고급 소프트웨어 실습 보고서 12주차

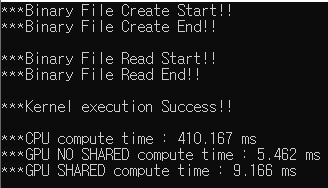
20171693 조병화

**실습 1.**

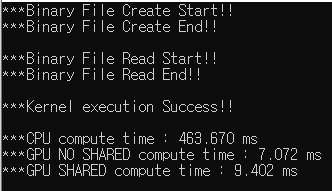
NF값을 변경시켜 가면서 gpu를 이용하였을 때와 cpu를 이용하였을 때의 속도를 비교해보고, shared memory를 사용한 경우와 사용하지 않은 경우에 대해서도 그 속도를 비교해 보도록 하겠다.

다음은 같은 block size일 때 nf 값을 변경시켜가면서 그 시간을 비교해 본 결과이다.

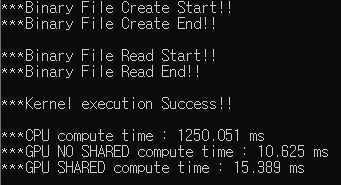
Block size 32 . Nf =1 일때



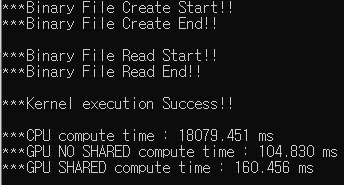
Block size 32 . Nf =4 일때



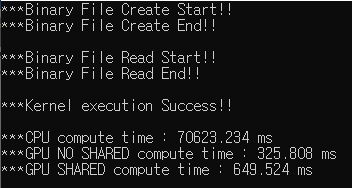
Block size 32 . Nf =16 일때



Block size 32 . Nf =256 일때



Block size 32 . Nf =1024 일때



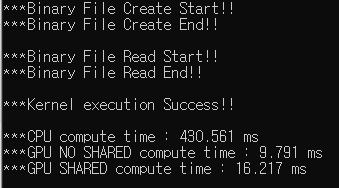
다음 결과를 분석해본 결과 예상한바와 같이 cpu 의 시간이 gpu의 시간이 더 짧게 걸렸음을 확인할 수 있었다. 하지만 shared memory 를 사용한 것과 사용하지 않은 경우에서 shared memory를 사용한 결과가 더 빠를 것이라 예측하였으나 실제로는 no shared memory의 경우가 시간이 더 짧게 걸렸음을 알 수 있었다. 이는 코드에서의 최적화가 제대로 이루어 지지 않았기 때문인데,

Shared 에서 buffer에 접근하는 횟수가 더 많아지면 그 만큼 시간이 더 소모되기 때문에 일어나는 문제로, 이를 최적화를 시켜주면 shared의 경우의 시간이 더 짧게 걸릴 것이다. Nf의 값이 커질수록 shared의 경우가 no shared의 속도를 조금씩 따라잡는 것을 확인할 수 있다.

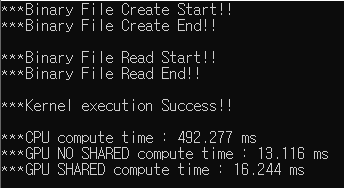
다음은 block size 의 크기를 변화시켜 주면서 shared 와 no shared 의 속도를 비교해 보았다.

Block Size를 16일 때와 32일 때를 서로 비교해 보도록 하겠다. 위에서 block size 32 일때의 결과를 보였고, 16일때의 결과를 보이면,

