응용 프로그램 특성 분석 기반 스케줄링 최적화 기법의 확장성 연구

최지은, 박근철, 노승우, 박찬열
한국과학기술정보연구원 슈퍼컴퓨터기술개발센터
(jieun1205, gcpark, seungwoo0926, chan)@kisti.re.kr

A Scalability Study for scheduling optimization method based on application characterization

Jieun Choi, Geunchul Park, Seungwoo Rho, Chan-Yeol Park
Center for Development of Supercomputing System
Korea Institute of Science and Technology Information

요 약

한정된 고성능 자원을 여러 사용자들에게 제공해야하는 슈퍼컴퓨터와 같은 시스템은 제한된 기간 내에 보다 많은 양의 작업이 실행되도록 시스템 활용률을 높이는 방안이 필요하다. 이를 위해 시스템 관리자가 수행할 응용 프로그램에 대한 사전 정보를 파악하는 것이 유용하다. 대부분의 고성능 컴퓨팅 시스템 운영에 있어 작업을 실행할 때 사용자로부터 실행 기간, 자원 요구사항들에 대한 정보를 제공 받거나 시스템 사용 통계 값을 사용하여 필요한 정보를 생성하는 등의 프로파일링 기술을 바탕으로 시스템 활용률을 높이는데 활용하고 있다. 본 논문의 선행연구에서 하드웨어 성능 카운터를 이용하여 응용 특성 분석을 실행하고, 이 결과를 바탕으로 작업 스케줄링을 최적화하는 기술을 개발한 바 있다. 본 논문에서는 슈퍼컴퓨터 최적 실행 지원을 위한 프로파일링 테스트베드를 구축하고 단일노드를 기반으로 분석한 응용 프로그램 특성 결과를 활용한 스케줄링 최적화 기법이 확장성 있게 동작함을 보이고자 하였다. 또한 중규모 클러스터에 개발한 스케줄링 최적화 기법을 적용한 결과 전체 응용 프로그램이 실행 시간을 단축함으로써 최대 33%의 성능 향상 효과를 얻었다.

1. 서론

전세계 슈퍼컴퓨터의 계산 성능 상위 500 시스템을 공표하는 Top500[1]에 따르면 현재 상위 10위에 속하는 슈퍼컴퓨터들은 최소 788개에서 최대 40,960개의 계산 노드로 구성되어 있으며, 100Gb/s 전송속도를 갖는 고성능 I/O 장비를 통해 대규모 클러스터 시스템 구조를 갖는다. 이와 같은 고성능 컴퓨팅 시스템은 한정된 고성능 자원을 여러 사용자들에게 제공해야하기 때문에 제한된 기간 내에 보다 많은 양의 작업이 실행되도록 시스템 생산성을 높이는 방안이 필요하다. 이를 위해 시스템 관리자가 수행할 응용 프로그램에 대한 사전 정보를 파악하는 것이 유용하다.

대부분의 고성능 컴퓨팅 시스템을 운영하는 센터들은 사용자로부터 작업 제출 시에 응용 프로그램의 실행 시간, 자원 요구사항들에 대한 기본 정보를 제공 받는다. 추가적으로 시스템 모니터링을 통해 자원 사용에 대한 모니터링 값이나 자원 상태 정보와 같은 통계 데이터를 수집하여 필요한 정보를 생성하는 등의 프로파일링 기술을 바탕으로 시스템 활용률을 높이기 위한 연구를 진행하고 있다[2, 3, 4]. 그러나 실제 사용자가 요청한 작업이 수행되어야할 대규모 고성능 컴퓨팅 시스템에서 작업의 프로파일링 데이터를 생성하기 위한 과정은 시스템 활용률을 낮출 수 있으므로 실제 시스템과 유사하면서 적은 규모의 테스트베드 운용의 필요성이 대두된다.

