작업관리 소프트웨어의 스케줄링 정책을 이용한 클러스터 시스템의 공정한 작업 실행 우선순위 관리 방안 연구

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

A study on fair job priority management methods of cluster system using scheduling policy of resource manager

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

요 약

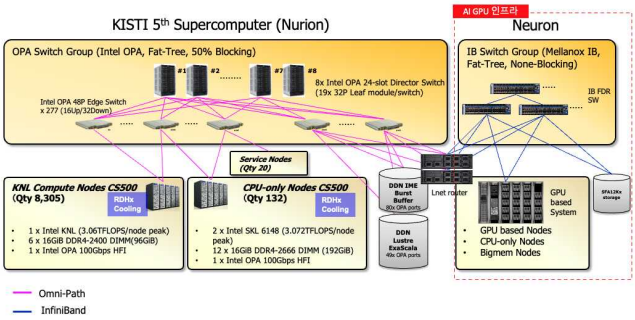
한국과학기술정보연구원(KISTI)의 슈퍼컴퓨터 보조시스템인 Neuron은 이기종 가속기인 GPU가 탑재된 클러스터 시스템으로 작업관리 소프트웨어인 SLURM을 통해 국내 연구자들에게 서비스되고 있다. 본 논문에서는 SLURM 작업관리 소프트웨어의 작업 스케줄링 정책을 이용하여 연구자들이 제출하는 복수 개의 대기작업을 공정하게 처리하는 방안에 대해서 소개한다.

1. 서론

최근 AI 기술을 활용한 연구가 활발히 수행되면서 AI 소프트웨어의 가속 성능이 뛰어난 GPU 기반 서버 수요가 급격히 늘어나고 있다. 국내 주요 대학 및 AI 대학원의 경우, 자체적인 GPU 기반 클러스터를 구축하여 운영 중이지만, 그 밖의 연구실에서는 고가의 장비 구축비, 운영 인력 및 유지보수비 확보의 부담으로 인해 자체 클러스터 운영에 어려움을 겪고 있다. KISTI에서는 국내 연구자들의 GPU 기반 서버 수요에 대응하기 위해 GPU 장착 서버가 다수 탑재된 클러스터 시스템인 Neuron을 슈퍼컴퓨터의 보조시스템으로 운영하고 있다[1]. 본 논문에서는 Neuron에서 사용 중인 작업관리 소프트웨어 SLURM의 작업 스케줄링 정책을 이용하여 다수의 연구자들이 제출하는 복수 개의 대기작업을 공정하게 처리하기 위한 기법을 소개한다.

2. 슈퍼컴퓨터 보조 시스템 Neuron

Neuron 시스템은 슈퍼컴퓨터인 Nurion의 보조시스템으로 그림 1과 같이 Lnet 라우터를 이용해 연결되어 있다. Nurion과 Neuron 시스템은 동일한 LDAP 서버를 사용하고 있어서, 슈퍼컴퓨터 계정을 하나 발급받게 되면, Nurion과 Neuron 시스템을 동시에 사용할 수 있다. 사용자는 필요에 따라 Intel Xeon x86 기반의 시스템인 Nurion을 사용할 수도 있고, GPU가 장착되어 있는 Neuron도 사용할 수 있다.



(그림 1) KISTI 슈퍼컴퓨터 구성도

표1은 Neuron 시스템에 포함되어 있는 GPU 장착 서버의 구성을 보여준다. Neuron 시스템에는 크게 NVIDIA V100과 A100이 장착된 서버가 있으며, V100 8, 4, 2 GPU 장착 서버는 각각 5, 19, 12대로 총 36대이며, A100 8, 4 GPU 장착 서버는 각각 14, 2대로 총 16대이다. 그 밖에 AI 데이터 전후 처리를

위해 CPU만 장착된 서버 10대와 대용량 메모리가 탑재된 서버 3대가 함께 운영되고 있다[1].

<표 1> Neuron GPU 장착 서버

V100 장착 서버		A100 장착 서버	
GPU 개수	서버 수	GPU 개수	서버 수
8	5	8	14
4	19		
2	12	4	2
합계	36	합계	16

