클러스터 시스템의 계산자원 활용률을 극대화하기 위한 작업배치스케줄러의 공유노드 정책 적용 방안 연구

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

A study on the application of the shared node policy of the job batch scheduler to maximize the utilization rate of computational resources of cluster system

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

요 약

작업배치스케줄러는 다수의 사용자에게 클러스터 시스템의 계산 자원을 효과적으로 제공하는 유용한 시스템 소프트웨어이다. 한국과학기술정보연구원에서는 작업배치스케줄러인 PBS와 SLURM을 이용하여 슈퍼컴퓨터 5호기 메인시스템인 누리온과 뉴론을 각각 공동활용서비스하고 있다. 본 논문에서는 뉴론의 제한된 계산자원을 다수의 연구자들에게 효율적으로 서비스하기 위해 SLURM 작업배치스케줄러의 공유노드 정책을 적용하는 방안과 작업통계 분석 기법을 소개한다.

1. 서론

한국과학기술정보연구원에서는 빅데이터 분석, Machine Learning, Deep Learning과 같은 인공지능 기술을 활용한 연구를 가속화하는데 최적화된 GPU를 장착한 고성능 공동활용서비스 인프라인 뉴론(Neuron)을 운영하고 있다. 뉴론은 59대의 서버와 200개의 GPU로 구성된 시스템으로 8,437대로 구성된 메인시스템인 누리온에 비해 제한적인 규모를 가지고 있다[1]. 본 논문에서는 뉴론과 같이 제한된 계산자원을 다수의 연구자들에게 효율적으로 서비스하기 위해 SLURM 작업배치스케줄러의 공유노드정책을 적용하고 작업통계 분석을 통해 시스템을 모니터링하는 기법을 소개한다.

2. 뉴론 시스템

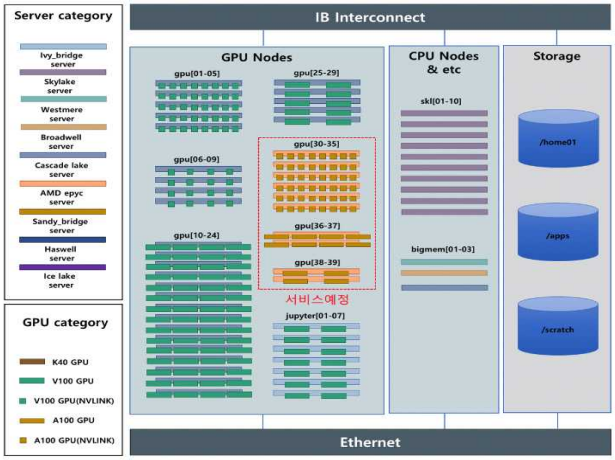
뉴론은 GPU 기반 시스템으로 슈퍼컴퓨터 5호기 메인 시스템인 누리온이 Intel Knight Landing 기반의 시스템으로 구축됨에 따라 사용자의 다양한 수요에 대응하기 위한 목적으로 운영되고 있다. 5년 단위로 구축이 되는 슈퍼컴퓨터 메인 플래그십 시스템과 달리 아래 표와 같이 사용자의 수요와 선호도를 고려하여 연차별로 증설이 이루어지고 있다. 2019년 7월 공식 서비스 이후, 노후 장비 퇴역과 신규 장비 증설 과정을 걸쳐 현재 이론성능 2.34PF 규모로 서비스되고 있다[1].

<표 1> 뉴론 시스템 연도별 규모

	2019	2020	2021
서버	63대	78대	59대
GPU	99개	163개	200개
성능	698.8TF	1.24PF	2.34PF

그림 1은 뉴론 시스템의 구성도를 보여준다. Jupyter 노트북을 이용한 수월한 인공지능 코드 개발 및 디버깅을 위한 Jupyter 서버 7대와 V100 GPU NVLink 타입 8개/4개 장착 서버 각각 5대/4대, PCIe 타입 4개/2개 장착 서버 각각 15개/5대로 구성되어 있으며, A100 GPU는 NVLink 타입 8개 장착 서버 6대, PCIe 타입 4개/2개 장착 서버 각각

2대로 구성되어 있다[1]. 뉴론 시스템은 SLURM 작업배치스케줄러를 이용하여 사용자가 요구하는 만큼 계산자원을 할당하는 방식으로 서비스되고 있다[2].



