

<http://dx.doi.org/10.7236/IIBC.2016.16.2.157>

IIBC 2016-2-20

비대칭적 멀티코어 프로세서의 통계적 모의실험에 관한 연구

A Study On Statistical Simulation for Asymmetric Multi-Core Processor Architectures

이종복*

Jongbok Lee*

요약 비대칭적 멀티코어 프로세서 구조의 성능을 분석하기 위하여 명령어 트레이스 모의실험이나 실행 위주 모의실험을 이용하는 경우, 시간이 과다 소요되고 대량의 데이터 저장 공간을 차지하는 문제점이 있다. 본 논문에서는 통계적 모의실험에 의하여 다양한 하드웨어의 사양을 갖는 비대칭적 멀티코어 프로세서의 성능을 측정하는 기법에 대하여 연구하였다. 이것을 위하여 SPEC 2000 벤치마크 프로그램의 특성을 통계적 프로파일링 기법으로 모델링하고, 여기서 얻은 통계적 프로파일을 바탕으로 벤치마크 트레이스를 합성하여 비대칭적 멀티코어 프로세서에 대한 모의실험을 수행하였다. 그 결과, 통계적 모의실험에 의하여 측정된 성능이 명령어 트레이스 모의실험에 의하여 측정된 성능에 근접한 결과를 가져왔으며, 모의실험 시간을 크게 단축시켰다.

Abstract If trace-driven or execution-driven simulation is used for the performance analysis of asymmetric multi-core processors, excessive time and much disk space are necessary. In this paper, statistical simulations are performed for asymmetric multi-core processors with various hardware configurations. For the experiment, SPEC 2000 benchmark programs are used for profiling and synthesis, which is supplied as input for the simulation of asymmetric multi-core processors. As a result, the performance of asymmetric multi-core processor obtained by statistical simulation is comparable to that of the trace-driven simulation with a tremendous reduction in the simulation time.

Key Words : asymmetric multi-core processor, statistical simulation

1. 서론

현재 멀티코어 프로세서가 서버 및 데스크탑 컴퓨터, 태블릿, 스마트폰, 노트북 등의 성능 향상을 높이기 위하여 광범위하게 쓰이고 있다^[1,2]. 비대칭적 멀티코어 프로세서는 동일한 명령어 집합을 가지면서 대규모 복잡도로 구성되는 적은 수의 코어 프로세서와, 작고 간단한 다수

의 코어 프로세서들로 구성된다. 이러한 비대칭적 멀티코어 프로세서는 동종의 코어들로 구성되는 대칭적 멀티코어 프로세서에 비하여 더욱 효율이 높은 것으로 알려져 있다^[3-5].

이러한 비대칭적 멀티코어 프로세서의 개발 단계에서 성능을 평가하기 위하여 명령어 트레이스 모의실험(trace-driven simulation) 또는 실행 위주 모의실험

*정회원, 한성대학교 정보통신공학과
접수일자 : 2016년 2월 16일, 수정완료 : 2016년 3월 16일
게재확정일자 : 2016년 4월 8일

Received: 16 February, 2016 / Revised: 16 March, 2016 /

Accepted: 8 April, 2016

*Corresponding Author: jblee@hansung.ac.kr

Dept. of ICs Engineering, Hansung University, Korea

(execution-driven simulation)이 주로 쓰이고 있다. 명령어 트레이스 모의실험은 마이크로프로세서 내부에서 실제 프로그램이 실행하는 값에 대한 계산을 시행하지는 않지만, 명령어 간의 종속성과 계산에 소요되는 연산유닛의 클럭 사이클을 이용하여 사이클 수를 계산하므로 비교적 정확하다. 그러나, 명령어 트레이스를 저장하기 위한 대량의 디스크 공간과 적잖은 모의실험 시간이 필요하다. 한편, 실행 위주 모의실험은 실제로 마이크로 프로세서 내부에서 프로그램이 실행하는 모든 값을 계산하기 때문에 명령어 트레이스 방법보다 더욱 정밀하지만, 그 이유로 인하여 실험 시간이 과다하게 소모된다.

