Problem 2 : Matrix Multiplication

* The processor/execution environment is described in Problem 1 report.

Analysis

* Result Talble
* The input data uses a given 500x500 matrix.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Thread# | 1 | 2 | 4 | 6 | 8 |
| Exec time | 309 | 217 | 136 | 134 | 134 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Thread# | 10 | 12 | 14 | 16 | 32 |
| Exec time | 139 | 142 | 152 | 150 | 253 |

Unit : ms

* Result Graph

차트이(가) 표시된 사진

자동 생성된 설명

<Execution Time>

차트이(가) 표시된 사진

자동 생성된 설명

<Performance Time>

* Static load balancing

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

<Thread #8> <Thread #10>

* The approach adopted here is Static load balancing BLOCK method. For a problem like Problem 1, where the time required for the task increases with the size of the input, block method may not be a good choice. However, matrix multiplication involves random numbers, and thus, even if divided into blocks, load balancing can still be achieved.
* Looking at the table above, we showed that the higher the number of threads, the better the performance improvement, but when there are more than 10 threads, it does not improve to some extent, and then suddenly decreases when there are 32. I think it's probably because the more threads there are, the more overhead there is, and these threads take matrix information, so it's more overhead.
* Overall, load balancing did not perform as well as expected. However, using a different approach may not necessarily improve load balancing, as I cannot predict how the matrix will be given.

텍스트이(가) 표시된 사진

자동 생성된 설명

* Here, Block\_Size is calculated as 500 divided by the number of threads since the result matrix has dimensions of 500 \* 500. Moreover, for matrix multiplication, we multiply each row of matrix A by each column of matrix B and sum up the products. Therefore, we assigned each thread to handle a block of rows from matrix A.

텍스트이(가) 표시된 사진

자동 생성된 설명

* In the run method of each thread, a specified range of calculations are performed and the results are stored in the result matrix.
* All Results Screenshot

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트, 명판이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명