Molecular Dynamics tutorial lab

Eduardo M. Bringa (ebringa@yahoo.com), Emmanuel Millán

Computational cost of MD simulations is usually measured by how much time (wall clock) does it take a given simulation, divided the number of atoms and the steps run in that simulation (TPAS). For simple pair potent ials, TPAS is in the range ~5-50 μs/atom/step. For complex potentials, or long range potentials it can be 1-2 orders of magnitude longer. You need to make sure you run enough steps to get meaningful timing, i.e. at least 1,000 steps.

0) Warm-up

a) Copy everything from "/home/ebringa/WHPC" to your local dir. There are several executables in the bin dir, including versions for packages GPU y CUDA, and directories with suitable inputs and submission scripts.

The lammps manual and detailed descriptions of commands can be found in the web:

http://lammps.sandia.gov/doc/Manual.html

http://lammps.sandia.gov/doc/Section_commands.html#comm

http://lammps.sandia.gov/doc/package.html (about gpu, cuda and omp packages)

b) Please bring output from code from *mendieta* to your local machine for plotting/analysis.

1) Lennard-Jones (LJ) simulations (without MPI)

Every MD tutorial should start with Lennard-Jones (LJ) simulations ☺

$$V(r) = 4\varepsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right]$$

You should first try to run in a single CPU and in a single GPU (will need a CPU assigned, even for a single task).

All of the "Melt" scripts create a LJ material, and then increase the temperature above the melting point.

There sample submission script has to change according to which input and run mode you want to use.

a) Run the input scripts, modifying the submission script as needed and saving the logs. Use a single core/GPU.

i) Compare the timing. Note the force loop timing, neighbor timing, etc.

ii) What is the TPAS for each case? Why?

b) Now increment the number of steps from 1,000 to 10,000. Does the "cost" of the simulation changes linearly with the number of steps? Why?

c) Now increment the number of atoms in the simulation by a factor of 2^3 (box size changes a factor of 2 along each direction) going back to 1,000 steps. Does the "cost" of the simulation changes? Why? What happens if you simulate an elongated box, where only one of the dimensions is extended 2^3?

d) Using your favorite plotting program -for instance, gnuplot (short tutorial at http://people.virginia.edu/~lz2n/mse627/Eduardo/)-, plot the total, kinetic and potential energies from the different logs. Are they different? Are they statistically different (calculate mean and standard deviation using for instance OpenOffice)? For fancy plotting of atomic configurations, you can use OVITO (http://www.ovito.org/) and/or VMD (http://www.ks.uiuc.edu/Research/vmd/), which are already installed.

e) Using the gpu version of the input, run changing neighbor list parameters in GPU and in CPU.

What happens if one increments the number of atoms in the simulation by a factor of 2^3? What happens if your neighbor "skin" is doubled?

f) Test how much the timing is affected by the disk I/O, un-commenting the "dump" commands in the script. Note that most benchmarks showing TPAS or equivalent do not include disk I/O, which will be needed in production runs. In this case you would be writing to a single file, but LAMMPS is capable to writing to one file/process to achieve better parallel efficiency. This starts mattering seriously for runs with +100 cores.

2) Parallelization: Lennard-Jones (LJ) simulations with MPI

This is crucial! Now you have the chance to test not only what happens with 2 GPUs from the same node, but 2 GPUs from different nodes, which will force MPI communication between the GPUs. This is usually a problematic bottleneck for GPU computing.

Run the input scripts, modifying the submission script as needed and saving the logs. Use:

a) Using the gpu version of the input, run using "split" option to split atoms between CPU and GPU. Are the results what you expected? Why?

b) Run in two GPUs.

c) Run in two cores.

d) Two cores and two GPUs

f) Four cores and two GPUs.

d) Do a table with the timing for the different cases. For a "strong" scaling case like this one, how good is the parallelization scaling?

e) Do a "weak" scaling test, increasing the system size by a factor of 2 and running case (b). Is the scaling good?

3) ADITIONAL MATERIAL

a) There are several examples within dir "examples" of the lammps distribution.

b) You can test weak/strong scaling in different examples, by changing number of atoms and cores. For instance, the LJ melt case runs up to several million atoms in one of the Tesla C2090s in *mendieta*.