논문 2014-09-39

이종 멀티코어 시스템의 전력 및 성능 분석을 위한 프레임워크 설계 및 구현

(A Systematic Power and Performance Analysis Framework for Heterogeneous Multiprocessor System)

김 형 준, 경 주 현, 임 성 수*
(Hyeong-Jun Kim, Joohyun Kyong, Sung-Soo Lim)

Abstract : Mobile computing devices such as smartphones, tablet computers have become the dominant personal computing platforms. Energy efficiency is a prime design requirement for smart devices. In order to reduce the energy consumption of the smart devices, analysis of performance and energy consumption has become important. However, so far, there is no framework for the analysis and systematic approach to improve the power consumption of the heterogeneous multi-core system. In this paper, we describe a new framework for the analysis of heterogeneous multi-core systems. Also, by use of an analysis tool, can be provide reliability and productivity of development results.

Keywords : Heterogeneous multiprocessor, Embedded system, System analyzer

I. 서 론

리눅스 기반의 스마트 디바이스에 대한 사용이 늘어남에 따라, 성능 개선과 더불어 전력 소모량 개선을 위해 많은 연구 및 개발을 하고 있다. 그동안 대부분의 전력 소모량의 개선은 하드웨어 관점으로 이루어 졌으나 최근 고성능 저전력의 이종 멀티코어 시스템이 등장하면서, 전력에 효율적인 스케줄러 개발[1]등 소프트웨어 관점으로 변경되고 있다. 이러한 이종 멀티코어 시스템은 고성능과 저전력에 모든 이점을 가지기 위해 고성능 코어와 저전력 코어로 구성되어 있는 특징이 있다[2]. 이처럼 이종 멀티코어 아키텍처의 등장과 더불어 전력 소모량 최적화를 위한 스케줄러 개선 등 시스템 소프트웨어적인 중요성이 증가 되었다. 동시에 이종 멀티코어 시스템 분석을 위한 툴에 대한 중요성이 증가되고 있다. 하지만 현재까지 시스템 소프트웨어의 전력 소모량을 개선하기 위한 환경과 도구에 대한 지원이 부족한 것이 현실이다. 그동안 전력 소모량을 개선하기 위해서, 기존 커널에서 지원하는 CPU 사용량을 분석하는 툴[3]과 커널의 각종 트레이스 툴[4]이 사용되고 있다. 작업 과정은 전력과 성능을 분석하기 위해서 많은 복잡한 과정을 거치게 된다. 먼저 소프트웨어 관점으로 성능 분석도구를 사용하여, 시스템을 분석 후 스케줄러 개선 등 시스템 레벨의 전력 개선을 수행한다. 개선된 결과를 확인하기 위해, 외부 전력 측정 장비를 이용해 특정 워크로드의 전력 소모량에 대한 통계를 구한다. 즉, 개발자가 시스템 상태를 성능 분석도구를 통해 먼저 확인해야 하고 같은 상황에 대해 전력 분석을 위해 다시 전력 측정해야 하는 추가적인 오버헤드를 준다. 특히 복잡한 사용자 입력에 의존한 응용프로그램 같은 경우, 매번 측정결과가 상이함에 따라, 해당 응용프로그램의 전력 개선 작업을 수행하기에는 힘든 문제점이 있다. 이처럼 전력 성능에 대한 체계적인 분석 방법이 없고, 이를 위한 분석도구의 부재로 인해, 전력 소모를 줄이기 위한 효율적인 개발이

*Corresponding Author(sslim@kookmin.ac.kr)
Received: 4 Feb. 2014, Revised: 25 Mar. 2014, 1 May 2014, Accepted: 9 Aug. 2014.
H.-J. Kim, J. Kyong, S.-S. Lim: Kookmin University
* 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No. 2010-0020731)
* 본 연구는 미래창조과학부가 지원한 2014년 기술혁신사업 (No. 10041752)의 연구결과로 수행되었음.

 2014 Dec.: 9(6) 315-321
ISSN : 1975-5066
http://dx.doi.org/10.14372/IEMEK.2014.9.6.315

힘든 상황이다.