한편, 본 논문의 선행연구[5]에서 하드웨어 성능 카운터를 이용하여 응용 프로그램의 시스템 활용에 대한 특성 분석으로 프로파일링 데이터를 생성하고, 생성된 데이터를 바탕으로 응용 프로그램 사이의 자원 간섭률 기반 동시 스케줄링 기법을 개발한 바 있다. 본 논문에서는 슈퍼컴퓨터 최적 실행 지원을 위한 프로파일링 테스트베드를 구축하고 단일 노드를 기반으로 분석한 응용 프로그램 특성 결과를 활용한 스케줄링 최적화 기법이 확장성 있게 동작함을 실험을 통해 보이고자 한다.

2. 관련연구

고성능 컴퓨팅 시스템에서 작업에 대한 정보 및 통계 값을 사용하여 생성된 대략적인 프로파일 데이터를 활용하는 연구[2]에서는 오프라인 스케줄링을 통해 자원 활용률 기반으로 프로파일을 재생성한 다음 수정된 프로파일을 기반으로 스케줄링 기법을 제안하였다. [2]연구의 결과로 CINECA 컴퓨팅 센터에서 운영하는 오로라(EURORA) 시스템의 PBS[7] 스케줄러의 성능을 개선시켰다.

동적 자원 관리 플랫폼을 제안한 연구[3]에서는 스케줄러가 사전에 노드수와 전력 사용 레벨의 조합에 대한 성능 프로파일 데이터 셋을 저장하도록 한다. 스케줄러는 프로파일이 없는 새로운 작업에 대해서 필수 전력특성을 바탕으로 성능 모델링을 통해 자원을 할당한다. 제안된 기법은 아르곤 국립연구소의 IBM BG/P 시스템에서 선입선출 및 backfiling 스케줄링 기법을 사용하는 Slurm[8] 스케줄러와 비교하여 성능을 최대 5.2배 향상시켰다.

전력 프로파일링 생성 방법을 연구한 논문[4]에서는 작업 모니터링 데이터 수집, cray 플랫폼을 활용한 out-of-band 데이터 수집, 하드웨어 성능 카운터를 활용한 In-band 데이터 수집 및 코드 인스트루멘테이션 기반의 코드 프로파일링 데이터 수집 방법을 비교 분석한다. 이 연구는 소규모 테스트베드에서 생성된 전력 프로파일을 이용하여 로스앨러모스 국립연구소의 Trinity 대규모 슈퍼컴퓨터를 대상으로 프로파일링 데이터의 확장성 실험을 진행하였다.

3. 슈퍼컴퓨터 최적 실행 지원을 위한 프로파일링 테스트베드 구축

2018년 도입된 국가 슈퍼컴퓨터 5호기(누리온)는 인텔 제온파이 프로세서 기반의 계산 노드 8,305대와 인텔 제온 프로세서 기반의 CPU-only 노드 132대로 구성되어 총 25.7PFlops의 계산 성능 갖추었다. 슈퍼컴퓨터 5호기는 지난 1년 여간 140개 기관과 2000여명이 넘는 연구자들을 대상으로 서비스되었으며, 본 연구에서는 누리온의 최적 실행 지원을 위해 응용 프로그램의 성능 프로파일링이 가능한 테스트베드 클러스터를 구축하였다.

프로파일링 테스트베드 클러스터는 누리온의 계산 노드와 동일한 인텔 제온파이 프로세서 기반의 'KISTI-GVP' 서버 16노드로 구축하였다. KISTI-GVP 서버는 한국과학기술정보연구원 슈퍼컴퓨터기술개발센터에서 진행 중에 있는 창의형 융합연구 사업 과제의 연구 결과인 자체 개발 메인보드에 인텔 제온파이 프로세서를 장착한 시제품으로, <표 1>은 KISTI-GVP 서버의 하드웨어 및 소프트웨어 세부 사양을 보여준다.