통계적 모의실험은 프로파일링 방법을 통하여 프로세서와 프로그램의 통계적 특성을 수집하고 그것을 바탕으로 새로운 입력 트레이스를 합성하여, 이것을 확률적으로 모의실험하는 것이다^[6,7]. 합성된 명령어 트레이스는 통계적 프로파일링 기법으로 발생시켰기 때문에, 길이가 매우 짧으면서도 각 벤치마크 프로그램의 특성을 함축적으로 나타낸다. 따라서, 보다 간단한 명령어 트레이스 모의 실험기에서 실행 가능하며, 기존의 일반적인 명령어 트레이스 모의실험 보다 훨씬 짧은 시간에 성능을 계산할 수 있다는 장점을 갖는다. 따라서, 비대칭적 멀티코어 프로세서에 대한 통계적 모의실험에 대한 연구가 필요하다.

본 논문에서는 여섯 개의 SPEC 2000 벤치마크의 정수형 프로그램을 대상으로 다양한 사양의 비대칭적 듀얼 코어와 비대칭적 쿼드코어 프로세서의 성능에 대하여 통계적 프로파일링과, 기존의 명령어 트레이스 모의실험으로 측정된 결과를 비교하여 그 정확도를 평가하였다. 본 논문은 다음과 같이 구성된다. 2 장에서는 통계적 프로파일링 기법에 대하여 논하고, 3 장에서는 모의실험 환경을 다룬다. 4 장에서 모의실험결과를 보이고, 5 장에서 결론을 맺는다.

II. 통계적 프로파일링 기법

통계적 모의실험의 전 과정은 그림 1과 같이 크게 4 단계로 나누어진다. 첫째, 일반적인 벤치마크 프로그램에 대한 명령어 트레이스의 발생, 둘째 통계적 프로파일링 기법에 의한 분석 및 통계 데이터의 수집, 셋째, 통계적 트레이스의 합성, 넷째 합성된 트레이스에 대한 통계적 트레이스 모의실험이다.

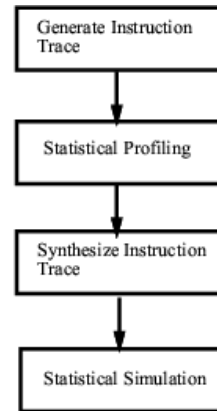


그림 1. 통계적 모의실험의 흐름도

Fig. 1. The flow of statistical simulation

첫 번째의 명령어 트레이스의 발생은 기존의 방법과 동일하다. 특정한 벤치마크 프로그램을 특정 명령어 트레이스 발생기를 통하여 명령어 트레이스를 생성한다. 두 번째의 통계적 프로파일링을 위한 분석 및 통계 데이터의 수집 단계는 다음과 같다. 우선, 첫 단계에서 얻은 프로그램의 명령어 트레이스를 분석하여 프로그램의 고유한 특성과 지역적 특성에 대한 통계값들을 추출해낸다. 프로그램의 고유한 특성은 프로그램을 구성하는 명령어의 유형별 구성비, 각 명령어에 대한 레지스터 피연산자의 개수 및 명령어 간의 데이터 종속에 대한 분포로 구성된다. 이 때, 명령어의 유형별 분포는 원래의 아키텍처가 갖는 명령어 집합보다 훨씬 적은 9 개의 간단한 명령어 집합으로 축소시킨다. 한편, 명령어 간의 상호 레지스터 종속은, 쓰레드 내에서 각 명령어의 목적 레지스터와 이에 종속인 소스 레지스터를 갖는 명령어 간의 거리로 정한다. 이러한 특성은 주어진 마이크로 프로세서의 구조와는 무관하고 단지 컴파일러와 마이크로 프로세서의 명령어 집합에만 영향을 받는 고유한 성질이다. 마지막으로, 지역적 특성은 분기 미스율이나 캐쉬 미스율 등을 의미하며, 이것은 마이크로 프로세서 하드웨어 구조에 의하여 영향을 받는다. 이 단계는 각 벤치마크 프로그램에 대하여 단 한 번만 시행한다.

