Assignment 4 - Parallel Matrix Multiplication

Ghazal Rafiei

April 25, 2022

1 Introduction

The objective in this assignment is to parallelize traditional matrix multiplication. Also, compare between different scheduling methods. In the rest of this document, we will discuss the platform specifications, implementation approach, experiment method and the results.

2 Platform

The program is written in C++ language using OpenMP library. Furthermore, here is the specification of the system.

```
Linux zenbookux434flcux433flc 5.15.25-1-MANJARO 1 SMP PREEMPT x86_64 GNU/Linux
CPU(s):  8
Vendor ID: GenuineIntel
Model name:  Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz
MemTotal:  16179008 kB
```

3 Parallel Implementation

The traditional algorithm of matrix multiplication is written and parallelized using `#pragma omp parallel for` before the outer `for`.

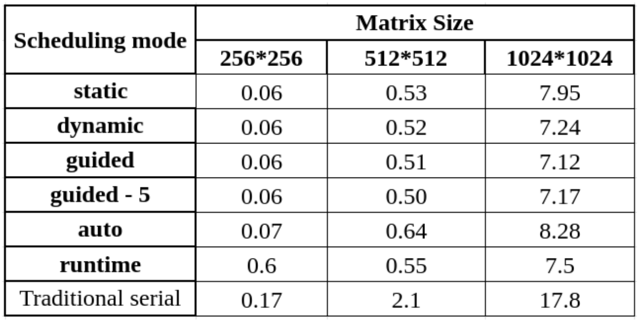
4 Experiment

Different scheduling methods of OpenMP is tested and compared with the traditional serial algorithm. Each duration is average of 50 runs of each algorithm. In the next section, we will see the run time of each one.

5 Results

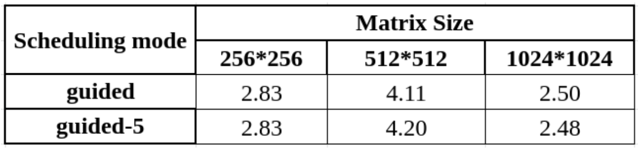
In the following table, we can see the results of explained experiment in the previous section.

Table 5.1 - Duration (second) for random matrices on 16 threads with different schedulings.



Scheduling mode	**Matrix Size**		
	256*256	**512*512**	**1024*1024**
static	0.06	0.53	7.95
dynamic	0.06	0.52	7.24
guided	0.06	0.51	7.12
guided - 5	0.06	0.50	7.17
auto	0.07	0.64	8.28
runtime	0.6	0.55	7.5
Traditional serial	0.17	2.1	17.8

Table 5.2 - Speedup for random matrices on 16 threads with different schedulings.



Scheduling mode	Matrix Size		
	256*256	512*512	1024*1024
guided	2.83	4.11	2.50
guided-5	2.83	4.20	2.48

As it is shown in the table 5.1 and 5.2, the best method of scheduling is `guided`. Using this method, the speedup rates for 256, 512 and 1024 sizes of matrices are approximately 2.83, 4.20 and 2.50 respectively.