슈퍼컴퓨터 5호기 사용자의 작업별 IO 통계정보 획득 방안에 대한 연구

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

A study on the method of acquiring IO statistical information for each user task of the KISTI-5 supercomputer

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

요 약

슈퍼컴퓨터 5호기 누리온은 8,437대의 계산노드와 33.88PB 규모의 병렬스토리지가 100Gbps의 Omni-Path(OPA) 인터커넥트로 연결되어 있는 초대형 클러스터 시스템이다. 누리온의 계산자원은 PBS 작업스케줄러를 통해 관리되고 있고 병렬 스토리지는 DDN사의 Exascaler Monitoring System(ESMON)을 통해 influxDB에 read/write IO 통계 데이터를 수집하고 있다. 본 논문에서는 PBS의 과금 데이터와 ESMON influxDB의 IO 통계 데이터를 활용하여 사용자의 작업별 IO 통계 정보를 생성하는 방안에 대하여 소개한다.

1. 서론

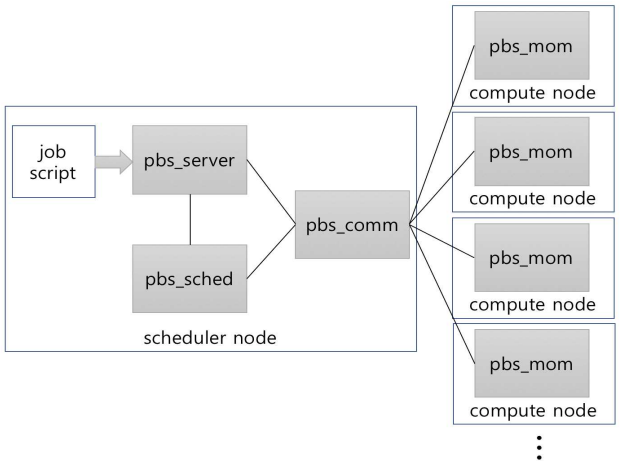
한국과학기술정보연구원에서 운영 중인 슈퍼컴퓨터 5호기 누리온은 Intel Xeon Phi 7250 (Knights Landing 계열) CPU가 장착된 계산노드 8,305대와 6148 (Skylake 계열) CPU가 장착된 계산노드 132대로 구성되어 있다. 모든 계산노드에는 33.88PB의 Lustre 병렬스토리지가 100Gbps 광대역 인터커넥트인 OPA를 통해 연결되어 있다. Lustre 병렬스토리지에는 사용자별로 홈디렉터리(/home01)와 스크래치(/scratch)가 존재하여 동일한 입출력 파일을 가지고 다수의 노드를 이용한 초병렬 프로그램의 수행이 가능하다. 사용자는 로그인 노드에 접속하여 PBS 작업스케줄러를 통해 계산 자원을 할당받아 자신의 작업을 수행할 수가 있다[1]. 이와 같은 방식으로 2018년 12월 정식 개통한 이후 현재까지 대략 660만개의 작업이 수행되었으며 수행된 작업에 대한 과금 데이터는 PBS 작업스케줄러의 DB에 저장되어 있다. Lustre 병렬스토리지는 DDN사의 Exascaler Monitoring System(ESMON)이라는 소프트웨어를 이용하여 실시간 read/write IO 통계 데이터를 수집하여 influxDB에 축적하고 있다.

본 논문에서는 PBS 스케줄러의 DB에 저장되어 있는 사용자 작업의 과금 데이터와 ESMON influxDB에 저장되어 있는 IO 통계 데이터를 활용하여 사용자의 작업별 IO 통계 정보를 생성하는 방안에 대하여 소개한다.

2. PBS 작업스케줄러의 과금데이터

그림1은 PBS 스케줄러의 서비스 데몬 구성도를 보여준다[2]. Server 데몬(pbs_server)은 로그인노드에 접속한 사용자로부터 그림2와 같은 작업스크립트를 입력받아 요청받은 계산 자원을 할당해준다[3]. 그림2의 작업스크립트는 MPI 라이브러리를 이용하여 개발된 멀티노드 작업을 4대의 노드를 점유해서 노드당 64프로세스(총 256 프로세스)를 이용해 수행하는 작업스크립트이다. Scheduler 데몬(pbs_sched)은 free 상태의 노드가 4대 생길 때까지 해당 사용자의 작업을 'Q'(대기) 상태에 두었다가 자원이 확보되면, 사용자의 작업을 'R'(실행) 상태로 전환시킨다. Communication 데몬(pbs_comm)은 계산노드에서 동작하고 있는 Mom 데몬(pbs_mom)과 통신하여 작업이 수행되는 동안의 과금정보(CPU사용시간, 메모리 사용량 등)를 수집한다. 작업이 종료되면 수집된

과금데이터를 Server 데몬이 DB에 저장한다.