최근 전력 분석을 위해 자동화된 툴을 개발하는 연구[5]가 진행되고 있으나 이러한 연구는 전력 측정을 위한 자동화 방법에 대해 초점을 두고 있는 문제점을 가지고 있다. 즉 성능과 에너지 소모량에 대한 통계적인 수치로 보여주기에는 한계가 있다. 이러한 문제점을 해결하기 위해 본 연구에서는 이종 멀티코어에 최적화된 분석 도구를 개발 하였다. 제안하는 분석 도구는 성능과 전력을 동시에 분석할 수 있도록 지원 하였으며, 통계적인 분석 결과를 통해 소프트웨어 개발자가 전력과 성능을 동시에 효율적으로 분석 할 수 있도록 지원해준다. 즉 본 연구의 결과를 통해 개발 생산성을 향상 가져왔으며, 실측기반의 다양한 결과를 가지고 통계적 기법을 적용하므로, 분석 결과에 대한 가독성과 결과물에 대한 신뢰성을 향상 시켰다.

이종멀티코어는 상업적으로 제공되는 Inter의 Sandy Bridge[6], AMD의 Fusion[7], NVidia의 Tegra[8] 등 GPU와 CPU를 통합하는 것과, NVidia의 Kal-El[9], ARM의 big.LITTLE[2] 등 다른 타입의 CPU를 통합하는 등 다양한 종류의 시스템을 포함할 수 있다. 본문에서 고려하는 이종멀티코어 시스템은 고성능-저전력 코어로 이루어진 다른 타입의 CPU를 통합한 것을 의미한다.

Ⅱ. 분석 도구 프레임워크

본 논문에서 고려하는 이종멀티코어 시스템은 고성능 코어와 저전력 코어를 사용하고 있다. 고성능 코어와 저전력 코어를 사용하는 이유는 높은 성능을 필요로 할 경우 전력 소모량이 높은 고성능 코어를 사용하도록 하고, 그 외의 경우 전력 소모량이 낮은 저전력 코어를 사용하게 함으로써 효율적인 전력 관리를 하고자함에 있다. 위와 같은 이종멀티코어의 최종적인 목표는 결국 고성능 시스템을 유지하며 전력 효율성은 높이고자 하는 것이다. 즉 분석도구는 시스템의 성능 지표를 나타낼 필요성이 있고, 또한 성능 지표 당 전력 소모량을 표기해야 한다. 따라서 우리는 성능지표를 위해 시스템의 CPU 정보를 사용한다. 전력의 경우 디바이스 전체적으로 소모한 전력과 부분적으로 소모한 전력의 소비내역을 사용한다.

본 논문에서 제안하는 이종 멀티코어 시스템의 전력 및 성능 분석을 위한 프레임워크를 도식화한 모습은 그림 1과 같다. 프레임워크는 전체 실행 흐름을 관리하고 스마트 디바이스의 성능, 즉 시스템 CPU 정보를 측정하기 위한 컨트롤 모듈 (Control Modules), 전력을 측정하는 전력 서비스 모듈 (Power Service Modules), 결과물을 분석하는 시스템 분석도구(System Trace Analyzer) 그리고 스마트 디바이스와 전력 측정 기기로 구성된다. 시스템 개발자는 컨트롤 모듈을 통해 전체 프레임워크의 실행 흐름을 관장한다. 분석 도구의 전체 실행흐름은 다음과 같다. 먼저 개발자에 의해 분석 도구가 시작되면, 컨트롤 모듈은 스마트 디바이스에 워크로드를 실행하면서 성능 측정을 시작하는 동시에 전력 서비스 모듈에게 전력 측정을 명령한다. 전력 서비스 모듈은 전력 측정기기를 실행시키고, 결과를 수집하며, 워크로드가 종료되면, 결과를 컨트롤 모듈에게 전송한다. 마지막으로 컨트롤 모듈은 수집된 전력 및 성능 데이터를 시스템 분석도구에게 전송하고, 시스템 분석도구에서 데이터를 시각화 및 통계 작업을 통해 시스템 개발자에게 유용한 정보를 보여준다. 본 장에서는 각 단계를 상세히 정의하고, 정의된 기능 및 실행 흐름을 기술한다.

