

안드로이드 플랫폼에서의 멀티코어 환경을 고려한 성능 및 전력 프로파일링 모듈의 설계와 구현

(Design and Implementation of a Performance and Power Profiling Module for Android Platform in Multi-core Systems)

송 옥, 전 상 우, 김 주 성, 김 학 봉, 김 지 흥
서울대학교 컴퓨터공학부

(Wook Song, Sang-Woo Jun, Joosung Kim, Hakbong Kim, Jihong Kim)
(School of Computer Science & Engineering, Seoul National University)

Abstract : Embedded systems are no longer the restricted performance special-purpose systems they were in the past. State-of-the-art embedded systems deliver over 1Ghz of processing speed to achieve multimedia processing as well as real-time mobile services and elegant user interfaces. However, mobile embedded systems usually need to run on limited battery power, which restricts the continuous increase in speed. The need for energy efficiency as well as processing power will most likely lead to the adoption of MPSoC architectures in the future. The rise in system complexity has created a need for more sophisticated embedded operating systems, most notably Google's Linux-based Android OS. Android differs from traditional Linux in that it uses a component-oriented application structure implemented on a Java virtual machine. This aspect renders traditional process-oriented profilers such as Oprofile less fit to use. This paper presents a novel tool to profile the performance and power usage of an Android system at the component level granularity. The tool augments Android's Dalvik VM to read the Hardware Performance Counters(HPC) at component boundaries, and utilizes an energy model to measure the power consumption of individual components. This tool is believed to present more insightful performance and energy consumption information compared to traditional performance profilers.

Keywords : 안드로이드, 프로파일러, 성능, 전력, 분석 도구

1. 서론

임베디드 시스템에 단순한 몇 가지의 기능과 그 기능을 수행하기에 무리 없는 성능만을 기대했던 기존의 페러다임과 달리 최근에는 멀티미디어 처리, 게임, 높은 실시간 성능을 갖는 모바일 서비스, 유려한 사용자 인터페이스 등의 처리를 요구하는 실정이다. 이렇게 성능과 기능에 대한 높아진 사용자의 요구로 인해 임베디드 시스템에서 쓰이는 마이크로

프로세서(Microprocessor)의 연산 처리 능력이 기존의 것에 비해 비약적으로 높아지는 추세다. 최근에는 이러한 경향이 더욱 심화되어 근래에 출시되는 최신 임베디드 장치에는 동작 클럭이 1Ghz에 가까운 고성능의 마이크로 프로세서를 채택하고 있다.

하지만 대부분의 임베디드 시스템은 배터리를 주 동작 전원으로 사용하고 있으므로, 점점 높아만 가는 사용자 요구에 맞추어 마이크로프로세서의 동작 클럭을 계속해서 높일 수는 없다. 이러한 제약 사항으로 인해 한정된 자원인 배터리의 효율적인 사용 역시 성능의 향상 못지않게 매우 중요하며 이러한 주제에 대해 많은 연구들이 계속하여 진행되고 있다. 높은 성능과 효율적인 에너지 사용이라는 두 가지 상충되는 임베디드 시스템에 대한 요구는 결국 임베디드 시스템에서도 멀티 코어를 기반으로 한 MPSoC(Multiprocessor System -on-Chip) 구조를 가질 것으로 예상된다.

* 본 논문은 BK21사업에 의하여 지원되었으며, 정부(교육과학기술부)의 재원으로 한국과학재단의 지원과 (No. 20100018873) 한국연구재단-미래기반기술개발사업(첨단융복합분야)의 지원을 받아 (No. 2010-0020724) 수행되었습니다. 본 연구를 위해 연구장비를 지원하고 공간을 제공한 서울대학교 컴퓨터 연구소에 감사드립니다.

