

고성능 컴퓨팅을 위한 데이터 집약적인 응용의 런타임 특성 분석

최지은 오지선⁰ 김윤희¹

한국과학기술정보연구원 슈퍼컴퓨팅본부

⁰숙명여자대학교 컴퓨터학과

jjeun1205@kisti.re.kr

⁰{jsoh8088,yulan}@sookmyung.ac.kr

A Runtime Characteristic Analysis of Data-Intensive Applications for High Performance Computing

Jieun Choi Jisun Oh Yoonhee Kim

Supercomputing, KISTI

⁰Dept. of Computer Science, Sookmyung Women's University

요 약

데이터 센터 자원의 효율적인 활용을 위하여 과학 응용의 특성을 파악하는 것이 필요하다. 본 논문에서는 성능 분석 도구를 사용하여 천문학, 신약 개발, 기상-대기과학, 물리학, 항공 우주 과학 계산 분야의 데이터 집약적인 HPC 응용 프로그램의 런타임 작업 특성을 분석한다. 작업 특성의 분석은 CPU, 메모리, I/O 와 같은 자원 사용량을 통해 실행 패턴을 파악하여 분석하며 자원 접근 패턴에 대한 응용의 공통 특성을 도출한다. 분석된 HPC 응용 프로그램은 자원 사용 패턴 및 특징에 따라 실행 응용의 변화를 예측하는데 참고할 수 있는 정보를 제공할 수 있다.

1. 서 론

최근 데이터 센터의 자원 낭비와 성능 하락으로 데이터 집약적인 HPC(High Performance Computing) 응용 프로그램의 특성 분석을 반영한 적응형 데이터 관리 및 실행 제어 연구가 확대되는 추세이다. 천문학, 신약 개발, 기상-대기과학, 물리학, 항공 우주 과학 계산 분야 등의 계산 과학 응용은 자원과 특성에 따라 성능이 좌우[1]되기 때문에 응용 특성 분석에 따른 I/O, CPU, 메모리 등의 자원 사용에 대한 응용 분류가 중요하다.

본 연구에서는 실제 데이터 집약적인 HPC 응용 프로그램의 작업 특성을 분석하여 자원 활용 패턴을 파악하고 공통 특성을 도출한다. Valgrind[7], Glances[8], gprof[9] 등과 같은 런타임 분석 도구를 사용하여 5가지 과학 계산 분야의 7개의 HPC 응용에 대해 I/O, CPU, 메모리 자원 사용 패턴을 분석한다.

2. 데이터 집약적인 응용의 런타임 특성

HPC 응용의 특성 분석은 응용의 자원 소비, 메모리, I/O 특징 추출을 통해 분류한다. 자원 소비 분석을 통해 HPC 응용을 CPU-intensive, Memory-intensive, I/O & Paging-intensive, Network-intensive, Disk-intensive 의 패턴[2]으로 분류할 수 있다. 메모리 특징 분석의 경우 데이터 위치, 데이터 구조, 메모리 주소등을 파악해서 메모리 접근 패턴을 분류한다. 일반적으로 sequential, nearest neighbor, random, linear, strided, scatter, gather, 2D spatially coherent 등의 패

턴[3]으로 분류한다. I/O 패턴의 경우 I/O 연산 발생 횟수, I/O 크기, 빈도 패턴을 파악해서 HPC 응용의 I/O 패턴을 분류한다. 이는 constant, bell-shaped, bursty, irregular, recurring, periodic 등의 패턴[4][5][6]으로 분류된다.

3. 실험 및 분석

Montage GALFA, CFD, AutoDock, HACC_I/O, LAMMPS MELT/PEPTIDE, WRF 응용을 분석한 실험환경은 표1과 같다.

표 1. HPC응용 분석 실험 환경

HPC응용	CPU	Core	RAM
Montage GALFA	Intel(R) Core(TM) i7-5820K CPU @ 3.30GHz	4	8GB
CFD	Intel(R) Core(TM) i7-5820K CPU @ 3.30GHz	12	32GB
AutoDock	Intel(R) Xeon(R) E552 CPU @ 2.27GHz	16	24GB
HACC_I/O, LAMMPS MELT, PEPTIDE, WRF	Intel(R) Core(TM) i7-5820K CPU @ 3.30GHz	12	32GB

