

논문 2011-06-37

# ePRO-OMP: OpenMP 응용 프로그램의 성능 및 에너지 분석 도구

## ePRO-OMP: A Tool for Performance/Energy PROfiler and Analyzer for OpenMP Applications

이영호\*, 김지홍

(Young-Ho Lee, Jihong Kim)

Abstract : As chip multiprocessors have been widely adopted in embedded systems, achieving both high performance and low power consumptions of parallel applications becomes challenging. In order to meet these requirements, it is crucial for developers to analyze the performance and energy consumption of parallel applications. In this paper, we propose a tool for profiling and optimizing the performance and energy consumption of OpenMP applications (energy PROfiler and analyzer for OpenMP: ePRO-OMP). The main advantage of ePRO-OMP is that it can analyze both the performance and energy consumption of each parallel region of an OpenMP application, which can help developers find the bottleneck of parallel applications in detail.

Keywords : Embedded systems, OpenMP, Hardware performance counters, Performance, Energy

### 1. 서 론

최근 모바일 임베디드 시스템에서도 멀티코어를 채택하는 비율이 높아지고 있는데, 이러한 멀티코어 환경에서는 응용 프로그램이 다중 코어의 성능을 최대한 활용할 수 있도록 병렬성을 높이는 것이 중요하다. 이를 위해 pthread, OpenMP, Cilk 등 멀티코어를 위한 다양한 프로그래밍 모델이 제안되어 왔다. 또한 모바일 임베디드 시스템은 대부분 배터리를 기반으로 하기 때문에 데스크톱이나 서버 시스템에 비해 에너지에 대한 제약이 크다. 따라서 응용 프로그램이 최대 성능을 발휘하면서도 에너지 소모를 최소화하는 것이 중요하다.

이러한 요구사항을 만족시키기 위해서는 먼저 개발자가 작성한 병렬 응용 프로그램의 성능 및 에너지를 분석하는 것이 필요하다. 프로그램 성능의 경우, 일반적인 방법은 실행 시간을 측정함으로써 해당 응용 프로그램의 성능을 평가하는 것이다. 단, 병렬 응용 프로그램의 경우 다양한 병렬 구간으로

구성되어 있기 때문에 정확한 성능 분석을 위해서는 병렬 구간 별 성능 분석이 요구된다. 또한 소모 에너지를 알기 위해서는 전력 소모를 측정할 수 있는 특별한 기기의 도움이 필요하다 [3,4,5]. 그러나 이러한 전력 측정 도구를 사용할 경우 비용의 제약이 크고, 프로그램의 성능 측정 또는 최적화와 별개로 이루어진다는 단점이 있다.

본 논문에서는 병렬 프로그래밍 모델 중 하나인 OpenMP를 사용하는 병렬 응용 프로그램의 성능 및 에너지 분석 도구 (energy PROfiler and analyzer for OpenMP: ePRO-OMP)를 제안하고, 이를 통한 응용 프로그램의 성능 분석 및 최적화 기법을 제안한다. 기존에 제안된 도구의 경우, 성능 및 전력 분석을 모두 지원하지 못했으며, 특히 OpenMP와 같은 고수준 (high-level) 병렬 프로그래밍 모델을 지원하지 제안하는 단점이 있었다. 예를 들면, 병렬 프로그램에서는 각 병렬 구간 별 성능 및 전력 정보를 분석하는 것이 매우 중요하다. 반면, 제안하는 도구는 OpenMP로 작성된 응용 프로그램의 각 병렬 구간 별 성능 및 에너지 소모 정보를 제공함으로써, 개발자가 병렬 응용 프로그램의 성능 및 에너지 소모를 최적화할 수 있도록 돕는다. 또한 에너지 소모를 분석하기 위한 방법으로 하드

\* 교신저자 (Corresponding Author)

논문접수 : 2011. 02. 09., 수정일: 2011. 02. 23.,

채택확정 : 2011. 04. 07.

이영호, 김지홍 : 서울대학교 컴퓨터공학과

웨어 성능 카운터 (Hardware Performance Counter) 기반 기법을 활용함으로써 추가적인 하드웨어 없이 병렬 응용 프로그램의 에너지 소모를 예측한다.