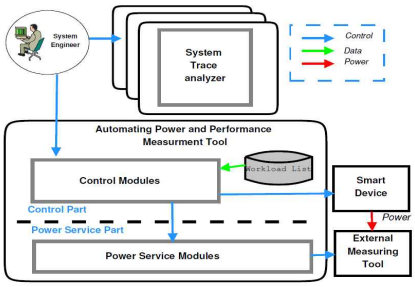


그림 1. 분석 도구 프레임워크
Fig. 1 System Analyzer Framework

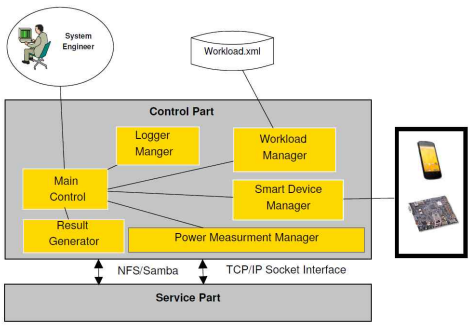


그림 2. 제어 모듈 구성도
Fig. 2 Control Module Structure

표 1. 메인 제어 명령어 정의
Table 1. Main Control Command List

제어 명령어	기능
Run	시작
Generate_workload	분석 시나리오 생성
Usb_on/off	스마트 디바이스 USB 연결/차단
Power_on/off	스마트 디바이스 전원 on/off
Exit	종료

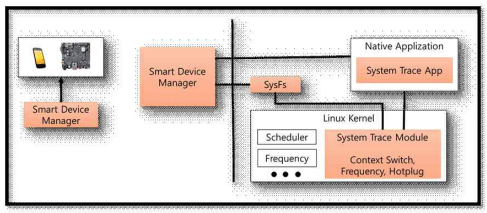


그림 3. 디바이스 내부 측정 모듈 구조도
Fig. 3 Device Embedded Module Structure

제어 모듈의 내부 구조는 그림 2와 같다. 내부적으로 사용자 인터페이스를 담당하고 시스템 전체 모듈을 호출하는 메인 제어(Main Control) 모듈이 있다. 분석 도구가 실행될 경우 먼저 실행되는 모듈이며, 명령을 커맨드 라인을 통해 입력을 받는 구조로 되어 있다. 명령을 통해 제어할 수 있는 기능은 표 1과 같이 정의한다. Run 명령은 스마트 디바이스에게 지정된 워크로드를 수행시키고 전력/성능 분석을 시작한다. 이때 워크로드는 Generate_workload 명령을 통해 만들어 진다. 시나리오를 저장하고, 관리하는 것은 워크로드 관리(Workload Manager) 모듈을 통해 이루어진다. 어떤 애플리케이션을 수행할 것인지 어떤 순서로 어떤 작업을 수행할 것인지 저장한다.

그림 2의 스마트 디바이스 관리(SmartDevice Manager) 모듈은 실제 디바이스에 명령을 내리는 것을 담당한다. ADB를 통해 워크로드를 수행시키며, 동시에 스마트 디바이스 내부적으로 성능 측정을 시작하도록 한다. 스마트 디바이스 내부의 성능 측정 프레임워크는 그림 3과 같다. 성능을 나타내는 내부 정보를 수집하는 모듈을 장치에 직접 삽입하고, 이를 가상파일시스템(sysfs)을 통해 제어한다. 수집된 데이터들은 메모리에 저장되며 수집이 완료되면 애플리케이션을 통해 다시 제어 모듈에 전송된다.

