멀티코어컴퓨팅(Spring, 2023)

20171248 안재형

# Multicore Project #3 - Problem02

## Environment

208 Building 6th floor laboratory PC

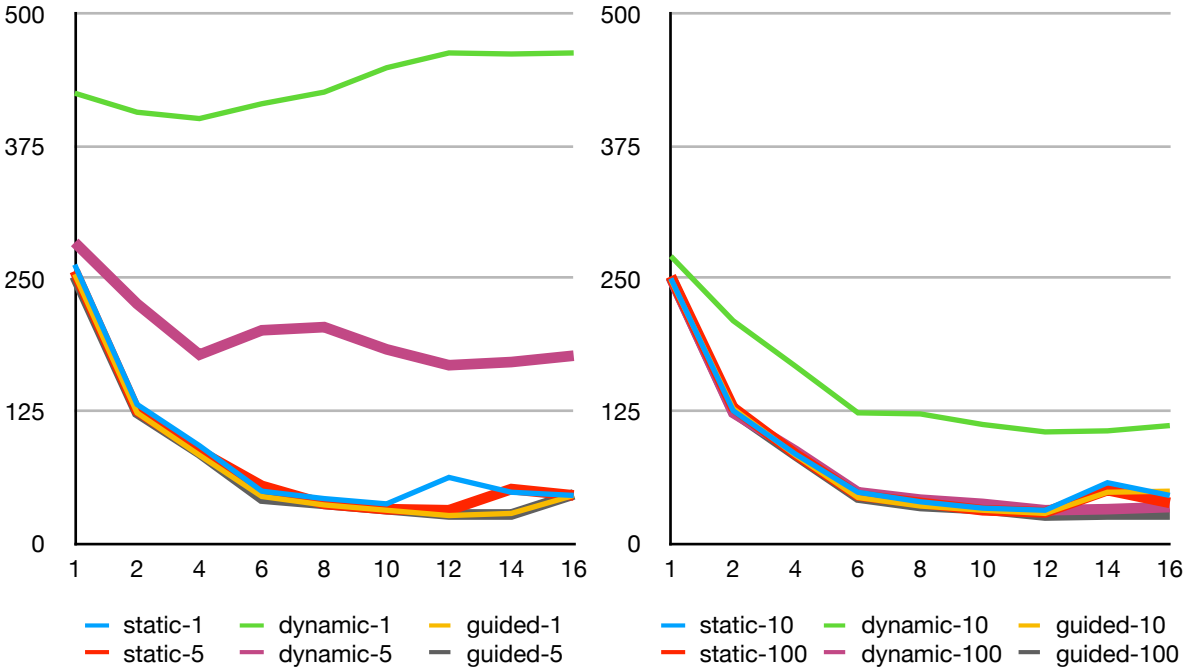
CPU : Intel Core i7-8700 (# of cores : 12)

Memory : 16GB

OS : Windows 10

IDE : Visual Studio 2019

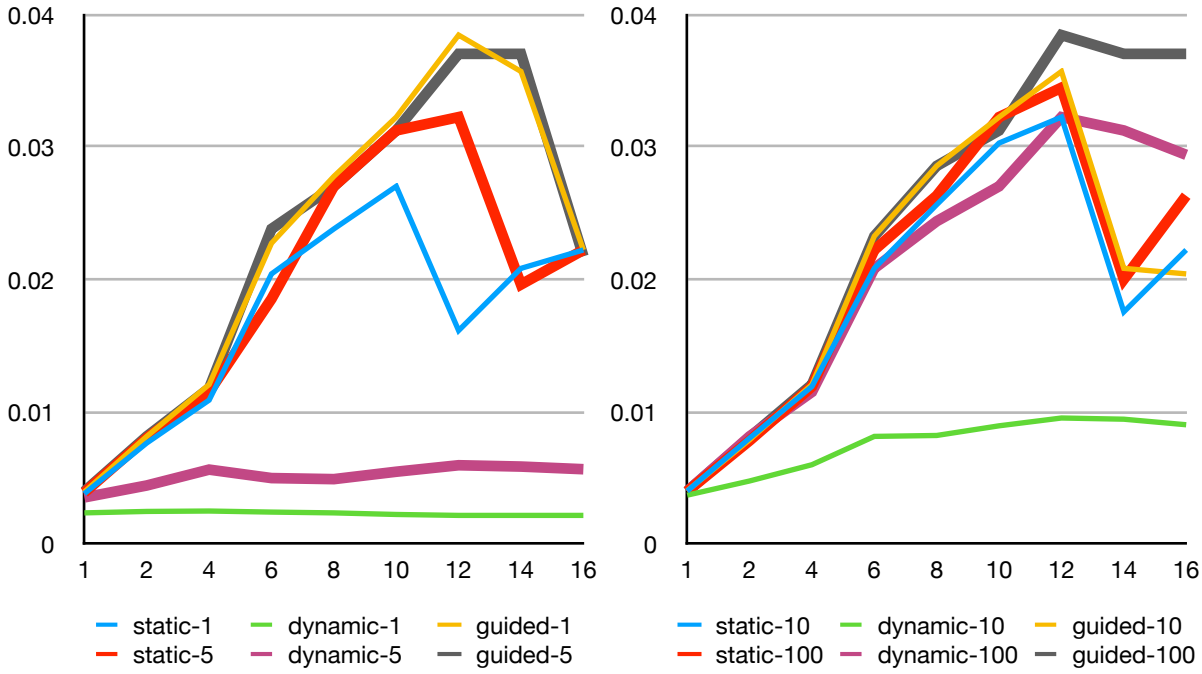
## Execution Result




(unit : ms)

