

# 사이클 정확도를 반영하는 멀티코어기반 전체 시스템 시뮬레이터<sup>\*†</sup>

김홍준<sup>○</sup>    김정현    이주환    이재진    최재영<sup>‡</sup>    임선영<sup>‡</sup>

서울대학교 컴퓨터공학부, <sup>‡</sup>삼성전자

{hongjune, junghyun, joothan, jlee}@aces.snu.ac.kr, <sup>‡</sup>{anton, sylim829}@samsung.com

## A Cycle Accurate Full System Simulator for Multicore Environment

Hongjune Kim<sup>○</sup>    Junghyun Kim    Joothan Lee    Jaejin Lee    Jaeyong Choi<sup>‡</sup>  
Sunyoung Lim<sup>‡</sup>

School of Computer Science and Engineering, Seoul National University

<sup>‡</sup>Samsung Electronics

### 1 서론

오늘날의 프로세서들은 한 개의 프로세서로는 물리적으로 더 높은 성능을 낼 수 없는 한계에 부딪쳐 멀티코어 구조로 대체되어 가고 있다. 이로 인해 코어간의 캐시 일관성을 보장하는 프로토콜의 구현이 필요하게 되었을 뿐만 아니라 프로세서와 메모리간의 대역폭 역시 늘어나게 되었다. 또한 인텔의 Nehalem 아키텍처에서는 메모리 컨트롤러가 CPU안으로 내장되었고 앞으로는 그래픽 프로세서까지 CPU안으로 집적되는 구조가 사용될 것으로 보인다.

이런 변화들은 시스템 전체 구조와 시스템 요소간의 연결관계에 대해서도 큰 변화를 가져오게 되며, 이러한 변화된 환경에서 성능을 평가하기 위해서 기존과는 다른 모델이 필요하게 되었다. 앞서 서술한 많은 변화들을 반영하는 전체 시스템 시뮬레이터의 개발이 필요하게 된 것이다. 이를 이용한다면 미래 멀티코어 시스템의 다양한 특성들을 반영하여 각 특성의 변화에 따른 전체 시스템 성능 변화를 효율적으로 측정할 수 있을 것이다.

본 논문에서는 x86 기반의 최신 프로세서 인텔 네할렘 프로세서 중심의 전체 시스템을 시뮬레이션하기 위한 시뮬레이터의 설계 및 구현에 대하여 논하고자 한다.

### 2 본론

PTLsim [1]은 사이클 정확성을 만족하는 오픈 소스 x86-64 마이크로 프로세서 시뮬레이터이다. PTLsim은 Xen을 통해서 전체 시스템을 시뮬레이션 할 수 있는 기능을 제공한다. 하지만 프로세서까지에 대해서만 사이클 정확성을 보장해 주고 메인 메모리는 정해진 상수 값으로 지연시간을 계산한다. 또한 나머지 I/O 서브시스템에 대해서는 기능적인 시뮬레이션만을 지원한다. 여기에 정확한 메인 메모리 시뮬레이션을 위해서 오픈 소스 DRAM 시스템 시뮬레이터인 DRAMsim [2]을, 디스크 시스템 시뮬레이션을 위해서 DiskSim [3]을 통합하였다.

PTLsim은 single SMT 코어만을 시뮬레이션 하기 때문에 한 개의 캐시 계층(cache hierarchy)만을 가지고 있다. 타깃 프로세서인 네할렘 Xeon E5520는 4개의 물리코어를 가지고 있다. 각 코어당 독립적인 L1, L2 캐시를 가지고 있으며 L3 캐시를 전체 코어가 공유하고 있다. 이런 구조에서는 L2 캐시간의 일관성(coherence)을 보장해 주어야 하기 때문에 캐시 일관성 프로토콜(cache coherence protocol)이 필요하다. 이에 PTLsim 코드를 한개의 캐시 계층을 공유하는 구조에서 여러 개의 캐시 계층을 가지는 구조로 변경하고 MESI 프로토콜을 구현하여 멀티코어 시뮬레이션이 가능하도록 하였다. Intel CPU에서 사용하는 MESI 프로토콜의 검증과 구현은 [4]에서 수행한

\*본 연구는 교육과학기술부/한국과학재단 창의적연구진흥사업(메니코어프로그래밍연구단, 0421-20090025)의 지원으로 수행되었습니다.

<sup>†</sup>본 연구를 위해 연구장비를 지원하고 공간을 제공한 서울대학교 컴퓨터연구소에 감사 드립니다.

연구를 기반으로 하였다.

또한 기존의 DRAMsim 코드를 wrapping하여 외부의 메모리 요청을 받아서 각 CPU cycle마다 해당하는 시뮬레이션을 수행하고 완료된 결과를 받아갈 수 있도록 수정하였다. 이를 위에서 복제된 각 캐시 계층에서 메모리 시스템을 참조하도록 하여 메모리 요청 시 DRAM 시스템의 요청 큐로 요청을 보내도록 하였고 DRAM 시스템에서 이 요청을 받아서 각 클락마다 해당하는 시뮬레이션을 하게 된다. 이후 각 코어의 캐시 시스템이 시뮬레이션을 수행하는 하는 도중 메모리 시스템의 요청 큐를 검사하여 완료된 요청이 있을 경우 이를 가져와 처리하게 된다.

