

공유노드 정책으로 운영 중인 클러스터 시스템에서 작업별 GPU 사용 통계 생성 방안에 대한 연구

권민우*, 윤준원*, 홍태영*
 *한국과학기술정보연구원 슈퍼컴퓨팅인프라센터
 mwkwon81@kisti.re.kr

A study on how to generate GPU usage statistics for each task in a cluster system operated by shared node policy

Min-Woo Kwon*, JunWeon Yoon*, TaeYoung Hong*
 *Dept. of Supercomputing Infrastructure Center, KISTI

요 약

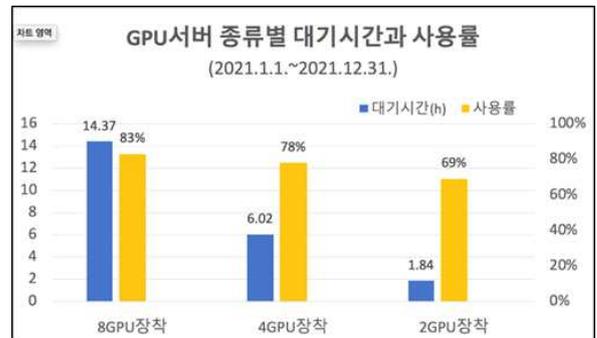
KISTI(한국과학기술정보연구원)는 슈퍼컴퓨터 5호기 메인시스템인 Nurion과 보조시스템인 Neuron을 연구자들에게 서비스하고 있다. Neuron은 메인시스템인 Nurion이 Intel Knights Landing 프로세서가 장착된 클러스터로 구성됨에 따라 인공지능, 빅데이터에 관한 연구 인프라 수요를 충족시키기 위해 GPU를 장착한 이기종 클러스터로 구성되어 있다. Neuron은 연구자들에게 효율적으로 계산 자원을 배분하기 위해 SLURM 작업배치스케줄러의 공유노드 정책을 이용하여 한 개의 계산노드에서 다수 개의 작업이 수행될 수 있는 환경으로 서비스되고 있다. 본 논문에서는 공유노드 정책으로 운영 중인 클러스터 시스템에서 작업별로 GPU 사용 통계 데이터를 생성하는 기법을 소개한다.

1. 서론

KISTI는 슈퍼컴퓨터 5호기 보조시스템 Neuron을 2019년 7월부터 공식서비스하고 있다. 메인시스템인 Nurion이 x86기반의 Intel Knights Landing 프로세서가 장착된 계산노드로 구성됨에 따라 메인시스템의 기능을 보장하는 신규 아키텍처를 탑재한 계산노드로 구성되어 있다. 최근 인공지능, 빅데이터를 이용한 연구 인프라 수요가 늘어나면서 가속성능이 뛰어난 GPU를 탑재한 계산 자원을 꾸준히 보장하고 있다[1]. Neuron과 같이 GPU가 장착된 이기종 클러스터의 계산 자원 활용률을 극대화하기 위한 작업배치스케줄러의 공유노드 정책 적용 방안이 이미 소개되었다[2]. 그리고 배타적인 노드 정책으로 운영 중인 GPU 클러스터 시스템의 작업별 GPU 사용 통계 데이터를 획득하는 방안이 소개되었다[3]. 작업별 GPU 사용 통계 데이터는 사용자별 GPU 사용 실태를 파악하는 자료로 사용될 수 있다. 본 논문에서는 작업배치스케줄러의 공유노드 정책으로 운영 중인 클러스터 시스템에서 작업별로 GPU 사용 통계 데이터를 생성하는 방안을 소개한다.

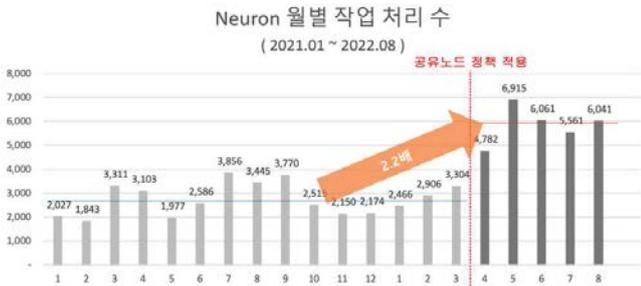
2. Neuron 공유 노드 정책

Neuron은 5년 단위로 구축이 되는 슈퍼컴퓨터 메인 시스템과는 다르게 연구 인프라 수요를 고려하여 연단위로 계산 자원을 꾸준히 보장하여 서비스하고 있다. 그림 1은 2021년 연간 GPU 서버 종류별 대기시간과 사용률을 보여준다. GPU 장착 개수가 많은 서버일수록 사용률과 대기 시간이 높은 것을 알 수 있다. 사용자의 선호도를 고려하여 2021-2022년 8GPU가 장착된 서버를 증설하였다.