<표 1> KISTI-GVP 노드 사양

플랫폼	Intel S7200AP
프로세서	Intel Xeon Phi CPU 7290@1.50GHz, 72 cores (288 cores by Hyper-threading)
메모리	192 GB (DDR4) 16GB (MCDRAM)
네트워크	Melanox Infiniband 100GB/s
운영체제	CentOS 7.3
커널	3.10.0-514.el7.x86_64
클러스터 모드	Quadrant Mode
메모리 모드	Cache Mode
파일 시스템	Network File system
플랫폼	KISTI-GVP(Groveport)

4. 응용 특성 분석 기반 스케줄링 최적화 기법의 확장성 실험

본 장에서는 KISTI-GVP 단일 노드에서 응용 프로그램의 특성을 분석하고 이를 바탕으로 16개 노드의 테스트베드 클러스터 시스템에서 스케줄링 최적화 기법의 확장성 실험을 진행하였다. 실험내용을 다루기에 앞서 실험에 사용된 응용 프로그램에 대해 설명하고자 한다.

4-1. NPB 응용 프로그램

실험에 사용된 응용 프로그램은 NASA에서 개발한 전산유체역학(CFD) 분야의 계산 및 데이터 연산을 벤치마크한 Nas Parallel Benchmark(NPB)[6] 프로그램을 대상으로 하였다. NPB는 5개의 커널 프로그램(IS, EP, CG, MG, FT)과 3개의 수도 어플리케이션(BT, SP, LU)으로 구성되어있다. 또한 I/O 연산 벤치마크를 위한 프로그램(BT_epio, BT_full)등을 포함한다. 각 프로그램은 가장 작은 문제크기(class)인 S, W부터 중규모 A, B, C 클래스와 대규모 D, E, F 문제크기로 규모를 변경하여 실행 가능하다.

본 실험에서는 하나의 물리서버가 갖는 최대 물리코어 수(72개)를 고려하여 노드 당 MPI 프로세스(ppn)를 64로 설정하였다. <표 2>는 실험에서 사용한 10개의 NPB 응용프로그램의 문제 크기와 실행

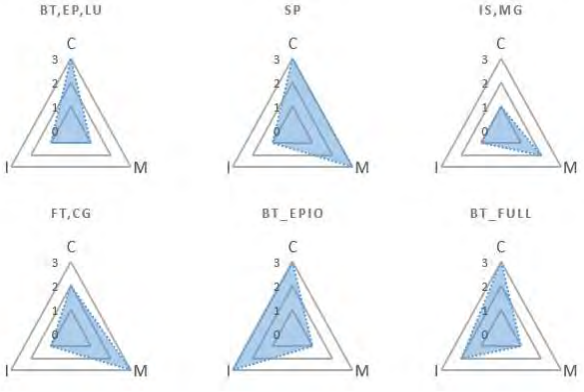
시간 측정 결과를 나타내고 있다. 단일 노드 프로파일링 실험에서 문제크기는 C 또는 D 클래스를 사용한다. 문제크기가 너무 클 경우 단일 노드에서 실행이 불가능한 경우가 있고, 응용 특성 분석에 있어 실행 시간이 미치는 영향을 줄이고자 다음과 같은 규모로 실행하였다. 클러스터 구성은 4노드, 8노드, 16노드로 확장해가며 실험하고 해당 규모의 클러스터에서 실행하기 적합한 D, E 클래스를 대상으로 하였다.

<표 2> 실험에 사용된 NPB 프로그램의 문제크기 및 실행시간

NPB	단일 노드 프로파일링		중규모 클러스터 실험 (16 nodes)	
	문제크기 (class)	실행시간 (seconds)	문제크기 (class)	실행시간 (seconds)
BT	C	51.38	D	118.55
EP	D	33.79	E	34.46
LU	C	31.09	D	47.77
SP	C	35.45	D	113.72
IS	D	34.92	E	188.97
MG	D	84.23	E	37.06
FT	C	10.39	D	151.04
CG	C	10.93	D	53.61
BT_epio	C	76.44	D	214.55
BT_full	C	120.23	D	595.93
Total	488.85 seconds		1555.66 seconds	

4-2. 하드웨어 성능 카운터를 이용한 응용 특성 분석

(그림 1)은 KISTI-GVP 단일노드에서 응용 프로그램이 실행되는 동안 하드웨어 성능 카운터를 수집하고 이후 군집분석을 수행하여 그 결과를 응용 프로그램별로 차트(kiviat chart)로 도식화한 결과이다.