세 번째, 위에서 얻은 통계적 특성을 기반으로 하여 각 벤치마크별로 고유한 특성을 갖는 새로운 명령어 트레이스를 합성한다. 통계적 트레이스를 합성시키는 방법은 0부터 1 사이의 난수를 발생시키고, 이 값을 통계적 프로파일링에 의하여 얻은 누적 분포 함수에 대응시키는 것이다. 이러한 방법으로 명령어의 유형 뿐만이 아니라,

피연산자의 개수, 피연산자의 종속거리를 결정하여 프로그램을 구성하는 명령어 코드를 만들어낼 수 있다. 그리고 이와같은 작업을 계속하여 반복하면 각 벤치마크 프로그램의 특성을 함축적으로 내포하는 짧은 길이의 통계적 트레이스의 합성이 가능하다.

마지막으로, 이렇게 합성된 트레이스를 분기 히트율 및 캐쉬 히트율에 대한 정보와 함께 비대칭적 멀티코어 프로세서 모의실험기로 입력하여 그 성능을 측정한다. 통계적 프로파일링에 의하여 자료를 확보한 후에는, 하드웨어 조건을 바꿔서 전체 설계 공간에 대하여 다양한 시도를 할 수가 있다.

III. 모의실험 환경

1. 비대칭적 멀티코어 프로세서의 구조

그림 2는 본 논문에서 모의실험 대상으로 하는 N 개의 코어로 구성되는 비대칭적 멀티코어 프로세서의 구조를 나타낸 것이다.

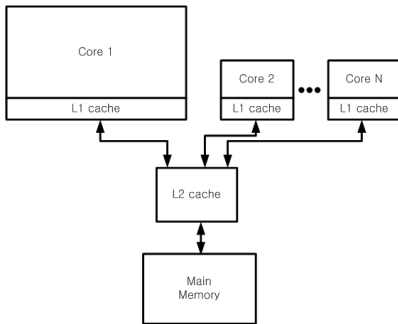


그림 2. 비대칭적 멀티코어 프로세서의 구조
 Fig. 2. The asymmetric multi-core processor architecture

한 개의 코어는 일정한 크기 이상의 쓰레드로 구성되는 비순차 (out-of-order) 수퍼스칼라 프로세서가 이용되며, 나머지는 여러 개의 작고 간단한 RISC 또는 순차 (in-order) 수퍼스칼라 프로세서가 이용된다. 표 1은 모의실험에 이용된 비대칭적 멀티코어 프로세서 아키텍처의 사양을 나타낸 것이다. 비대칭적 멀티코어의 개수는 듀얼코어에서 2 개, 쿼드코어에서 4 개를 대상으로 하였다. 각 코어는 수퍼스칼라 방식으로 운영되므로, 매 사이클 마다 1 개에서 M 개의 명령어를 인출, 이슈, 실행

및 종료한다. 실행되는 프로그램은 예측 기법을 이용하여 쓰레드로 분할되어 각 코어에 동적으로 할당된다. 이때, 마이크로프로세서에서 이용되는 2 단계 분기 예측기와 유사한 2 단계 쓰레드 예측기를 이용한다^[8].

표 1. 모의실험에 이용된 비대칭적 멀티코어 프로세서 아키텍처 하드웨어의 사양

Table 1. The architecture specification of asymmetric multi-core processor for the simulation

항목	값
멀티코어의 수	2, 4
코어의 구조	순차/비순차 수퍼스칼라
쓰레드의 크기	2, 4, 8
인출율, 이슈율, 퇴거율	2, 4
명령어 캐쉬 및 데이터 캐쉬의 공통 사항	64 KB, 2 차 연관, 16 B 미스 페널티 10 사이클
연산유닛 개수	산술논리(1/2/4/8), 분기(1), 로드(1/2), 스토어(1),
쓰레드 어드레스 캐쉬	2 K 엔트리
쓰레드 예측기	2 단계 14 비트 전역 히스토리 방식 미스 페널티 6 사이클
이슈 지연 사이클	산술논리(1), 분기(1), 로드(1), 스토어(1),
결과 지연 사이클	산술논리(1), 분기(1), 로드(1), 스토어(1),

각 코어의 연산유닛은 산술논리 유닛, 로드 스토어 유닛, 분기 유닛으로 구성된다. 한편, 각 코어는 자체적으로 1 차 명령어 캐쉬와 1 차 데이터 캐쉬를 가지며, 또한 메인 메모리와 연결되는 공통의 2 차 통합 캐쉬를 공유한다. 각 코어에 설치된 1 차 데이터 캐쉬의 일관성 (cache-coherency)을 위하여 MESI 프로토콜을 이용하여, 어느 코어에서 공유된 캐쉬 메모리에 쓰기 작업을 하였을 때, 나머지 코어에서는 해당 데이터를 무효화 (write-invalidate) 시킨다.