(그림 1) GPU서버 종류별 대기시간과 사용률

하나의 작업이 계산노드의 전체 자원을 점유하여 사용하는 배타적노드 정책에서는 작업에서 사용하는 계산자원의 요구량에 따라 자원 낭비가 발생할 수 있다. 따라서 GPU 장착 개수가 많은 서버에서 복수개의 작업이 수행될 수 있도록 2022년 3월부터는 Neuron의 작업배치스케줄러인 SLURM의 공유노드 정책을 적용하여 서비스하고 있다[2]. 그림 2와 같이 공유노드 정책 적용 이후 월별 작업 처리 건 수가 2배 이상 증가하여 시스템의 활용률이 높아진 것을 확인할 수 있다.



(그림 2) Neuron 월별 작업 처리 수

3. 작업별 할당된 GPU Index 식별 방안

기존의 배타적 노드 정책에서는 하나의 작업이 계산노드의 전체 GPU를 점유하여 사용하므로 작업이 수행되는 동안의 할당된 노드의 전체 GPU의 코어 사용률과 메모리 사용률의 평균/최대값/최소값을 구하는 방식으로 간단하게 사용 통계 획득이 가능하였다[3]. 그러나 공유노드 정책에서는 여러 개의 작업이 계산노드를 공유하여 사용하므로 작업별로 할당된 노드 정보와 함께 각각의 노드에서 할당된 GPU Index를 식별해야 한다. SLURM 스케줄러에서는 'scontrol show jobid=(작업ID) -dd' 커맨드를 이용하여 작업에 할당된 GPU Index 정보를 제공하고 있으며 [4], Neuron에서는 SLURM의 프로로그에 그림 3과 같은 작업 관련 정보를 추출하는 스크립트를 등록하여 작업이 시작될 때 로그로 저장을 하고 있다.

```
if [ -d ${JOB_BACKUP2} ]; then
scontrol show jobid=${SLURM_JOB_ID} -dd > "${JOB_BACKUP2}/${SLURM_JOB_ID}.job"
set | grep SLURM >> "${JOB_BACKUP2}/${SLURM_JOB_ID}.job"
fi
```

(그림 3) SLURM 프로로그 작업 정보 추출 스크립트

그림 4는 작업 정보 추출 스크립트를 통해 저장된 작업의 정보이다. SLURMD_NODENAME 변수에 저장되어 있는 값(gpu12)을 통해 할당된 계산노드의 정보를 획득할 수 있고 SLURM_JOB_GPUS 변수에

저장되어있는 값(2, 3)을 통해 4개의 GPU가 장착된 계산노드에서 GPU Index 2, 3을 사용하였음을 확인할 수 있다. 그러면 작업이 시작된 시점부터 종료된 시점까지 gpu12에 장착된 GPU 2, 3의 사용률 통계를 수집하여 작업의 GPU 사용 통계 정보를 획득할 수 있다.

```
SLURMD_NODENAME=gpu12
SLURM_CLUSTER_NAME=neuron
SLURM_CONF=/etc/slurm/slurm.conf
SLURM_JOBID=138404
SLURM_JOB_CONSTRAINTS='(null)'
```

(그림 4) 작업 정보

4. GPU 사용률 수집 방안

기존에는 nvidia-smi 커맨드를 이용하여 5분 간격으로 사용률 로그를 저장한 후에 Perl 스크립트를 이용하여 사용률 데이터를 계산해내는 방식을 사용하였다[3]. 본 논문에서는 NVIDIA에서 제공하는 클러스터 관리 도구인 DCGM(Data Center GPU Manager)를 이용하여 사용률 획득하는 방안을 소개한다[5]. 그림5는 Neuron의 SLURM 프로로그에 등록된 DCGM 작업통계정보 저장 스크립트이다. 작업이 시작되는 시점에 이 스크립트가 수행이 되면서 DCGM이 작업통계정보를 수집하기 시작한다.

```
# DCGM job statistics - root
group=$(dcgmi group -c allgpus --default)
if [ $? -eq 0 ]; then
groupid=$(echo $group | awk '{print $10}')
dcgmi stats -g $groupid -e
dcgmi stats -g $groupid -s $SLURM_JOBID
fi
```

(그림 5) SLURM 프로로그 DCGM 작업통계정보 저장 스크립트

그림 6은 Neuron의 SLURM 에필로그에 등록되어 있는 DCGM 작업통계정보 저장 스크립트이다. 작업이 종료되는 시점에 이 스크립트가 수행이 되면서 DCGM이 작업통계정보 수집을 종료한다.

```
# DCGM job statistics - root
dcgmi stats -x $SLURM_JOBID
dcgmi stats -v -j $SLURM_JOBID | \
tee ${JOB_BACKUP}/${SLURM_JOBID}.dcgm
```

