
  “package      kokkos neigh half/thread comm/forward
no comm/exchange no” &
  “newton on”
if “${kokkos} == 7” then &
  “package      kokkos neigh half/thread comm/forward
device comm/exchange device” &
  “newton on”

units           lj
atom_style      atomic

lattice         fcc 0.8442
region          box block 0 64 0 64 0 64
create_box      1 box
create_atoms    1 box
mass            1 1.0

velocity        all create 1.44 87287 loop geom

pair_style      lj/cut 2.5
pair_coeff      1 1 1.0 1.0 2.5

neighbor        0.3 bin
neigh_modify    delay 0 every 20 check no

fix             1 all nve
thermo          100
run             20000
---
```