표 2는 본 모의실험에서 사용하는 비대칭적 듀얼코어, 쿼드코어 프로세서 내부의 각 조합을 나타낸 것이다. 본 논문에서 코어의 비대칭성은 코어의 복잡도의 차이에 따라 구분된다. 이 때, 복잡도는 코어가 처리할 수 있는 쓰레드의 최대 개수, 쓰레드당 최대 명령어의 개수 및 명령어에 대한 순차 수행 또는 비순차 수행의 여부로 결정된다. 비대칭적 멀티코어 프로세서에서는 처리할 수 있는 쓰레드의 길이가 2, 4, 8, 16이며 비순차 실행되는 수퍼스

칼라 코어를 각각 T_2, T_4, T_8, T_{16} 로 표기하였다. 또한, 쓰레드의 길이가 2, 4이고 순차 실행되는 슈퍼스칼라 코어를 각각 I_2, I_4 로 표기하였으며, 마지막으로 R은 간단한 RISC 코어를 의미한다.

표 2. 비대칭적 멀티코어 프로세서의 세부 사양
Table 2. The specific architecture configurations of asymmetric multi-core processors

코어수	비대칭적 구조	
듀얼코어	$T_2 \times 1$	$R \times 1$
	$T_4 \times 1$	$I_2 \times 1$
	$T_8 \times 1$	$I_4 \times 1$
쿼드코어	$T_4 \times 1$	$R \times 3$
	$T_8 \times 1$	$I_2 \times 3$
	$T_{16} \times 1$	$I_4 \times 3$

2. 비대칭적 멀티코어 프로세서 모의실험기

본 논문에서는 그림 3과 같은 명령어 트레이스 모의실험기를 개발하여 모의실험에 이용하였다^[9]. 비대칭적 멀티코어 프로세서는 제 1 단계 명령어 트레이스의 발생, 제 2 단계 명령어 트레이스에 대한 비대칭적 멀티코어 프로세서의 실행으로 나누어진다. 제 1 단계에서 명령어 트레이스는 SimpleScalar를 이용하여 SPEC 벤치마크 프로그램으로부터 임의의 차수의 비대칭적 멀티코어에 적합하도록 발생되었다^[10].

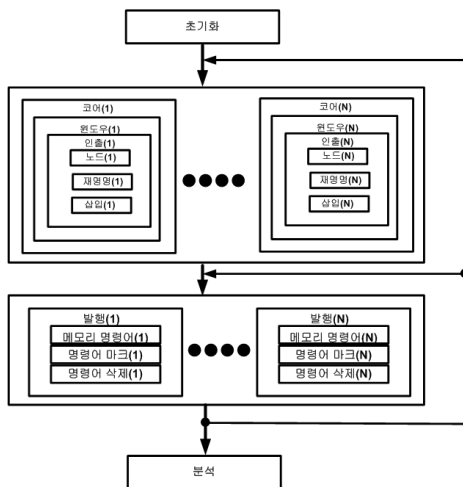


그림 3. 비대칭적 멀티코어 프로세서 모의실험기의 흐름도
Fig. 3. The flow chart of asymmetric multi-core processor simulator

이것은 제 2 단계에서 각 코어가 슈퍼스칼라로 동작하는 비대칭적 멀티코어 프로세서에 입력된다.