또한 본 논문에서는 실제 임베디드 멀티코어 프로세서 중 하나인 MPCore 기반 개발환경 상에서 ePRO-OMP를 활용한 다양한 OpenMP 응용 프로그램의 성능/에너지 분석 결과를 제시하고, 응용 프로그램 수준에서의 성능/에너지 최적화 방안에 대해서도 논의한다. 특히, 병렬 프로그램의 병렬성 (i.e., 동시에 수행되는 thread의 개수)을 높이는 것이 에너지 측면에서는 최적이지 않음을 보임으로써, 에너지 소모를 최소화하기 위해 개발자가 OpenMP 프로그램 작성 시 최적의 병렬성을 명시하는 것이 중요함을 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 성능 및 에너지 분석 도구 (ePRO-OMP)의 전체적인 구조를 설명하고, 3장에서는 ePRO-OMP에서 활용한 성능 및 에너지 분석 기법에 대해 살펴본다. 4장에서는 MPCore 대상 환경에서의 OpenMP 응용 프로그램 성능 및 에너지 분석 결과를 보인다. 마지막으로 5장에서는 결론 및 향후 연구 방향을 제시한다.

## 2. ePRO-OMP 구조

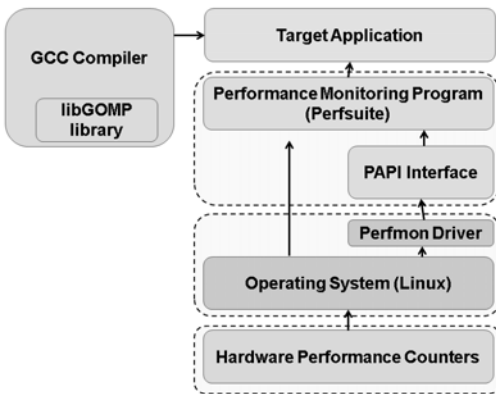


그림 1. ePRO-OMP 구조

Fig. 1. An overview of ePRO-OMP

그림 1은 ePRO-OMP의 전체 구조를 나타낸다. ePRO-OMP는 크게 운영체제 확장 부분 (Hardware Performance Counter 드라이버), 성능

및 에너지 분석 부분 (perfsuite 및 PAPI), gcc 컴파일러 등으로 구성되며, OpenMP 응용 프로그램의 성능 및 에너지 분석을 위해 하드웨어 성능 카운터를 활용한다.

### 1. 운영체제 확장 부분

일반적으로 사용자 영역에서는 하드웨어 성능 카운터에 직접 접근할 수 없기 때문에 운영체제 수준에서 디바이스 드라이버 등을 통한 지원이 필요하다. 이를 위해 ePRO-OMP에서는 하드웨어 성능 카운터를 측정하기 위한 perfctr [9] 디바이스 드라이버를 추가되었다.

### 2. 성능/에너지 분석 부분

성능 및 에너지 분석 부분의 경우, 응용 프로그램의 성능 및 에너지를 함수별/쓰레드별로 분석하기 위해 수정된 버전의 perfsuite [8] 응용 프로그램 도구를 사용한다. perfsuite의 경우, 병렬 응용 프로그램이 수행될 때 각 코어 (스레드) 별 하드웨어 성능 카운터 정보를 측정하는 기능을 제공한다. 사용자는 OpenMP 응용 프로그램의 성능/에너지를 측정하기 위해 perfsuite 응용 프로그램 도구 중 psrun 프로그램을 수행하게 되는데, psrun 프로그램은 대상 OpenMP 응용 프로그램을 호출하고 해당 프로그램의 성능/에너지를 측정하여 분석 결과를 xml 파일로 출력한다.

기존의 perfsuite 응용 프로그램 도구는 단일 스레드 및 pthread 라이브러리로 작성된 병렬 프로그램에 대해 하드웨어 성능 카운터를 측정할 수 있도록 구현되어 있기 때문에 OpenMP 응용 프로그램의 병렬 구간 별 하드웨어 성능 카운터를 측정하는 데는 적합하지 않다. 가장 큰 이유는 대부분의 OpenMP 컴파일러에서 사용하는 스레드 풀 기법 [11] 때문이다. 대부분의 OpenMP 컴파일러에서는 각 병렬 구간이 시작될 때마다 내부적으로 pthread 라이브러리를 호출하여 여러 스레드를 생성하게 되는데, pthread 라이브러리 호출 오버헤드를 줄이기 위해 병렬 구간이 종료하더라도 스레드를 종료하지 않고 스레드 풀에 저장한 후, 이후 병렬 구간에서는 스레드 풀에 있는 스레드를 활용한다. 스레드 풀은 프로그램 종료 시 해제된다. 따라서 기존의 perfsuite 응용 프로그램 도구를 활용할 경우, OpenMP 응용 프로그램이 여러 개의 병렬 구간으로 구성되어 있더라도 하나의 병렬 구간으로 구성되어 있다고 가정하기 때문에 올바른 성능/에너지 정