(그림 1) PBS 스케줄러 서비스 데몬 구성도

```
#!/bin/sh
#PBS -N IntelMPI_job
#PBS -V
#PBS -q normal
#PBS -A {PBS 옵션 이름} # Application별 PBS 옵션 이름표 참고
#PBS -l select=4:ncpus=64:mpiprocs=64
#PBS -l walltime=04:00:00

cd $PBS_O_WORKDIR

module purge
module load craype-mic-knl intel/18.0.3 impi/18.0.3

mpirun ./test_mpi.exe
```

(그림 2) 사용자 작업스크립트

3. ESMON influxDB의 IO 통계데이터

DDN사의 ESMON은 성능 모니터링 및 분석을 위해 Lustre 및 기타 파일 시스템의 시스템 통계를 수집할 수 있는 오픈 소스 기반의 모니터링 소프트웨어이다. 그림 3은 ESMON의 influxDB에서 실시간으로 수집하고 있는 누리온 Lustre 파일시스템의 read/write IO 통계 데이터를 보여준다. 그래프를 통해 Lustre의 OST별로 IO Throughput의 통계를 확인할 수 있다. 또한 ESMON의 influxDB에서는 4분 간격으로 작업별 IO 통계 정보를 수집하고 있다. 본 논문에서는 PBS 스케줄러에서 수집 불가능한 작업별 IO 통계 정보를 ESMON의 influxDB에 저장되어 있는 통계 데이터를 이용하여 추가하는 기능을 구현하였다.

Read Throughput per OST

Metric	Avg	Max	Current
OST0000	7.63 MB/s	33.29 MB/s	5.55 MB/s
OST0001	6.74 MB/s	52.49 MB/s	6.36 MB/s
OST0002	7.33 MB/s	32.65 MB/s	11.84 MB/s
OST0003	6.37 MB/s	34.98 MB/s	2.10 MB/s
OST0004	5.34 MB/s	24.35 MB/s	10.96 MB/s
OST0005	5.04 MB/s	18.50 MB/s	4.02 MB/s
OST0006	5.80 MB/s	28.96 MB/s	1.78 MB/s

1 2 3 4

Write Throughput per OST

Metric	Avg	Max	Current
OST0006	11.49 MB/s	119.35 MB/s	119.35 MB/s
OST008f	10.02 MB/s	110.25 MB/s	110.25 MB/s
OST0085	9.72 MB/s	104.01 MB/s	104.01 MB/s
OST0076	10.72 MB/s	103.52 MB/s	103.52 MB/s
OST0086	8.79 MB/s	102.56 MB/s	102.56 MB/s
OST008e	10.19 MB/s	99.92 MB/s	99.92 MB/s
OST00f2	14.84 MB/s	101.69 MB/s	96.42 MB/s

1 2 3 4

(그림 3) ESMON IO 통계데이터

4. 펄스크립트를 이용한 작업별 IO 통계데이터 생성

ESMON의 influxDB는 influx라는 커맨드를 이용하여 SQL 쿼리를 입력받아 DB에 저장되어 있는 정보를 획득할 수 있게 해준다. 그림 4는 리눅스의 cron 데몬을 이용하여 전날에 수행된 모든 작업의 read IO 통계 정보를 수집한 raw 파일을 보여준다.

```
8863935,20210906235200,307
8863935,20210906235600,123392
8863936,20210906234800,7571792
8863936,20210906235200,307
8863936,20210906235600,123575
8863937,20210906234800,8136842
8863937,20210906235200,443
8863937,20210906235600,123404
8863938,20210906234800,6557280
8863938,20210906235200,307
8863938,20210906235600,123457
8863939,20210906234800,8387336
```

(그림 4) ESMON IO 통계데이터

표 1은 PBS 과금정보에 저장되어 있는 데이터를 보여준다. PBS 뿐만 아니라 유사한 다른 작업 스케줄러(SLURM 등)에서도 동일한 형태의 과금 정보를 제공하고 있다[4].