성능 측정에 대한 시작을 명령하는 동시에 전력 측정 관리(Power Measurement Manager) 모듈이

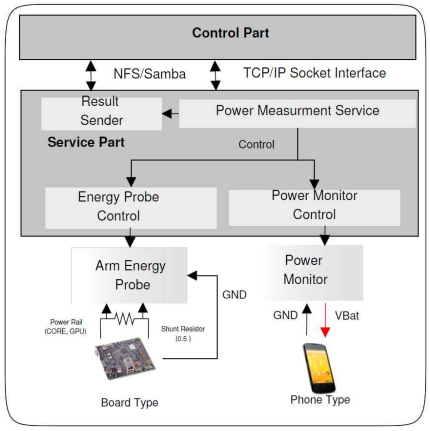


그림 4. 서비스 모듈 구조도
Fig. 4 Service Module Structure

서비스 모듈 파트에 전력 측정 시작을 명령한다. 모든 측정이 끝나면 결과 생성자 (Result Generator) 모듈에서 데이터들을 수집하고, 각 결과를 분석이 쉬운 형태로 재가공하여 저장한다.

서비스 모듈의 내부 구조는 그림 4과 같다. 전력기기를 제어하는 부분과 결과를 전송하는 부분으로 구성되어 있다. 전력 분석이란 전력을 어느 시점에 어느 위치에서 어떻게 소비되었는지를 파악하는 작업이다. 전력 측정 종류는 측정 대상과 방법에 따라서 하드웨어 서브 시스템에 대한 전력 소비를 측정하는 방법[10], 모든 전력 소비를 측정하는 방법[11], 어플레이케이션을 통한 전력 소비 측정 방법[12], 실시간으로 전력을 측정하고 수집하는 방법[13] 그리고 안드로이드 운영체제에 전력 모니터링 기능을 추가하는 방법[13] 등으로 분류할 수 있다.

본 논문에서 제안하는 전력 측정 방법은 크게 2가지로 나뉜다. 첫 번째는 스마트 디바이스 기기의 배터리로부터 입력되는 전류를 측정하는 방법이다. 이는 디바이스가 사용하는 전체 전력을 의미한다. 두 번째는 레일을 이용하여 원하는 부분에 대한 전력을 측정하는 방법이다. 디바이스를 이루는 각 구성요소가 소비하는 전력을 의미한다. 부분 전력 측정의 경우 디바이스에 레일을 설치하는 추가적인 작업이 필요하며, 해당 디바이스가 이를 지원해야 한다.

제어 모듈로부터 전력 측정 시작 명령을 받으면 전력 측정 서비스 모듈은 각 전력 측정 기기를 제어하는 모듈을 동작한다. 그림 4에서 ARM Energy Probe, Power Monitor는 전력 측정 기기명을 의미한다. 전체 전력과 부분 전력을 동시에 측정하며 위

크로드가 동작하는 동안 측정된 결과를 결과 전송자(Result Sender) 모듈을 통해 제어 모듈로 전송한다. 전력 측정 모듈이 제어 모듈의 내부에 포함되어 있지 않고, 분리되어 있는 이유는 다음과 같다. 상당히 많은 전력 측정 장비가 운영체제에 의존적이기 때문이다. 즉 운영체제를 측정기기에 맞추어 구현해야 하기 때문에 Linux/Window 등 플랫폼 독립성을 위해 전력 측정 모듈을 분리하여 설계하였다. 본 논문에서는 전력 측정 장치 중 Arm Energe Probe는 Linux/Window 모든 지원이 가능했으나 Power Monitor의 경우 Window에서만 사용이 가능했기 때문에 서비스 모듈은 Window에서 구현하였다. 제어 모듈의 경우 Linux 기반 운영체제를 기반으로 구현되었으며, 서비스 모듈과 제어 모듈은 TCP/IP 통신을 이용하여 명령을 전송하며, 수집된 데이터들은 NFS/Samba를 이용하여 공유하도록 설계하였다.

전력과 성능을 분석하는 이유는 이종 멀티코어 시스템의 전력최적화를 위함이다. 위 이종 멀티코어의 경우, 고성능 코어는 성능이 뛰어나지만 전력 소모가 크다. 따라서 고성능이 필요하지 않은 경우 저전력 코어를 사용하는 것이 전력사용에 효율적이다. 전력 효율을 위해 상황에 따라 시스템적으로 코어를 결정하는 것이 전력 최적화라고 할 수 있다.

