

CMOR 421/521, Homework #1: L^AT_EX Submission

amc50

February 29, 2024

1 Compilation

1.1 Accessing NOTS Cluster

It is important to note that the following process is done on Rice Owls Network. If this is not the case, this would be unsuccessful.

Command used to ssh into NOTS:

```
MacBook-Pro-95:cmor-421-521-submissions antoniocrivello$ ssh amc50@nots.rice.edu
```

Command used to activate interactive node on NOTS:

```
[amc50@login1 ~]$ srun --pty --partition=interactive --ntasks=1 --mem=1G --time=00:30:00 $SHELL
```

Command used to load modules needed for compilation:

```
[amc50@bc9u7n1 ~]$ module load GCC/13.1.0
```

Accessing files on local desktop by using GitHub Repository:

```
[amc50@bc9u7n1 ~]$ ssh-keygen -t ed25519 -C "amc50@rice.edu"
```

```
\begin{comment}
```

Generating public/private ed25519 key pair.

Enter file in which to save the key (/home/amc50/.ssh/id_ed25519):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/amc50/.ssh/id_ed25519.

Your public key has been saved in /home/amc50/.ssh/id_ed25519.pub.

The key fingerprint is:

```
SHA256:1Xu4gV32DaSy0mP1qHGao7fMWMgxI6UWwx7PYAILwSY amc50@rice.edu
```

The key's randomart image is:

```
+--[ED25519 256]--+
```

```
| .o..      . |
|E o. o      . o |
| o  . o * o + + |
|      X B * B o. |
|      = S X B o o |
|      . o B B +   |
|      o * .       |
```

```
|          *..      |
|          o.+      |
+-----[SHA256]-----+
\end{comment}
```

Command used to access public key:
[amc50@bc9u7n1 ~]\$ emacs ~/.ssh/id_ed25519.pub

The key was then added to list of SSH Keys on GitHub.

Command used to clone cmor-421-521-submission GitHub repository:
[amc50@bc9u7n1 ~]\$ git clone git@github.com:AntonioCrivello/cmor-421-521-submissions.git

```
\begin{comment}
Cloning into 'cmor-421-521-submissions'...
Warning: Permanently added the ECDSA host key for IP address '140.82.114.3' to the list of known h
\end{comment}
```

For compilation on the NOTS Cluster I am utilizing a Makefile.
[amc50@bc9u7n1 homework-1]\$ make clean
rm -f matmul_recursive ./obj/*.o *~ *.o

```
[amc50@bc9u7n1 homework-1]$ make
g++ -c src/matrix.cpp -o obj/matrix.o -I./include -O3 -std=c++11
g++ obj/matrix.o main.cpp -I./include -O3 -std=c++11 -o matmul_recursive
```

```
[amc50@bc9u7n1 homework-1]$ ./matmul_recursive 512
```

Describe generate-timings.sh

2 Matrix-Matrix Multiplication

a

3 Optimizing Matrix-Matrix Multiplication

Table 1: Blocked Matrix-Matrix Multiplication on NOTS

	Number of Iterations	NOTS Timing
Matrix 4 x 4		
Matrix 8 x 8		
Matrix 16 x 16		
Matrix 32 x 32		
Matrix 64 x 64		
Matrix 128 x 128		
Matrix 256 x 256		
Matrix 512 x 512		
Matirx 1024 x 1024		