보를 얻기 어렵다. 따라서 ePRO-OMP에서는 스레드 시작/종료 시점이 아니라 각 병렬 구간이 시작/종료할 때마다 하드웨어 성능 카운터를 측정하도록 perfsuite 응용 프로그램 도구 및 컴파일러 도구를 수정하였다. 또한 성능/에너지 분석을 위한 하드웨어 성능 카운터 정보를 운영체제에 전달하기 위해 PAPI 인터페이스를 사용한다.

### 3. gcc 컴파일러 부분

ePRO-OMP에서는 OpenMP 응용 프로그램의 컴파일을 위해 gcc 컴파일러 (버전 4.5)를 사용한다. 최근 배포된 gcc 컴파일러 (버전 4.1 이상)의 경우 OpenMP 응용 프로그램의 컴파일 기능 및 실행 라이브러리 (i.e., libgomp) 등을 포함하는데, ePRO-OMP에서는 OpenMP 응용 프로그램의 병렬 구간 별 성능/에너지 분석을 위해 다음과 같이 gcc 컴파일러 일부를 수정하였다. 실행 라이브러리인 libgomp 내부에서 각 병렬 구간이 시작/종료할 때마다 perfsuite의 프로파일링 관련 함수 (e.g., 프로파일 시작/종료 등)를 호출함으로써, 각 병렬 구간에 대해 스레드 별로 하드웨어 성능 카운터를 측정할 수 있도록 했다.

### 3. ePRO-OMP를 활용한 성능 및 에너지 분석

ePRO-OMP에서는 OpenMP 응용 프로그램의 성능 및 에너지를 분석하기 위해 하드웨어 성능 카운터를 활용한다. ePRO-OMP를 사용하여 OpenMP 응용 프로그램을 수행한 후 사용자는 해당 OpenMP 프로그램의 병렬 구간 별 하드웨어 성능 카운터를 얻게 된다. 예를 들어 임의의 OpenMP 응용 프로그램이 2개의 병렬 구간으로 구성되어 있고, 각 병렬 구간은 4개의 스레드로 나누어 동작한다고 가정할 때, 응용 프로그램 수행 후에는 총  $(N/2) \times (1 + 3 \times 2)$ 개의 xml 파일이 출력된다. 이때 N은 측정하고자 하는 하드웨어 성능 카운터의 개수이다. 본 논문에서 대상으로 하는 ARM MPCore의 경우 각 코어마다 동시에 2개씩의 하드웨어 성능 카운터를 측정할 수 있는 반면, ePRO-OMP에서는 OpenMP 응용 프로그램의 성능/에너지 분석을 위해 8개의 하드웨어 성능 카운터를 요구한다. 따라서 ePRO-OMP에서는 응용 프로그램을 N/2회만큼 반복하여 수행하여 전체 하드웨어 성능 카운터 정보

를 얻게 된다. 또한 매 수행마다 1개는 마스터 스레드에 대한 파일이, 나머지는 2개의 병렬 구간에 대한 각 스레드 (3개)에 대한 파일이 생성된다  $(1 + 3 \times 2)$ .

성능 분석은 각 병렬 구간 별 cycle 값과 master thread의 cycle 값을 통해 분석하였다.

