High Performance Computing

2023 Fall

Lab 2. C Programming in Linux

General Matrix Multiplication

Yongzhuo Ma

September 17, 2023

Chapter 1

Introduction to the Environment

1.1 Host Machine

Item	Value
OS version	macOS Ventura 13.5.2
Apple clang version	14.0.3
CPU	Apple M2 Max
CPU Frequency	3.54 - 3.70 GHz
CPU Cores	12
Memory	64 GB

1.2 Virtual Machine

Item	Value
Virtualization	Parallels
OS version	Ubuntu 22.04.3 LTS
gcc version	11.4.0
CPU	Apple M2 Max
CPU Frequency	3.66 GHz
CPU Cores	8
Memory	32 GB

Chapter 2

Matrix Multiplication

In this chapter, I will compare the efficiency between two algorithms of matrix multiplication. They are **general matrix multiplication : naive** and **general matrix multiplication : OpenBlas**.

2.1 General Matrix Multiplication : Naive

It is easy to implement this algorithm, relevant code has been attached to **Appendix**.

Here the math is,

$$\boldsymbol{A} = \alpha \boldsymbol{L} \times \boldsymbol{R} + \beta \boldsymbol{C}$$

where the size of $\boldsymbol{A}, \boldsymbol{L}, \boldsymbol{R}, \boldsymbol{C}$ is $M \times K$,$M \times N$,$N \times K$ and $M \times K$.

Time is recorded in microsecond(μs).All numbers in the matrices are generated randomly. Here,

$$\alpha = 3, \beta = 2$$

Virtual Machine

M	N	K	Duration(μs)	GFlops
4	4	4	1	0.128000
16	16	16	12	0.682667
64	64	64	802	0.653726
256	256	256	44750	0.749820
1024	1024	1024	2383867	0.900840
2048	2048	2048	19046422	0.902000
4096	4096	4096	153913963	0.892960
8192	8192	8192	1235454909	0.889965
16384	16384	16384	9883639272*	———
32768	32768	32768	———	———
65536	65536	65536	———	———

For M = N = K = 16384, the duration is estimated.
For M = N = K = 32768, it raised a memory error.
For M = N = K = 65536, it raised a memory error.

2.2 General Matrix Multiplication : OpenBlas

Relevant code has been attached to **Appendix**.

Nothing changed,

$$\boldsymbol{A} = \alpha \boldsymbol{L} \times \boldsymbol{R} + \beta \boldsymbol{C}$$

where the size of $\boldsymbol{A}, \boldsymbol{L}, \boldsymbol{R}, \boldsymbol{C}$ is $M \times K$,$M \times N$,$N \times K$ and $M \times K$.

Time is recorded in microsecond(μs).All numbers in the matrices are generated randomly. Here,

$$\alpha = 3, \beta = 2$$

M	N	K	Duration(μs)	GFlops
4	4	4	466	0.000275
16	16	16	84	0.097524
64	64	64	177	2.962079
256	256	256	673	49.857997
1024	1024	1024	15604	137.623920
2048	2048	2048	99692	172.329467
4096	4096	4096	507859	270.624235
8192	8192	8192	3417775	321.703924
16384	16384	16384	24669406	356.558768
32768	32768	32768	191583411	367.300821
65536	65536	65536	———	———

For M = N = K = 65536, it raised a memory error.