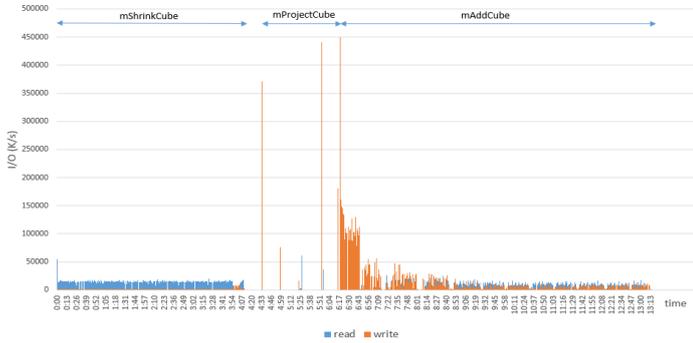
3.1 Montage GALFA[10]

Montage GALFA 응용은 천문학 분야의 Mosaic 이미지 생성 엔진으로 대용량의 입력데이터를 이용하여 많은 수의 중간 데이터와 하나의 출력데이터를 생성한다. GALFA는 27.5GB의 입력데이터를 사용하며 데이터 축소 단계에서 planes의 크기에 따라 중간, 최종 생성되는 데이터의 사이즈가 달라진다.

Montage GALFA 응용의 각 모듈 별 사용된 최대 메모리 사이즈는 Valgrind 툴을 사용하여 측정하였다. mProjectCube 모듈의 경우 전체 응용 실행 시간의 16%를 차지하나 다른

¹ 교신저자

모듈이 요구하는 메모리에 비해 최대 201배의 메모리 사용량을 보인다. 이는 어느 특정 순간에 높은 메모리를 요구하는 bursty 메모리 패턴을 가진 응용으로 분석된다.



[그림 1] Montage GALFA I/O 패턴

그림 1은 Montage GALFA를 15planes으로 실행했을 때 초당 read, write, I/O size를 나타내며 특정 모듈에서 write I/O가 급격하게 증가하는 bursty 패턴을 보인다. mShrinkCube 모듈의 경우 15palnes 단위로 데이터를 읽어 축소하여 사용하고 있으며, mProjectCube는 축소된 큐브를 통해 생성된 이미지 파일과 테이블 및 헤더파일을 이용하여 새로운 fit파일을 생성하므로 write의 양이 다른 모듈에 비해(최대 write size = 448,978.99K/s) 크다.

3.2 CFD[11]

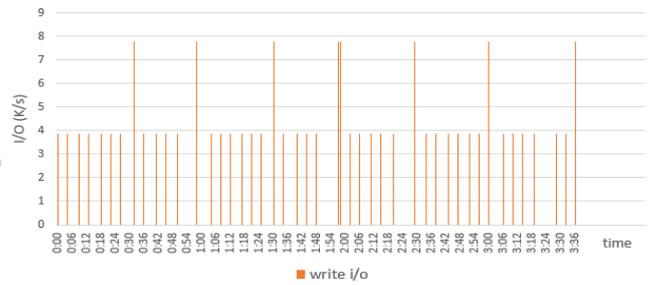
본 실험에서는 항공우주 분야의 비정상 유동해석 역학 시뮬레이션(CFD)중 2차원 Euler 비정상 유동 해석 프로그램을 대상으로 런타임 분석을 수행하였다. CFD응용은 2-D shock tube problem으로 고압관과 저압관의 가운데 격막을 제거하였을 경우 발생하는 비정상 유동을 예측한다. Euler 비정상 유동해석 응용의 평균 CPU 사용량은 1,137%이며, 평균 메모리 사용량은 0%로 완전 계산 집약적(Pure CPU-intensive) 응용이다.

Euler 비정상 유동 해석 프로그램을 3만 2천개 cell로 이루어진 격자를 사용했을 때 전체 실행 시간은 21분 7초이며, 1초당 read, write I/O를 분석했을 때 각각 0으로 constant 패턴을 보이며 disk I/O가 전혀 발생하지 않는 zero constant 패턴을 보인다.

3.3 AutoDock[12]

신약 개발 분야의 단백질과 지질 결합모드를 알아내는데 사용되는 분자 도킹 응용 중 하나인 Autodock3 응용에 대한 분석을 진행하였다. Autodock3 응용은 각 지질 분자 데이터가 하나의 입력 파일이 되며, 한 개의 입력 데이터 크기는 0.2MB 정도로 보통 수천, 수만 개의 지질 분자 데이터가 사용된다. 이는 99.12%의 평균 CPU 사용량과 0.7% 메모리 사용량을 보인다.