(1) 명령어 인출, 재명명 및 이슈

초기화 작업을 거친 후에, 각 코어는 매 사이클마다 정해진 인출율에 따라서 1 개 또는 M 개의 명령어를 인출받는다. 인출한 명령어는 재명명 (renaming) 작업을 거치면서 명령어 종속에 의한 타임스탬프(timestamp) 값을 설정받는다. 타임스탬프 방식은 명령어 자취를 이용하는 모의실험에서 데이터 종속성을 신속하고 효율적으로 부여할 수 있는 방법이다. 레지스터 화일의 타임스탬프 값에 의하여, 비대칭적 멀티코어 프로세서 명령어 간의 종속성이 유지되고 성능을 구하는데 반영된다. 명령어 디코딩 단계에서, 재명명을 거친 명령어는 각 코어의 쓰레드에 삽입되며, 사이클이 증가함에 따라서 쓰레드 내의 명령어는 자체의 타임스탬프 값이 현재 사이클 보다 작거나 같을 때 삭제될 수 있다.

(2) 비대칭적 멀티코어 시뮬레이션

N 개의 비대칭적 멀티코어에 대하여 해당 코어의 윈도우 공간에 적절하게 1 개 또는 M 개의 명령어를 인출해서 채우고, 역시 N 개의 비대칭적 멀티코어에 대하여 각 코어에 대하여 명령어를 실행하면서 종속성에 의하여 부여된 명령어의 타임스탬프가 충족되면 삭제한다. 이 과정은 코어 내부 및 코어 간의 레지스터 종속 및 메모리 종속 검사에 적용되며, 입력으로 주어진 벤치마크 프로그램의 모든 명령어가 소진될 때까지 반복된다. 위 과정이 한번 실행될 때 마다 사이클이 증가하므로, 매 사이클 당 명령어의 실행 및 삭제가 가장 오래 걸리는 코어가 전체 사이클 수를 결정한다. 모의실험에 입력으로 쓰인 명령어의 총 개수를 처리하기 위하여 소요된 총 사이클 수로 나누어, 비대칭적 멀티코어 프로세서 시스템의 성능의 척도인 IPC(Instruction Per Cycle)를 계산할 수 있다.

3. 벤치마크의 사양

표 3은 모의실험에 이용된 여섯 개의 정수형 SPEC 2000 벤치마크 프로그램이다.