(그림 1) KISTI-GVP 단일 노드에서 NPB 응용(C 또는 D 클래스) 특성 분석 결과

차트에서 삼각형의 각 꼭짓점은 CPU(C), Memory(M), I/O(I) 자원을 나타낸다. 삼각형의 크기는 응용 프로그램별 상대적인 자원 사용률에 따라 자원 사용이 낮은 군집(1), 중간 군집(2), 높은 군집(3)에 속함을 보여준다. 예를 들어 NPB의 BT, EP, LU 응용 프로그램의 경우 계산 자원의 사용이 다른 응용에 비해 상대적으로 높게 나타났으며 메모리, I/O 자원의 사용은 가장 낮은 군집에 속한 것으로 분석되었다.

응용 프로그램 특성 분석 결과를 활용하면 자원 간섭률 기반의 스케줄링 최적화 전략을 세울 수 있다[5]. 선행 연구[5]에 따르면, 응용 프로그램별 특성 결과 차트를 두 개씩 중첩시켰을 때 중첩되는 면적이 응용 프로그램 사이의 자원 간섭률로 계산되고 시스템 큐에서 대기중인 작업 간에 작업 간섭률을 고려하여 스케줄링 최적화가 가능하다.

4-3. 확장성 실험

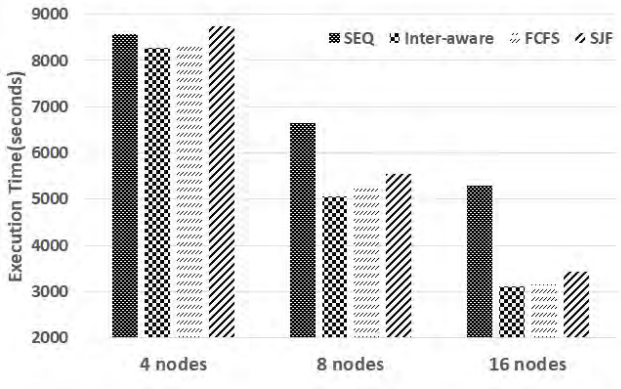
(그림 1)의 단일 노드에서 문제크기가 작은 응용 프로그램의 특성 분석 결과를 활용한 자원 간섭률 기반의 스케줄링 최적화 기법(Inter-aware)을 문제크기가 큰 응용을 대상으로 중규모 클러스터에서 스케줄링 최적화시에 시스템 실행 최적화 효과를 분석하기 위한 실험을 진행하였다.

스케줄링 실험은 랜덤으로 생성된 NPB 작업 30개가 대기중인 작업큐를 대상으로, 두개의 작업 실행 스레드(Worker)가 스케줄링 기법에 따라 작업을 실행한다. 제안된 자원 간섭 기반 스케줄링 최적화 기법(Inter-aware)은 간섭률이 낮은 작업을 동시에 우선 스케줄링한다. 이와 비교군으로 선입선출 스케줄링(FCFS) 기법과 작업 실행시간이 짧은 작업 우선 실행 스케줄링(SJF) 기법을 수행하였다. 또한 자원 간섭 없이 하나의 worker로 작업을 하나씩 순차적으로 실행한 결과는 'SEQ'으로 나타내었다.