(그림 1) 뉴론 시스템 구성도

2. SLURM의 공유 노드 정책 적용

SLURM 작업배치스케줄러의 계산자원 운영 정책은 아래 표와 같이 2가지가 있다. 배타적(Exclusive) 노드정책은 한 개의 서버에서 하나의 작업이 모든 계산자원을 독점적으로 사용할 수 있는 방식으로 서버의 계산자원에 최적화된 작업이 아닌 경우에 자원이 낭비될 수 있다. 공유(Shared)노드정책은 여러 개의 작업이 하나의 서버에서 수행될 수 있으며, 제한된 계산자원을 다수의 사용자가 효율적으로 사용할 수 있다. 공유 노드 정책 적용 시에 사용자의 작업 간에 계산 자원에 대한 침해가 발생하지 않도록 리눅스 cgroup 기능을 통한 격리가 필요하다. SLURM 스케줄러는 이를 위해 두 가지 SelectType을 제공하는데, select/cons_res의 경우 CPU와 Memory만을 제한/격리하고 select/con_tres의 경우 GPU까지 제한/격리해주는 기능을 제공한다[3].

<표 2> SLURM 계산자원 운영 정책

Policy	Setting
Exclusive	SelectType=select/linear
Shared	SelectType=select/cons_res
	SelectType=select/cons_tres

그림 2는 계산노드에서 동작하는 slurmd 데몬이 구동 시에 참조하는 cgroup 설정 파일로 CPU/Memory/GPU에 대한 cgroup 기능을 ConstrainCores/ConstrainRAMSpace/ConstrainDevices 파라미터를 이용하여 Enable/Disable시킬 수 있다.

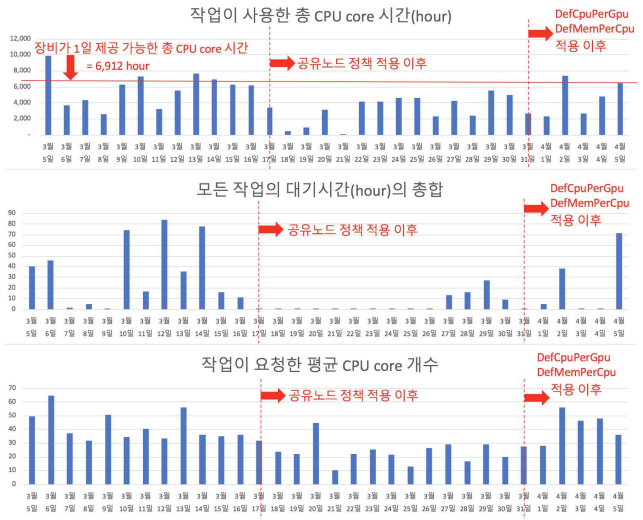
```
###
#
# Slurm cgroup support configuration file
#
# See man slurm.conf and man cgroup.conf for further
# information on cgroup configuration parameters
#--
CgroupAutomount=yes

ConstrainCores=yes
ConstrainRAMSpace=yes
ConstrainDevices=yes
```

(그림 2) 계산노드 cgroup.conf 파일

뉴론 시스템은 시스템 소프트웨어의 업그레이드나 인프라 자원의 장애 교체를 위해 정기적인 예방점검을 수행하고 있다. 2022년 3월 17일에 수행된 예방점검 이전에는 배타적노드정책을 사용하고 있었으나 예방점검 이후 공유노드정책인 “SelectType=select/cons_tres”로 변경하였다. 그림 3은 예방점검 전후인 3월 5일 ~ 4월 5일까지 V100 NVLink 타입 GPU가 탑재된 계산노드 8대에서 수행된 472개의 작업에 대한 통계를 분석한 결과이다.