전화와 그에 필요한 기능들과 같이 단순한 기능만이 목적인 기존의 모바일 단말에서의 운영체제와 응용의 개발은 개발자가 그 장치의 하드웨어 맞춤형 소프트웨어를 만드는 일련의 과정이었으며, 동영상 재생과 같이 원래 그 장치의 목적과는 상관이 없는 기능들을 하나 둘씩 추가됨에 따라 수반되는 개발의 과정에서도 이러한 개념은 생산성 향상을 위하여 각 벤더마다 자신만의 독자적인 플랫폼을 구축하는 형태로 개발의 편의를 도모하기는 하였지만, 크게 달라지지는 않았다. 그러나 최근에는 단말에서 동작하는 응용을 그 단말의 벤더에서만 개발하는 것이 아닌 하나의 에코시스템(Eco-System)으로써 외부의 다양한 개발자들이 다양한 응용의 개발에 적극 참여하는 시스템이 크게 각광을 받게 됨에 따라 모바일 플랫폼의 표준이 필요하게 되었다. 이러한 표준 플랫폼의 등장은 비단 모바일 단말의 응용 개발자의 편의뿐만 아니라 벤더 입장에서 플랫폼 이하의 계층에 대해서는 큰 개발의 부하 없이 새로운 제품을 만들 수 있다는 장점을 함께 갖고 있어, 점차 많은 회사들이 몇 개의 표준 모바일 플랫폼을 채택하여 제품을 개발하고 있다. 2007년 구글(Google)이 모바일 기기의 표준 플랫폼의 일환으로 발표한 안드로이드(Android) [1]가 대표적인 예이며, 현재 시장에서 빠르게 점유율을 높여가고 있다.

멀티 코어와 표준 모바일 플랫폼을 탑재한 차세대 임베디드 시스템에서 높은 성능과 에너지 효율을 갖는 응용의 개발을 위해서는 대상 응용의 성능과 에너지 소모를 정확하게 수집하고 분석할 수 있는 도구가 필요하다. 정확한 성능 및 에너지 소모에 대한 정보는 비단 응용의 개발에만 사용될 뿐 아니라, 모바일 플랫폼 자체의 성능 및 에너지 소모에 대한 피드백으로 활용함으로써 최적화 기법 개발에 기여할 수 있다.

본 논문에서는 이미 제안된 특정 아키텍처에 대한 하드웨어 성능 카운터(Hardware Performance Counter, HPC) 기반의 전력 모델을 활용하여 커널 수준에서의 태스크 별 성능과 전력 프로파일링을 모듈을 설계하고 구현하였다 [2]. 각 태스크 별로 수집한 정보를 안드로이드 플랫폼에서의 사용할 수 있도록 모듈을 설계하고 구현하였으며, 안드로이드에서 기 제공되는 디버그 관련 클래스를 확장하여 안드로이드 플랫폼에서도 성능 및 전력 소모 정보를 수집할 수 있도록 하였고, 그 수집 결과를 분석하였다. 향후 본 모듈을 확장하여 수집 가능한 정보의 수를 늘리고, 모바일 플랫폼 및 운영체제 성능 및 에너지 소모 최적화에 활용할 계획이다.

II. 관련 연구

본 논문에서 대상으로 하고 있는 안드로이드 플랫폼의 응용의 성능을 프로파일링 할 수 있는 도구는 크게 안드로이드 플랫폼 위에서 안드로이드 응용의 프로파일링을 대상으로 하는 도구와 기존의 리눅스 환경에서 현재 수행 중인 프로세스를 프로파일링 할 수 있는 도구로 구분할 수 있다.

안드로이드 플랫폼 위에서 안드로이드 응용의 성능을 프로파일링 할 수 있는 대표적인 도구는 구글에서 제공하는 안드로이드 SDK(Software Development Kit)에 포함된 도구이다 [3]. 구글은 응용 내부의 각 메소드 호출 정보와 각 메소드가 수행된 시간, 메모리 사용 정보 등을 쉽게 추출할 수 있도록 프로파일링 및 트레이싱에 관련된 메소드들을 디버그 클래스에 따로 구현하였으며, 이러한 메소드들을 플랫폼 내부에 미리 추가해두었다. SDK에서 제공하는 간단한 툴을 사용하여 각 메소드가 제공하는 정보를 쉽게 추출할 수 있으며, 그래픽 인터페이스를 기반으로 한 분석 또한 가능하다 [4]. 하지만 하나의 응용에 대한 정보의 수집이 목적으로서 시스템 전체에서 수행 중인 다른 응용의 프로세스나 스레드들과 어떠한 관계를 맺고 서로 영향을 주는지에 대한 정보를 알 수 없으며, 소모 전력에 대한 정보도 제공하지 않는다는 단점이 있다.

