CSCI 596: HW 1

M. Oneeb H. Khan

September 6, 2021

Question 1

1.1 Plot

A python file using the matplotlib library was created to plot the log-log plot of values of the number of atoms, N, and the corresponding running time, T, provided in the MDtime.out data file. The values in the data file were collected from the simulation results of the *md.c* program.

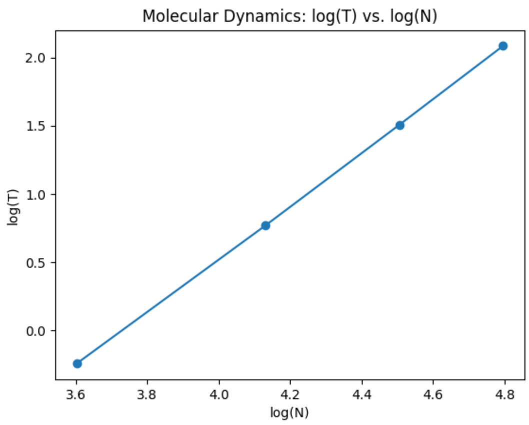


Figure 1: Plot of log(T) vs. log(N)

1.2 Fitted value of α

Using the equation:

$$log\,T = \alpha log N + \beta$$

and the first and last points of the plot in figure 1, we were able to determine the values of both α and β .

$$\boldsymbol{\alpha = 1.951} \quad \beta = -7.277$$

1.3 Least Square Fit of a Line

To determine the best linear fit for $y = ax + b$ such that the square error is minimized, we determine the values of a and b using the following system of equations:

$$\begin{cases} a = \dfrac{N\sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{\sum_{i=1}^{N} x_i^2 N - \left(\sum_{i=1}^{N} x_i\right)^2} \\[2ex] b = \dfrac{-\sum_{i=1}^{N} x_i \sum_{i=1}^{N} x_i y_i + \sum_{i=1}^{N} x_i^2 \sum_{i=1}^{N} y_i}{\sum_{i=1}^{N} x_i^2 N - \left(\sum_{i=1}^{N} x_i\right)^2} \end{cases}$$

Figure 2: Equations to calculate values of a and b for least square fit

The same equation was implemented in python using the numpy library as follows:

```
a = ((4 * np.dot(logN, logT)) - (np.sum(logN) * np.sum(logT))) / \
    ( np.sum(np.multiply(np.square(logN), NUM_ELEMENTS)) - np.square(np.sum(logN)) )

b = ( -1*(np.sum(logN)*np.dot(logN, logT)) + (np.sum(np.square(logN))*np.sum(logT)) ) / \
    ( np.sum(np.multiply(np.square(logN), NUM_ELEMENTS)) - np.square(np.sum(logN)) )
```

Figure 3: Numpy implementation of system of equations for least square fit

The resulting values were:

$$\mathbf{a = 1.951,\ b = -7.277}$$

Question 2

Given a computer with 1 octa-core processor operating at a clock speed of 2.3 GHz, with each core having 1 FMA circuit with vector registers holding 4 double-precision operands,the theoretical peak performance in terms of $flop/s$ is determined as follows:

$$\text{Peak flop/s} = \overset{1}{\#\text{ multi-core processors}} \times \overset{2}{\#\text{ cores}} \times \overset{3}{\text{Peak flop/s per core}}$$

$$3\ \text{Peak flop/s per core} = \overset{4}{\text{clock speed}} \times \overset{5}{\#\text{ operations executable per clock cycle}}$$

$$5\ \#\text{ops executable per clock cycle} = (2 \times \#\text{FMA units}) \underset{3}{\times} \#\text{double precision operands}$$

$$\therefore \text{Peak flop/s} = \overset{1}{1} \times \overset{2}{8} \times \left[\overset{4}{2.3 \times 10^9} \times \overset{5}{\left((2 \times 1) \times 4\right)}\right]$$

$$\boxed{\text{Peak flop/s} = 147.2\ \text{G flop/s}}$$

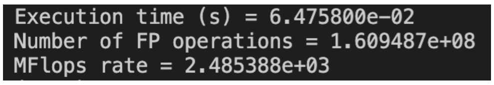
Figure 4: Equation to calculate theoretical peak performance

Therefore,

$$\textbf{Theoretical Peak flop/s} = \mathbf{147.2 \;\; Gflop/s}$$

Question 3 (Optional)

After compiling and running the *lmd_sqrt_flop.c* file, the following output is observed:



```
Execution time (s) = 6.475800e-02
Number of FP operations = 1.609487e+08
MFlops rate = 2.485388e+03
```

Figure 5: Resulting output of *lmd_sqrt_flop.c*

The resulting flop/s performance is: $2.485388 \;\; Gflop/s$

As a percentage of theoretical peak performance, the value is:

$$\%\ of\ theoretical\ peak\ performance = \frac{2.485388\ Gflop/s}{147.2\ Gflop/s} \times 100$$

$$\%\ \textbf{of theoretical peak performance} = \mathbf{1.69}\%$$