멀티코어 아키텍처에서 전구 대기과학 시뮬레이션의 성능분석

최정식O 한환수
성균관대학교 전자전기컴퓨터공학과
chjs@skku.edu, hhan@skku.edu

Performance analysis of GEOS-Chem on multi-core architecture platform

Jungsik ChoiO Hwansoo Han
Department of Electrical and Computer Engineering, Sungkyunkwan University

고성능 컴퓨팅 기술의 발전과 더불어 현대 과학 기술 또한 예전에는 수행할 수 없었던 실험을 시뮬레이션을 통해서 수행함으로써 크게 발전해 오고 있다. 유전체 연구 및 신약 연구 등의 생물학 분야, 첨단 반도체나 탄소 나노 튜브 연구 등의 나노 과학 분야, 기후 변화/예측 및 지질 연구 등의 환경 분야 등이 대표적인 분야이다[1][2][3][4]. 이와 같이 주요한 과학 분야의 발전을 지속하기 위해서는 컴퓨터 구조에 따른 성능향상을 최선으로 활용하는 것이 필요하다. 2002년 이후 프로세서의 성능 향상률은 매년 약 20% 정도로 둔화되었다. 이는 칩 전력 소비의 증가, 효율적으로 이용할 수 있는 명령어 수준 병렬성의 고갈, 상대적으로 성능 차이가 커지는 메모리 지연시간 등 세 가지 장벽에 기인한다. 결국 2004년부터 컴퓨터 시장은 단일 프로세서 보다는 한 칩에 여러 프로세서를 집적하여 성능을 향상시키기 시작한다. 이러한 멀티코어(multi-core)를 통해 쓰레드 수준의 병렬성(TLP)이 제공된다[5]. 결국 코어 수의 증가로 현재의 데스크탑 컴퓨터는 예전의 슈퍼컴퓨터 성능 수준의 멀티코어 시스템이 되었고, 많은 과학 분야의 과학계산 프로그램이 멀티코어 시스템에서 동작하고 있다. 본 논문에서는 앞서 언급한 과학 분야에서 사용되는 과학계산 프로그램 중 하나인 GEOS-Chem을 소개하고, GEOS-Chem과 같은 과학 계산용 코드의 성능 분석을 통해 멀티코어 아키텍처 상에서의 실행 특성을 분석한다.

대기 시뮬레이션 모델인 GEOS-Chem은 Harvard 대학이 중심이 되어 개발하고 있으며, 전지구적 대기의 3차원 화학 이동 모델(chemical transport model, CTM)을 시뮬레이션 하는 응용프로그램이다. GEOS-Chem은 NASA 고다드 지구 관측 시스템(Goddard Earth Observing System, GEOS)의 기상 관측 데이터를 입력으로 대기 화학 시뮬레이션을 수행하여 오존을 포함한 여러 기체상의 화학물질들과 황산염, 질산염을 포함한 무기 에어로졸, 탄소 성분 에어로졸 그리고 해염과 먼지입자와 같은 다양한 종류의 에어로졸의 시간적 공간적 분포를 모사한다[6]. GEOS-Chem의 현재 버전(v8-02-01)에서는 OpenMP 프로그래밍 모델로 개발되어 공유메모리를 제공하는 대칭 다중프로세서(symmetric multiprocessor, SMP) 컴퓨터 상에서 수행되도록 작성되어 있다. OpenMP는 쓰레드를 기반으로 하는 공유 메모리 프로그래밍 모델이며 프로그램의 병렬실행을 위해 Fork-Join 모델을 사용한다. GEOS-Chem은 Fortran 90형식으로 작성되어 있고, 226개의 파일, 총 194,355라인으로 구성되어 있으며 이 중, 20,056라인(10%)이 OpenMP 코드이다.

멀티코어 시스템에서 GEOS-Chem의 성능분석을 위해 GEOS-Chem의 최신 공개 버전인 v8-02-01과 Intel® Core™ Processor i7-860(2.66GHz, Memory 6GB)을 사용하였다. 운영체제는 CentOS 5.4 배포판, 컴파일러는 Intel® Fortran Compiler v10.1을 사용하였고, 프로파일러는 Intel® VTune™ Performance Analyzer v9.1을 사용하였다. 성능평가의 주요 관심은 GEOS-Chem의 총 실행시간을 이루는 구성요소의 분포도가 쓰레드 개수와 시뮬레이션 시간에 따라 어떻게 변화되는지 관찰하는 것이다.

실험에서 GEOS-Chem을 통해 12시간 동안의 대기 화학물질 변화를 시뮬레이션 하되 쓰레드는 1개, 2개와 4개로 진행하였다. 실험 결과, 쓰레드의 수가 증가할수록 전체 실행 시간은 줄어든다. 그러나 모든 경우에서 실제로 계산에 사용된 시간은 약 5%정도밖에 되지 않았다. 스톨로 인한 지연시간이 전체

본 논문은 교육과학기술부 재원으로 한국연구재단의 지원을 받아 연구되었음 (NRF 2009-0084870)

실행시간의 95% 이상을 차지하고 있었고 쓰레드 수가 증가해도 같은 양상을 보여주었다. 가장 많은 부분을 차지하고 있는 메모리 스톨 영역을 세분화하면 DRAM, L3, L2 순서로 많은 비중을 차지하고 있었고, 특히 DRAM에서 발생하는 지연시간이 70~80%를 차지해 개선의 필요성을 보여주고 있다. 데이터 지연은 응용프로그램의 성능을 결정하는 중요한 요소 중에 하나이다[7].