2. 작업관리 소프트웨어 SLURM의 스케줄링 정책

Neuron에서 사용 중인 작업관리 소프트웨어인 SLURM은 다수의 사용자가 제출하는 복수 개의 작업을 효율적으로 처리하기 위해 Muti-factor기반의 스케줄링 정책을 제공하고 있다. 그림 2는 대기 작업의 실행 우선순위(Job_priority)를 결정하기 위한 계산수식을 보여준다. 대기 작업의 Job_priority가 높을수록 먼저 자원을 할당받게 된다[2].

```
Job_priority =
    site_factor +
    (PriorityWeightAge) * (age_factor) +
    (PriorityWeightAssoc) * (assoc_factor) +
    (PriorityWeightFairshare) * (fair-share_factor) +
    (PriorityWeightJobSize) * (job_size_factor) +
    (PriorityWeightPartition) * (partition_factor) +
    (PriorityWeightQOS) * (QOS_factor) +
    SUM(TRES_weight_cpu * TRES_factor_cpu,
        TRES_weight_<type> * TRES_factor_<type>,
        ...)
    - nice_factor
```

(그림 2) SLURM Job_priority Formula

<표 2> Job_priority factors

Factor	설명
site_factor	관리자 정의값
age_factor	대기시간에 대한 factor
fair-share_factor	fair-share에 따른 factor
job_size_factor	요청자원 크기에 따른 factor
partition_factor	요청 partition에 따른 factor
QOS_factor	partition/user의 QOS factor
nice_factor	사용자 정의값

표 2는 Job_priority를 계산하기 위한 factor의 구성을 보여준다. 이 factor값은 모두 0~1사이의 값을 갖는다. 그림 3과 같이 Neuron에서는 대기시간에 따른 age_factor와 공정한 자원 분배를 위한 fair-share_factor를 사용하여 Job_priority를 계산한다.

```
Job_priority =
    10,000 * (age_factor) +
    3,000 * (fair-share_factor) +
```

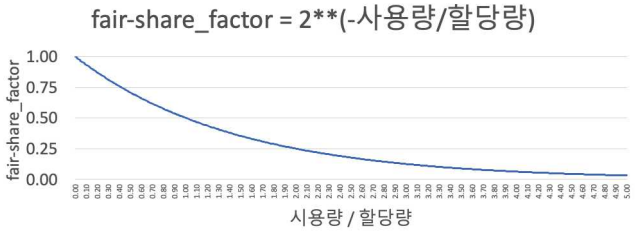
(그림 3) SLURM Job_priority Formula of Neuron

age_factor는 PriorityMaxAge라는 설정값을 기반으로 factor값이 계산된다. PriorityMaxAge는 대기시간을 최대 몇일까지 계산할 것인지를 결정하는 값으로 'PriorityMaxAge=7-0'으로 설정시, 작업이 제출된 직후에는 '0'값을 갖게 되고, 대기시간이 점점 늘어나 작업이 7일을 대기하게 되면 age_factor는 '1'이 된다. 그림 2와 같이 PriorityWeightAge값이 곱해져서 Job_priority가 결정되게 되는데, Neuron 시스템에서는 PriorityWeightAge가 '10,000'으로 설정되어 있어 7일이상 대기한 작업에 대해서는 Job_prioryty가 '1*10,000=10,000'이 되게 된다. 7일은 분으로 환산하면 10,080분으로 대략 1분에 Job_priority가 1씩 증가하게 된다[3].

fair-share factor는 Neuron 시스템의 공정한 자원 분배를 가능하게 해주는 값으로서, SLURM 스케줄러에 등록된 사용자의 수에 따라 할당량이 결정되고 할당량 대비 사용자의 실제 사용률의 비율로 계산이 된다. fair-share factor는 그림 3과 같은 수식을 통해 계산이 된다.

$$\text{fair-share\_factor} = 2^{(-\text{사용량} \ / \ \text{할당량})}$$

(그림 4) SLURM fair-share_factor Formula



(그림 5) fair-share_factor 그래프

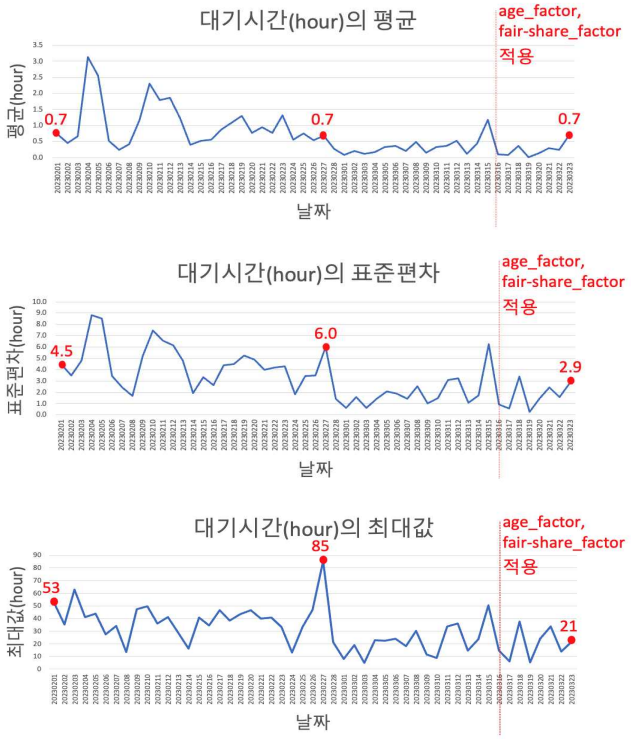
그림 5와 같이 2의 지수함수에서 입력값이 항상 음수값이기 때문에 factor값은 어떤 경우에서든 0~1까지의 값을 갖게 된다. 사용자의 할당량이 동일할 경우, 결국 사용자 본인의 사용량에 따라 fair-share