에너지 분석의 경우, 하드웨어 성능 카운터를 활용한 에너지 예측 기법을 활용하였다. 기본 아이디어는 프로세서의 소모 전력을 전력 소모와의 상관관계 (correlation)가 매우 높은 몇 개의 하드웨어 성능 카운터의 가중치 합 (weighted sum)으로 나타낼 수 있다는 것이다. 예를 들면, 단위 시간당 수행된 명령어 수는 전력 소모와 비례 (strong positive correlation)하는 반면, 메모리 접근 횟수나 파이프라인 스톨 등은 전력 소모와 반비례하는 경향 (strong negative correlation)이 있다. 본 논문에서는 ARM MPCore를 대상으로 한 에너지 모델 [7]을 활용한다. [7]에서는 전력 소모와의 상관관계가 높은 5개의 이벤트를 활용하여 전력 소모를 제안하였다.

$$Power = A \times (Instr/time) + B \times (DL1Access/time) + C \times (L2Access/time) + D \times (DataDep/time) + E \times (CohTrans/time) + F_{const}$$

식 1. MPCore 전력 모델

Equ. 1. MPCore power model

제안한 전력 모델은 식 1과 같으며, 식 1의 각 가중치 A~F는 표 1에 나타나 있다.

표 1. MPCore 전력 모델 계수

Table 1. Coefficient values of MPCore power model

| 이벤트       | 계수        |
|-----------|-----------|
| Instr     | 2.56E-07  |
| DL1Access | 9.32E-07  |
| L2Access  | 2.76E-05  |
| DataDep   | -3.97E-07 |
| CohTrans  | -1.99E-07 |
| Const     | 7.89E+ 02 |

### 4. 실험 결과

본 절에서는 ePRO-OMP를 활용한 OpenMP 응용 프로그램의 성능 및 에너지 분석 결과를 기술한다.

#### 1. 실험 환경

실험 환경은 다음과 같다. 먼저 하드웨어는 임베디드 멀티코어 중 하나인 4코어 ARM MPCore [6]가 탑재된 개발보드를 사용하였다. 프로세서는 210MHz로 동작하며, 32KB 코어 별 L1 캐시, 1MB 공유 L2 캐시로 구성된다. 또한 각 프로세서 코어는 2개씩의 하드웨어 성능 카운터를 동시에 측정할 수 있다. 그림 2는 하드웨어 환경을 보인다.

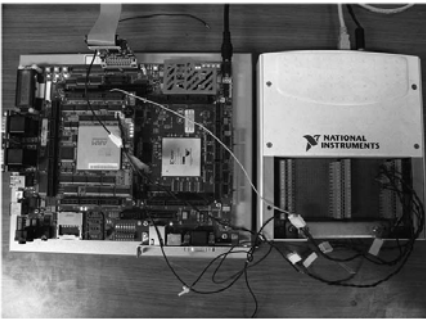


그림 2. ARM MPCore 기반 환경  
Fig. 2. ARM MPCore-based platform

소프트웨어 환경은 다음과 같다. 운영체제는 Linux 2.6.23을 사용하였으며, 대상 환경 내에 ePRO-OMP를 구현하였다. 단, 하드웨어 성능 카운터 접근을 위한 perfctr 디바이스 드라이버의 경우 ARM MPCore 프로세서를 지원하지 않기 때문에, 본 논문에서는 perfctr 디바이스 드라이버를 ARM 프로세서에 맞게 포팅하여 사용하였다.

대상 응용 프로그램은 행렬 곱셈, 병합 정렬 및 OpenMP용으로 변환된 버전의 NPB 벤치마크 [10]를 사용하였다.

#### 2. 성능 분석

그림 3과 4는 각각 행렬 곱셈 및 병합 정렬 OpenMP 응용 프로그램을 수행하였을 때의 성능 분석 결과를 보인다. 그래프 상에서 x축은 병렬화 정도(i.e., 스레드 개수)를, y축은 각 병렬 구간에서 소모된 cycle 값을 나타낸다. 병렬화 정도가 1인

경우는 단일 스레드로 수행하였을 때를 나타낸다. 분석 결과를 보면, 두 프로그램 모두 하나의 병렬 구간으로 이루어진 것을 알 수 있으며, 전체 수행 시간 중 각 병렬 구간이 차지하는 비율도 확인할 수 있다. 또한 병렬화 정도가 커질수록 각 병렬 구간에서 소모하는 cycle 수가 스레드 개수에 거의 비례하게 감소한 것을 통해 병렬성이 높은 응용 프로그램임을 확인할 수 있다.

