제목 : cuda를 이용한 aes알고리즘의 성능분석

서론 : 아직 작성하지 못했습니다.

GPU구조 : 실험에 사용된 rtx2060의 아키택처인 nvidia turing architecture 의 대한 설명

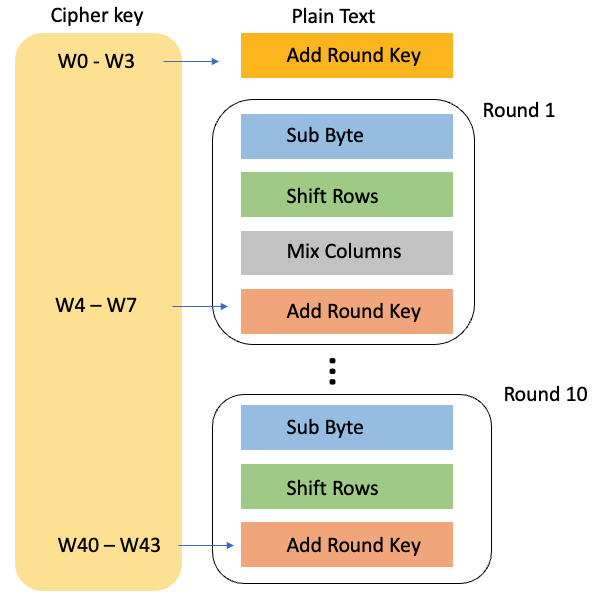
CUDA : **cuda 언어에 대한 설명**

Cuda(compute unified device architecture)는 NVIDIA의 GPU를 이용하여 프로그래밍 하기 위한 기술이다. CUDA는 GPU에서의 프로그래밍을 제공하기 위해 하드웨어와의 드라이버, 다양한 API, 컴파일러(NVCC)와 라이브러리를 제공한다. CUDA프로그래밍에서 CPU는 호스트(host), GPU는 디바이스(device)로 정의된다.

CUDA프로그래밍에서 쓰레드 모델을 이해하는 것은 매우 중요한데 최상위 레벨로 GRID가 구성되며 이는 응용프로그램이 동작할때 할당된다. 그리드는 2차원 블록을 포함하는데 하나의 블록은 하나의 단일 멀티프로세서에서 동작하고, 이때 다른불록과의 동기화가 불가능하고, 같은 시간에 동시에 실행됨을 보장받지 못한다. 블록의 내부는 3차원 쓰레드의 집합으로 구성되는데 각 쓰레드는 멀티프로세서 내부의 서로 다른 코어에 의해 할당되고, 쓰레드는 WARP라는 단위로 할당되어 SIMT형태로 명령어를 수행한다. 본 논문에서는 WARP단위를 32로 실험했다. + 메모리 설명

AES알고리즘 : **aes알고리즘의 대한 설명**

AES는 미국표준기술연구소(NITS)에 의해 재정된 암호화 방식이다. 2001년에는 미국 연방정보처리 표준으로 공표되었고, 이후 전세계적으로 널리 쓰이고 있는 암호화 방식이다. AES는 대칭키 알고리즘이 사용되며 암호화가 블록단위로 이루어 지며, 블록의 크기를 선택할 수 있다. 본 논문에서는 128bit를 기준으로 하였다. 이후 블록단위로 평문을 나누어 암호화 과정을 거친다. 암호화 과정은 Sub Byte(바이트치환), ShiftRows(행의 쉬프트), MixColumns(열의 혼합)을 거치는데 이가 한 라운드로 구성되고 평문의 길이에 따라 라운드 횟수가 정해진다. 128bit의 경우 라운드 횟수는 10이 된다.

