Assignment 1- Molecular Dynamics
CSCI 596: Scientific Computing & Visualization

Anup V Kanale
September 8, 2017

1 Scaling using Linked-List MD

In this section, we shall compare the effect of scaling on the linked-list cell implementation program `lmd.c`, and the regular molecular dynamics program `md.c`. The two programs were run for different number of unit cells `InitUcell` (from {4,4,4} to {10,10,10}). The corresponding number of atoms, `nAtom` is calculated as nAtoms = 4atoms/cell $*$ (InitUcell)^{3}cells, and it varies from 256 to 4000.

The time of execution T was measured for each case and fit as per the formula $T = C \times (\text{nAtoms})^p$. Upon taking the logarithm, we see that $\log T = p \log(\text{nAtoms}) + K$. Therefore, the slope of the line in the log plot gives us the order of scaling. The results are summarized in the plot below.

As we can see from the equations of fit lines, the regular md program scales almost quadratically, whereas the linked-list cell implementation scales linearly with the number of atoms. This agrees with the theoretical prediction.

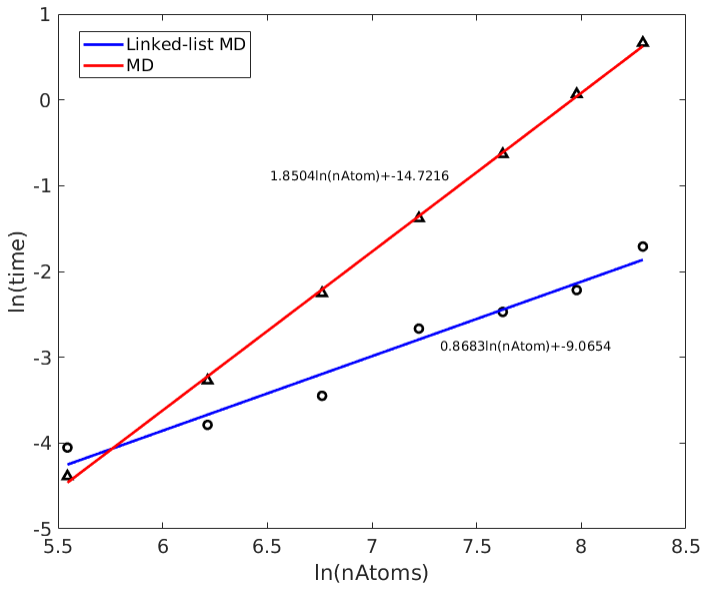

Figure 1: Scaling of regular and linked-list MD programs

2 Flop/s Performance

Performance of a program is often measured in MFlop/s (megaflop/s = million floating point operations per second). To get a "*feel*" of this performance measure, the linked-list md program `lmd.c` was run for `InitUcell` = {10,10,10} (or `nAtom` = 4000). The number of floating point operations (+, −, ∗, /) executed were counted and this result was divided by the value obtained for elapsed time (in seconds) to obtain the program's Mflop/s performance. (for simplicity, sqrt() counts as 1 operation).

This test was performed on a laptop with Intel core i5 processor with a clock speed of 2.3GHz. The results obtained were as follows:

- **Execution time**: $1.787950E-01$ s
- **Number of floating-point operations**: $1.506044E+08$
- **Mflop/s performance**: $8.423302E+02$

A Programs

A python script was written to automate the running of the programs for different values of `InitUcell`. The script is as shown below.

```
#!/usr/bin/python

import os
import subprocess
import matplotlib.pyplot as plt
import numpy as np

UCellSize = np.arange(4,11)
nAtoms = 4*np.power(UCellSize, 3)
fnames = ["md", "lmd"]
for j in range(2):
        fname = fnames[j]
        infile = fname + '.in'
        cfile = fname + '.c'
        exfile = fname
        fp = open(infile, 'r+')
        strOld = fp.read()
        fp.close()

        time = []
        for i in range(4,11):
                fp  = open(infile, 'w+')
                newSize = str(i) + ' '
                strNew =  3*newSize + '\n'+ strOld[9:]
                fp.write(strNew)
```

```
                fp.close()

                # Run MD program
                subprocess.call(['gcc', '-O', '-o', exfile, cfile, '-lm'])
                comm = './' + exfile + '<' + infile + '>out.txt'
                os.system(comm)
                time.append(float(open('out.txt','r').read()))
        print(time)
```

The matlab script used to generate these plots is shown below.

```
% Script to generate plot and fit to a curve
clear
close all;

nAtoms = log(4* (4:10).^3);
timelmd = log([0.017377, 0.022644, 0.031774, 0.06952,...
    0.084479, 0.108944, 0.180908]);

plmd = polyfit(nAtoms, timelmd, 1);
timeFit = polyval(plmd, nAtoms);

set(figure, 'position', [1000 300 800 600], 'color', 'w')
h1 = plot(nAtoms, timelmd, 'ko', 'linewidth', 2);
hold on;
h2 = plot(nAtoms, timeFit, 'b-', 'linewidth', 2);

% MD plot
timemd = log([0.012428, 0.037953, 0.105062, 0.251538,...
    0.530956, 1.068831, 1.9434]);
pmd = polyfit(nAtoms, timemd, 1);
timeFit = polyval(pmd, nAtoms);

hold on;
h3 = plot(nAtoms, timemd, 'k^', 'linewidth', 2);
hold on;
h4 = plot(nAtoms, timeFit, 'r-', 'linewidth', 2);

txt = strcat(num2str(plmd(1)), 'ln(nAtom)+', num2str(plmd(2)));
text(nAtoms(4)+0.1, timelmd(4)-0.2, txt);
txt = strcat(num2str(pmd(1)), 'ln(nAtom)+', num2str(pmd(2)));
text(nAtoms(2)+0.3, timemd(4)+0.5, txt);

xlabel('ln(nAtoms)', 'fontsize', 14)
ylabel('ln(time)', 'fontsize', 14)
legn = legend([h2,h4], 'Linked-list MD', 'MD', 'Location', 'best');

set(gca, 'fontsize', 14)
```