Ⅲ. 분석 도구 구현

본 장에서는 제안한 이종 멀티코어 시스템을 위한 분석 도구를 구현한 결과에 대해 설명한다. 실험 환경은 보드는 쿼드코어에 젤라빈을 탑재하여 진행한다. CPU사용량을 높이기 위해 벤치마크 어플을 실행하도록 하였다. 제안하는 시스템 분석 도구는 그림 5와 같이 두 가지로 구분하였다. 하나는 수집된 데이터들를 통계적인 수치로 보여주는 웹 기반의 뷰어(Web Based Viewer)이고 다른 하나는 특정 시점의 자세한 정보를 보기 위한 타임 라인 뷰어(Time Line Based Viewer)이다. 각 뷰어의 특징에 따라 나타내고자 하는 데이터를 정의하고, 얻고자하는 시스템 정보를 추출하는 방법을 설명한다. 시스템 내부적 정보 추출에 있어서 구현 방법에 대해서는 기존의 성능 분석방법 [4, 14, 15] 중 Ftrace와 같이 원하는 측정 지점에 명령어를 삽입하여 커널 레벨에서 로고를 기록하는 방법을 사용하였다.

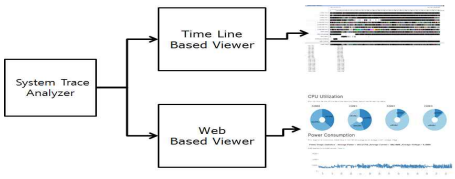


그림 5. 분석 도구 뷰어
Fig. 5 System Trace Analyzer Viewer

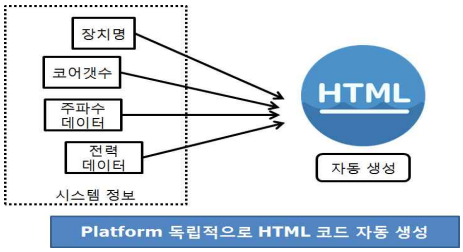


그림 6. 시스템 정보와 웹 뷰어의 관계
Fig. 6 Relationship between System Information and Web Viewer

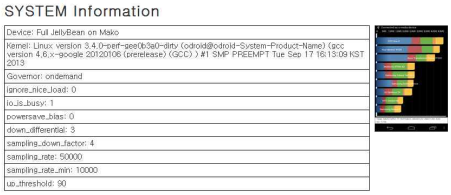


그림 7. 시스템 정보 및 스크린샷
Fig. 7 System Information and Screenshot

1. 웹 기반 뷰어

웹 기반 뷰어는 모든 데이터를 종류에 따라 구분하고 이를 시각화하여 통계적인 수치를 웹 브라우저를 통해 보여준다. 수집되는 시스템 정보는 이종 멀티코어 시스템을 위한 CPU 사용량, 에너지소비량, CPU 주파수 정보가 있으며, 부과적인 정보로는 측정 되는 장치의 하드웨어 정보와 스마트폰 화면을 보여주는 스크린샷, 그리고 장치가 동작하면서 실행하는 시스템 정보 등이 있다. 이러한 웹 기반 뷰어는 모두 수집되는 데이터의 결과를 기반으로 그림 6와 같이 HTML파일을 자동으로 생성되도록 구현하였다. 수집된 시스템 정보를 기반으로 웹 뷰어가 자동 생성되기 때문에 플랫폼에 독립적으로 HTML파일이 생성된다.