(그림 6) SLURM 에필로그 DCGM 작업통계정보 저장 스크립트

그림 7은 DCGM을 통해 수집된 GPU 사용률 데이터이다. GPU Index별로 사용률 데이터가 수집되며, 앞에서 추출한 작업별 할당된 GPU Index 정보를 사용하여 작업별 GPU 사용률 데이터 획득이 가능하다.

```

GPU ID: 1
----- Execution Stats -----
| Start Time           | Thu Jul 7 10:28:37 2022 |
| End Time             | Thu Jul 7 10:30:34 2022 |
| Total Execution Time (sec) | 117.77 |
| No. of Processes     | 8 |
----- Performance Stats -----
| Energy Consumed (Joules) | 19619 |
| Power Usage (Watts)      | Avg: 229.161, Max: 236.11, Min: 218.634 |
| Max GPU Memory Used (bytes) | 609222656 |
| SM Clock (MHz)          | Avg: 1410, Max: 1410, Min: 1410 |
| Memory Clock (MHz)      | Avg: 1215, Max: 1215, Min: 1215 |
| SM Utilization (%)      | Avg: 100, Max: 100, Min: 100 |
| Memory Utilization (%)  | Avg: 42, Max: 43, Min: 42 |
| PCIe Rx Bandwidth (megabytes) | Avg: N/A, Max: N/A, Min: N/A |
| PCIe Tx Bandwidth (megabytes) | Avg: N/A, Max: N/A, Min: N/A |
    
```

(그림 7) DCGM GPU 사용률

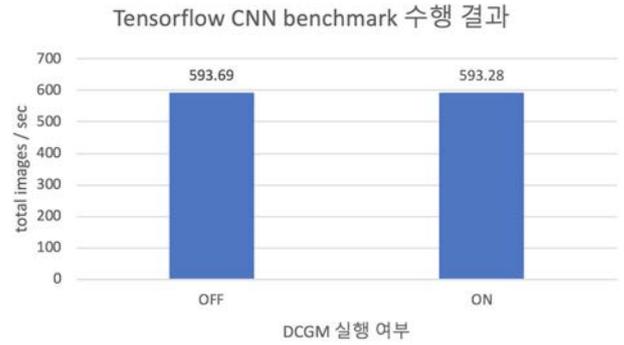
Neuron에서는 그림 8과 같이 Perl 스크립트를 사용하여 작업 정보 추출 스크립트에서 작업이 사용한 노드 정보와 GPU Index 정보를 획득하고 DCGM을 통해 저장된 작업통계정보를 이용하여 작업이 사용한 GPU 카드의 코어 사용률(SM Utilization)과 메모리 사용률(Memory Utilization)의 평균(Avg), 최대(Max), 최소(Min)값을 생성하여 SLURM 스케줄러의 과금 테이블에 포함시켜 사용자의 사용 패턴을 모니터링하고 있다.

```

for node in Nodes_used_by_job
  for gpu in GPUs_used_by_job
    accumulate GPU core/memory usage
    find max/min GPU core/memory usage
    calculate average GPU core/memory usage
    include results in the SLURM accounting table
    
```

(그림 8) Perl 스크립트 Pseudo 코드

DCGM은 low-overhead, non-invasive 모니터링 툴로서 계산노드에서 수행되는 사용자 작업에 큰 영향을 주지 않는다[5]. 그림 9와 같이 계산노드에서 Tensorflow CNN benchmark(Model:resnet50, Dataset:imagenet, Batch size:128, Num batches:200)를 수행하여 모니터링 툴의 오버헤드를 측정된 결과, 모니터링 툴의 실행과 관계없이 결과 값에 큰 차이를 보이지 않았다.



(그림 9) DCGM 실행에 따른 오버헤드 측정

4. 결론 및 향후 연구 방향

다수의 GPU 카드가 장착된 계산노드에서는 작업 배치스케줄러의 공유노드정책으로 서비스할 때 활용성을 극대화 시킬 수 있다. 본 논문에서는 SLURM 스케줄러의 프로로그/에필로그와 NVIDIA의 클러스터 관리 도구인 DCGM을 사용하여 공유노드 정책으로 단일노드에서 복수개의 작업이 수행되는 경우, 작업별 GPU 사용률을 획득하는 방안에 대하여 소개하였다. 향후에 축적된 작업별 GPU 사용률 데이터를 분석하여 사용자의 GPU 사용실태를 파악하고 이를 기반으로 앞으로 도입할 시스템의 CPU와 GPU의 탑재 비율을 결정하는 기초 데이터로 사용할 예정이다. 또한 본 논문에서 제안하는 기법을 사용하여 공유노드 정책에서 작업별 IO 통계 및 전력 사용량 정보 등의 다양한 정보를 획득하는 방안에 대한 연구들을 수행할 예정이다.

참고문헌

- [1] 한국과학기술정보연구원 국가슈퍼컴퓨팅센터, 보유자원 뉴론, <https://www.ksc.re.kr/ggspcpt/neuron>
- [2] 권민우, 윤준원, 홍태영 “클러스터 시스템의 계산자원 활용률을 극대화하기 위한 작업배치스케줄러의 공유노드 정책 적용 방안 연구” ASK 2022 학술발표대회, 4-7, 2022.
- [3] 권민우, 윤준원, 홍태영 “클러스터 시스템에서 GPU 사용 통계정보 획득 방안에 대한 연구” 2018년 추계학술발표대회, 476-477, 2018.
- [4] SLURM Workload Manager 21.08, scontrol, <https://slurm.schedmd.com/scontrol.html>
- [5] NVIDIA, DCGM, <https://developer.nvidia.com/dcgm>