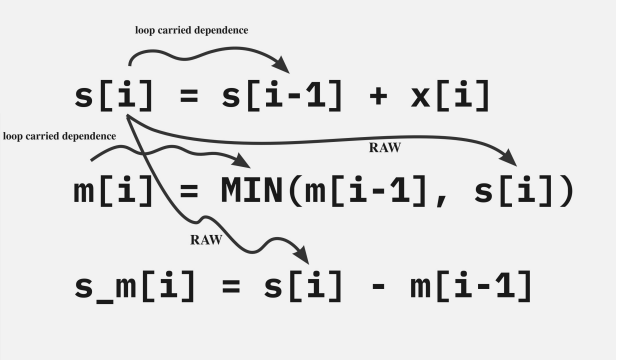
CENG444 | Programming Assignment 2

It seemed more intuitive to me to implement the serial version first and then consider where we can parallelize.

In the serial implementation I wrote, I realized the following relations.



I put in as much effort as I could for parallelization and uploaded my project to the UHeM login node.

After updating the `gcc` used for compiling on the login node using `module load gcc/11.2.0` command, I compiled my program with `gcc mcs-par.c -fopenmp -o mcs-par`. —For other commands such as profiling, you can examine the `basit.sh` script in the folder I submitted.—

I prepared the job script that will run for 2,4,8,16 threads at a time and delivered the job to one of the `core40q` computation nodes.

```
[yatmaca@sariyer ~ ]$ isler
             JOBID PARTITION     NAME     USER    STATE
                    TIME  TIME_LIMIT NODES CPUS  NODELIST(REASON)
            827454   core40q basit.sh  yatmaca CONFIGUR
                    0:02  7-00:00:00     1   40  s207

                                  0 PENDING JOBS (  0 Nodes,    0 CPUs),
                                  0 RUNNING JOBS (  0 Nodes,    0 CPUs).
```

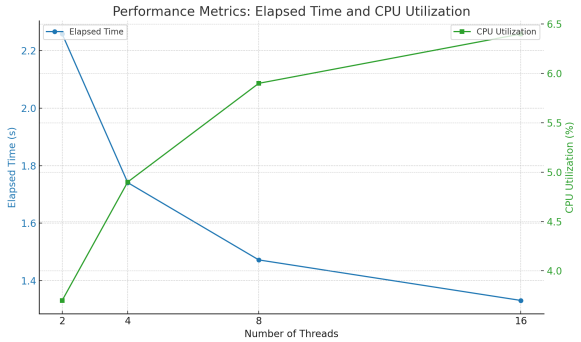
Some necessary information about the machine on which my program is running is as follows:

```
[yatmaca@s207 ~ ]$ lscpu
...
CPU(s):                40
On-line CPU(s) list:   0-39
Thread(s) per core:    1
Core(s) per socket:    20
Socket(s):             2
```

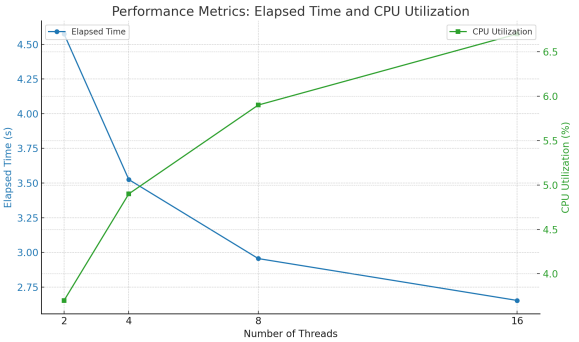
```
...

Model name:            Intel(R) Xeon(R) Gold 6148 CPU
                                                @ 2.40GHz
...
```

I am adding graphs prepared using performance metrics to understand how the performance of the program changes as the number of threads changes.



Total Number of Element: 100M



Total Number of Element: 200M