<표 3> 계산노드 8대 1일 제공 가능 core 시간

노드수	core수	GPU수	1일 제공 가능 core 시간
4	32	8	3,072
4	40	4	3,840
합계			6,912



(그림 3) 뉴론 시스템 예방점검 전후 작업 통계 분석

예방점검을 수행한 3월 17일 이후, 모든 작업의 대기시간의 총합이 급격히 감소한 것을 확인할 수 있다. 이는 작업이 사용한 총 CPU core 시간이 계산노드 8대가 제공 가능한 6,912시간에 미치지 못하면서, 작업이 즉시 처리되었기 때문입니다. 이는 작업이 요청한 평균 CPU core 개수가 예방점검 이전보다 감소한 것에서 원인을 찾을 수 있다. 배타적 노드 정책으로 운영될 때는 하나의 작업이 계산노드의 전체 계산자원을 독점적으로 점유하기에 작업이 요청한 CPU core 개수가 계산노드의 총 CPU core 개수와 동일하다. 반면에 공유 노드 정책으로 변경된 이후에는 다수의 작업이 하나의 계산노드에서 수행될 수 있으므로 작업이 요청한 CPU core 개수는 사용자가 작업 제출 시에 요청한 CPU core 개수가 된다. 이에 따라 공유노드정책을 적용함으로 제한적인 계산자원을 최소의 대기시간으로 사용가능한 효율적인 서비스 환경이 조성된 것이다.

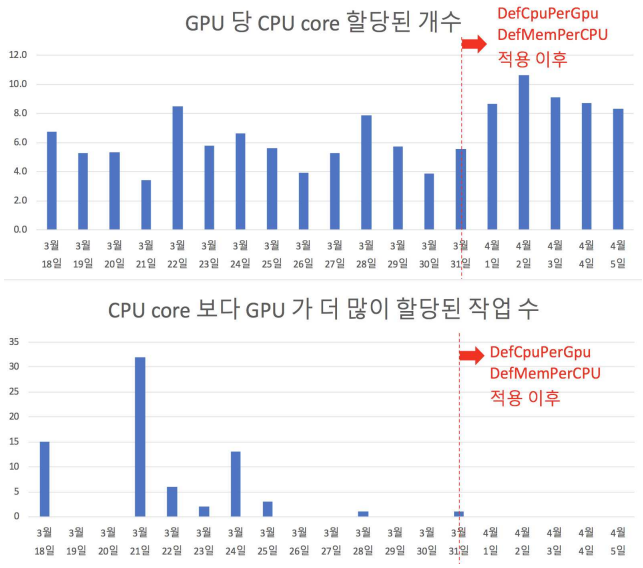
3. 사용자 작업의 성능 보장을 위한 스케줄러 추가 설정

SLURM 작업배치스케줄러가 공유노드정책으로 변경되면서 사용자는 서버 내의 계산자원을 정확하게 파악하고 요청해야 하는 책임을 갖게 되었다. 아래 표에서는 작업제출 시에 sbatch 명령어를 사용하여 계산자원을 요청하기 위한 SLURM 작업배치스케줄러의 옵션을 보여준다[4].

<표 4> SLURM sbatch 옵션

옵션	설명
--nodes=*N*	작업을 수행할 노드 수
--ntasks-per-node=*N*	노드당 수행될 프로세스 수
--cpus-per-task=*N*	프로세스 당 할당될 core 수
--gres=gpu:*N*	작업에 할당할 GPU 수
--cpus-per-gpu=*N*	GPU 당 할당될 core 수
--mem=*N*	작업에 할당될 memory(MB)

