Homework 1
COSC 594

Jiahui Guo

Jan. 22, 2014

1 Program Implementation

1.1 Test Data

The vector and matrix are initialized with random number ranging from 0 to 100.

1.2 Result verification

To verify the correctness of the program, I use CBLAS library to compute an accurate result, and then calculate the error between the accurate result and computed result, and the error is measured in 2-norm. The criterion to state a result is accurate needs to meet

$$\left\|(accurate - computed)\right\| \leq 10^{-16}$$

1.3 Time Measurement

C timer is used to measure the execution time of each executable. Data generation, memory allocation and deallocation, and initialization are not included in the execution time, so as to flops.

1.4 Organization of the program

The program is organized in the following way:

src /

- **twoNorm.c** 2-norm of vector
- **vecMul.c** Vector-matrix multiplication
- **matMul.c** Matrix-matrix multiplication
- **c_timer.c**
- **Makefile**

include /

- **c_timer.h**
- **cblas.h**

bin /

- **run.sh** Run the executable files and generates plots
- **avg.py** Average the time and flops obtained from several trials
- **plotTime.gnu**
- **plotFlops.gnu**

- **plotAvgTime.gnu**
- **plotAvgFlops.gnu**

pdf /

- **hw1.tex**
- **Makefile**

Makefile

ReadMe Illustrating how to execute the program.

2 Experiment Platform

The experiment was conducted on a hydra machine. The configuration of the machine is list below:

Processor Intel Core i7-3770 @ 3.40GHz

Memory 16.0GB DDR3

Hard drive 500GB 7200 RPM SATA @ 6Gbps, 16MB cache

3 Performance Analysis

3.1 Operation Analysis

1. 2-norm of vector
   The operations for 2-norm of a size N vector are $2N$, there is also one multiplication and one addition for each data.

2. Vector-matrix multiplication
   The operations for vector-matrix multiplication is $2N^2$, there is also $\theta(N)$ multiplication and $\theta(N)$ addition for N element in the result vector.

3. Matrix-matrix multiplication
   The Operations for matrix-matrix multiplication is $2N^3$, there is also $\theta(N)$ multiplication and $\theta(N)$ addition for N element in the result matrix.

3.2 Theoratical Peak Performance

For the machine used in the experiment, the peak flop per second for one core is calculated as:

$$4\, flops/cycle \, * \, 3.40\, G\, cycles \, = \, 13.6\, Gflops$$

3.3 Experiment Results

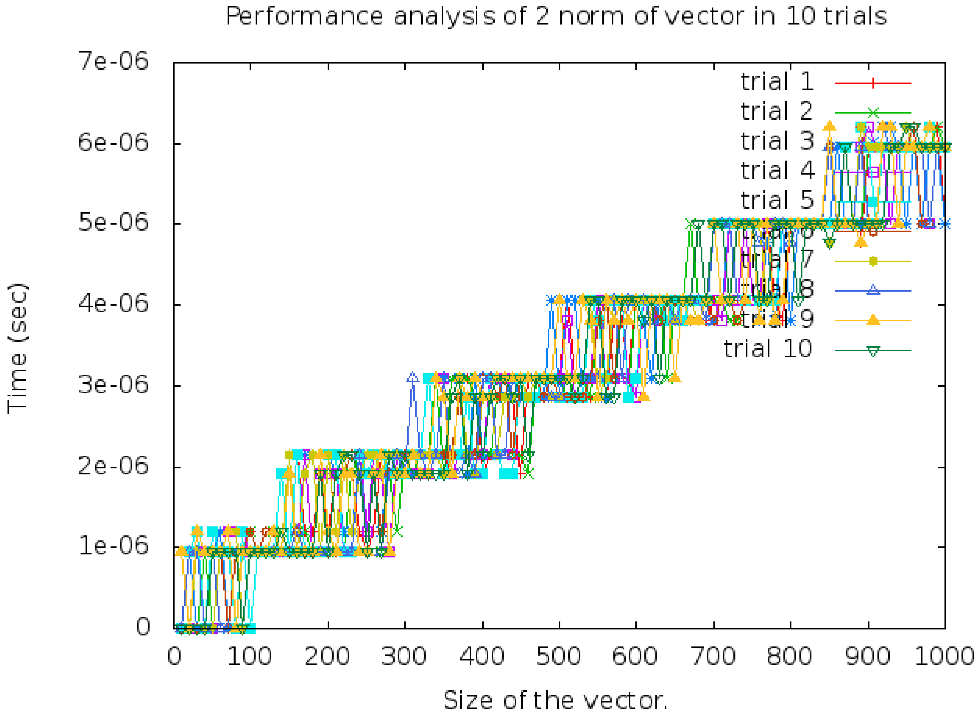

Figure 1: Execution time of 2-norm of vector in 10 trials

1. 2-norm of vector In Fig.1 and Fig.2, the execution time and flops of 2-norm of vector computation is shown. The peak Gops/sec is around 0.6 GFlops. Execution time keeps increasing with larger input data size.