그림 7는 웹 기반 뷰어의 결과 중 측정 장치에

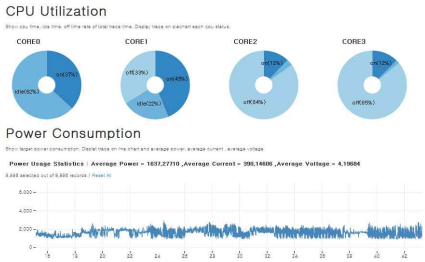


그림 8. CPU 정보 및 전력 소모율 결과
Fig. 8 CPU Utilization and Power Consumption Result

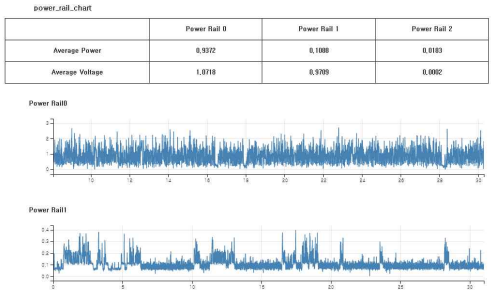


그림 9. 파워 레일 측정 결과
Fig. 9 Power Rail Result

대한 하드웨어 정보와 스마트폰 화면에 대한 스크린샷에 대한 결과 화면이다. 시스템 하드웨어 정보는 측정 장치에 대한 장치명, 운영체제, CPU Frequency의 가버너(Governor)에 대한 정보를 나타내고, 스크린샷은 해당 응용프로그램의 실행을 모두 종료한 마지막 화면을 보여준다. 측정 장비나 운영체제에 대한 기본 정보를 확인하는 용도로 사용된다.

그림 8는 웹 기반 뷰어의 결과 중 CPU 사용정보와 전력소모율에 대한 결과 화면이다. CPU 사용량에 대한 정보는 각 코어별 on/off/idle에 대한 비율을 파이차트로 나타낸다. 이 정보를 통해 어떤 코어가 얼마나 실행하였는지를 통해 해당 코어가 소모한 전력량을 확인할 수 있다. 위 결과의 경우 전체 워크로드 실행 시간 동안 CORE0와 CORE1은 각각 37%, 40% 동작했다. 그리고 CORE2와 CORE3의 경우 각각 12%만 사용했다는 것을 확인할 수 있다. 또한 전체 사용 전력이 약1637W 인 것을 확인할 수 있다. 특히 이종 멀티코어 시스템의 전력 소모량의 분석을 위해서는 통합 전류보다 각 이종 멀티코어 간 전력 소모량을 분석하는 것이 중요하다 그러므로 그림 9과 같이 파워 레일별 전력

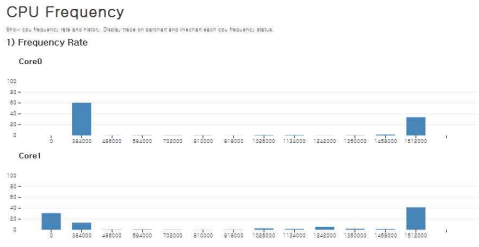


그림 10. CPU 주파수 비율 결과
Fig. 10 CPU Frequency Rate Result

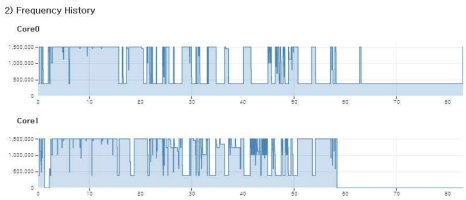


그림 11. CPU 주파수 변화 기록 결과
Fig. 11 CPU Frequency History Result

측정 결과를 보여주도록 구현하였다. 그림 9의 경우 CPU 캐시의 전력사용량과 개별 코어의 전력사용량이 보인다.

그림 10은 CPU 주파수 비율에 대한 정보를 나타낸다. x축은 CPU에서 설정가능한 주파수 목록이고, y축은 전체 워크로드 실행하는 시간 동안 해당 주파수에서 CPU가 동작한 시간 비율을 나타낸다. 그림 10의 워크로드의 경우 가장 낮은 주파수와 가장 높은 주파수 사용 비율이 높은 것으로 CPU 사용량이 급증했다는 것을 확인할 수 있다.

그림 11은 CPU 주파수 변화 기록에 대한 정보를 나타낸다. 이종 멀티코어 각각의 변화를 시간단위로 확인하고 분석하는 것이 가능하다. 그림 11의 경우, CORE0는 60초 동안 일정 주파수 이상으로 계속적으로 실행되었고, CORE1은 59초까지 높은 주파수로 계속 실행되었다는 것을 확인할 수 있다. 본 연구에서 구현한 결과는 이외에도 Regulator 정보, GPIO 정보, Clock 상태, FPS, 온도 정보 등을 출력이 가능하다.