배타적노드정책으로 시스템이 운영될 때는 한 작업이 하나의 노드에 있는 계산자원을 점유하므로 CPU core/GPU 수, 메모리량에 대한 요청을 할 필요가 없이 노드 수(nodes)와 프로세스 수(ntasks-per-node)만을 요청하면 되었다. 그러나 공유노드정책에서는 노드 수와 프로세스 수만을 요청하게 되면, 프로세스 당 할당될 core 수(cpus-per-task)가 자동으로 '1'로 할당이 되면서, 결과적으로 1 core에서 사용자 작업이 수행 되어 사용자는 동일한 작업을 배타적노드정책에서 수행했을 때보다 느려지는 결과를 얻게 된다. 그림 4를 보면, 예방정비 이후 작업 통계를 분석하던 중에, CPU core보다 GPU가 더 많이 할당된 작업들이 발견되었다. 이러한 작업들의 성능 저하를 해결하기 위해 SLURM 작업배치스케줄러의 설정 파일인 slurm.conf에 DefCpuPerGpu와 DefMemPerCpu를 추가하였다[5]. DefCpuPerGpu는 사용자가 프로세스 당 할당될 core 수(cpus-per-task)나 GPU당 할당될 core 수(cpus-per-gpu)를 정의하지 않는 경우, 이 두 개의 파라미터에 대한 기본값을 스케줄러가 할당해주는 설정이다. DefMemPerCpu는 사용자가 작업에 할당될 메모리량(mem)을 정의하지 않는 경우, 기본값을 스케줄러가 할당해주는 설정이다. 그림 4에서 볼 수 있듯이 DefCpuPerGpu 설정에 의해 GPU당 CPU core에 할당된 개수가 증가하였으며, 이에 따라 성능저하를 유발할 수 있는 CPU core보다 GPU가 더 많이 할당된 작업은 사라진 것을 확인할 수 있다. 물론 그림 3에서 볼 수 있듯이 작업이 요청한 평균 CPU core 개수가 DefCpuPerGpu 설정 적용 이후 증가하여, 작업이 사용한 총 CPU core 시간이 증가하면서 동시에 대기 시간도 상승하였다. 한국과학기술정보연구원에서는 사용자가 최적의 작업제출옵션을 사용할 수 있도록 지침서를 통해 다양한 상황의 작업제출옵션을 예시로 제공하고 있다[2]. 따라서 동일한 작업량이 시스템에서 수행된다는 가정 하에, 지침서를 충분히 숙지한 사용자가 최적의 작업제출옵션을 사용함에 따라 대기시간은 점점 개선되고 시스템의 활용률은 올라갈 것으로 예상된다.



(그림 4) 작업별 할당된 CPU core 및 GPU 개수

3. 결론 및 향후 연구 방향

뉴론 시스템과 같이 제한된 규모에서 다수의 연구자들에게 공동활용서비스를 수행하는 클러스터 장비의 경우 작업배치스케줄러의 공유노드정책을 적용하여 운영할 때 서버 단위에서 다수의 작업을 동시에 수행할 수 있어 시스템의 활용성을 높일 수 있다. 반면에 사용자가 작업제출 시에 서버 내의 계산자원을 정확하게 파악하고 요청해야 하는 책임을 갖게 되었으며, 그렇지 않으면 성능저하를 유발할 수 있다. 공유노드정책에서는 본 논문에서 분석한 사례인 GPU 당 CPU core 할당 개수 뿐 아니라 CPU core/GPU/Infiniband 네트워크 포트의 affinity와 같은 이슈들이 사용자 작업의 성능저하를 유발시킬 수 있다. 향후에 다양한 작업 환경에서 사용자의 작업 성능을 최적화하기 위해 작업배치스케줄러와 같은 시스템 소프트웨어의 설정을 고도화하고 계산자원을 공유노드정책에 맞게 튜닝하는 연구들을 수행할 예정이다.

참고문헌

[1] 한국과학기술정보연구원 국가슈퍼컴퓨팅센터, 보유자원 뉴론, https://www.ksc.re.kr/ggspcpt/neuron
[2] 한국과학기술정보연구원 국가슈퍼컴퓨팅센터, 뉴론 사용자 지침서, https://www.ksc.re.kr/gsjw/jcs/hd
[3] SLURM Workload Manager 21.08, Sharing Consumable Resources, https://slurm.schedmd.com/cons_res_share.html
[4] SLURM Workload Manager 21.08, sbatch, https://slurm.schedmd.com/sbatch.html
[5] SLURM Workload Manager 21.08, slurm.conf, https://slurm.schedmd.com/slurm.conf.html