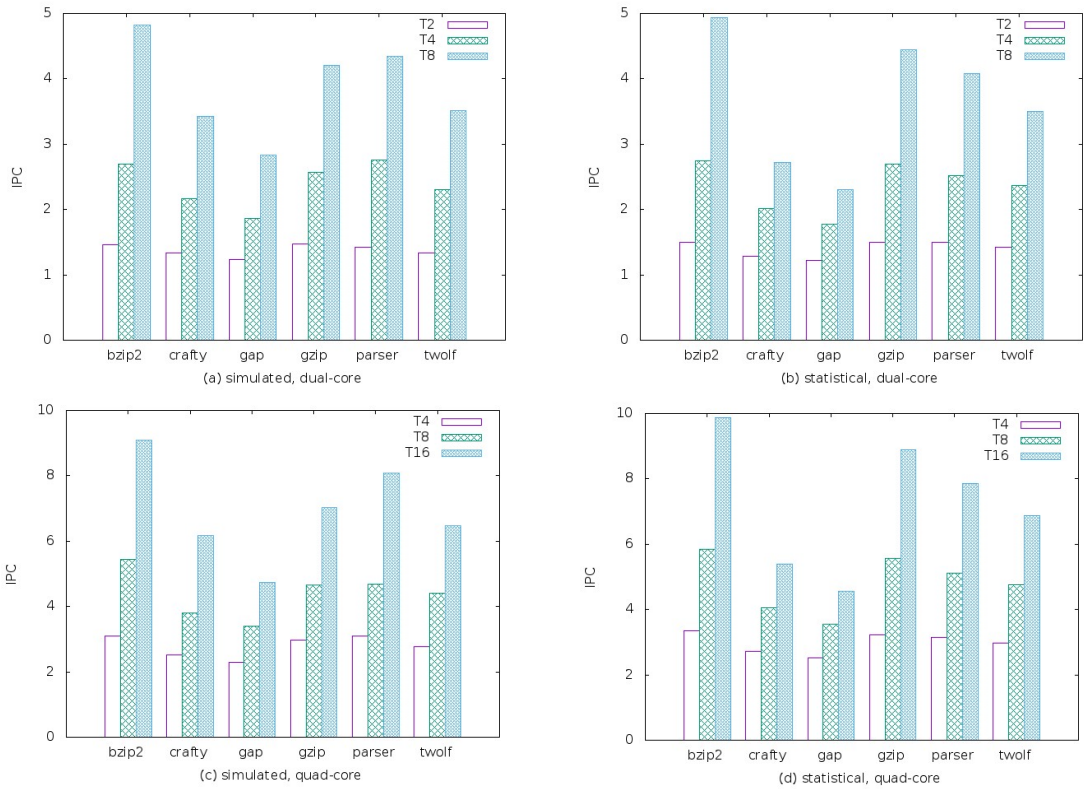


그림 4. 비대칭적 멀티코어 프로세서 구조의 명령어 트레이스 모의실험과 통계적 모의실험에 의하여 측정된 성능의 비교
 Fig. 4. The performance comparison of the trace-driven simulation and statistical simulation for asymmetric multi-core processor architectures

표 3. SPEC 2000 정수형 벤치마크 프로그램
 Table 3. SPEC 2000 integer benchmark programs

벤치마크	설명
bzip2	압축
crafty	체스 경기 놀이
gap	그룹이론 해석기
gzip	압축
parser	워드 프로세서
twolf	배선 및 배치 모의실험기

SimpleScalar를 통하여, 10억 개의 MIPS IV 명령어 트레이스를 임의의 개수의 비대칭적 멀티코어 프로세서에 적합하도록 발생시켜서 모의실험기에 입력하였다. 한편, Gcc는 명령어 캐쉬 미스에 의한 성능 저하의 폭이 크기 때문에 명령어 병렬성에 의한 고유한 성능을 제대로 얻기가 어려워서 모의실험에서 제외하였다.

IV. 모의실험 및 결과

그림 4에 세 가지의 스레드 길이에 대한 비대칭적 듀얼 코어 프로세서와 비대칭적 쿼드코어 프로세서의 성능을 명령어 트레이스 모의실험으로 측정된 값과, 본 논문에서 제안하는 통계적 모의실험에 의하여 측정된 값을 비교하여 나타냈다. 명령어 트레이스를 이용한 모의실험은 기존에 널리 사용되던 방식이므로, 일반 모의실험이라고 간단히 칭하기로 한다.

그림 4(a)와 4(b)는 비대칭적 듀얼코어 프로세서에 대하여 스레드의 길이가 2, 4, 8일 때의 모의실험 결과이다. 스레드의 길이가 2일 때, 일반 모의실험에 의하여 측정된 성능의 기하평균은 1.37 IPC를 기록하였으며, 통계적 모의실험에 의한 값은 1.40 IPC를 나타내어, 3.75 %의 상대 오차를 기록했다. 스레드의 길이가 4일 때는, 일반 모의실험에 의한 결과의 기하평균이 2.37 IPC를, 통계적 모의

실험에 의한 결과가 2.33 IPC를 나타내어 4.83 %의 상대 오차를 나타냈다. 마지막으로 쓰레드의 길이가 8일 때, 일반 모의실험에 의한 기하평균은 3.79 IPC를 나타냈고, 통계적 모의실험에 의한 결과는 3.54 PC를 나타내어, 상대오차는 다소 증가한 8.60 %를 기록하였다. 쓰레드의 길이가 증가할 수록 상대오차가 증가하였으나 상대오차가 10 % 미만임을 알 수 있다.

그림 4(c)와 4(d)는 비대칭적 쿼드코어 프로세서일 때의 일반 모의실험과 통계적 모의실험 결과를 각각 나타낸 것이다. 쓰레드의 길이가 2, 4, 8로 증가함에 따라서, 일반 모의실험에 의한 기하평균은 각 2.78, 4.42, 6.47 IPC로 측정되었으며, 이에 대응하는 통계적 모의실험에 의한 기하평균은 각 2.98, 4.75, 6.87 IPC를 나타내어, 상대오차는 7.53, 9.20, 9.96 %를 각각 기록하였다.