안드로이드는 기본적으로 리눅스 운영체제를 기반으로 하고 있으므로, 리눅스를 대상으로 한 시스템 성능 프로파일러들은 대부분 그대로 사용 가능하다. 특히 Oprofile[5]은 안드로이드 소스 트리에 포함된 원시 어플리케이션 성능 프로파일러로써 하드웨어 성능 카운터의 정보를 직접 수집할 수 있다는 점에서 그 성능이 매우 강력하다. 하지만 리눅스 운영체제의 프로세스의 성능 수집이 주목적이기 때문에 안드로이드 플랫폼 위에서 동작하는 응용의 성능 수집에 있어서 프로세스 수준보다 작은 단위인 컴포넌트 대해 인지하지 못하며, 전력 정보 역시 제공되지 않는다.

본 논문에서는 안드로이드 응용의 구성 요소인 컴포넌트 수준에서 하드웨어 성능 카운터(HPC) 정보를 수집하고 그 값을 기존에 제안된 전력 모델에 적용하여 각 컴포넌트 별 소모 전력을 수집하는 도구의 구현을 위하여 커널과 안드로이드 플랫폼에 필요한 모듈을 설계하고 구현하는 것을 목표로 하였다.

III. 성능 및 전력 프로파일러 구현

1. 프로파일러 설계 개관

안드로이드는 앞서 언급한 것과 같이 리눅스를 기본 운영체제로 그 상위 계층에 안드로이드 플랫폼이 존재하는 형태이다. 따라서 하드웨어 성능 카운터를 직접 접근하여 제어하기 위해서는 먼저 리눅스 커널에 관련 모듈을 구현한 후, 안드로이드 플랫폼에 구현된 정보 추적 모듈에 커널 수준에서 수집한 정보를 전달하는 방식을 사용하여야 한다.

그림 1은 리눅스 커널과 안드로이드 프레임워크 사이에 하드웨어 성능 카운터 제어를 위해 구현된 디바이스 드라이버 모듈과 JNI를 이용하여 구현된 디바이스 드라이버 제어 자바 메소드, 그리고 실제 안드로이드 응용에서 정보를 수집하기 위해 확장된 디버그 클래스로 구성되어 있는 프로파일러의 구조를 나타낸 것이다.

2. 커널 수준의 동적 태스크 프로파일링 모듈

안드로이드 플랫폼에서 대상 장치의 하드웨어를 제어하기 위해서는 하위 계층인 리눅스에서의 디바이스 드라이버와 시스템 콜과 같은 지원이 필요하다. 커널 수준의 동적 태스크 프로파일링 모듈은 기본적으로 커널 내부에서 각 리눅스 LWP(Light Weight Process)의 스케줄링 시점에 개입하여 수행 중인 LWP의 하드웨어 성능 카운터 정보를 수집하여 커널 내부 공간에서 관리하는 역할과 사용자 영역에 이러한 정보를 제공하는 역할을 담당하고 있다.

디바이스 드라이버와 시스템 콜과 같은 지원이 필요하다. 커널 수준의 동적 태스크 프로파일링 모듈은 기본적으로 커널 내부에서 각 리눅스 LWP(Light Weight Process)의 스케줄링 시점에 개입하여 수행 중인 LWP의 하드웨어 성능 카운터 정보를 수집한 후, 커널 내부 공간에서 관리하는 역할과 사용자 영역에 이러한 정보를 제공하는 역할을 담당하고 있다.