2. 시간 축 기반 뷰어

시간 축 기반 뷰어는 모든 데이터를 하나의 시간 축에 기반하여 보여준다. 웹 기반 뷰어가 좀 더 통계적인 자료를 나타낸다면, 시간 축 기반 뷰어는 특정 시점에서의 좀 더 세밀한 시스템 상태를 나타

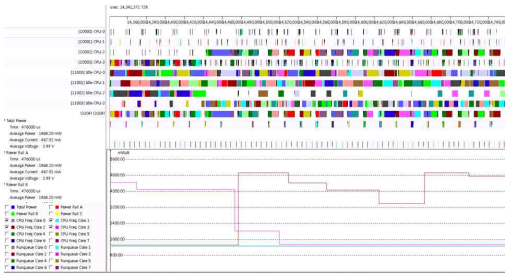


그림 12. 시간 축 기반 뷰어 결과
Fig. 12 Time Line Viewer

낸다. 그림 12 는 시간 축 기반 뷰어의 전체 모습을 나타낸 것이다. 최상단에 가로로 시간이 표기되며, 크게 상단 부분에 CPU와 프로세스의 정보가 가로 바 형태로 출력되며, 하단 부분에 전력과 CPU 주파수에 대한 그래프가 출력된다.

상단의 CPU와 프로세스 정보에서는 특정 시점에서 CPU를 기준으로 어떤 프로세스가 동작했는지 나타내고, 또한 프로세스를 기준으로 어떤 CPU에서 동작했는지를 시간 축을 기반으로 나타낸다. 나타내는 시간 범위를 확대/축소가 가능하여, 원하는 시점의 초, 마이크로 초 단위를 세밀한 분석이 가능하다.

하단의 전력 및 CPU 주파수 그래프의 경우, 자신이 확인하고자 하는 CPU를 선택하면 출력이 된다. 이 또한 시간 축을 확대/축소하는 것을 통해 변화율을 정확하게 확인할 수 있다.

Ⅳ. 결 론

본 연구는 최근 스마트 디바이스의 에너지 소모량을 줄이기 위해 에너지 소모량과 성능 분석을 위한 프레임워크를 설계 및 구현하였다. 특히 이종 멀티코어 시스템을 위한 전력 소모량과 성능간의 차이를 분석할 수 있는 도구를 개발하였다. 이는 전력과 성능을 동시에 체계적으로 분석하는 것을 통해 개발 결과물의 신뢰성과 개발 생산성을 향상시킨다.

본 연구 결과는 CPU 가버너 알고리즘 개발과 이종 멀티코어 시스템의 스케줄러 알고리즘 개발에서 유용하게 사용될 수 있다. CPU 가버너의 알고리즘 개발에 있어서 시스템의 CPU 사용량과 CPU 주파수 그리고 전체 전력 소모율에 대한 데이터를 통해 개발이 가능하다. 분석 프레임워크를 통해 출력되는 정보를 토대로 현재 CPU 가버너의 동작 경향을 파악하고, 이에 따라서 정책을 변경한다. 마지막으로 변경된 정책에 따라서 어떠한 전력 및 성능 변화를 보이는지 통계 결과를 확인한다. 비록 커널의 일부 기능에 대한 소프트웨어 알고리즘 개발에 사용되었지만 현재 분석 프레임워크가 분명히 활용될 수 있다는 점에서 충분한 의미를 갖는다.

이종 멀티코어 중 big.LITTLE 시스템과 같이 전력 최적화 모델의 경우 전력 소모율과 성능간의 관계를 분석할 수 있는 도구는 필수적이다. 본 논문에서 제안하는 프레임워크는 전력과 성능을 동시에 체계적인 분석이 가능하기 때문에 big.LITTLE 시스템에서의 핵심 기능을 담당하는 스케줄러에 대한 개발 또한 가능하다. 스케줄러 알고리즘의 변화에 따라 CPU 사용량과 CPU 로드밸런스 현황 그리고 전체 전력 소모율에 대한 결과를 확인할 수 있다. 또한 결과를 알고리즘에 반영함으로써, 전력 효율에 있어서 보다 최적화된 알고리즘의 개발이 가능하다. 향후 연구로 이종 멀티코어 개발 보드에 실제 적용하고 분석 도구를 사용하여 개선 과정과 개선된 점을 증명하도록 한다.

