Introduction to Molecular Dynamics on ARCHER: Exploring parallel performance of NAMD

1 Introduction

The purposes of this practical are for you to:

- gain experience running parallel jobs on ARCHER.
- understand how NAMD scales for a particular benchmark system.

You may wish to refer to the "Instructions for running jobs on ARCHER" handout for details of how to connect and construct batch jobs scripts.

2 Choosing a benchmark system

For the purposes of the exercise, you can work with any system you like, for example:

- The solvated HEWL protein system you set up in the previous practical sessions.
- Any other system of your choice.
- The ApoA1 benchmark system for which benchmark results are provided on the NAMD website.

The ApoA1 benchmark consists of a protein (27,850 atoms), solvated in explicit water for a total of 92,224 atoms. The force-field has a 12 Åcut-off for pair interactions, and uses PME for long-range electrostatics. The PME calculation is done every 4 timesteps, and a total of 500 1fs timesteps will be carried out in an NVE (microcanonical) ensemble. A tar-ball containing the files can be obtained from `http://www.ks.uiuc.edu/Research/namd/utilities/apoa1.tar.gz`. To download and extract the files on ARCHER do:

```
ibethune@eslogin001:~> mkdir /work/z01/z01/ibethune/namd-tests

ibethune@eslogin001:~> cd /work/z01/z01/ibethune/namd-tests

ibethune@eslogin001:/work/z01/z01/ibethune/namd-tests> wget
  http://www.ks.uiuc.edu/Research/namd/utilities/apoa1.tar.gz

--2015-06-25 11:08:01--  http://www.ks.uiuc.edu/Research/namd/utilities/apoa1.tar.gz
Resolving www.ks.uiuc.edu... 130.126.120.35
Connecting to www.ks.uiuc.edu|130.126.120.35|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2890839 (2.8M) [application/x-tar]
Saving to: `apoa1.tar.gz'
```

```
100%[======================================>] 2,890,839   1.05M/s   in 2.6s

2015-06-25 11:08:04 (1.05 MB/s) - ‘apoa1.tar.gz’ saved [2890839/2890839]

ibethune@eslogin001:/work/z01/z01/ibethune/namd-tests> tar -xzvf apoa1.tar.gz

apoa1/
apoa1/apoa1.pdb
apoa1/apoa1.psf
apoa1/par_all22_prot_lipid.xplor
apoa1/par_all22_popc.xplor
apoa1/apoa1.namd
```

3 Running jobs

To run the benchmark, start with a batch script given in the "Instructions for running parallel jobs on ARCHER". You will need to modify the `aprun` line as follows:

```
aprun -n 48 -N 24 namd2 apoa1.namd
```

This will run NAMD using 48 MPI processes, 24 per node. To run on more than one node you will also need to modify the `select` option:

```
#PBS -l select=2
```

When NAMD runs it will generate an output file containing information about the setup of the calculation, including how many MPI processes are being used, details relating to the simulation parameters and force-field, and then a summary of the computed energies, temperature, pressure etc. at each timestep. You will also see intermittent messages to indicate that NAMD is performing dynamic load balancing. Periodically, NAMD prints timing information, for example:

```
TIMING: 480  CPU: 22.6909, 0.0448748/step  Wall: 22.6909, 0.0448748/step,
 0.000249304 hours remaining, 672.472656 MB of memory in use
```

You should observe that after a few rounds of dynamic load balancing, the time per step will decrease and converge to some value.

Conveniently, NAMD prints out a `Benchmark time`, which is designed to provide an accurate estimate of the performance for long runs, ignoring any data before the initial load balancing steps. In the apoa1 benchmark this is printed after 400 and 500 steps. Finally NAMD prints the total time it has used:

```
WallClock: 24.523418  CPUTime: 24.523418  Memory: 838.140625 MB
End of program
```

Warning: the total `Wallclock` time includes the time taken for initialisation (reading the input files, and also performancing some initial calculations to find the most efficient FFT strategy for the PME calculation). The first time this is done, NAMD will write a file `FFTW_NAMD_CVS-2013-11-13_CRAY-XC-MPI.txt` which it can re-read in later runs to save time. Thus the wallclock time is not a good indicator of performance, the time per step is a better metric.

3.1 Measuring Performance

Run your benchmark for a range of different processor counts, and record the `Benchmark time`. For each processor count, calculate the speedup and the parallel efficiency using the expressions given in the lecture slides. You may also calculate other performance metrics, like the number of ns of simulation time per day of wallclock time.

Plot your data against the number of processors (you may need to use a logarithmic scale). How many processors can you use before the performance starts to drop off? If you were running extended MD calculations with this system, how many MPI processors would you choose to use?

Some additional suggestions for tuning performance are provided at the NAMD website: `http://www.ks.uiuc.edu/Research/namd/wiki/?NamdPerformanceTuning` - try experimenting with some of these and see how they affect the performance of your benchmark.