CMOR 421/521, Homework #1: LaTeX 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@nlogin1 ~]$
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
```

To access files created on local desktop, a GitHub Repository is used. As such,
this repository must be cloned on the cluster. This process requires a password-protected key.

Command used to generate SSH key :

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

Command used to access public key:

```
[amc50@bc9u7n1 ~]$ emacs ~/.ssh/id_ed25519.pub
```

After accessing the SSH key, it is added to the GitHub 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
```

For compilation on the NOTS Cluster I am utilizing a Makefile.

Set of commands to compile project:

```
[amc50@bc9u7n1 homework-1]$ make clean
```

```
Output of make clean:
rm -f matmul_recursive ./obj/*.o *~ *.o

[amc50@bc9u7n1 homework-1]$ make
Output of 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
```

In order to efficiently generate timings for matrix sizes of 2^i for i = 4,5,6 ... 10, I have created a bash script to run the aforementioned make clean and make commands with the matrix size as an argument. Additionally, at the initial phase of the project to determine the optimal block size for each matrix-matrix multiplication the block size is also supplied as an argument to the executable.

Command used to generate timings and compile project:

```
[amc50@bc9u7n1 homework-1]$ ./generate-timings.sh
```

2 Matrix-Matrix Multiplication

2.1 Column Major Storage for Matrix-Matrix Multiplication

If the matrix was stored in column major format instead of row major format for matrix-matrix multiplication a few changes should be made. The first is to change how the memory is accessed during the impementation. With row major storage the memory is efficiently access by looping through the rows and then the columns. For the column major formatting, this should be changed, looping through the column first and then the row. This will allow the required data values to be located adjacent to each other. There is also a benefit in unrolling the innermost loop to reduce overhead and utilize the location of the column values.

2.2 Matrix Transpose

For matrices stored in column major format it is expected that $A^T B$ would be faster than AB. The primary reason this is the case is because when applying the transpose to matrix A and then multiplying to matrix B, the matrix−matrix multiplication involve looping through the rows of

3 Optimizing Matrix-Matrix Multiplication

3.1 Timing

3.2 Discussion

Does this depend on the size of the matrix?

Table 1: Naive Matrix-Matrix Multiplication Timings on NOTS

Matrix Size	n = 4	n = 8	n = 16	n = 32	n = 64	n = 128	n = 256	n = 512
n = 16	2.9041e-05	3.3614e-05	3.2159e-05					
n = 32	0.000222842	0.00025741	0.000259517	0.00028529				
n = 64	0.00205113	0.00200312	0.00167115	0.00158465	0.00150467			
n = 128	0.0120923	0.0121557	0.0115599	0.0111493	0.0119782	0.014615		
n = 256	0.158795	0.131669	0.140042	0.137167	0.134298	0.153463	0.149298	
n = 512	1.27346	1.14611	1.0844	1.07697	1.23491	1.19583	1.15814	1.05354
n = 1024	11.4279	14.3977	12.306	11.8852	10.7746	12.3554	12.6266	14.937

Table 2: Blocked Matrix-Matrix Multiplication Timings on NOTS

Matrix Size	n = 4	n = 8	n = 16	n = 32	n = 64	n = 128	n = 256	n =
n = 16	2.3312e-05	2.0671e-05	1.8851e-05					
n = 32	0.000196252	0.000195223	0.000194868	0.000180096				
n = 64	0.00178605	0.00150001	0.0011403	0.00112655	0.00115903			
n = 128	0.0105116	0.00939145	0.00799496	0.00823802	0.00810977	0.0105357		
n = 256	0.102929	0.0823619	0.0791603	0.0769003	0.0808152	0.0836151	0.0974383	
n = 512	0.753265	0.621876	0.580653	0.555099	0.679861	0.751526	0.77674	0.77
n = 1024	6.20977	5.36031	4.86852	4.55242	6.04679	6.17169	6.34725	6.25