2.3 Comparison

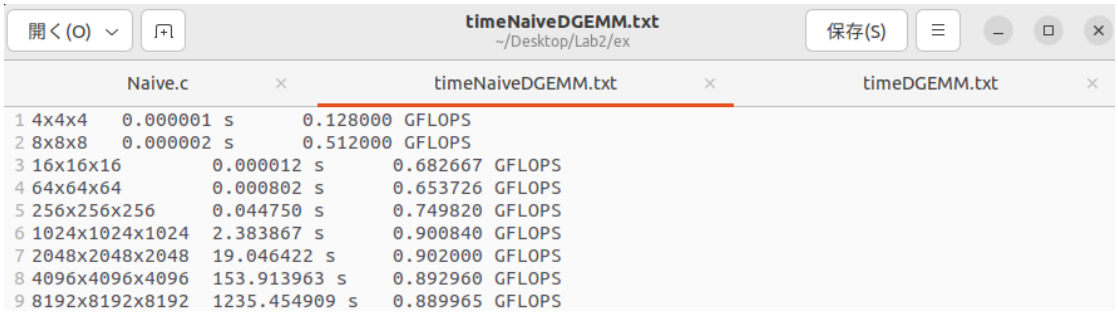

Figure 2.1: timeNaiveDGEMM.txt

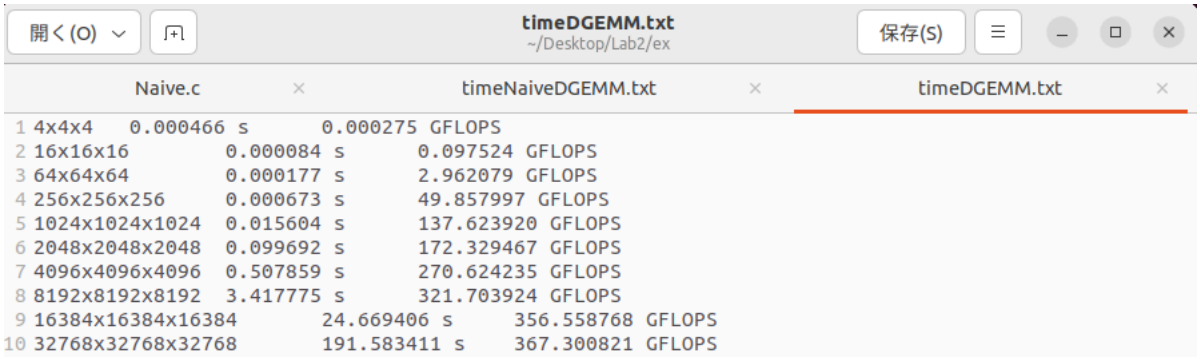