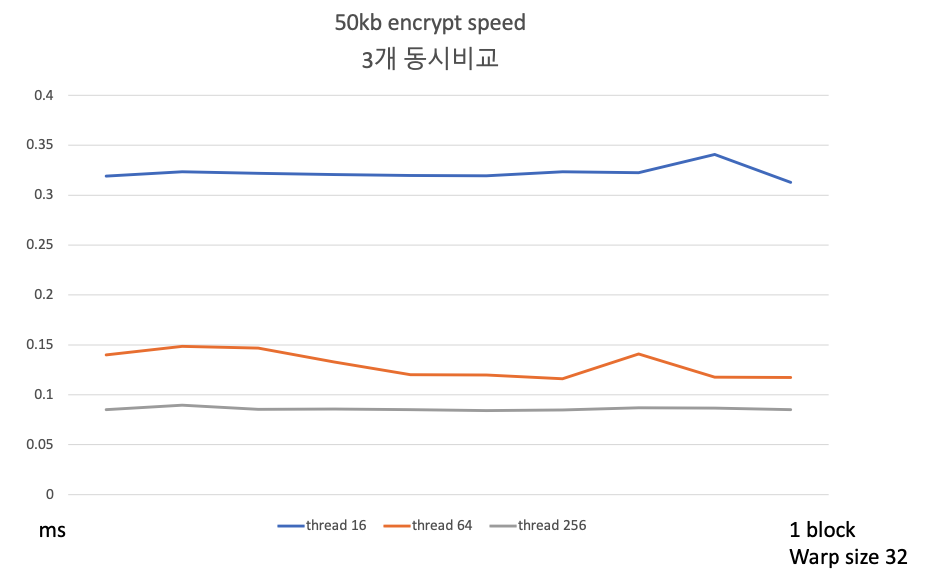


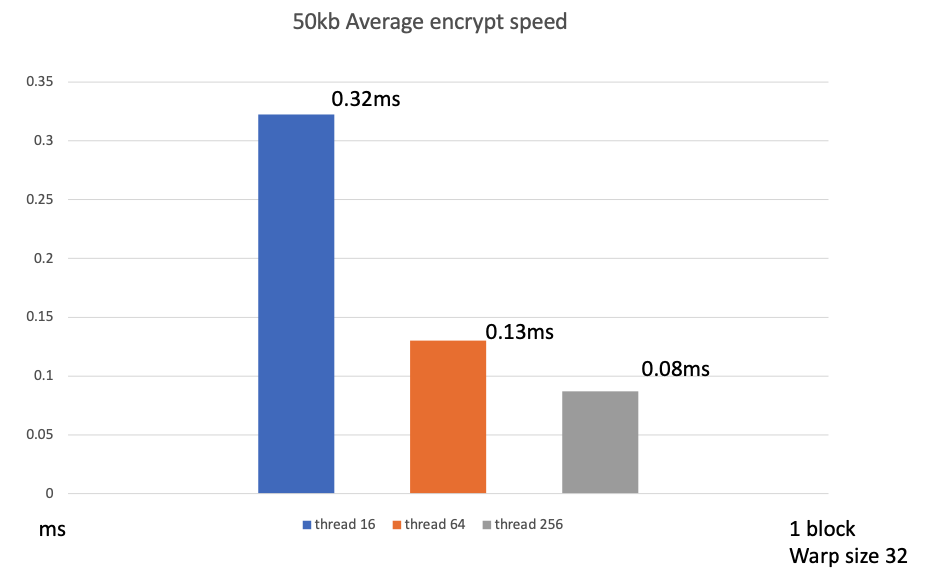
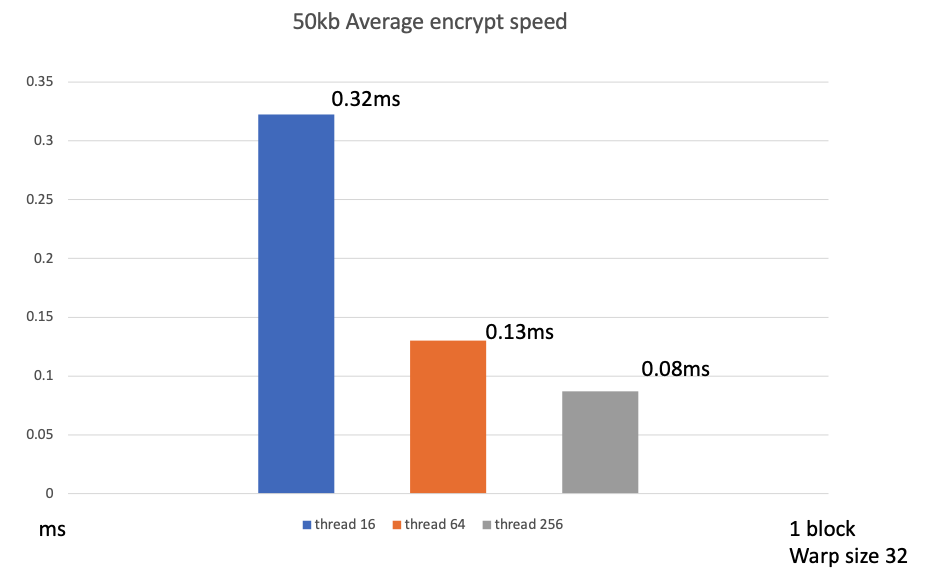
+ 각 라운드별 설명

실험결과 :

실험에 사용된 gpu는 rtx 2060이고, cuda version 11.4를 사용했다.

50kb크기의 평문을 시작으로 암호화 속도를 측정했다. 이때 wrap size는 32로, block은 1개만 할당했다.