References

[1] http://www.linaro.org/linaro-blog/2013/07/10/big-little-software-update/
[2] Big.LITTLE Processing with ARM Cortex-A15 & Cortex-A7. http://goo/gl/7mgbL
[3] https://01.org/powertop/
[4] T. Bird, “Measuring Function Duration with Ftrace,” Proceedings of the Japan Linux Symposium, 2009
[5] M. Milosevic, A. Dzhagaryan, E. Jovanov, A. Milenkovic, "An Environment for Automated Power Measurements on Mobile Computing Platform," Proceedings of the ACM Southeast Conference 2013.
[6] Intel. 2nd generation Intel Core vPro processor family.http://www.intel.com/content/dam/doc/white-paper/corevpro-2nd-generation-core-vpro-processor-family-paper.pdf, 2008.
[7] AMD, The future is fusion: The industrychanging impact of accelerated computing, 2008.
[8] NVidia. The benefits of multiple CPU cores in mobile devices, 2010. http://www.nvidia.com/content/PDF/tegra_whit

e_papers/Benefits-of-Multi-core-CPUs-in-Mobile-Devices_Ver1.2.pdf

[9] NVidia. Variable SMP - a multi-core CPU architecture for low power and high performance, 2011. http://www.nvidia.com/content/PDF/tegra_white_papers/Variable-SMP-A-Multi-Core-CPU-Architecture-for-Low-Power-and-High-Performance.pdf

[10] A. Carroll, G. Heiser, "An analysis of power consumption In a smartphone," in Usenix technical conference, pp. 1-14, 2010.

[11] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," Proceedings of IEEE International conference on HW/SW codesign and system synthesis, pp. 105-114, 2010.

[12] Y.-F. Chung, C.-Y. Lin, C.-T. King, "Aneprof: Energy profiling for android java virtual machine and applications," Proceedings of International Conference on Parallel and Distributed Systems, Vol. 0, pp. 372-379, 2011.

[13] F. Ding, F. Xia, W. Zhang, X. Zhao, C. Ma, "Monitoring energy consumption of smartphones," Proceedings of International Conference on Cyber, Physical and Social Computing, pp. 610-613, 2011.

[14] M. Desnoyers, M. Dagenais, "The LTTng Tracer: A Low Impact Performance and Behavior Monitor of GNU/Linux," Proceedings of the Ottawa Linux Symposium, 2006.

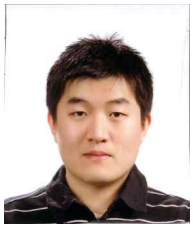
[15] B. Cantrill, M. Shapiro, A. Leventhal, "Dynamic Instrumentation of Production Systems," Proceedings of USENIX conference, 2004.

저 자 소 개

김 형 준

2012년 국민대학교 컴퓨터공학부 학사.
2014년 국민대학교 컴퓨터공학부 석사.
관심분야: 임베디드 소프트웨어, 저전력 소프트웨어, 리눅스 스케줄링
Email: dudnwjs@gmail.com

경 주 현



2007년 한성대학교 정보시스템 공학과 학사.
2011년 한성대학교 컴퓨터공학과 석사.
2007년-2014년 ㈜ 세븐코아 선임연구원.
2011년-현재, 국민대학교 컴퓨터공학과 박사과정.
관심분야: NUMA 메모리 최적화, 리눅스 Scalability, 임베디드 소프트웨어, 저전력 소프트웨어
Email: joohyun0115@gmail.com

임 성 수

1993년 서울대학교 컴퓨터공학과 학사.
1995년 서울대학교 컴퓨터공학과 석사.
2002년 서울대학교 전기컴퓨터공학과 박사.
1999-2004년 ㈜ 팜팜테크 기술총괄이사.
현재, 국민대학교 컴퓨터공학부 교수.
관심분야: 임베디드 실시간 시스템, 이동단말 소프트웨어, 저전력 소프트웨어, 가상화 기술
Email: sslim@kookmin.ac.kr