각 태스크의 소비 전력 값을 수집하기 위해서는 대상 장치에서 제공하는 하드웨어 수준의 전력 측정 모니터를 이용하거나, 태스크의 수행 패턴에 근거한 전력 예측 모델을 이용하는 방법을 사용할 수 있다.본 논문에서 대상으로 하고 있는 장치에는 직접 소비 전력을 측정할 수 방법을 내부적으로 제공하고 있지 않으므로, [2]에서 제안한 멀티프로세서용 에너지 예측 모델을 사용하여 태스크 별 소비

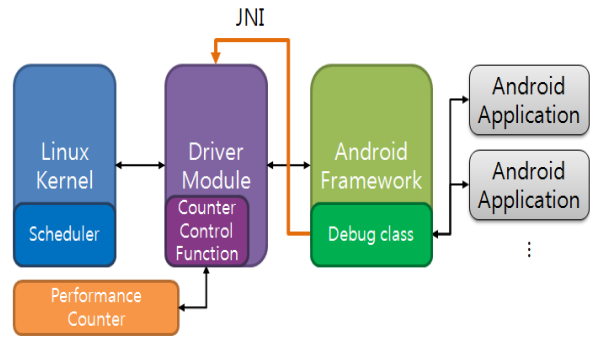


그림 1. 안드로이드 응용의 성능 및 전력 프로파일러 구조

Fig. 1. The structure of the performance and power profiler for Android applications

전력 프로파일을 구성하였다. 사용한 에너지 예측 모델은 ARM11 MPCore[6]에서 제공하는 하드웨어 성능 카운터 이벤트를 수행한 명령어의 수, 지역 1차 데이터 캐시 접근 수, 공유 2차 캐시 접근 수, 데이터 의존으로 인한 지연, 캐시 일관성 트랜잭션의 5가지 이벤트를 선형 회귀 분석을 통해 구성된 모델이다.

본 모델을 활용하여 소비 전력을 예측하기 위해 필요한 하드웨어 성능 카운터 이벤트의 수는 총 5개이다. 대상 장치의 프로세서가 5개 이상의 하드웨어 성능 카운터를 지원한다면, 각각의 카운터 값을 읽는 것만으로 태스크 별 소비 전력의 예측이 가능하다. 하지만 본 논문에서 대상으로 하고 있는 장치는 단 2개의 하드웨어 성능 카운터만을 제공하고 있기 때문에, 동적으로 소비 전력을 예측하기 위해서는 멀티플렉싱 기법이 필요하다. 본 논문에 적용한 멀티플렉싱 기법은 이전 연구인 [7]을 참고하여 안드로이드 플랫폼의 리눅스 커널에 적용하였다.

3. 안드로이드 프레임워크 디버그 클래스 확장

커널 수준의 성능 및 전력 프로파일링 모듈이 태스크 별 정보 수집의 일체를 담당하여도, 실제 안드로이드에서 관리하는 프로세스나 스레드는 리눅스 태스크와 다르기 때문에 안드로이드 수준의 프로세스, 스레드, 컴포넌트와의 연관성을 파악하는 것이 중요하다. 안드로이드 스레드의 경우 자바 스레드를 사용하여 구현되어 있기 때문에 가상 머신 상위 계층에서 그 생성과 소멸을 추적하기 어렵다. 따라서 Dalvik 가상 머신의 실제 내부 pthread를 생성하는 부분을 추적하여 프로파일링이

4 안드로이드 플랫폼에서의 멀티코어 환경을 고려한 성능 및 전력 프로파일링 모듈의 설계와 구현

가능하도록 설계하였다. 컴포넌트의 생성과 소멸, 통신하는 과정의 분석을 통해 각각의 컴포넌트가 생성되거나 소멸할 때 발생하는 이벤트를 추적하였으며 이 과정에서 적절한 프로파일링 시작 및 종료 시점을 찾아서 프로파일링 메소드를 삽입하였다. 프로파일링 메소드는 구글에서 기본적으로 제공하는 디버그 클래스를 확장하여, 안드로이드 전체적인 설계 철학을 위배하지 않도록 하였다. dalvik 가상 머신 상위 계층에서 프로파일링 하기 위해서는 커널이 제공하는 API를 그대로 활용할 수 없기 때문에 JNI를 활용하여 가상 머신 상위에서도 사용할 수 있도록 하였다. 수집된 결과는 디버그 클래스에서 기록하는 다른 로그와 같이 보관된다.