Figure 2.2: timeDGEMM.txt

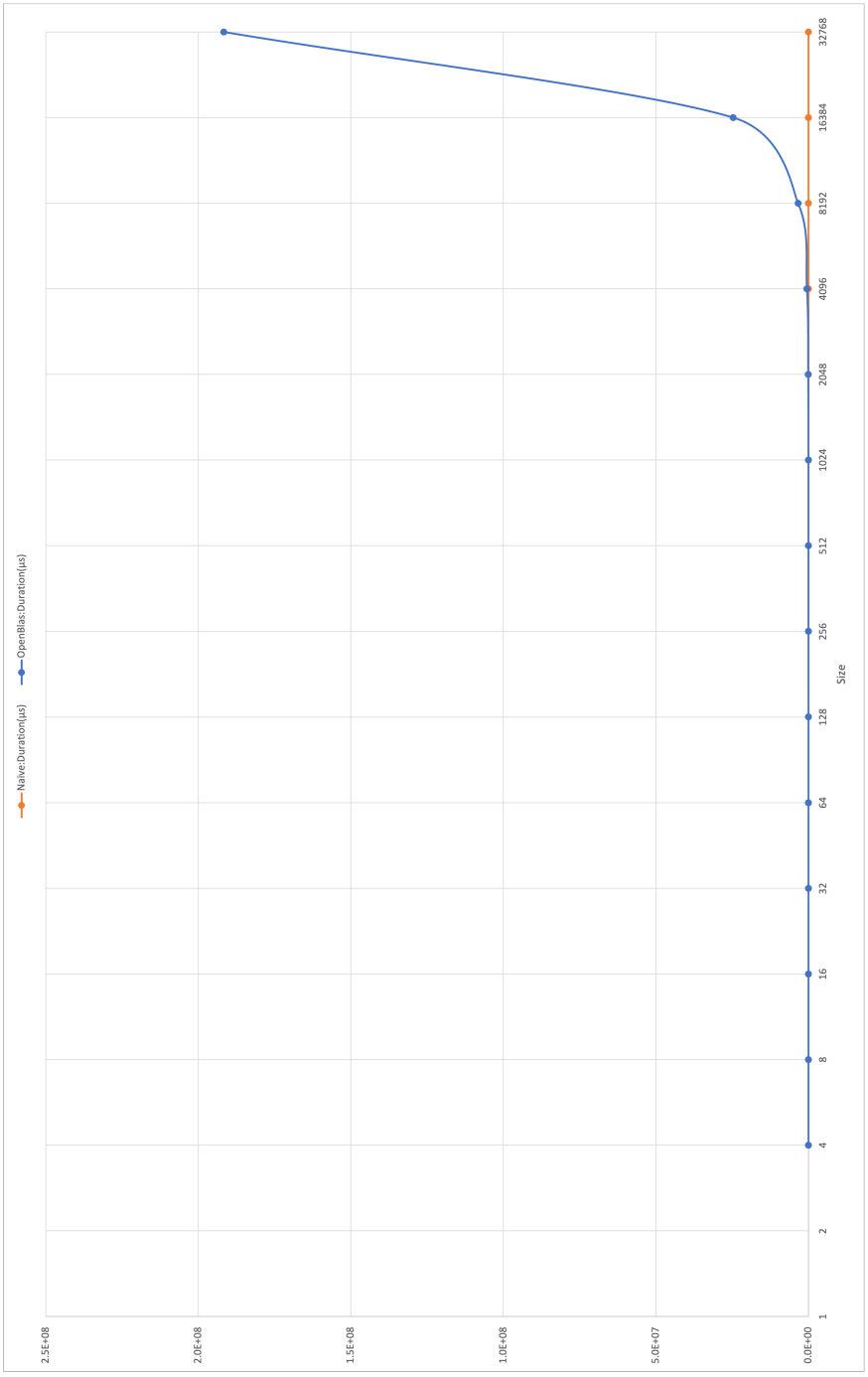

Figure 2.3: Duration Differences

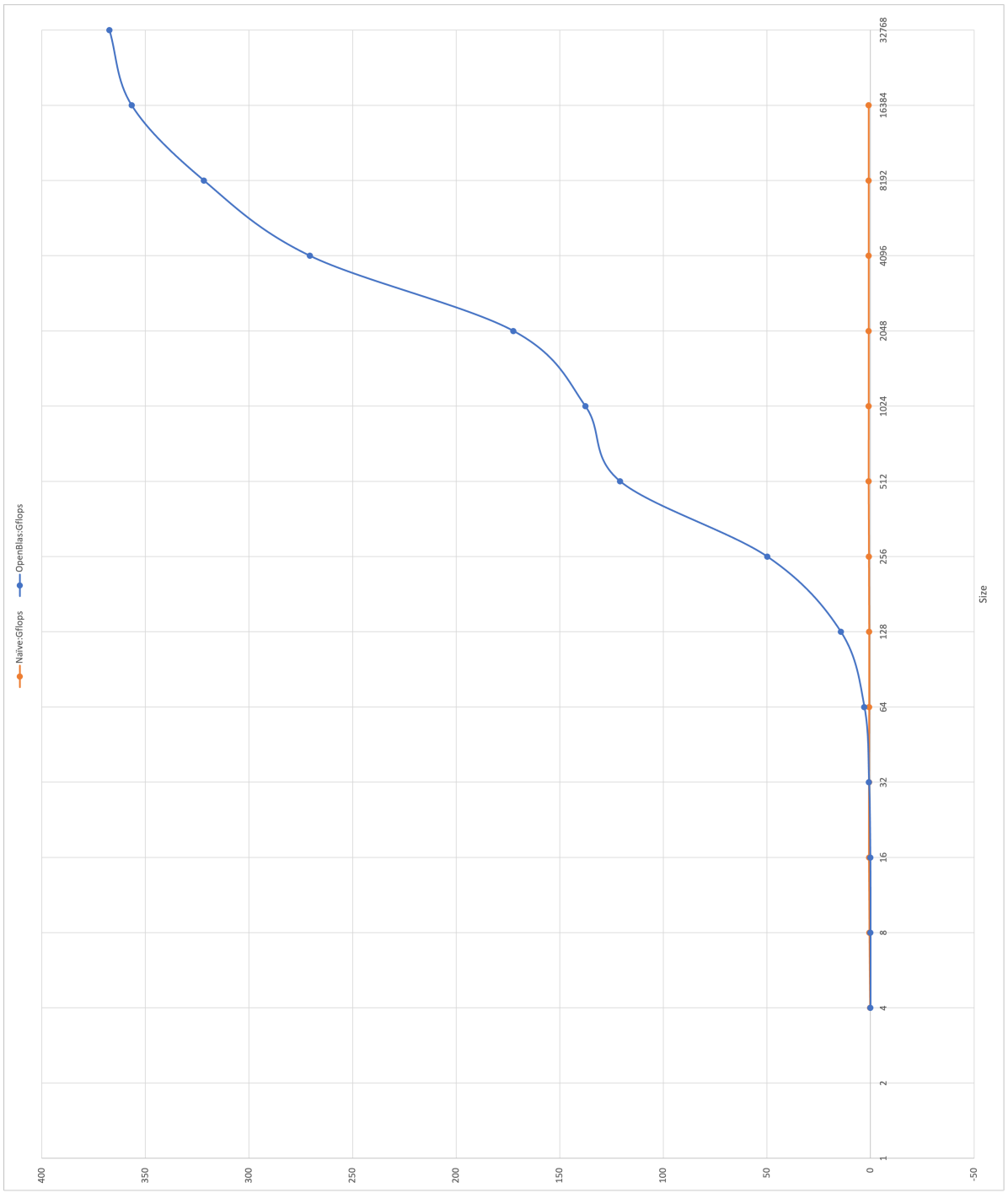