3.3 Roofline Plot Results

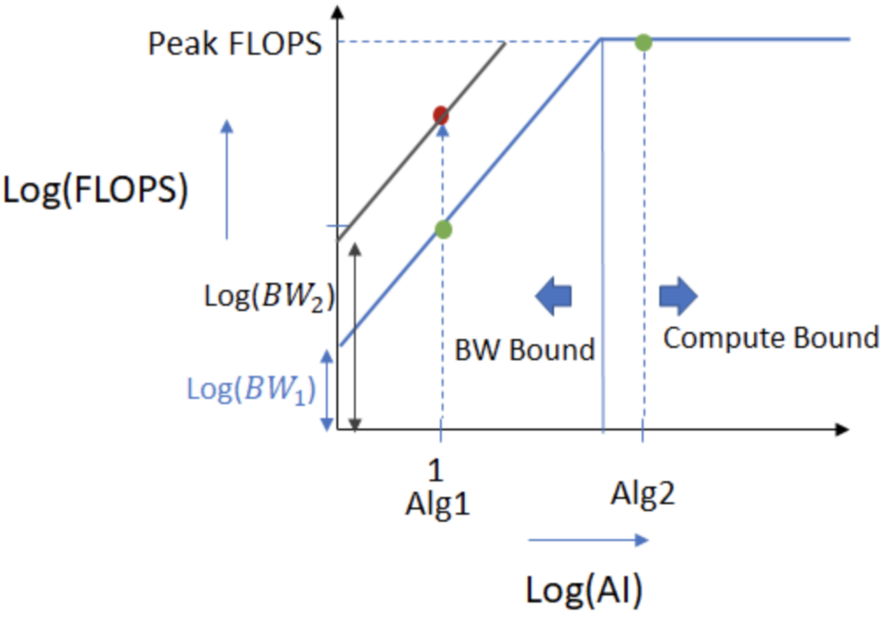

Figure 1: Roofline Plot for Naive Matrix-Matrix Multiplication

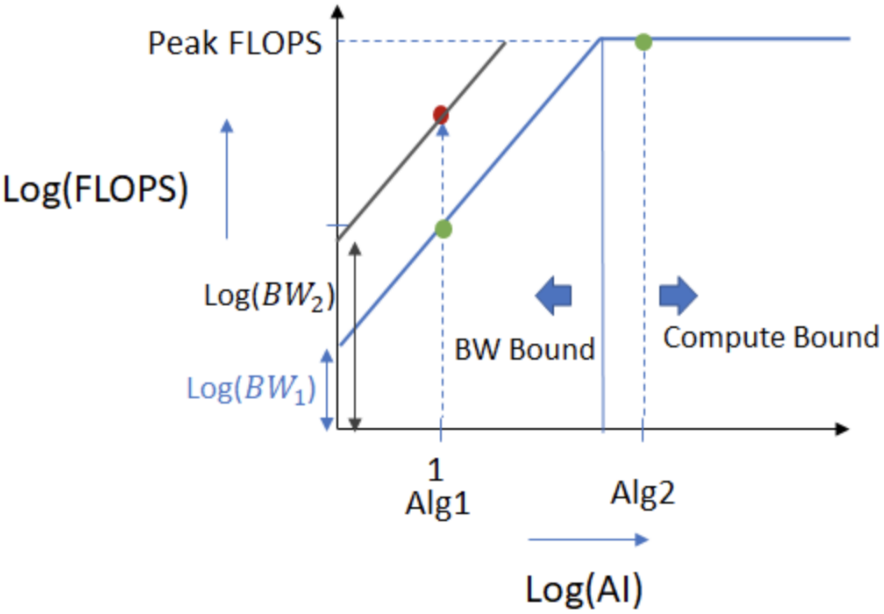

Figure 2: Roofline Plot for Blocked Matrix-Matrix Multiplication

4 Recursive Matrix-Matrix Multiplication

4.1 Timing

Table 3: Blocked Matrix-Matrix Multiplication Timings on NOTS

Matrix Size	n = 4	n = 8	n = 16	n = 32	n = 64	n = 128	n = 256	n =
n = 16	2.3312e-05	2.0671e-05	1.8851e-05					
n = 32	0.000196252	0.000195223	0.000194868	0.000180096				
n = 64	0.00178605	0.00150001	0.0011403	0.00112655	0.00115903			
n = 128	0.0105116	0.00939145	0.00799496	0.00823802	0.00810977	0.0105357		
n = 256	0.102929	0.0823619	0.0791603	0.0769003	0.0808152	0.0836151	0.0974383	
n = 512	0.753265	0.621876	0.580653	0.555099	0.679861	0.751526	0.77674	0.77
n = 1024	6.20977	5.36031	4.86852	4.55242	6.04679	6.17169	6.34725	6.25

4.2 Implementation Analysis

`Determine Optimal Block Size`