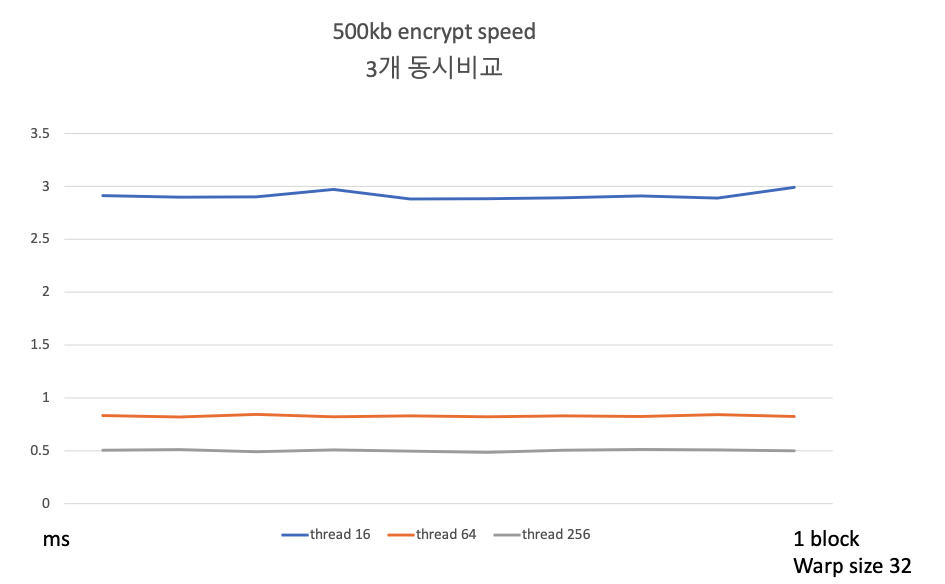
50kb파일을 암호화 할때 thread를 16개를 할당 후 암호화 할때 평균 0.32ms의 속도를 보여주었다. Thread를 64개 쓸때는 평균 0.13ms의 속도를 보여주었는데 이는 16개 대비 2.46배 빠른 속도였다. 그리고 Thread를 256개를 쓸때는 평균 0.08ms속도를 보여주었는데 이는 64개 대비 1.62배 빠르고, 16개 대비 4배 빠른 속도였다.

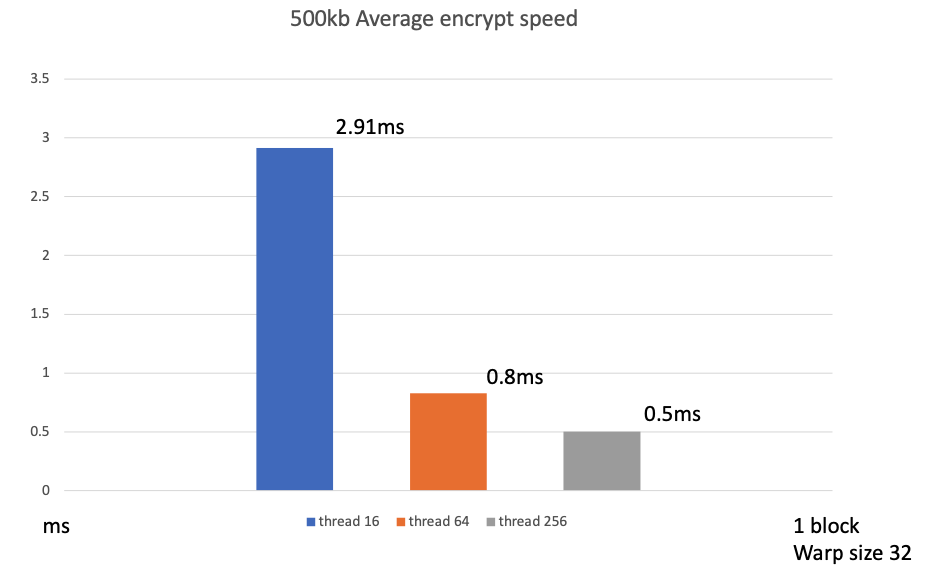
thread 64 > thread 16보다 2.46배 빠르다.

Thread 256 > thread 64보다 1.62배 빠르다.

Thread 256 > thread 16보다 4배 빠르다.

500kb파일을 암호화 할때 thread를 16개를 할당 후 암호화 할때 평균 2.93ms의 속도를 보여주었다. Thread를 64개 쓸때는 평균 0.8ms의 속도를 보여주었는데 이는 16개 대비 3.63배 빠른 속도였다. 그리고 Thread를 256개를 쓸때는 평균 0.5ms속도를 보여주었는데 이는 64개 대비 1.6배 빠르고, 16개 대비 5.82배 빠른 속도였다.



