

Project : Swaption 벤치마크 가속화

본 과제의 목표는 PARSEC 벤치마크¹에 포함된 Swaption² 구현의 성능을 OpenCL을 사용하여 가속화하는 것이다. 아래의 4가지 케이스를 구현하여 보고서와 함께 제출한다.

- a) OpenCL을 사용하여 1개 노드의 모든 CPU를 사용하는 구현
- b) OpenCL을 사용하여 1개 노드의 모든 GPU를 사용하는 구현
- c) MPI + OpenCL을 사용하여 4개 노드의 모든 GPU를 사용하는 구현
- d) SnuCL을 사용하여 4개 노드의 모든 GPU를 사용하는 구현

사용하는 SnuCL 버전은 1.3.2, 컴파일러는 gcc 4.4.6 (천둥 로그인노드의 컴파일러)를 사용하며, 그 외의 실행 환경(OpenCL 라이브러리 버전 등)은 천둥을 기준으로 한다. 프로젝트 결과물의 성능 평가 또한 천둥에서 수행한다.

첨부된 기본 소스 코드는 pthread를 사용하여 병렬화 되어있으며, 다음과 같이 컴파일 및 테스트 할 수 있다.

- 시퀀셜 버전 컴파일 : `make version = seq`
- pthread 버전 컴파일 : `make version = pthread`
- 샘플 실행 : `./run.sh`

4가지 구현물은 각각 아래와 같은 명령으로 컴파일 및 실행(thor enqueue)되도록 해야 함
컴파일 시

- a) 단일 노드 OpenCL CPU : `make version = cpu`
- b) 단일 노드 OpenCL GPU : `make version = gpu`
- c) 4 노드 MPI + OpenCL GPU: `make version = mpi`
- d) 4 노드 SnuCL GPU : `make version = snucl`

실행 시

- a) 단일 노드 OpenCL CPU : `./run_cpu`
- b) 단일 노드 OpenCL GPU : `./run_gpu`
- c) 4 노드 MPI + OpenCL GPU: `./run_mpi`
- d) 4 노드 SnuCL GPU : `./run_snucl`

¹ <http://parsec.cs.princeton.edu>

² <http://en.wikipedia.org/wiki/Swaption>

프로젝트 평가와 관련된 유의사항은 아래와 같다.

- 모든 구현물의 수행 결과는 Sequential 구현과 같아야 하며, 이는 화면에 출력되는 결과가 같은지의 여부로 판단한다.
- 평가 시 Swaption의 개수(-ns 옵션으로 설정)는 128로 고정하며, 시뮬레이션의 개수(-sm 옵션으로 설정)는 충분히 큰 수를 사용한다. 이를 고려하여 병렬화를 수행하여야 한다.
- 평가 시 네 가지 구현 모두 동등한 비중으로 점수에 반영된다.
- 평가 시 점수는 **성능 90% + 보고서 10%**로 구성된다. 보고서에는 구현 시 시도한 병렬화 아이디어 및 최적화 기법에 대해 서술하여야 한다.
- 프로젝트 관련 주요 추가 공지사항이 있을 경우 ETL을 통해 공지될 예정이니 ETL 공지사항 게시판을 자주 확인할 것

프로젝트 결과물은 **6월 12일 밤 11시 59분**까지 조교 메일로 제출할 것.

Grace day가 적용되지 않으니 주의!