IV. 실험 및 평가

1. 실험 환경

제안한 프로파일러의 구현을 위해 본 논문에서는 4개의 코어를 장착한 ARM11 MPCore 평가 보드에 안드로이드 버전 2.2 froyo를 이식하여 사용하였다. ARM11 MPCore는 코어마다 각 2개의 HPC를 제공하며, 210MHz의 클럭으로 동작한다.

2. 실험 결과

그림 2는 안드로이드 기본 내장 어플리케이션 중 사용 빈도가 높은 두 개의 어플리케이션의 소모 전력의 각 액티비티 별 코어 전력 소모 비율을 나타낸 것이다. 각각의 어플리케이션은 네트워크의 사용 비중이 높을수록 코어에서 소모 전력을 낮아지며 설정이나, 새로운 뷰를 여는 경우에는 상대적으로 코어의 전력을 많이 소모함을 알 수 있다. 액티비티 별 소모 전력에 차이가 명확하게 나타나기 때문에 기존의 oprofile과 같은 리눅스 프로세스를 대상으로 한 프로파일러와는 다른 접근 방향을 갖는 프로파일러가 필요하다는 것을 확인 할 수 있었다.

V. 결론

사용자의 경험과 높은 성능, 그리고 효율적인 에너지 사용을 위해 최근에는 여러 모바일 플랫폼 표준이 제안되고 있으며, 빠르게 시장 점유율을 높여가고 있다. 기존의 리눅스 프로세스를 대상으로 한 성능 또는 전력 프로파일러는 리눅스를 기본으로 하여 개발된 모바일 플랫폼일지라도 태스크 모

델이나, 사용자의 사용 패턴이 많이 다르기 때문

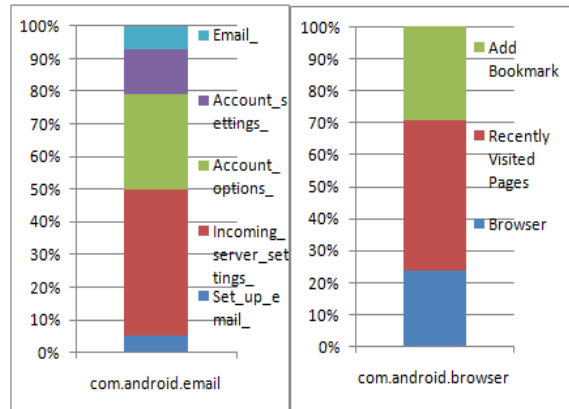


그림 2. 기본 내장 응용의 액티비티 별 전력 소모 비율

Fig. 2. Power consumption breakdown of default applications on Android

에 그대로 적용할 수 없으므로, 새로운 접근 방향을 갖는 프로파일러의 필요성이 대두되고 있다. 본 논문에서는 구글 안드로이드 플랫폼을 대상으로 구글이 제공하는 프로파일러의 기능을 확장하여 안드로이드 어플리케이션의 구성 요소 중 하나인 액티비티 별 성능 및 전력 정보를 수집할 수 있는 도구의 설계 및 구현 방법을 제안하였으며, 각 컴포넌트 별로 성능 정보와 소비 전력이 다름을 확인하였다.

참고 문헌

- [1] <http://www.android.com/>
- [2] 최원일, 김현희, 김지홍, “하드웨어 성능 카운터를 이용한 임베디드 멀티프로세서 환경에서의 에너지 모델링 예측 기법”, 한국정보과학회 추계학술대회, 2008.
- [3] <http://developer.android.com/sdk/index.html>
- [4] <http://developer.android.com/guide/developing/tools/traceview.html>
- [5] <http://oprofile.sourceforge.net/about/>
- [6] 송욱, 송지석, 김지홍, “하드웨어 성능 카운터에 기반한 커널 수준의 동적 태스크 프로파일링 모듈의 설계 및 구현”, 2008 정보통신분야학회 합동학술대회 논문집, pp.263-267, 20082005.