PTLsim은 프로세서와 메모리 시스템을 제외한 나머지 시스템을 시뮬레이션하기 위해서 Xen [5]의 가상화 기능을 이용한다. PTLsim의 모니터 프로그램은 Domain 0상에서 실행되며 시뮬레이션 하려는 가상 머신은 Domain U상에서 실행되게 된다. 이 때 수정된 리눅스 커널이 가상화가 필요한 시스템 요청들을 Xen통해서 수행하게 되고 이 사이에 PTLsim의 확장들이 관여하게 되어 필요한 부분을 시뮬레이션 해주고, PTLsim에서 지원하지 않는 기타 장치들에 대해서는 Domain 0를 거쳐 실제 하드웨어에서 수행되어 결과를 건네 받게 된다. DiskSim과의 통합 시 디스크 요청을 처리하기 위해서 blkfront - PTLsim - Xen - blkback - 실제 드라이버로 이루어 지는 요청 전달 통로에 대한 분석과 수정이 필요하였다. PTLsim에서는 디스크 I/O 요청을 그대로 Xen의 hyper-call을 통해서 그대로 전달한다. 이 안의 정보를 확인할 수 있도록 domain U의 리눅스 커널에 포함되어 있는 Xen의 domain U 측 가상화 디스크 드라이버인 blkfront에 새 Xen 이벤트 코드를 추가하고 이벤트 채널을 열어 요청 정보를 전송하도록 하였다. 이를 PTLsim에서 열어서 필요한 요청 정보에 따라 디스크 시뮬레이터에 요청을 보내게 된다. 이후 실제 디바이스 요청에 대한 응답이 돌아오면 이를 pending시키고 시뮬레이션된 결과가 나오면 이 지연 시간에 맞게 더 지연을 시킨 후 최종적으로 이벤트를 커널로 올려 보내게 된다.

세 가지의 부분적인 시뮬레이터를 통합하여 전체 시스템을 시뮬레이션 하도록 결합한 후 최신 x86 아키텍처인 네할렘 기반의 실제 시스템과 비교 검증(cross-validation)을 통해서 각 부분별 사이클 소요시간을 맞추었다. 타깃 시스템과 시뮬레이터 양측에서 종류별 CPU 명령어(instruction)들을 반복 수행하고 시간을 측정하여 PTLsim의 micro-operation 별 수행시간을 보정하였고, LMBench [6]등의 벤치마크를 이용하여 CPU 캐시와 DRAM 시스템의 지연시간을 측정하고 조정하였다. 또한 IOzone 파일 시스템 벤치마크 [7]를 이용하여서 HDD와 SSD등 디스크 시스템에 대한 검증도 수행하였다.

구현과 지연시간 조정을 마친 시뮬레이터를 SPEC OMP 2001에 포함된 4개의 벤치마크를 이용하여서 멀티코어 어플리케이션에 실행시의 전체적인 정확도를 검증하였다. 이런 계산 중심의 어플리케이션 벤치마크는 디스크를 많이 사용하지 않고, 또한 운영체제의 페이지 캐시의 영향으로 실제 디스크 접근은 거의 일어나지 않아서 구현된 디스크 시스템의 영향은 거의 볼 수 없다고 할 수 있으나 나머지 부분에 대한 전체적인 검증을 수행할 수 있었다. 실험 결과 수행 사이클의 최대 오차가 14.1% 평균 9.3%를 보임을 알 수 있었다.

### 3 결론

검증 결과 평균 오차 9.3%의 사이클 정확도를 만족하는 전체 시스템 시뮬레이터를 구현하였음을 알 수 있었다. 이는 멀티코어 프로세서와 메모리 시스템, 디스크를 포함한 전체 시스템을 정확한 사이클로 시뮬레이션하는 최초의 x86 아키텍처 시뮬레이터이다. 이 시뮬레이터를 이용하면 다양한 멀티코어 어플리케이션의 동작이나 하드웨어 설정에 따른 어플리케이션 실행의 영향을 분석할 수 있을 뿐만 아니라, 다른 추가적인 소프트웨어/하드웨어 부품을 쉽게 구현하여 통합함으로써 시스템 전반에 걸친 다양한 연구에 이용할 수 있을 것이다.

### 참고 문헌

- [1] M. Yourst, "PTLsim: A cycle accurate full system x86-64 microarchitectural simulator," in *IEEE International Symposium on Performance Analysis of Systems & Software, 2007. ISPASS 2007*, 2007, pp. 23-34.
- [2] D. Wang, B. Ganesh, N. Tuaycharoen, K. Baynes, A. Jaleel, and B. Jacob, "DRAMsim: a memory system simulator," *ACM SIGARCH Computer Architecture News*, vol. 33, no. 4, p. 107, 2005.
- [3] J. Bucy, J. Schindler, S. Schlosser, and G. Ganger, *The DiskSim simulation environment version 4.0 reference manual*, Parallel Data Laboratory, Carnegie Mellon University, 2008.
- [4] J. Kim, S. Seo, J. Lee, M. Park, J. Lee, and T. Ko, "The Design and Implementation of a Cache Simulator for Multicore Systems," *Proceedings of Korea Computer Congress 2009*, vol. 36, no. 1, pp. 327-332, 2009.
- [5] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *Proceedings of the nineteenth ACM symposium on Operating systems principles*. ACM, 2003, p. 177.
- [6] L. McVoy and C. Staelin, "Imbench: Portable tools for performance analysis," in *Proceedings of the 1996 annual conference on USENIX Annual Technical Conference*. Usenix Association, 1996, p. 23.
- [7] W. Norcott and D. Capps, *Iozone filesystem benchmark*. [Online]. Available: <http://www.iozone.org/>