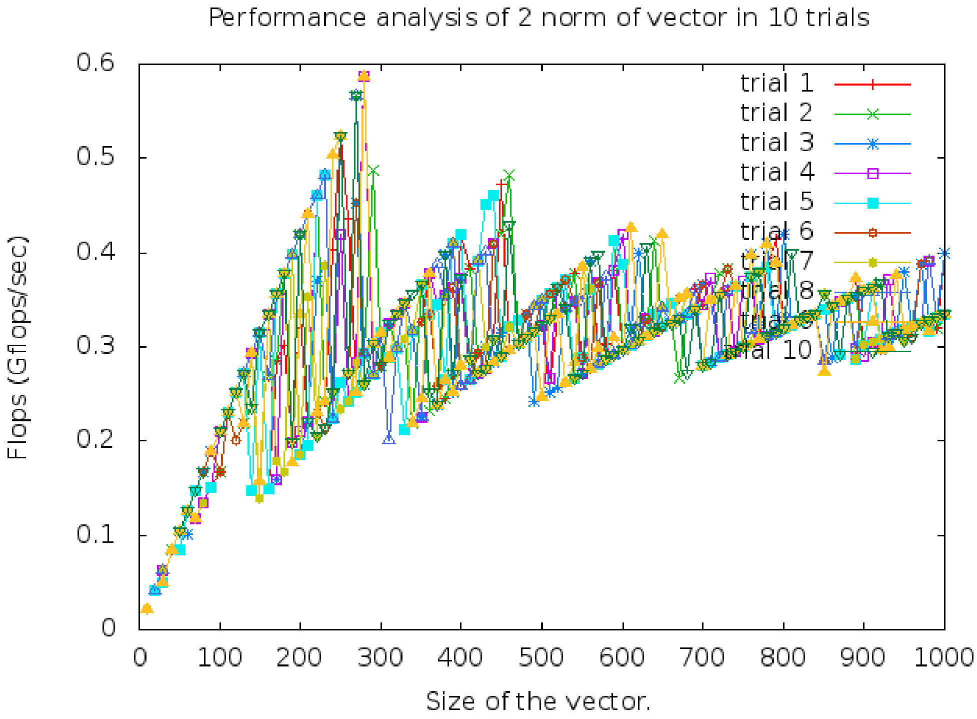

Figure 2: Flops of 2-norm of vector in 10 trials

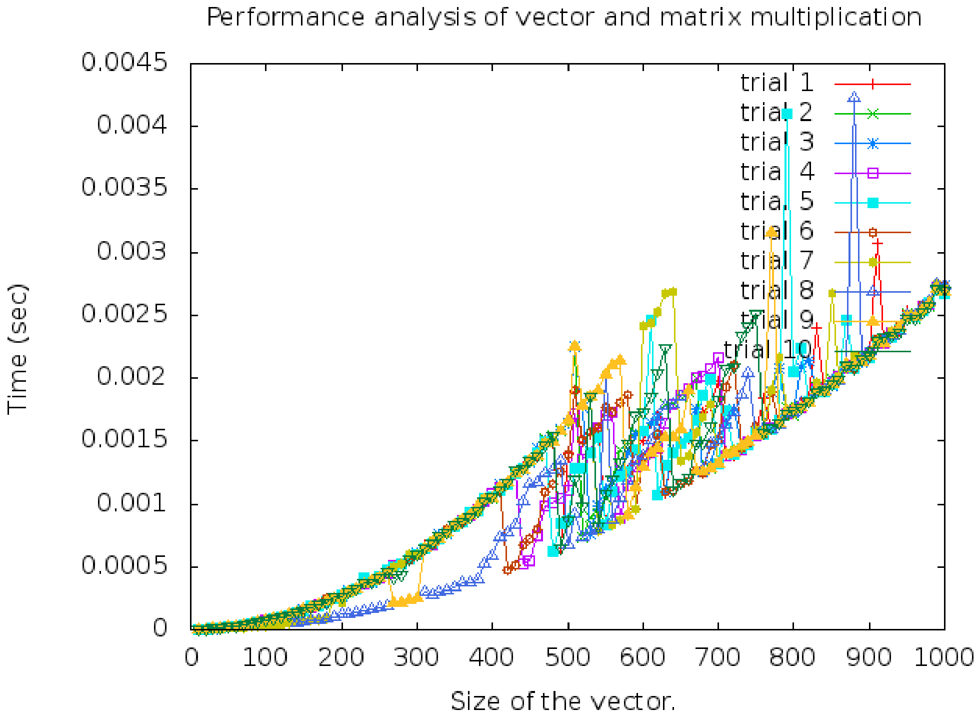

Figure 3: Execution time of vector-matrix multiplication in 10 trials

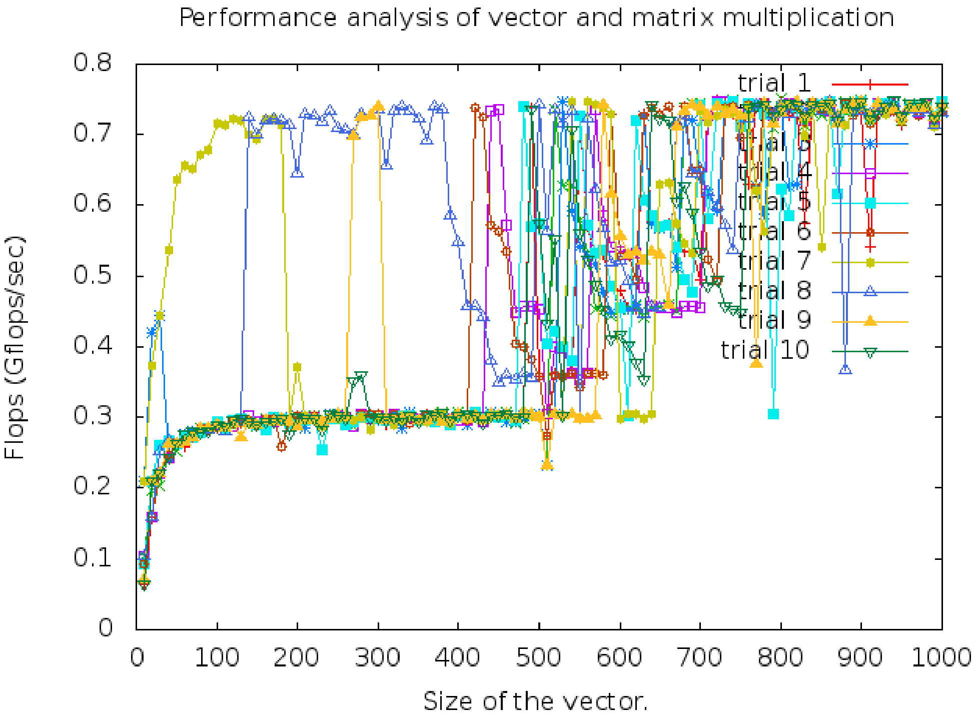

Figure 4: Flops of vector-matrix multiplication in 10 trials

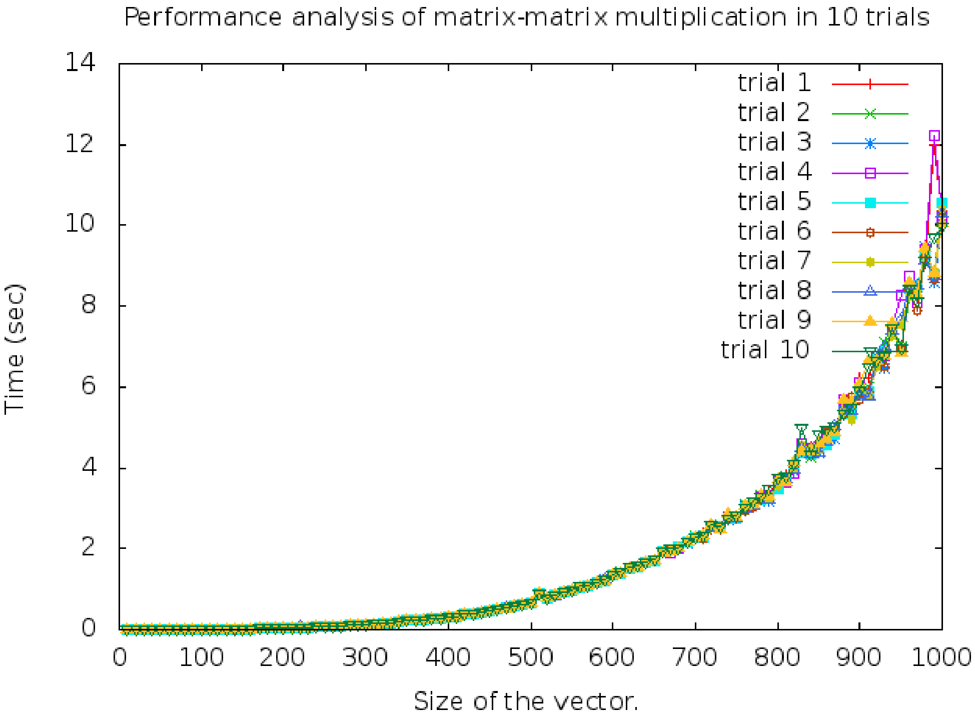

Figure 5: Execution time of matrix-matrix multiplication in 10 trials

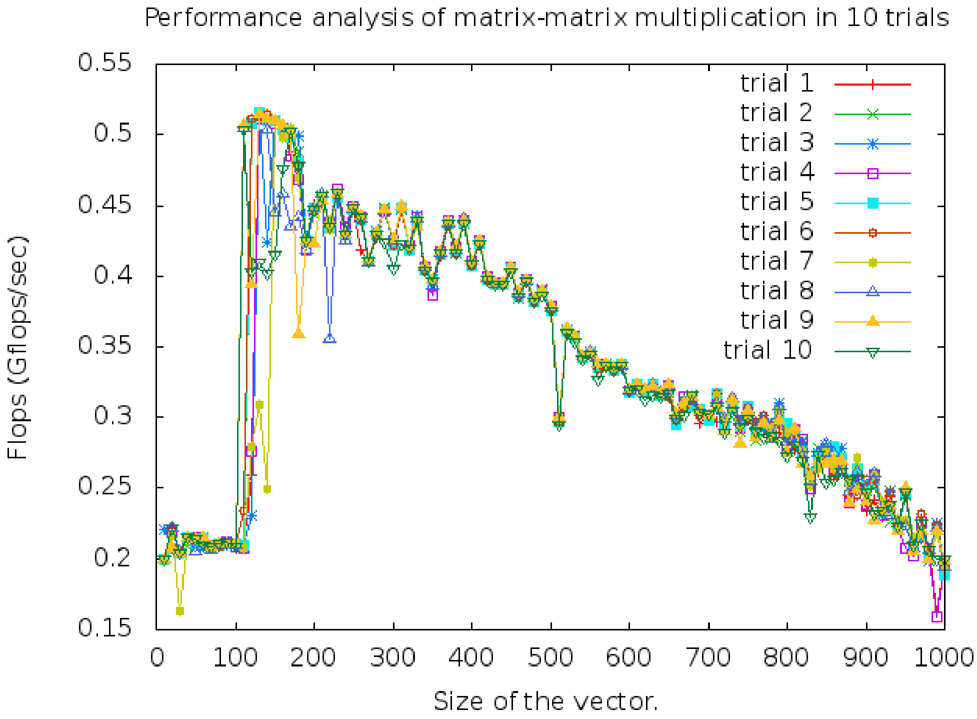

Figure 6: Flops of matrix-matrix multiplication in 10 trials