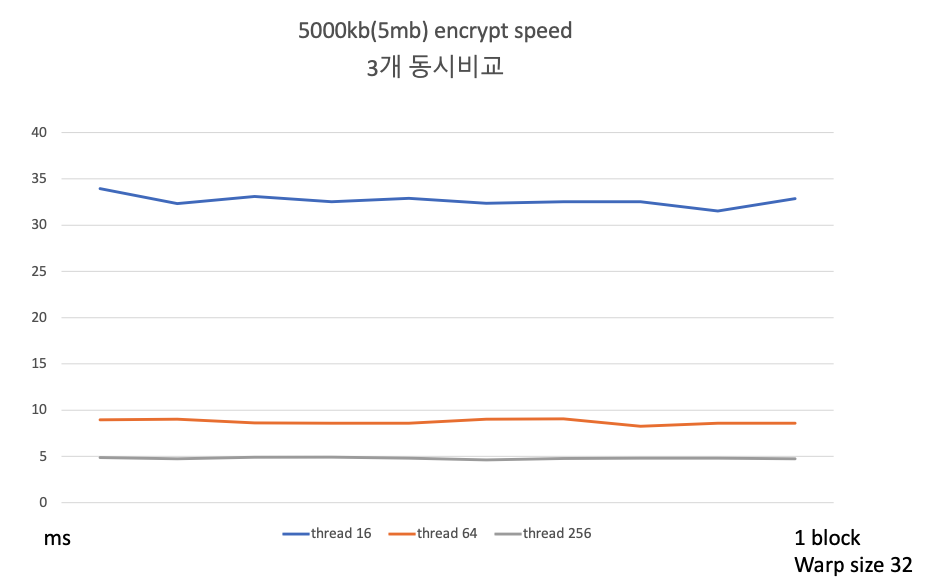


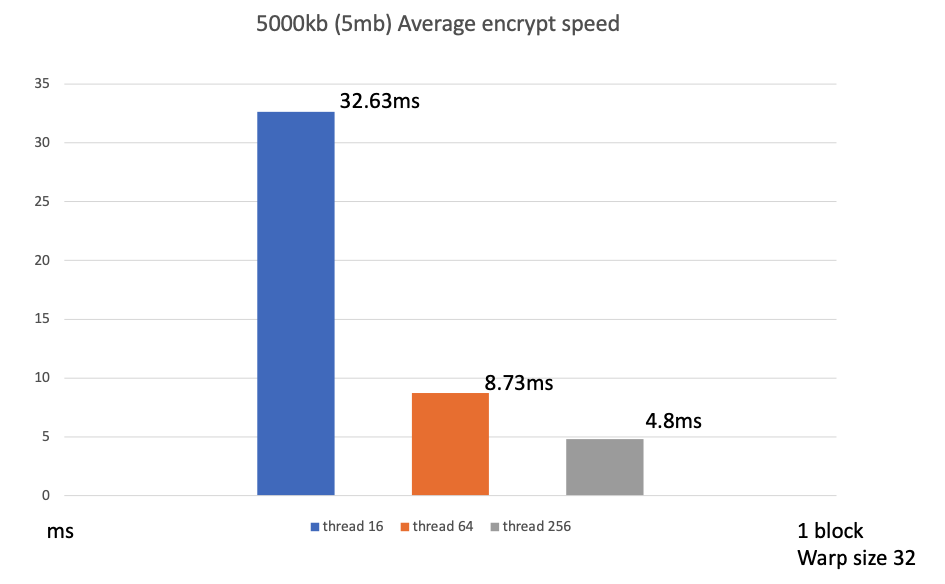
Thread 64 > thread 16보다 3.63배 빠르다.

Thread 256 > thread 64. 보다 1.6 배 빠르다.

Thread 256 > thread 16 보다 5.82 배 빠르다.

5000kb(5mb)





5000kb파일을 암호화 할때 thread를 16개를 할당 후 암호화 할때 평균 32.63ms의 속도를 보여주었다. Thread를 64개 쓸때는 평균 8.73ms의 속도를 보여주었는데 이는 16개 대비 3.74배 빠른 속도였다. 그리고 Thread를 256개를 쓸때는 평균 4.8ms속도를 보여주었는데 이는 64개 대비 1.8배 빠르고, 16개 대비 6.8배 빠른 속도였다.

Thread 64 > thread 16보다 3.74배 빠르다.

Thread 256 > thread 64. 보다 1.8배 빠르다.

Thread 256 > thread 16 보다 6.8배 빠르다.