`How to check correct implementation`

4.3 Results

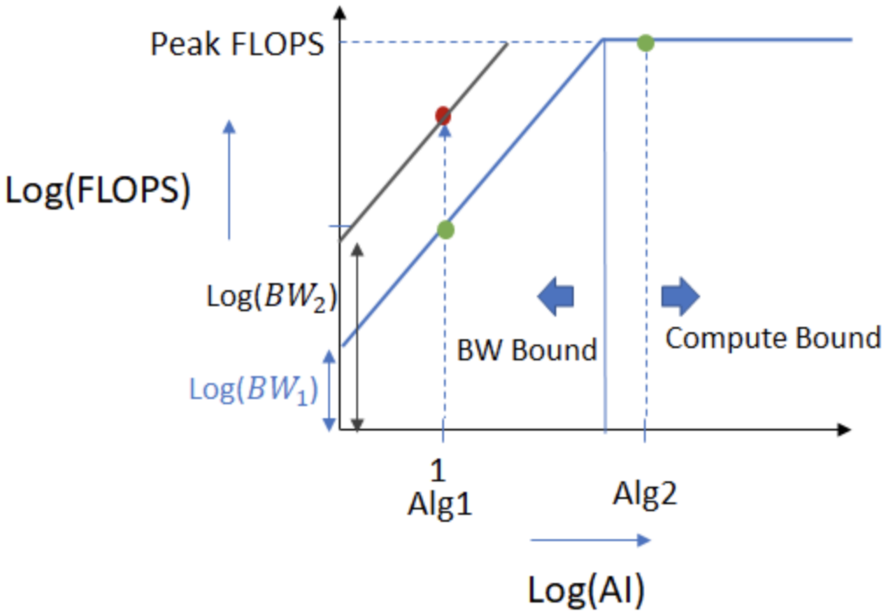

Figure 3: Roofline Plot for Recursive Matrix-Matrix Multiplication

4.4 Discussion

```
Timing Comparison

1.1. not much improvement
1.2.Column column. when transpose row column

2.1. does not depend on matrix size. it depends on cache size. unless block size is smaller tha
2.2. slide 4 is helpful for roofline

3.1.2 what is a microkernel
3.2. & in recursion?
```

Generating public/private ed25519 key pair. Enter file in which to save the key (/home/amc50/.ssh/id$_ed25519)$:
$Enterpassphrase(emptyfornopassphrase) : Entersamepassphraseagain : Youridentificationhasbeensavedin/home$
$SHA256 : 1Xu4gV32DaSy0mP1qHGao7fMWMgxI6UWWx7PYAILwSYamc50@rice.eduThekey'srandomartim$
$+--[ED25519256]--+|.o...||Eo.o.o||o.o*o++||XB*Bo.||=SXBoo||.oBB+||o*.||*..||o.+$
$.|+----[SHA256]-----+$

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 hosts.