| exec_time | chunc size | 1 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|
| static | 1 | 263 | 131 | 92 | 49 | 42 | 37 | 62 | 48 | 45 |
| dynamic | 1 | 425 | 407 | 401 | 415 | 426 | 449 | 463 | 462 | 463 |
| guided | 1 | 254 | 123 | 83 | 44 | 36 | 31 | 26 | 28 | 45 |
| static | 5 | 257 | 125 | 87 | 54 | 37 | 32 | 31 | 51 | 45 |
| dynamic | 5 | 284 | 226 | 178 | 201 | 204 | 183 | 168 | 171 | 177 |
| guided | 5 | 252 | 123 | 85 | 42 | 37 | 32 | 27 | 27 | 46 |

| exec_time | chunc size | 1 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|
| static | 10 | 250 | 125 | 84 | 48 | 39 | 33 | 31 | 57 | 45 |
| dynamic | 10 | 271 | 210 | 167 | 123 | 122 | 112 | 105 | 106 | 111 |
| guided | 10 | 249 | 127 | 83 | 43 | 35 | 31 | 28 | 48 | 49 |
| static | 100 | 252 | 128 | 84 | 45 | 38 | 31 | 29 | 50 | 38 |
| dynamic | 100 | 251 | 123 | 87 | 48 | 41 | 37 | 31 | 32 | 34 |
| guided | 100 | 251 | 124 | 83 | 43 | 35 | 32 | 26 | 27 | 27 |




performance = 1/exec_time

| perfor-mance | chunc size | 1 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|
| static | 1 | 0.00380 | 0.00763 | 0.01087 | 0.02041 | 0.02381 | 0.02703 | 0.01613 | 0.02083 | 0.02222 |
| dynamic | 1 | 0.00235 | 0.00246 | 0.00249 | 0.00241 | 0.00235 | 0.00223 | 0.00216 | 0.00216 | 0.00216 |
| guided | 1 | 0.00394 | 0.00813 | 0.01205 | 0.02273 | 0.02778 | 0.03226 | 0.03846 | 0.03571 | 0.02222 |
| static | 5 | 0.00389 | 0.00800 | 0.01149 | 0.01852 | 0.02703 | 0.03125 | 0.03226 | 0.01961 | 0.02222 |
| dynamic | 5 | 0.00352 | 0.00442 | 0.00562 | 0.00498 | 0.00490 | 0.00546 | 0.00595 | 0.00585 | 0.00565 |
| guided | 5 | 0.00397 | 0.00813 | 0.01176 | 0.02381 | 0.02703 | 0.03125 | 0.03704 | 0.03704 | 0.02174 |
| static | 10 | 0.00400 | 0.00800 | 0.01190 | 0.02083 | 0.02564 | 0.03030 | 0.03226 | 0.01754 | 0.02222 |
| dynamic | 10 | 0.00369 | 0.00476 | 0.00599 | 0.00813 | 0.00820 | 0.00893 | 0.00952 | 0.00943 | 0.00901 |
| guided | 10 | 0.00402 | 0.00787 | 0.01205 | 0.02326 | 0.02857 | 0.03226 | 0.03571 | 0.02083 | 0.02041 |
| static | 100 | 0.00397 | 0.00781 | 0.01190 | 0.02222 | 0.02632 | 0.03226 | 0.03448 | 0.02000 | 0.02632 |
| dynamic | 100 | 0.00398 | 0.00813 | 0.01149 | 0.02083 | 0.02439 | 0.02703 | 0.03226 | 0.03125 | 0.02941 |
| guided | 100 | 0.00398 | 0.00806 | 0.01205 | 0.02326 | 0.02857 | 0.03125 | 0.03846 | 0.03704 | 0.03704 |

* Because there were a lot of executions to do, I modified the code to automatically execute and print the time and result.

# Analysis

The size of default chunk is 1.

I could see the performance is like :

**(scheduling) dynamic < static =< guided**

**(chunk size) 1 < 5 < 10 =< 100**

If other conditions are the same,

- Static scheduling was faster than dynamic scheduling
  - In this problem, it is assumed that there was large overhead in dynamic scheduling. Especially, dynamic scheduling with smaller chunk size had an obvious performance decrease.
  - Guided scheduling had a slight higher performance because of the better load-balance
- It became faster as chunk size increases
  - As chunk size gets smaller, overhead increases and it leads to decrease of performance.

Also, I could see that as the number of threads exceeds the actual cores, there was a disadvantage of performance.