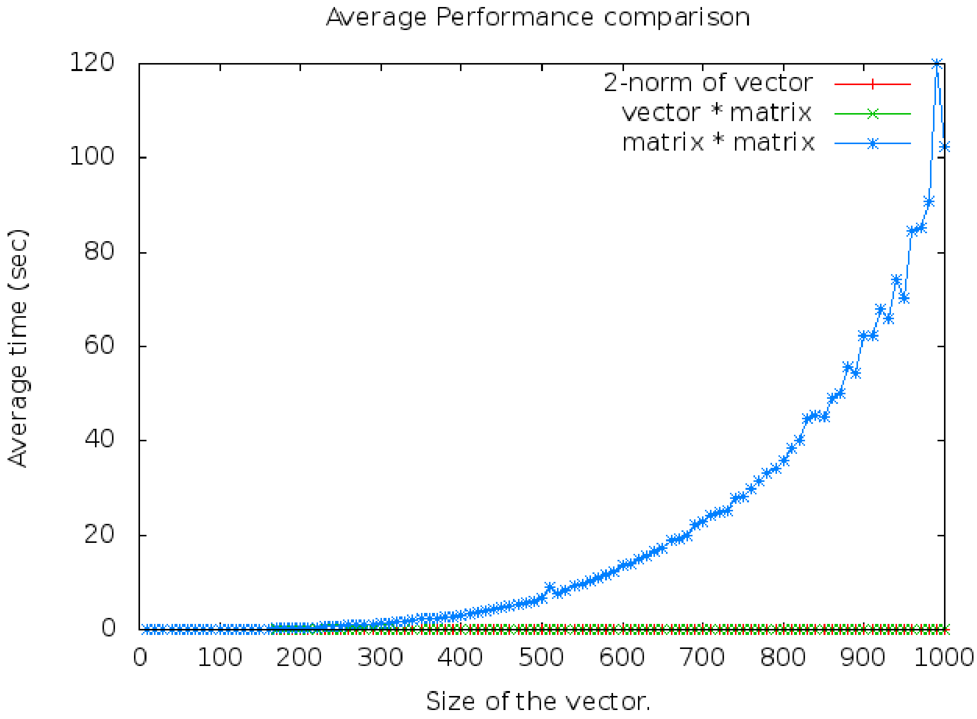

Figure 7: Comparison of execution time

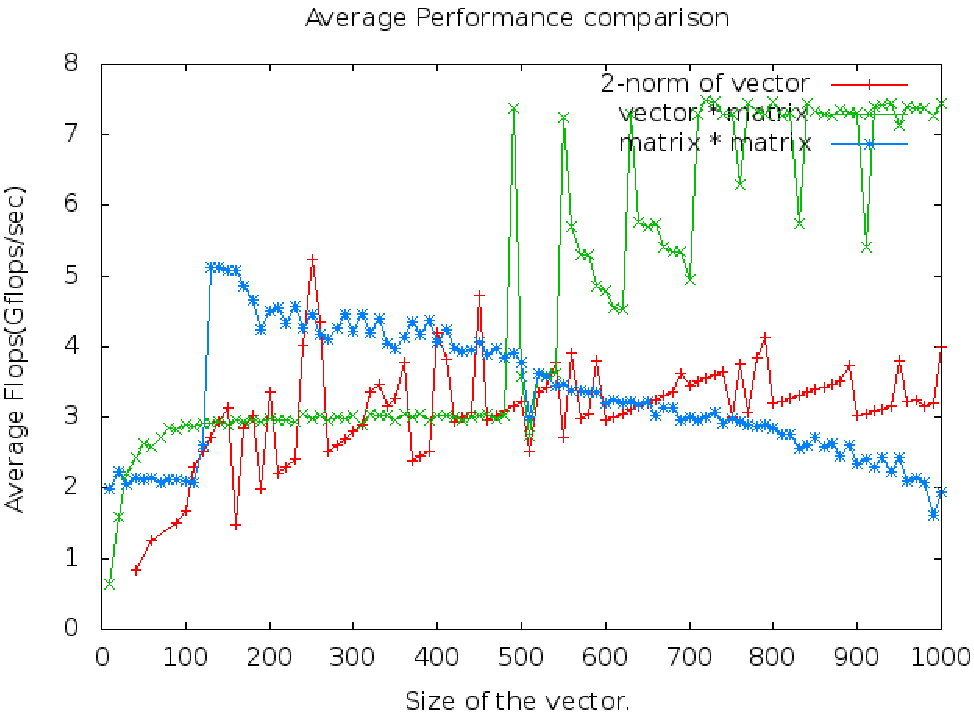

Figure 8: Comparison of flops

2. Vector-matrix multiplication In Fig.3 and Fig.4, the execution time and flops of vector-matrix multiplication computation is shown. The peak Gops/sec is around 0.75 GFlops. Execution time keeps increasing with larger input data size exponentially.

3. Matrix-matrix multiplication In Fig.5 and Fig.6, the execution time and flops of matrix-matrix multiplication computation is shown. The peak Gops/sec is around 0. 5 GFlops. Execution time keeps increasing with larger input data size exponentially.

4. Average execution time and flops comparison In Fig.7 and Fig.8, the execution time and flops of these three programs are compared. It can be seen that with the increasing of input data size, matrix-matrix multiplication will consuming much more time than the other two, and the execution time is incresing exponentially. Also, one thing needs to be addressed is that the flops for matrix-matrix multiplication would get worse with the increasing of input data size because of the memory issue.

3.4 Discuss

1. Why the achieved performance is different than the peak?
   (a) Program implementation is not efficient enough;
   (b) System resources are occupied by other programs at the same time;
2. How to improve the performance?
   (a) Increasing core frequency level;
   (b) Improving instruction level;
   (c) Optimizing the program;
   (d) Multithreading or parallel;
   (e) Larger cache.