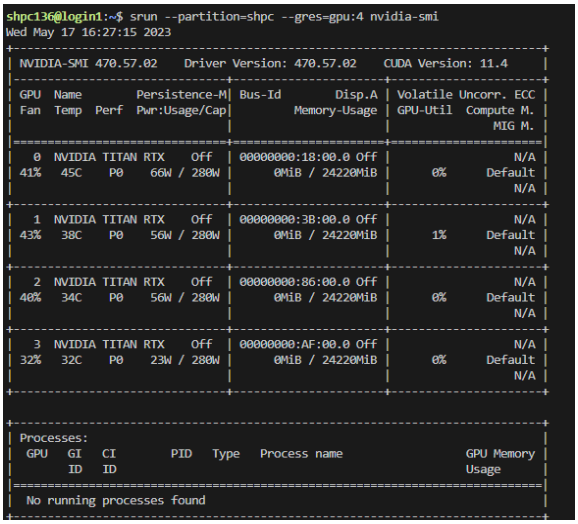
1. GPU정보 확인하기

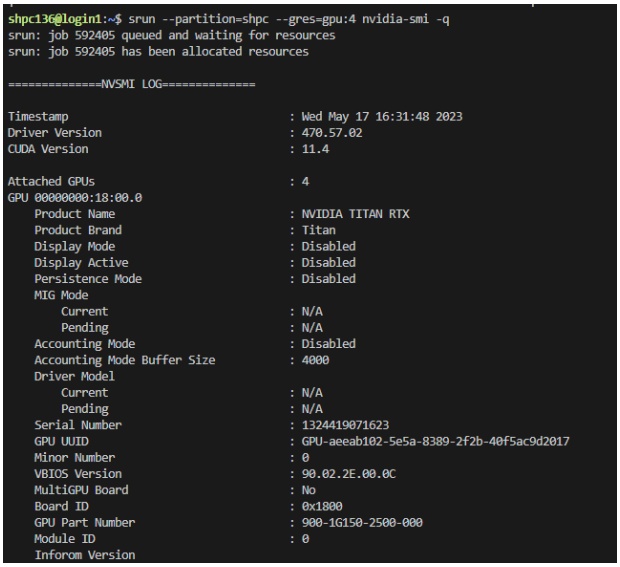
(a) srun --partition=shpc --gres=gpu:4 nvidia-smi



계산노드에 설치된 GPU 4개의 상태를 알 수 있음

(GPU 이름, driver version, Memory-usage, GPU-util, 온도, 전력소비 및 GPU에서 돌아가는 process의 정보 등등)

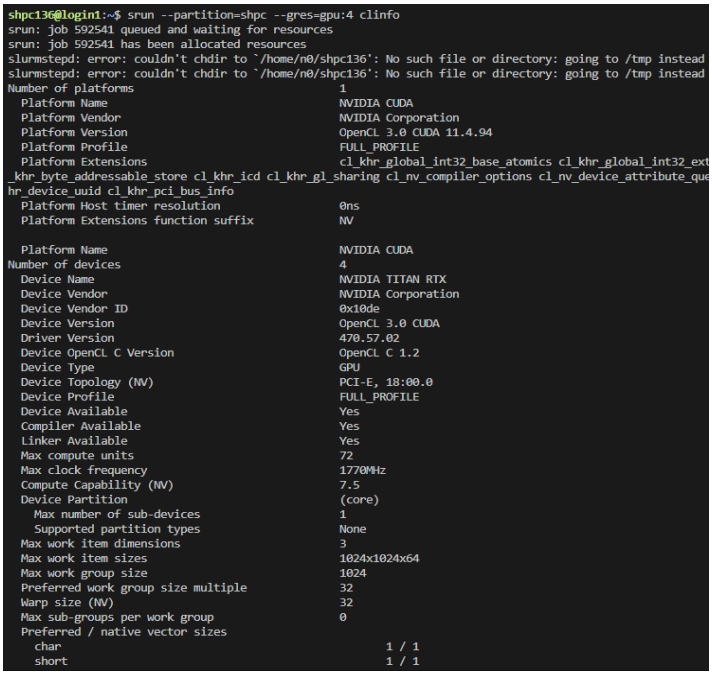
srun --partition=shpc --gres=gpu:4 nvidia-smi -q



계산노드에 설치된 4개의 GPU에 대해 더 상세한 정보에 대해 알 수 있음

(Product Name, Product Brand, Display Mode, Display Active, Persistence Mode, MIG Mode, Accounting Mode, Accounting Mode Buffer Size, Driver Model, Serial Number, GPU UUID, PCI 등등)

srun --partition=shpc --gres=gpu:4 clinfo



계산 노드 GPU 4개를 할당 받아 OpenCL 플랫폼과 디바이스에 대한 자세한 정보가 표시됨.

(플랫폼 이름, 버전, 지원하는 OpenCL버전, 디바이스 이름, 타입, 메모리 크기 등등)

(b) 모델명 : NVIDIA TITAN RTX, 노드당 GPU 개수 : 4개

(c) GPU 메모리 크기 : 24220MiB

(d) Max power limit : 320W, Max SM clock speed : 2100MHz

(e) Max work item dimension : 3, Max work item size : 1024X1024X64, Max work group size : 1024

2. Matrix Multiplication with OpenCL

(a) 병렬화 방식 :

1. Output Matrix의 크기인 M * N 스레드에 할당
2. global memory coalescing을 위해 커널 인덱스 공간을 {N, M}으로 함
3. local(shared) memory를 활용하기 위해 타일링 적용

(b) matmul.c에 대한 설명

1. matmul_initialize : 커널 코드 실행에 필요한 객체 생성

- clGetPlatformIDs : 플랫폼 개수와 ID 얻어오기
- clGetDeviceIDs : 디바이스 개수와 ID 얻어오기
- clCreateContext : 컨텍스트 생성
- clCreateCommandQueue : 커맨드큐 생성
- clCreateProgramWithSource : 소스 코드로부터 프로그램 오브젝트 만들기
- clBuildProgram : 프로그램 빌드
- clCreateKernel : 커널 오브젝트 만들기
- clCreateBuffer : 버퍼 오브젝트 만들기

2. matmul : 커널코드 실행

- clEnqueueWriteBuffer : 버퍼 쓰기
- clSetKernelArg : 커널 인자 설정
- clEnqueueNDRangeKernel : 커널 실행
- clEnqueueReadBuffer : 버퍼 읽기

3. matmul_finalize : 메모리 leaking 방지를 위해 활용이 끝난 메모리 free

- clReleaseMemobject: 버퍼 free
- clReleaseKernel : 커널 free
- clReleaseProgram : 프로그램 free
- clReleaseCommandQueue : 커맨드큐 free
- clReleaseContext : 컨텍스트 free

4. 그 외 API

- clGetplatforminfo : OpenCL 플랫폼 정보
- clGetdeviceinfo : OpenCL 디바이스 정보
- clGetProgramBuildInfo : OpenCL 프로그램빌드 정보

(c) 최적화에 따른 성능 실험(M=N=K=4096)

1. Naïve한 OpenCL 적용 : 140GFlops
2. Global Memory Coalescing : 1160GFlops
3. Local (shared) Memory Tiling : 1450GFlops