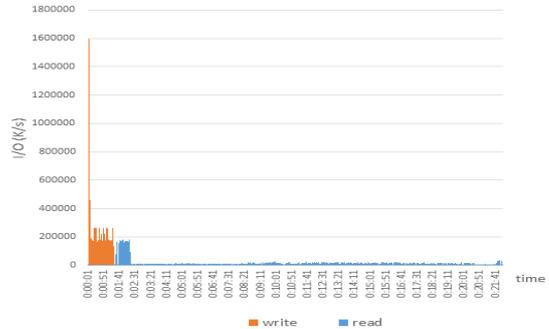
그림 2는 1개의 리간드 파일을 사용하여 단백질 도킹을 수행했을 때 1초당 write I/O size를 보여준다. 한편, read I/O의 경우 0으로 응용의 실행 시 disk read 연산을 수행하지 않는다. AutoDock 응용은 write I/O가 bursty한 패턴이다.



[그림 2] AutoDock write I/O 패턴

3.4 HACC_I/O[13]

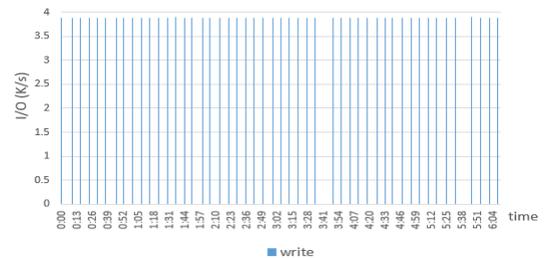
무충돌 유체 구조 형성 시뮬레이션(HACC)는 암흑 물질과 암흑 에너지를 연구하기 위한 입자 시뮬레이션을 수행한다. HACC_I/O는 시스템 성능평가를 위한 벤치마크로 HACC가 실행될 때 사용자가 정의한 데이터에 대해 체크포인팅을 수행한다. Glances 툴을 이용하여 분석하였으며, 평균 CPU 사용량은 10.77%이며, 평균 메모리 사용량은 46.34%이다.



[그림 3] HACC_I/O의 I/O 패턴

그림 3은 HACC_I/O의 I/O 패턴을 보여준다. particle 사이즈는 45,019,982, rank는 1로 설정하여 실행하였다. HACC_I/O의 전체 실행시간은 21분 41초이며, 체크포인트 파일사이즈는 18,646,684,456Byte(17GB)이다. 본 응용에서는 HACC의 체크포인트를 위한 데이터를 쓰는 동안 write 연산을 수행하다 restart 지점(1:41) 부터는 체크포인팅 파일을 통해 read 연산을 수행하며 진행된다. 최대 write I/O 크기는 1,594,433K/s 이며, 최대 read I/O 크기는 181,290.15K/s 이다.

3.5 LAMMPS[14] MELT/PEPTIDE



[그림 4] LAMMPS PEPTIDE write I/O

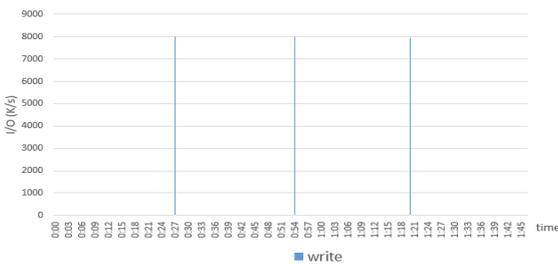
LAMMPS 분자 동역학 오픈소스 응용 프로그램은 rapid melt of 3d LJ system을 수행하는 MELT와 dynamic of a small solvated peptide chain (5-mer)을 수행하는 PEPTIDE를 수행하였다. 본 응용 프로그램은 Glances 툴을 이용하여

분석하였다. PEPTIDE, MELT 응용 프로그램은 각각 CPU 사용량은 99.49%, 99.77%이며, 평균 메모리 사용량은 0%로 완전 계산 집약 응용이다.

각 응용의 thermo를 50,000, 반복횟수(run)를 30,000으로 했을 때의 전체 실행시간은 MELT와 PEPTIDE 각각 11분 45초, 7분 6초이다. MELT는 실행시간 동안 read/write가 발생하지 않았다. 반면, PEPTIDE의 write I/O는 평균 3.8K/s으로 constant 패턴을 보이고 read I/O는 0으로 일정하다. 그림 4는 PEPTIDE의 write I/O 패턴이다.

