

Multi-threading

Dynamic Load Balancing

1. Experiment Environment

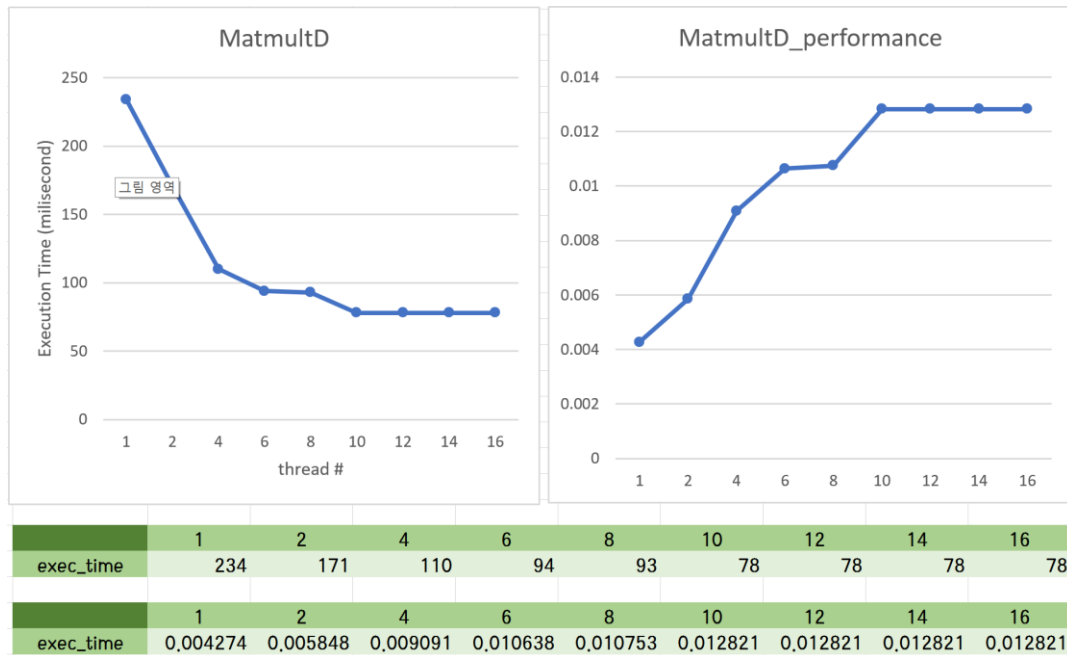


1. CPU Type: i7-6700
2. RAM: 16GB
3. OS Type: Windows
4. Clock Speed: 3.4 GHz
5. Core#: Octa-core

2. Matrix Multiplication

- A. 구현에 Dynamic load balancing을 사용하였다
- B. 매트릭스 A, B가 있을 때 스레드는 A의 x번째 행, B의 y번째 열을 선택하여 결과 매트릭스 C의 $C[x][y]$ 에 곱셈 결과를 저장한다.
- C. 스레드 i가 선언되고 행/열 하나씩 가져가서 계산을 진행한다.
- D. 처리가 끝나면 계산을 해야하는 부분이 있는지 확인한다. 확인할 때 synchronized 키워드가 선언된 함수를 호출한다.
- E. 남은 일이 있으면 마찬가지로 synchronized가 선언된 함수를 통해 일을 가져오고 행/열을 매트릭스 크기에 맞게 증가시켜준다.
- F. 행렬 곱셈이 끝나면 무한루프가 종료되고 스레드 이름과 실행 시간을 출력한다.

3. 실행시간 그래프



- MatmultD 그래프: x축은 스레드 개수, y축은 실행 시간을 의미한다.
- MatmultD_performance 그래프: x축은 스레드 개수, y축은 성능(1/실행시간)을 의미한다.

1. Static/Dynamic 둘 다 실행시간이 점점 감소(성능이 증가)함을 알 수 있다.
2. 스레드가 어느정도 커지면 실행 시간 감소량이 줄어드는(성능 향상이 더디어지는) 것을 볼 수 있다. 이는 스레드를 처음 선언하는 비용이 있기 때문에 감소량이 줄어든다.
3. 스레드 개수가 2배 늘어난다고 실행 시간이 2배 줄어드는 것은 아님을 확인할 수 있다. 이는 멀티 스레딩을 통해 성능을 향상시키는 데에는 한계가 있고, 나머지 부분 (메인 스레드 혼자 처리하는 부분)이 성능 향상에 많은 영향을 미침을 알 수 있다.

4. 별첨(실행 결과)

1. 스레드 1부터 순서대로 증가, 확대해서 자세히 확인할 수 있습니다