<표 1> PBS 과금 정보

이름	내용
user	사용자
jobid	작업 아이디
jobname	작업 이름
start_time	작업 시작 시간
end_time	작업 종료 시간

본 논문에서 소개하는 기법과 유사하게 SLURM 스케줄러의 과금 정보를 이용하여 GPU 카드의 사용 통계 정보와 작업별 전력 정보를 수집하는 연구가 이미 수행되었다[5,6]. 그림 5는 펄스크립트를 이용하여 작업별 평균/최대 read/write IO 통계 데이터를 PBS 과금 테이블에 포함시키는 기능을 구현한 Pseudo 코드를 보여준다. 그림 4에서 수집된 일일 IO 통계 파일을 열어서 작업이 시작한 시간과 종료된 시간 사이에 데이터의 평균과 최대값을 구하여 PBS의 작업별 과금 데이터에 포함시킴으로 작업별 IO 통계 데이터를 수집할 수 있다.

```
for from start_time to end_time
  open read/write IO file
  if start_time <= time_stamp <= endtime then
    accumulate read/write IO value
    find maximum read/write IO value
calculate average read/write IO value
include average/max read/write IO value in the PBS
accounting table
```

(그림 5) 펄스크립트 Pseudo 코드

그림 6은 PBS의 작업별 데이터에 새롭게 추가된 IO 통계 정보 결과를 보여준다.

MEAN_RIO_BS	MAX_RIO_BS	MEAN_WIO_BS	MAX_WIO_BS
15,215,357	42,802,865	21,635,069	52,653,105
15,265,968	42,812,059	21,789,080	43,491,600
16,491,313	42,789,415	21,967,804	34,922,135
15,238,232	42,798,220	21,689,393	43,797,752
22,441,968	42,810,529	32,712,553	32,933,706
15,353,377	42,818,264	21,703,746	41,547,803
11,466,478	42,825,522	16,261,639	34,505,706
14,653,319	41,299,833	20,891,999	48,831,104
15,077,560	42,806,184	21,447,688	42,515,807
22,884,770	42,814,481	32,459,651	34,121,998
22,901,429	42,813,543	32,286,340	34,143,184
15,224,983	42,806,610	21,850,934	43,664,424
15,299,637	42,811,559	21,498,273	42,877,970
15,841,089	42,811,962	22,042,844	47,030,001
15,306,592	42,814,271	21,710,393	51,189,518
15,191,427	44,993,093	21,664,537	62,291,185
9,196,906	42,799,477	13,145,757	44,335,803

(그림 6) PBS 과금테이블에 포함된 IO 통계 데이터

5. 결론 및 향후 연구 방향

작업별 IO 통계 정보는 차후 시스템 도입 시에 병렬스토리지의 스펙을 결정하는 기초자료로 활용이 될 수 있다. 또한 사용자 작업의 과도한 IO는 누리온에 설치된 OPA 네트워크와 같은 인터커넥트의 부하를 유발시키는데, 이는 전체적인 사용자 작업의 성능 저하를 일으킬 수 있다. 향후 본 논문에서 생성된 작업별 IO 통계 정보를 분석하여 PBS 작업 스케줄링에 반영시킴으로 인터커넥트의 부하분산을 유도하는 연구들이 수행될 예정이다.

참고문헌

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

[2] PBS Professional 2020.1 Installation and Update Guide, Altair, https://www.altair.com/pbs-works-documentation/

[3] 누리온 사용자 지침서, KISTI 국가슈퍼컴퓨팅센터 홈페이지, https://www.ksc.re.kr/gsjw/jcs/hd

[4] slurm workload manager, sacct command manual, https://slurm.schedmd.com/sacct.html

[5] 권민우, 윤준원, 홍태영 “클러스터 시스템에서 GPU 사용 통계정보 획득 방안에 대한 연구” 2018년 추계학술발표대회, 476-477, 2018.

[6] 권민우, 고동건, 윤준원, 홍태영 “공동 활용 컴퓨팅 시스템에서의 사용자 작업별 전력 사용량 생성 기능 구현” 2020년 추계학술발표대회, 24-25, 2020.