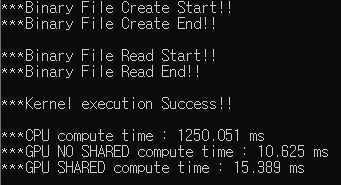
Block size 16 . Nf =1 일때



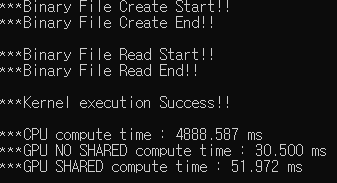
Block size 16 . Nf =4 일때



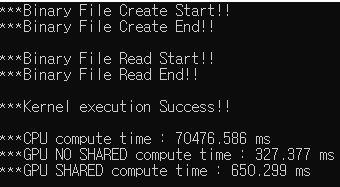
Block size 16 . Nf =16 일때



Block size 16 . Nf =64 일때

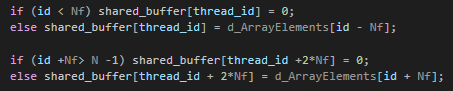


Block size 16 . Nf =1024 일때

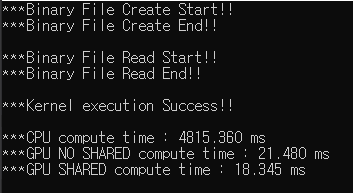


이 블록사이즈를 16에서 32로 키웠을 떄 shared 와 no shared 일 때 속도가 모두 다 느려지는 것을 확인할 수 있다. 이는 GPU 에 따라 어느 사이즈의 블락으로 나누는 것이 가장 효율적인지가 다르기 때문이다.

위에서 최적화를 통해 시간을 줄일 수 있다고 하였는데,



기존의 for문을 이용한 방식에서 다음과 같이 바꾸어 보았다. 이를 통해 최적화를 한 결과는 다음과 같다.



이를 통해 shared 일 때의 결과가 더 빨라졌음을 확인 할 수 있었다. 하지만, 이를 오류없이 제대로 된 결과를 얻기 위해선 BLOCK SIZE를 1024로 설정해 주어야 한다.

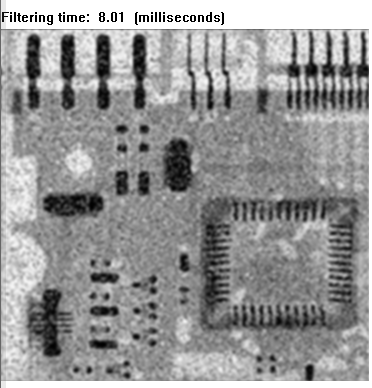
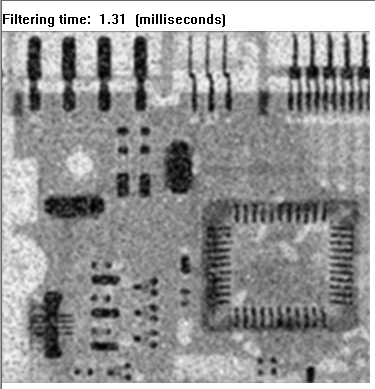
기존의 코드는 주석으로 처리해 주었다.

과제 1

이번 과제는 gaussian filter를 적용하여 영상에 blurring을 넣는 과정을 cpu, gpu 각각을 이용하여 실행해 보았다. Block size를 변화하여 실행해본 결과를 각각 살펴보자.

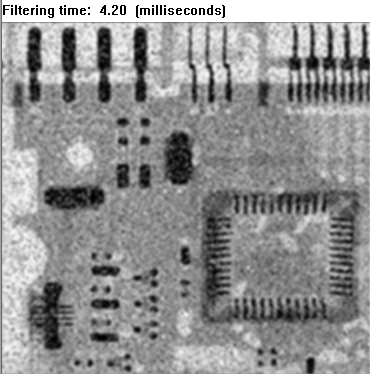
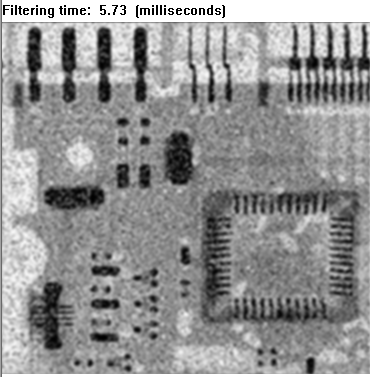
Block Size 16 일 때

No Shared 와 Shared

Block Size 32 일 때

No Shared 와 Shared

위 결과를 분석해본 결과 항상 CPI를 이용한 계산 결과보다 GPU의 계산결과가 훨씬 빨랐고, Shared 메모리를 이용하였을 경우가 no shared 보다 시간이 더 오래걸렸는데. 이는 실습에서 한 것과 같은 이유로 최적화가 진행되지 않았기 때문인것같다. BLOCK SIZE가 커지면 시간이 더 오래 걸린 것으로 확인 되었는데, 이 측정 결과는 처음 프로그램을 실행시켰을 때에는, 시간이 매우 짧게 나오고 이후로 그 측정값의 편차가 심해 정확히 확인할 수 없었다.