각 스케줄링 기법은 동일한 작업큐를 사용하여 비교 실험하였으며, <표 3>과 (그림 2)는 다섯 개의 작업 큐의 평균 실행 시간을 보여준다. 가장 적은 수의 노드 4개를 사용한 경우, Inter-aware 기법은 FCFS, SJF, SEQ 대비 0.30%, 5.54%, 3.66%로 작업 실행 시간을 단축시켜 시스템 성능을 향상시켰다. 마찬가지로 8노드를 대상으로 했을 때 1.59%, 10.99%, 15.55%씩 실행 성능이 향상되었고, 16노드에서는 1.68%, 9.59%, 33.34% 성능 향상을 보였다.

<표 4> 단일 노드 프로파일링 기반의 스케줄링 최적화 확장성 실험 결과

	SEQ	Inter-aware	FCFS	SJF
4 노드	8579.01	8264.50	8289.65	8749.75
8 노드	6640.79	5069.57	5225.64	5536.06
16 노드	5285.12	3107.05	3160.09	3436.76



(그림 2) 단일 노드 프로파일링 기반의 스케줄링 최적화 확장성 실험 결과 비교

선행 연구[5]에 따르면 단일 노드에서 응용 특성 분석 후 단일 노드 스케줄링 최적화시 SEQ 대비 29.85%, FCFS 대비 6.22%, SJF 대비 9.98%의 성능 향상을 보였다. 이와 비교했을 때, 본 실험에서의 8 노드, 16노드에 비해 4노드 성능 향상률이 상당히 낮게 측정되었고, SJF 실행의 경우 SEQ 실행보다 성능이 떨어지는 것으로 보인다. 이는 NPB 프로그램 D, E 클래스의 실행이 가능한 최소 노드수인 4 노드가 응용 실행은 가능하나 시스템 부하가 상당하기 때문으로 분석된다.

5. 결론

본 연구에서는 자체 개발한 KISTI-GVP 시스템 기반 클러스터 환경에서 응용 특성 분석 기법의 스케줄링 최적화 확장성 실험을 진행하였다. 실험 결과 프로파일링 대상 응용과 스케줄링 최적화 대상 응용 프로그램의 문제 크기가 다를지라도 시스템 성능 향상의 효과가 유지됨을 보였다.

향후 중규모 클러스터에서 실행할 응용 프로그램과 같은 문제 크기의 응용을 대상으로 최소한의 노드를 활용하여 프로파일링 정보를 생성하고 프로파일링 효과를 분석하고자 한다. 또한 중규모 클러스터에서 응용 프로그램의 런타임 전체를 프로파일링 하는 것이 아닌 일부분만 프로파일링 하는 경우의 성능 변화도 분석할 예정이다.

Acknowledgement

이 논문은 2020년도 한국과학기술정보연구원(KISTI)의 주요사업 과제(No. K-20-L02-C08-S01) 및 정부의 재원으로 진행된 창의형 융합연구사업(No. G-19-GT -CU01-S01)의 지원을 받아 수행된 연구임.

참고문헌

[1] (Online) Top500, https://www.top500.org

[2] Bridi Thomas, “Scalable optimization-based Scheduling approaches for HPC facilities.”, PhD Thesis, alama, Bologna, 2018.

[3] Sarood Osman et al. “Maximization throughput of overprovisioned hpc data centers under a strict power budget.” SC’14, New Orleans, LA, USA, 2014, pp. 807-818

[4] Younge Andrew J. et al. “Small scale to extreme: Methods for characterizing energy efficiency in supercomputing applications.” Sustainable Computing: Informatics and Systems, vol. 21, pp. 90-102, 2019.

[5] Jieun Choi et al. “Interference-aware co-scheduling method based on classification of application characteristics from hardware performance counter using data mining.” Cluster Computing, Vol. 23, No.1, pp. 57-69, 2020.

[6] (Online) NPB, https://www.nas.nasa.gov/publications/npb.html

[7] (Online) PBS, https://www.pbspro.org

[8] (Online) Slurm, https://www.slurm.schedmd.com