

시스템 프로그래밍 멀티 쓰레딩 실습

작성자: 김영진, 김정현

Github Repo: <https://github.com/smilu97/system-hyu>

목표

4000 by 4000 행렬 곱셈을 멀티쓰레딩을 이용해서 속도 개선함

방법

현재 대부분의 컴퓨터에는 한 개의 CPU에 다수의 코어가 들어있다.

그렇기 때문에 운영체제에서는 동시에 여러 개의 쓰레드를 처리할 수 있도록 지원해주므로, 독립적으로 실행가능한 작업들은 그것들을 각각 다른 쓰레드에 분리시켜줌으로써 여러 코어를 사용가능하게 만들어줌으로써, 이론적으로는 싱글 쓰레드로 처리하는 것에 비해 배로 성능을 향상시킬 수 있다.

‘ $A * B = C$ ’를 수행한다고 할 때, A와 B에서는 Read밖에 일어나지 않으므로 C에 Write하는 부분을 여러 조각으로 쪼갤 수 있다. 쪼개는 방법은 단순히 C에 들어갈 각 원소의 자리마다 나눌 수 있는데, 이번 과제에서는 Row마다 나누는 것으로 한다.

[pthread_create](#)

쓰레드를 생성하는 순간부터 새로운 작업을 시작한다.

[pthread_join](#)

쓰레드에 등록된 작업이 끝날 때 까지 기다린다

[pthread_barrier](#)

pthread_barrier_t 에 특정 개수의 wait가 들어가기 전까지 wait를 건 모든 쓰레드가 대기하는 것으로 보인다.

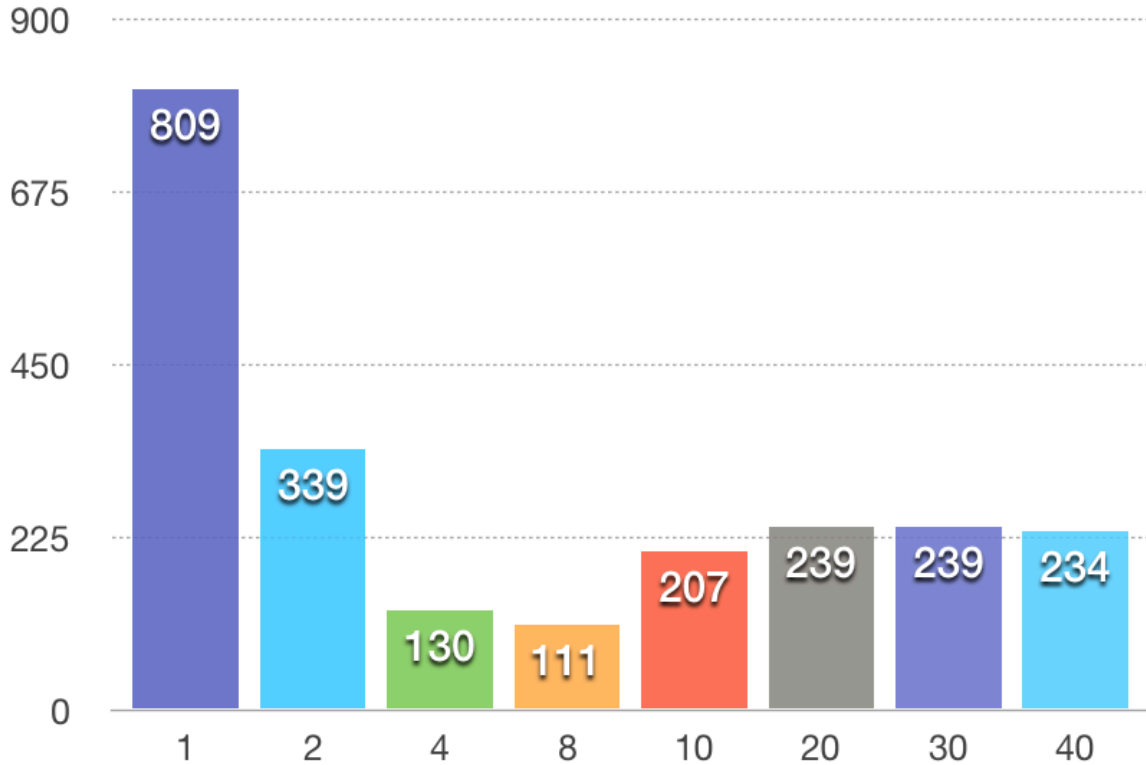
또한 OSX의 gcc에서 지원하지 않고 man7.org에서 나오지 않는 것으로보아 운영체제마다 지원하지 않는 경우도 있는 것으로 보인다.

결과

[스크린샷](#) <- 스크린샷 크기가 너무 커서 브라우저에서 확인해주세요

쓰레드가 너무 적어도, 많아도 성능이 좋지 않았으며, 8쓰레드에서 가장 성능이 좋았고, 20쓰레드 이상부터는 비슷한 성능을 보여주었다.

속도 차이



기타 개선할 수 있는 점들

행렬 곱셈의 경우 Naive한 알고리즘의 경우 ' $O(N^3)$ '의 시간이 드는 것으로 알려져있다. 하지만 알려진 [몇몇 알고리즘들](#)은 이것을 약 ' $O(N^{2.795})$ ' 정도로 줄였고, 현재 가장 빠른 알고리즘은 ' $O(n^{2.3737})$ ' 정도 시간에 처리할 수 있다고 한다.

슈트라센

행렬의 곱셈에서는 ' $A * B = C$ '를 수행한다고 했을 때, A와 B를 적절히 쪼개어 곱하는 것으로 곱셈의 횟수를 줄일 수 있는데, 슈트라센 알고리즘의 경우 이러한 전략중의 하나라고 볼 수 있다. 또한, 이 알고리즘도 충분히 멀티쓰레딩을 활용할 수 있을 것으로 보인다.

4000x4000 행렬에서 사용할 경우 이 알고리즘을 사용하기 위해 강제로 4096x4096행렬로 변환해도 ' $4096^{2.7} = 5667243323.58...$ ' 이고 ' $4000^3 = 64000000000$ ' 이므로 11.3배 정도 성능을 향상 시킬 수 있다.

유명한 파이썬 수학 연산 라이브러리 중의 하나인 numpy의 소스코드를 보면, 이 라이브러리에서도 matmul함수를 호출했을 시에 이 알고리즘을 쓰는 것으로 보인다. 실제로, 테스트해보면 본 과제에서 구현한 ' $O(N^3)$ '의 프로그램보다 numpy에서의 matmul이 더 좋은 성능을 내는 것으로 보인다.

CW(Coppersmith-Winograd)

위와 같은 행렬곱에서의 전략을 전처리를 통해 최대한 최적화 한 뒤에 곱하는 것으로 보인다. CW의 경우 Implementation을 찾을 수는 없었으며, 직접 구현해보지는 않았다.