_factor가 결정되며 자원을 많이 사용할수록 낮은 값을 갖게 된다[4].

Neuron에서는 PriorityWeightFairshare의 값을 3,000으로 설정하였다. 이는 age_factor와의 관계를 고려하여 설정한 값으로서, fair-share_factor가 '1'인 경우, 즉 Neuron은 한번도 사용하지 않은 사용자(사용량=0)의 작업이 제출된 시점에 Job_priority는 fair-share_factor에 따라 3,000이 되게 되는데, 이는 PriorityWeightAge*age_factor가 3,000을 넘는 작업, 즉 대략 2일(분으로 환산시 2,880분) 이상 대기한 작업보다 높은 우선순위를 갖지 않도록 하기 위함이다. 다시 말해, 사용량이 높은 사용자의 작업이 2일 이상 대기한 경우에 fair-share_factor로 인해 더 이상 작업이 후순위로 밀리지 않도록 설정을 한 것이다. 이를 통해 Fair-share 스케줄링을 사용 시에 사용량이 높은 사용자의 작업이 오랜시간 자원을 할당받지 못하는 상태(Starving Job)를 방지하였다.

3. 대기시간 통계분석을 통한 스케줄링 정책 검증

그림 6은 Neuron의 일자별 작업 대기시간의 통계를 보여준다. 2023년 3월 16일 이전에는 fair-share_factor만을 이용하여 스케줄링 정책을 운영했으며, 이후에는 위 단락에서 언급한 age_factor와 fair-share_factor를 동시에 적용하여 운영하고 있다.



(그림 6) Neuron 일자별 작업 대기시간 통계

첫 번째 그래프인 대기시간(hour)의 평균을 보면, 2월 1일, 2월 27일, 3월 23일에 동일한 대기시간의 평균(0.7시간)이 나온 반면에, 두 번째 그래프의 표준편차는 각각 4.5, 6.0, 2.9시간으로 나타났고, 세 번째 그래프의 최대값은 각각 53, 85, 21시간으로 나타났다. 이 결과를 토대로 age_factor를 적용한 이후, 작업마다 균등한 대기시간을 가지게 되었음을 확인할 수 있다. 이를 통해 fair-share_factor 값이 낮은, 즉 시스템의 사용률이 높은 사용자의 작업이 지나치게 오래 기다려야 하는 상황(Starving Job)을 방지할 수 있게 되었다.

4. 결론 및 향후 연구 방향

본 논문에서는 Neuron에서 사용 중인 작업관리 소프트웨어인 SLURM의 Job_priority factor인 age_factor와 fair-share factor를 적용하여 다수의 연구자들이 제출하는 복수 개의 대기작업을 공정하게 처리하기 위한 기법에 대해 소개하였다. 기존에 fair-share factor만을 적용했을 때 Starving Job이 발생할 가능성이 있었으나 age_factor를 함께 적용함으로써 작업마다 비교적 균등한 대기시간을 가지게 되었음이 확인되었다. 앞에서 언급한 것과 같이 Neuron은 슈퍼컴퓨터에 비해 작은 규모의 클러스터 시스템이다. 향후에 구축될 슈퍼컴퓨터 6호기 역시 GPU 기반의 시스템이 구축될 예정으로 보다 큰 규모 클러스터 시스템에서의 다양한 스케줄링 정책을 적용하고 고도화시키는 연구를 수행할 예정이다.

참고문헌

[1] KISTI 국가슈퍼컴퓨팅센터 홈페이지 보유자원, https://www.ksc.re.kr/ggspcpt

[2] SLURM Manual Site, Multifactor Priority Plugin, https://slurm.schedmd.com/priority_multifactor.html

[3] SLURM Manual Site, slurm.conf, https://slurm.schedmd.com/slurm.conf.html

[4] SLURM Manual Site, Classic Fairshare Algorithm, https://slurm.schedmd.com/classic_fair_share.html