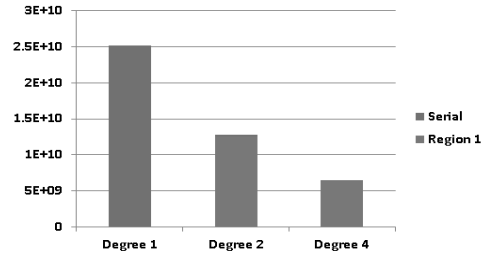


그림 3. 행렬 곱셈 프로그램의 성능 분석 결과  
Fig. 3. Performance analysis of matrix multiplication application

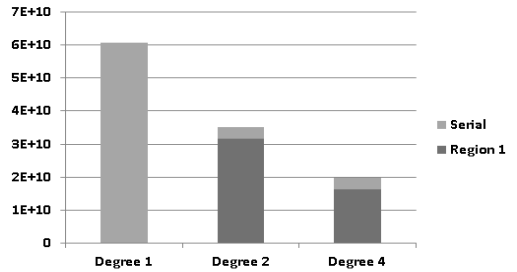


그림 4. 병합 정렬 프로그램의 성능 분석 결과  
Fig. 4. Performance analysis of merge sort application

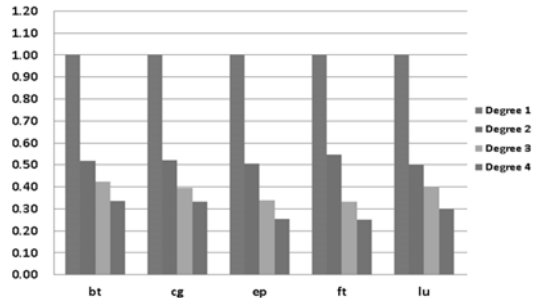


그림 5. NPB 벤치마크 성능 분석 결과  
Fig. 5. Performance analysis of NPB benchmark

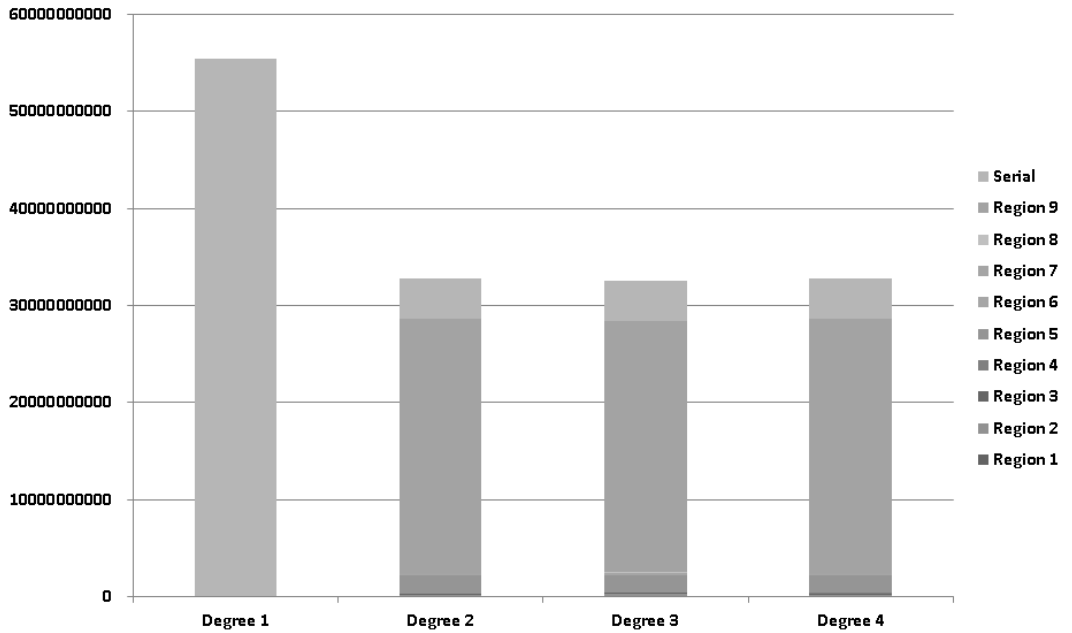


그림 6. mg 벤치마크의 세부 성능 분석 결과

Fig. 6. A detailed performance analysis of mg benchmark

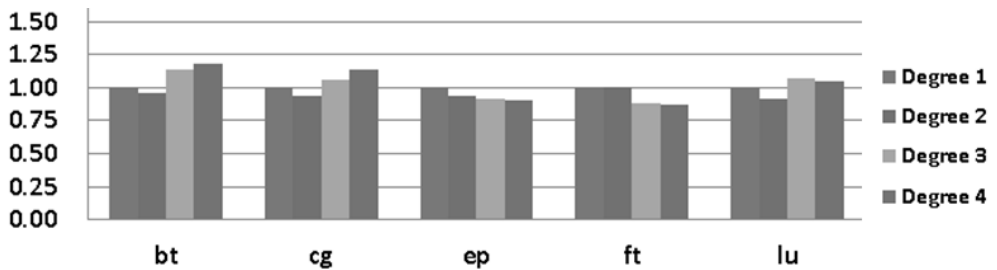


그림 7. NPB 벤치마크의 에너지 분석 결과

Fig. 7. An analysis of energy consumption of NPB benchmark

그림 5는 NPB 벤치마크의 성능 분석 결과를 나타낸다. 그림 4에서 x축은 각 벤치마크를 나타내며, y축은 병렬화 정도가 1일 때 (i.e., 단일 스레드)의 전체 수행시간 대비 수행 시간을 나타낸다. ep나 ft 벤치마크의 경우 병렬화 정도가 증가함에 따라 성능이 비례하는 것을 확인할 수 있으며, bt 벤치마크의 경우 성능 증가가 높지 않음을 확인할 수 있다.

그림 6은 mg 벤치마크의 성능 분석 결과를 병렬 구간 별로 보여준다. 그림 6을 통해, mg 벤치마크는 9개의 병렬화 구간으로 구성된 것을 확인할 수 있다. 실제 분석 결과를 보면 직렬 구간 (serial)의 비율은 매우 낮음을 확인할 수 있다. 그러나 병

렬화 수준을 높더라도 각 병렬 구간에서의 성능 증가가 거의 이루어지지 않아 병렬화 정도가 2 이상인 경우 성능 향상이 거의 발생하지 않음을 확인할 수 있다.