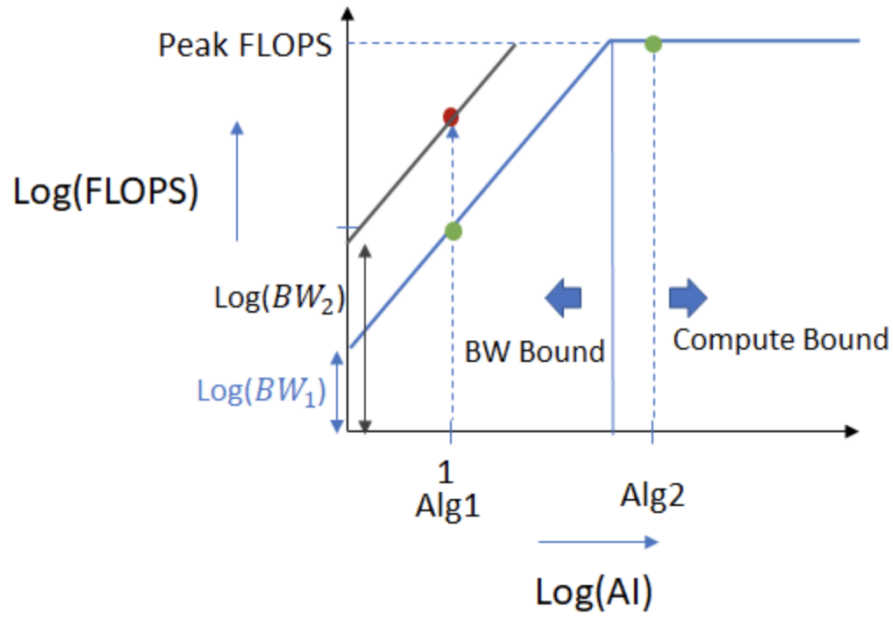


Figure 1: Roofline Plot for Naive Matrix-Matrix Multiplication

3.1 Timing

3.2 Results

4 Recursive Matrix-Matrix Multiplication

4.1 Results

4.2 Discussion

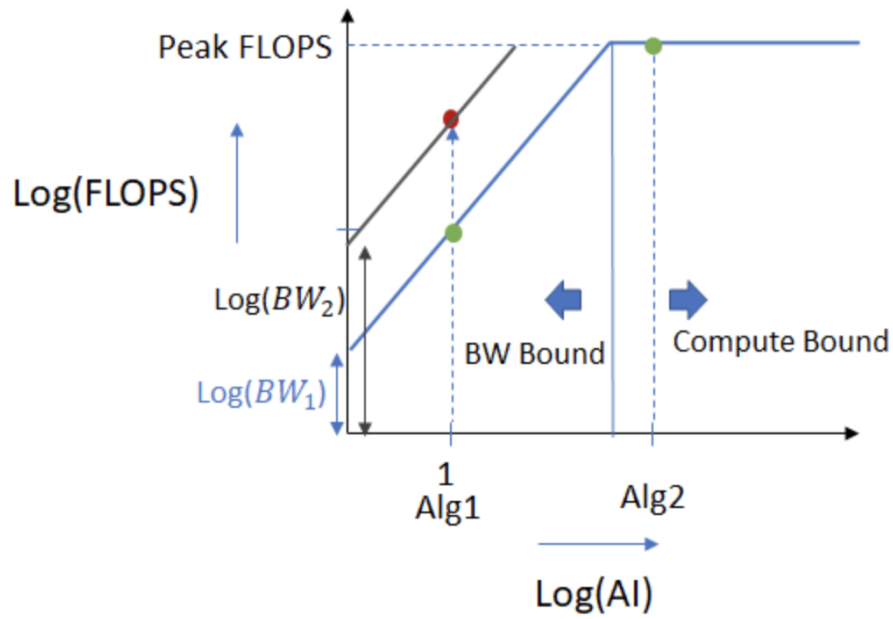


Figure 2: Roofline Plot for Blocked Matrix-Matrix Multiplication

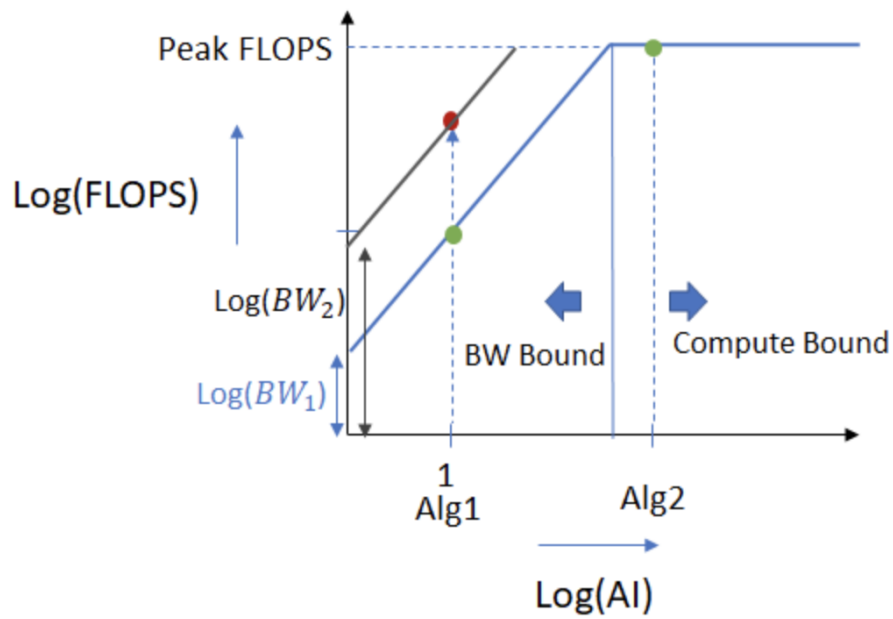


Figure 3: Roofline Plot for Recursive Matrix-Matrix Multiplication