2022.1 Multicore Computing, Project #3

Problem 1

Document

소프트웨어학부

20176342 송민준

(a) in what environment (e.g. CPU type, memory size, OS type ...) the experimentation was performed

CPU : AMD Ryzen 5 2600X Six-Core Processor (12 CPUs), ~3.6GHz

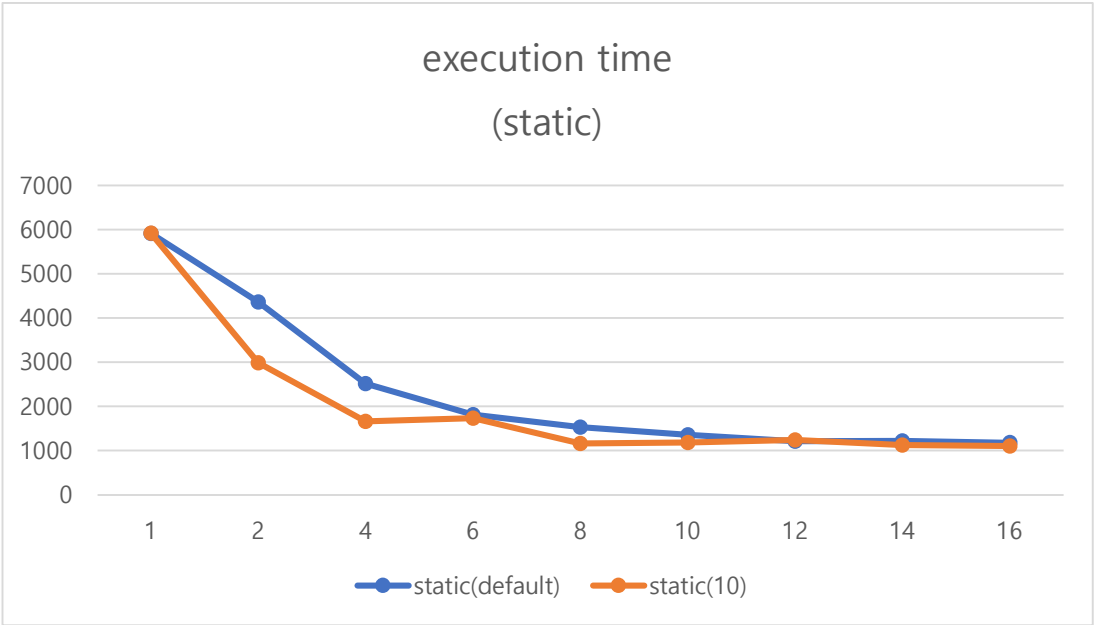
Memory : DDR4 16384MB RAM

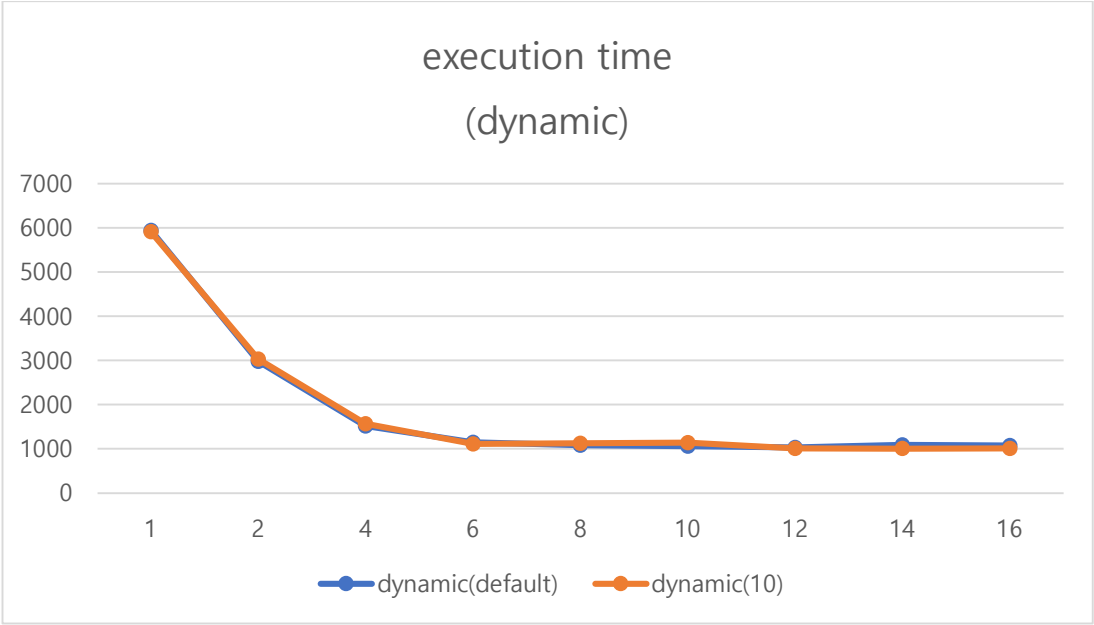
OS : Windows 10

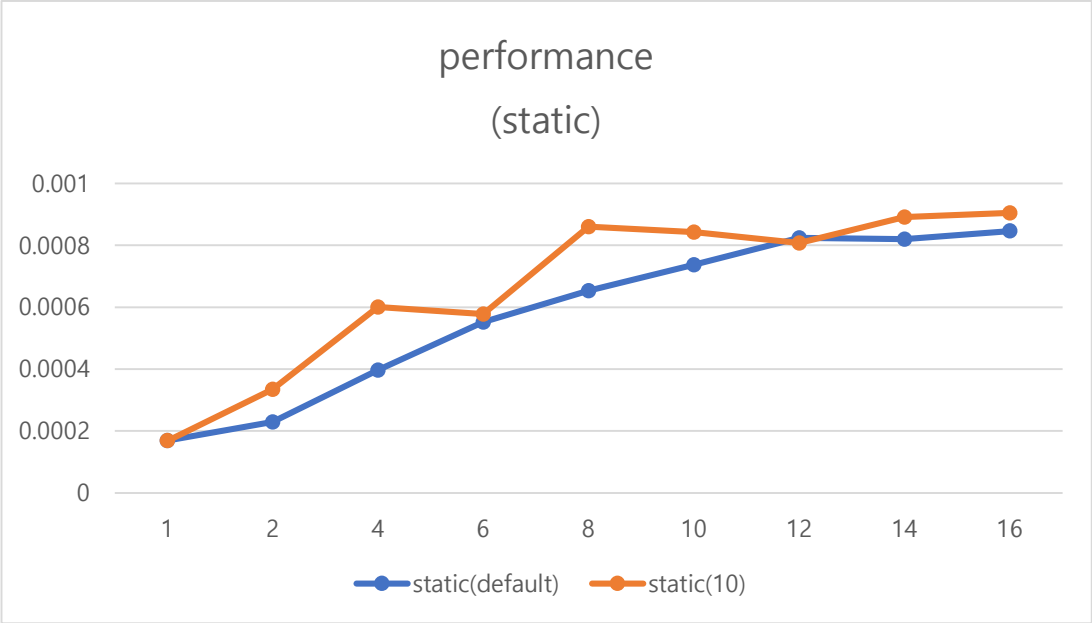
(b) tables and graphs that show the execution time (unit:milisecond) for thread number = {1,2,4,6,8,10,12,14,16}