### 3. 에너지 분석

그림 7은 NPB 벤치마크의 에너지 분석 결과를 나타낸다. 그림 7에서 x축은 각 벤치마크를 나타내며, y축은 병렬화 정도가 1일 때 (i.e., 단일 스레드)의 전체 에너지 소모 대비 에너지 소모를 나타낸다. 즉, y축 값이 낮을수록 에너지 소모가 낮음을

의미한다. 그림 7은 병렬성을 높이는 것이 에너지 소모 측면에서 최적이지 않음을 보여준다. ep 벤치마크의 경우에는 병렬성이 높아짐에 따라 에너지 소모가 감소하였으며, 병렬성을 최대 (degree 4)로 할 경우 에너지 소모가 최소로 나타났다. 반면, bt, cg 벤치마크의 경우에는 병렬성을 최대 (degree 4)로 할 경우 에너지 소모가 가장 높은 것으로 나타나 병렬성을 최대한으로 하는 것이 에너지 측면에서는 좋지 않음을 알 수 있다.

표 2. NPB 벤치마크의 세부 에너지 분석 결과  
Table 2. A detailed analysis of energy consumption of NPB benchmark

| 벤치마크 | 최적 병렬성 정도 | 에너지 감소 |
|------|-----------|--------|
| bt   | 2         | 22%    |
| cg   | 2         | 19%    |
| ep   | 4         | 10%    |
| ft   | 4         | 14%    |
| lu   | 2         | 15%    |

표 2는 그림 7의 결과를 세부적으로 나타낸 것으로, 에너지 소모가 가장 낮을 때의 병렬성 정도와, 에너지 소모 정도를 나타낸 것이다. 대부분의 벤치마크에서 병렬성을 2로 할 때 에너지 소모가 최소가 되었으며, 일부 벤치마크 (e.g., ep, ft)의 경우에만 병렬성을 최대한으로 할 때 에너지 소모가 가장 적었다. 표 2로부터 병렬성 정도에 따라 최대 22%까지 에너지 소모가 감소할 수 있음을 확인할 수 있다. 이는 OpenMP 응용 프로그램의 에너지 효율을 최적화 하기위해 최적의 병렬성을 찾는 것이 매우 중요함을 보여준다.