3.6 WRF[15]

본 실험에서 사용한 기상 및 대기질 예측 응용인 WRF는 전처리 단계의 WPS와 WRF 모형 본체로 구성되어 이상실험(ideal case) 또는 실제 기상 상황에 대한 실험과 예보(real case)를 위한 응용이다. 본 응용에서는 Glances 툴을 사용하여 자원 활용 패턴을 분석하였으며 지형 데이터를 전체 49GB에서 모형격자에 맞는 726MB 크기의 입력 데이터로 사용하였다. 또한 해당 분석에서는 WRF 모형을 구동하는 부분인 wrf.exe 실행 부분에 대해서만 패턴 분석을 수행하였다.



[그림 5] WRF write I/O 패턴

본 응용은 97.25% CPU 사용과 0.55%의 메모리 사용을 보이는 CPU intensive 응용이다. 3분(arc min) 마다 적분 시에 결과를 파일에 작성을 수행하여 그림 5와 같은 일정 기간 반복적인 write I/O 패턴을 보이며 write I/O 사이즈는 약 8,000 K/s이다.

4. 실험 결과 요약

표 2. 데이터 집약적인 응용의 런타임 특성 분석

HPC 과학 응용	타입	데이터		자원 사용 패턴	I/O 패턴
		입력	출력 및 중간		
Montage GALFA	Workflow	27GB	3.5~11GB	Memory-intensive	Bursty
CFD	Workflow	104KB	0	CPU-intensive	Constant
AutoDock	Bag of tasks	0.2MB * 10 ² ~10 ³ 개	0.2MB * 10 ² ~10 ³ 개	CPU-intensive	Bursty
HACC I/O	Bag of tasks	2KB	17GB	Memory-intensive	Bursty
LAMMPS Melt/Peptide	Workflow	2MB	5GB	CPU-intensive	Constant
WRF	Workflow	49GB~1TB	1.5GB~27GB	CPU & Memory intensive	Constant

표 2는 본 실험에서 데이터 집약적인 응용 7개에 대해 I/O, CPU, 메모리 자원 사용 패턴을 분석한 결과이다.

5. 결론 및 향후 연구

본 논문에서는 고성능 컴퓨팅을 위한 데이터 집약적인 응용인 Montage GALFA, CFD, AutoDock, HACC_I/O, LAMMPS MELT/PEPTIDE, WRF 의 런타임 특성을 분석하였다. 응용의 작업 타입(bag of tasks, workflow)을 파악하고, 런타임 도구를 사용하여 I/O, 메모리, CPU와 같은 자원 사용량에 대한 작업 특성을 분석하였다.

향후에는 응용 프로그램의 자원 소비 패턴 분석을 통해 응용의 특성을 분석하여 데이터의 특성 및 이해에 기반한 지능적 배치 기법 연구와 런타임 특성을 고려한 작업 스케줄링 기법 및 실행 프레임워크 연구를 진행할 계획이다.

Acknowledgement

이 논문은 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2017R1A2B4005681)

참고문헌

- [1] A. Losup et al. "Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing", IEEE transactions on parallel and distributed systems, VOL. 22, NO. 6, JUNE 2011
- [2] Zhang et al. "Application classification through monitoring and learning of resource consumption patterns." Proceedings 20th IEEE International Parallel & Distributed Processing Symposium. IEEE, 2006.
- [3] Memory access pattern, https://en.wikipedia.org/wiki/Memory_access_pattern
- [4] Comparing Decoupled I/O Kernels versus Real Traces in the I/O Analysis of the HACC Scientific Application on Large-Scale Systems by Sean McDaniel
- [5] Reid, Fiona JL, and Lorna A. Smith. "Performance and Profiling of the LAMMPS Code on HPCx." private communication (2005).
- [6] Ferreira, Kurt B., et al. "libhashckpt: hash-based incremental checkpointing using gpu's." European MPI Users' Group Meeting. Springer Berlin Heidelberg, 2011.
- [7] Valgrind, <http://valgrind.org/>
- [8] Glances, <https://nicolargo.github.io/glances/>
- [9] gprof, <https://sourceware.org/binutils/docs/gprof/>
- [10] Peek et al., The GALFA-HI Survey: Data Release 1, The Astrophysical Journal Supplement, Volume 194, Issue 2, article id. 20, 13 pp. 2011
- [11] CFD, Seoul national university Aerodynamic Simulation & Design Laboratory
- [12] Autodock3, <http://autodock.scripps.edu>
- [13] Habib, Salman, et al. "HACC: Simulating sky surveys on state-of-the-art supercomputing architectures." New Astronomy 42, 49-65, 2016
- [14] LAMMPS, <http://lammps.sandia.gov/>
- [15] WRF, <http://www2.mmm.ucar.edu/wrf/users/>