위에서 살펴본 것과 같이, 비대칭적 멀티코어 프로세서에 대하여 통계적 모의실험에 의하여 측정된 성능이 일반적인 명령어 트레이스 모의실험에 의하여 측정된 성능과 같은 경향을 보이며, 세 가지 쓰레드의 길이에 대하여 측정된 성능의 정확도가 평균 10 % 미만으로 비교적 우수함을 알 수 있다. 아울러, 통계적 모의실험에 소요되는 시간은 명령어 트레이스 모의실험에 소요되는 시간보다 평균 30 배 빠른 것으로 측정되었다.

V. 결론

본 논문에서는 비대칭적 멀티코어 프로세서 아키텍처에 대하여 통계적 모의실험을 통하여 성능을 측정하고 결과를 분석하였다. 통계적 모의실험은, 입력으로 이용되는 벤치마크 프로그램의 명령어 트레이스에 대한 사전 프로파일링을 통하여 합성되고 압축된 새로운 명령어 트레이스를 이용하기 때문에, 명령어 데이터 크기가 작고 모의실험 시간을 대폭 단축할 수 있다는 장점이 있다. 세 가지 서로 다른 쓰레드 길이를 갖는 비대칭적 듀얼코어 프로세서와 쿼드코어 프로세서에 대하여 모의실험을 시행한 결과, 평균 상대오차가 10 % 미만을 기록함과 동시에 모의실험 시간을 30 배 빠르게 개선시킬 수 있었다.

향 후의 연구과제로서, 매니코어 프로세서 아키텍처에 통계적 모의실험을 적용하는 것과, 비대칭적 임베디드 및 비대칭적 디지털 신호처리 멀티코어 프로세서에 대한 통계적 모의실험을 시행하는 것이다.

References

- [1] S. W. Keckler, K. Olukotun, and H. P. Hofsee, "Multicore Processors and Systems," Springer. 2009.
- [2] T. Ungerer, B. Robic, and J. Silk, "Multithreaded Processors," *The Computer Journal*, Vol. 45, No. 3, pp. 320-348, 2002
- [3] R. Kumar et al, "Single-ISA heterogeneous Multicore Architectures for Heterogeneous for Multithreaded Workload Performance," *Annual International Symposium on Computer Architecture*, pp. 64, Mar. 2004.
- [4] Hourd, Jon, et al. "Exploring Practical Benefits of Asymmetric Multicore Processors," *Workshop on Parallel Execution of Sequential Programs on Multi-core Architectures*, pp. 55, Apr. 2009.
- [5] J. Lee, "Performance Study of Asymmetric Multicore Processor Architectures," *Journal of The Institute of Internet, Broadcasting and Communication*, vol. 14, no. 3, pp. 163-169, Jun. 2014.
- [6] R. Carl and J. E .Smith, "Modeling Superscalar Processors via Statistical Simulation," *Parallel Architectures and Compilation Techniques*, pp. 15-24, Sep. 2001.
- [7] J. Lee, "The Study of Statistical Simulation for Multicore Processor Architectures" *Journal of The Institute of Internet, Broadcasting and Communication*, vol. 14, no. 6, pp. 259-265, Dec. 2014.
- [8] T-Y. Yeh and Y. N. Patt, "Alternative Implementations of Two-Level Adaptive Branch Prediction," in *Proceedings of the 19th International Symposium on Computer Architecture*, pp.124-134, May. 1992.
- [9] J. Lee, "A Study of Trace-driven Simulation for Multi-core Processor Architectures," *Journal of The Institute of Internet, Broadcasting and Communication*, vol. 12, no. 3, pp. 9-13, Jun. 2012.
- [10] T. Austin, E. Larson, and D. Ernest, "SimpleScalar : An Infrastructure for Computer System Modeling," *Computer*, vol. 35, no. 2, pp. 59-67, Feb. 2002. 3

저자 소개

이 중 복(정회원)



- 1964년 : 8월 20일생.
- 1988년 : 서울대 컴퓨터공학과 졸업.
- 1998년 : 동 대학 전기공학부 졸업 (공학박).
- 1998년~2000년 : LG반도체 선임연구원.
- 2000년~현재 : 한성대 전자정보공학과 교수

• E-Mail : jblee@hansung.ac.kr

※ 본 연구는 한성대학교 교내학술연구비 지원과제임.