## 5. 결론

본 논문에서는 OpenMP 응용 프로그램의 성능 및 에너지 분석 도구 (ePRO-OMP)를 제안하고, 이를 활용하여 다양한 응용 프로그램의 병렬 구간 별 성능 분석 결과를 보였다. 또한 분석 결과를 통해, 병렬 응용 프로그램의 성능 및 에너지 효율을 높이기 위해서는 응용 프로그램에서의 병렬 비율을 높이는 것도 중요하지만, 각 병렬 구간 내에서의 성능 및 에너지 효율을 높이는 것이 더 중요함을 보였다.

향후 OpenMP 응용 프로그램의 에너지 최적화

를 위한 온라인 및 오프라인 기법을 연구할 예정이다.

## 감사의 글

본 논문은 BK21사업에 의하여 지원되었으며, 정부(교육과학기술부)의 재원으로 한국과학재단과 한국연구재단-미래기반기술개발사업(첨단융합분야)의 지원을 받아 (No. 20110020426, No. 2011-0020514) 수행되었습니다. 본 연구를 위해 연구장비를 지원하고 공간을 제공한 서울대학교 컴퓨터 연구소와 IDEC에 감사드립니다.

## 참고문헌

- [1] H. Blume, J. von Livonius, L. Rotenberg, OpenMP-based parallelization on an MPCore multiprocessor platform - A performance and power analysis, *Journal of Systems Architecture*, 2008.
- [2] R. Chandra, *Parallel Programming in OpenMP*, Morgan Kaufmann, 2001.
- [3] W. Baek, Y. Kim and J. Kim, ePRO: A tool for energy and performance profiler for embedded applications, in: *Proceedings of International SoC Design Conference*, Seoul, Korea, 2004, pp. 372-375
- [4] C. Hsu and W. Feng, A feasibility analysis of power awareness in commodity-based high-performance clusters, in: *Proceedings of International Conference on Cluster Computing*, Boston, MA, pp. 1-10, 2005.
- [5] D. Shin, H. Shim, Y. Joo, H. Yun, J. Kim and N. Chang, Energy-monitoring tool for low-power embedded programs, *Design and Test of Computer* 19(4), pp. 7-17, 2002.
- [6] ARM11 MPCore, available at: <http://www.arm.com/products/CPUs/ARM11MPCoreMultiprocessor.html>.
- [7] W. Choi, H. Kim W. Song, J. Song, J. Kim, ePRO-MP: A tool for profiling and optimizing energy and performance of mobile multiprocessor applications, *Scientific Programming*, 2009.

- [8] R. Kuftrin. perfsuite: An Accessible, Open Source Performance Analysis Environment for Linux. In Proc. of the Linux Cluster Conference, Chapel Hill, North Carolina, April 2005.
- [9] M. Pettersson. Perfctr: Linux Performance Monitoring Counters Driver. Technical report, Computing Science Department, Uppsala University, Jan. 2005. <http://user.it.uu.se/mikpe/linux/perfctr>
- [10] NAS Parallel Benchmarks, available at: <http://www.nas.nasa.gov/Resources/Software/npb.html>
- [11] Novillo, D.: OpenMP and automatic parallelization in GCC. In: Proc. of the 2006 GCC Summit, 2006.
- [12] 이영호, 김지홍, "하드웨어 성능 카운터를 활용한 Open MP 응용 프로그램의 성능/전력 분석 및 최적화", 대한임베디드공학회 추계학술대회, 2010.

**저 자 소 개**

**이 영 호**



2005년 : 국민대학교  
전산학과 학사.  
2007년 : 국민대학교  
전산학과 석사.  
현재, 서울대학교  
컴퓨터공학과 박사과정.

관심분야 : 저전력/저발열 스케줄링 기법,  
멀티코어 시스템.

Email : buriburi205@davinci.snu.ac.kr

**김 지 홍**



1986년 : 서울대학교  
전산통계학과 학사.  
1988년 : University of  
Washington 컴퓨터공학과  
석사.  
1995년 : University of  
Washington 컴퓨터공학과  
박사.

현재, 서울대학교 컴퓨터공학과 교수.  
관심분야 : 임베디드 소프트웨어, 저전력 시스템,  
컴퓨터 구조.

Email : jihong@davinci.snu.ac.kr