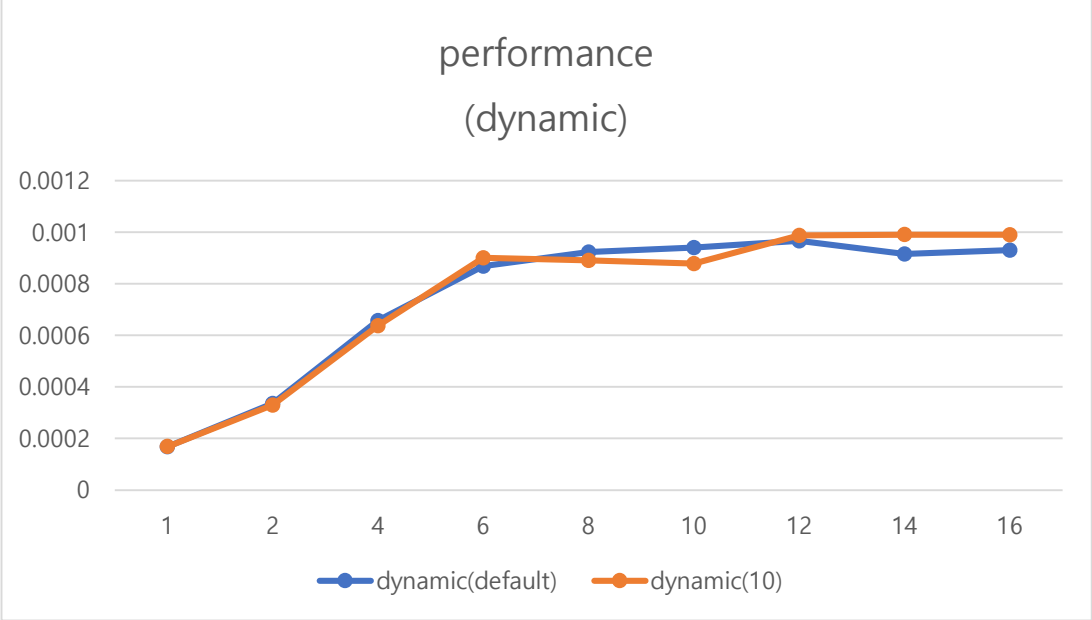
Exec time(unit: ms)	Chunk Size	1	2	4	6	8	10	12	14	16
static	Default	5914	4360	2519	1811	1530	1356	1213	1220	1181
Dynamic	Default	5944	2983	1520	1151	1083	1063	1034	1092	1075
Static	10	5919	2989	1665	1731	1162	1187	1238	1122	1105
Dynamic	10	5913	3033	1568	1110	1123	1139	1012	1009	1010

Performance (1/exec time)	Chunk Size	1	2	4	6	8	10	12	14	16
static	Default	0.00016909	0.0002294	0.000397	0.0005522	0.0006536	0.0007375	0.0008244	0.0008197	0.0008467
Dynamic	Default	0.00016824	0.0003352	0.0006579	0.0008688	0.0009234	0.0009407	0.0009671	0.0009158	0.0009302
Static	10	0.00016895	0.0003346	0.0006006	0.0005777	0.0008606	0.0008425	0.0008078	0.0008913	0.000905
Dynamic	10	0.00016912	0.0003297	0.0006378	0.0009009	0.0008905	0.000878	0.0009881	0.0009911	0.0009901


execution time
(static)
7000
6000
5000
4000
3000
2000
1000
0
1
2
4
6
8
10
12
14
16
static(default)
static(10)


execution time
(dynamic)
7000
6000
5000
4000
3000
2000
1000
0
1
2
4
6
8
10
12
14
16
dynamic(default)
dynamic(10)


performance
(static)
0.001
0.0008
0.0006
0.0004
0.0002
0
1
2
4
6
8
10
12
14
16
static(default)
static(10)


performance
(dynamic)
0.0012
0.001
0.0008
0.0006
0.0004
0.0002
0
1
2
4
6
8
10
12
14
16
dynamic(default)
dynamic(10)

(c) The document should also contain explanation on the results and why such results can be obtained.

First, static and dynamic load balancing gradually increases performance as the number of threads increases regardless of chunk size.

However, the performance increased until the number of threads was about 6, and the performance increase was insignificant because the overhead of context switching of the thread was greater than that of multi-tasking performance.

The reason for this result is that my computer has 6 cores of cpu and 12 threads.

In dynamic load balancing, the results were almost the same when the chunk size was default and at 10.

This is because there is little overhead for finding prime numbers in chunks, and if dynamic load balancing is good for obtaining prime numbers one by one or 10 by one, the execution speed is not much different.

Therefore, in the case of cyclic load balancing, unequal work distribution may occur when obtaining a few. For example, certain threads are always distributed in even numbers. Therefore, in the case of static 10, when the number of threads increases from 4 to 6, the execution time rather increases.

All execution results are no longer faster when there are more than 12 threads. The point at which the context switching overhead becomes larger is considered to be that point.