Figure 2.4: GFlops Differences

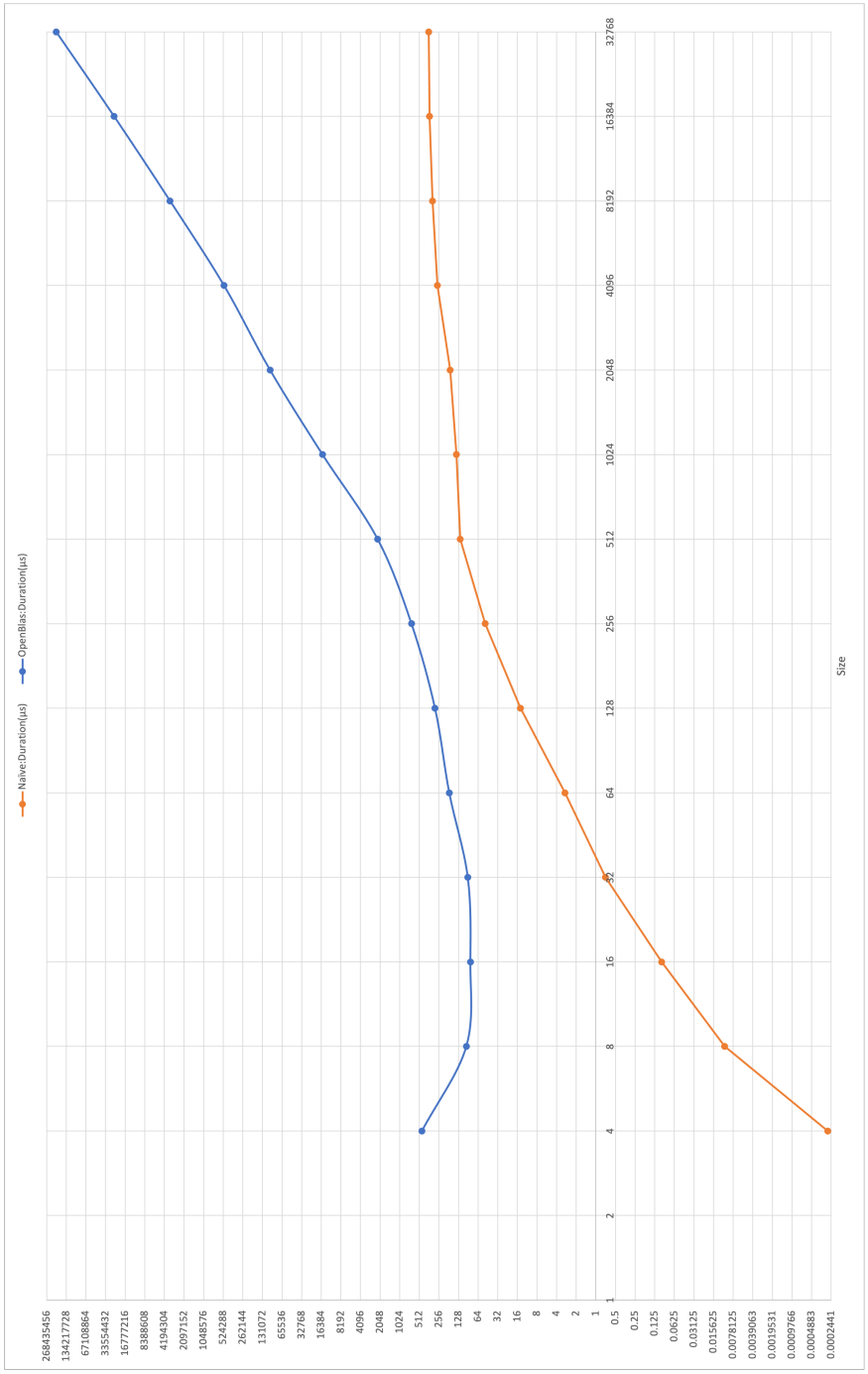

Figure 2.5: Duration Differences (base 2)