GEOS-Chem의 함수 중에서 self-time이 가장 긴 함수 10개의 self-time변화를 관찰하였다. 주요 함수의 self-time은 쓰레드 개수가 증가함에 따라 눈에 띄게 줄어들었지만, 각 비율의 큰 변화는 없었다.

또한 GEOS-Chem을 쓰레드 4개로 수행하면서, 시뮬레이션 시간을 1시간, 12시간, 그리고 24시간으로 구분하여 실험하였다. 실험 결과, 시뮬레이션 시간을 달리한 실험에서도 실제 연산에 사용된 시간은 3~4%로 매우 적고, 스톨에 의한 지연시간이 대부분을 이루었다. 여전히 메모리 스톨 영역이 가장 컸으며, 메모리 스톨의 가장 큰 비중은 역시 DRAM의 지연시간으로 가장 많은 시간을 소비하는 것으로 나타났다. 시뮬레이션 시간에 관계없이 같은 양상을 보인다.

앞서 설명한 실험결과에서 우리가 주목하는 것은 쓰레드 수를 늘려도 변하지 않는 싸이클 타임의 배분(%)이다. 이는 GEOS-Chem의 병렬성이 매우 크기 때문이라 판단된다. 또 총 실행시간에서 메모리 스톨로 인한 지연시간이 차지하는 비율이 모두 65~70%로 매우 크게 나타났다. 이는 실제 계산에 필요한 데이터가 하나의 멀티코어 프로세서로는 감당하기 어려울 정도의 큰 상태이기 때문이라고 판단된다. 특히 DRAM에서 읽혀지는 데이터로 인한 스톨이 차지하는 비율이 70~80%로 나타나고 있다. 이 문제를 극복하기 위해서는 많은 노드를 동원해서 사용되는 노드별 워킹셋(working-set) 사이즈를 줄여야 GEOS-Chem의 성능향상을 크게 볼 수 있을 것이다.

이를 위해 더 많은 프로세서를 이용한 병렬화가 필요하다. 그러나 멀티코어 아키텍처는 구조상 프로세서 노드의 개수를 크게 증가시키기 어려운 한계를 가지로 있다. 따라서 멀티코어 클러스터를 동원한 계산이 필요하다. Top500 슈퍼컴퓨터에서 제공하는 고성능 컴퓨터의 목록[8]을 살펴보면 2000년대 들어서면서 고성능 컴퓨터구조는 대량의 컴퓨터 노드를 고성능 인터커넥션 네트워크나 고대역폭의 이더넷으로 연결함으로써 쉽게 최대 성능치를 증대시킬 수 있는 클러스터 컴퓨터 구조로 발전해오고 있다. 더구나, 최근 클러스터 컴퓨터는 각 컴퓨터 노드가 멀티 코어 프로세서로 구성되어 있다. 과거의 주요 병렬컴퓨터는 cc-NUMA의 공유 메모리를 제공해왔기 때문에, GEOS-Chem도 OpenMP 프로그래밍 모델을 제공하고 있다. 기존에 존재하는 병렬 프로그래밍 모델은 공유메모리 혹은 분산메모리를 대상으로 개발되었기 때문에 멀티코어 클러스터와 같이 공유메모리와 분산메모리가 동시에 존재하는 컴퓨터 구조에서는 각각 한계를 가지고 있다. 그러므로 공유 메모리와 분산 메모리를 동시에 지원할 수 있는 하이브리드 프로그래밍 모델의 연구가 필요하다.

참고 문헌

1. The Landscape of Parallel Computing Research: A View from Berkeley. K. Asanovic, et al. Technical Report No. UCB/EECS-2006-183. University of California at Berkeley. Dec 2006.
2. Computational Science: Ensuring America's Competitiveness – Report to the President. President's Information Technology Advisory Committee (PITAC). Jun 2005.
3. Cyberinfrastructure Vision for 21st Century Discovery.National Science Foundation(NSF) Cyber infrastructure Council. Mar2 007.
4. 계산과학기술의 육성방안 연구. 서울대학교 신동우 외. 국가과학기술자문회의 2004년 5월
5. John L. Hennessy and David A. Patterson. Computer Architecture: A Quantitative Approach. Morgan Kaufmann. 1990.
6. GEOS-Chem Sites. http://acmg.seas.harvard.edu/geos/
7. James Reinders. VTune™ Performance Analyzer Essential. Intel Press. 2005.
8. Top 500 Supercomputer Sites. http://www.top500.org. Statistics from Nov 2008.