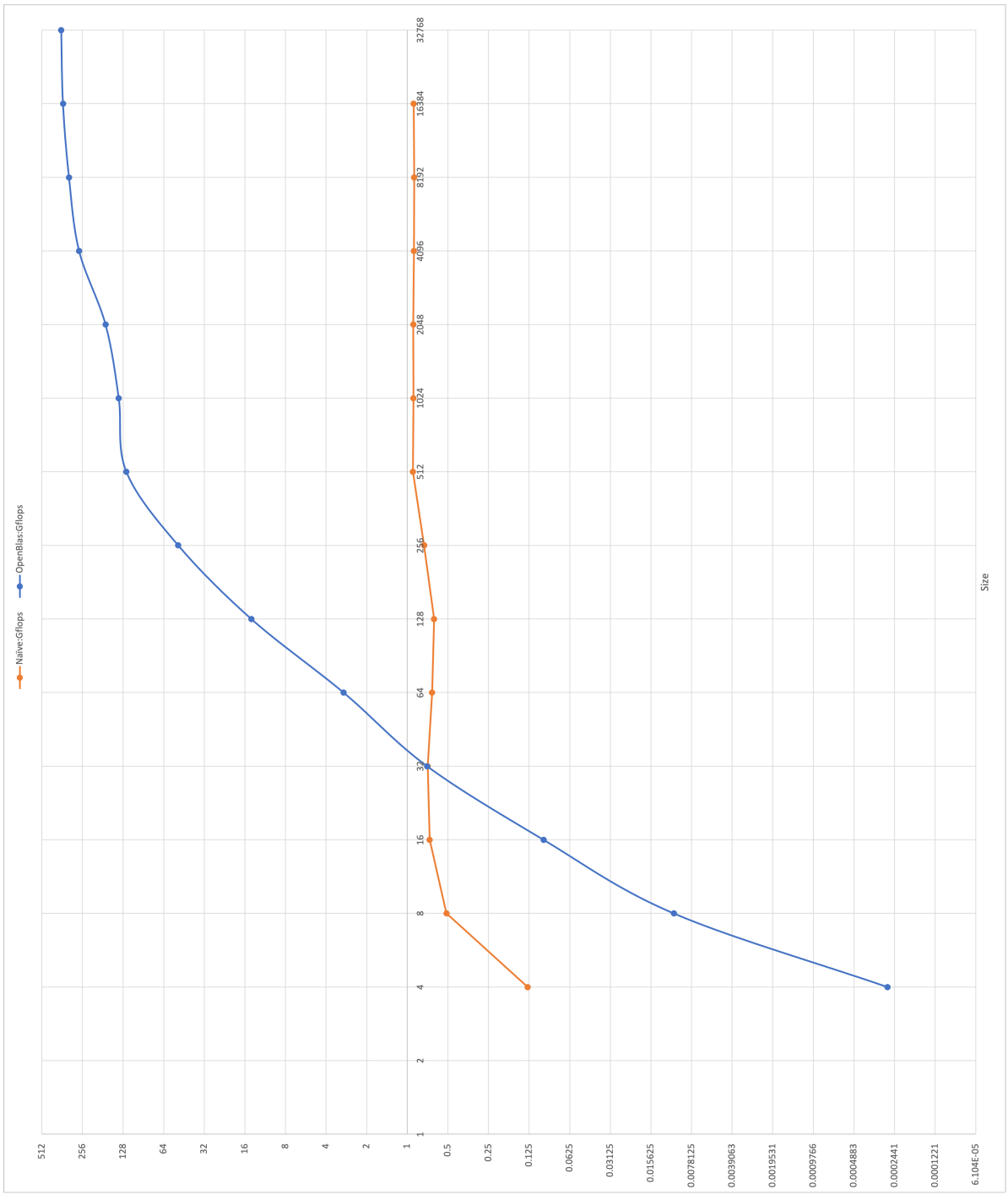

Figure 2.6: GFlops Differences (base 2)

Appendix

Two copies of source code attached.