
안드로이드 플랫폼을 위한 소프트웨어 기반의 전력 소비 측정 프레임워크 비교

이준민 · 노동건

송실대학교

Survey on Software-based Power-Metering Framework for Android Platform

Jun-min Yi · Dong-kun Noh

Soongsil University

E-mail : {jmyi, dnoh}@ssu.ac.kr

요 약

최근 스마트기기의 보급률이 점점 증가하면서 기기의 활용도가 계속해서 증가함과 동시에 사용되는 응용프로그램도 점차 고급화하고 있다. 그러나 이로 인해 기기의 사용시간은 점차 줄어들게 되는데 이러한 문제를 해결하기 위하여 하드웨어/소프트웨어적으로 수많은 연구들이 이루어지고 있다. 그 중 하나는 프로세스 단위 별로 전력 소모량을 프로파일링하는 것인데 이렇게 측정된 에너지 소모 데이터에 기반하여 프로세스들을 관리함으로써 잔존 에너지를 효율적으로 사용할 수 있다. 아울러 응용프로그램이 소모하는 에너지 사용 트레이스를 프로그램 설계 단계에서 분석하고 고려함으로써 저전력 프로그램 설계도 가능케 한다. 본 논문에서는 안드로이드 플랫폼을 위한 소프트웨어 기반의 프로세스 레벨 전력 소비량 측정 기법들을 조사하고 그 기법들의 장점과 단점을 서로 비교 분석하여 개선 방안을 제안한다.

ABSTRACT

Recently, the supply ratio of smart devices application has become increasable, utilization of device increases constantly. At the same time, used application is more gentrified. However, using time of devices is decreased. To solve these problems, many research is studying about the hardware/software. One of them is profiling power consumption by process units. The process can be managed, based on measured energy consumption data. These means that it can efficiently use the residual energy. Application at the stage of program design can analyze and used-energy using the trace by considering the low-power can design. In this paper, we studied software-based power-metering framework for android platform. We survey each process-level power consumption measurement techniques, compare advantages and disadvantages of the technique and propose improved measures.

키워드

안드로이드, 에너지 소비, 파워 소비, 에너지 프로파일러, 에너지 측정 도구, 프로세스 레벨

I. 서 론

최근 스마트기기가 널리 보급되면서 이와 관련된 산업과 연구는 나날이 발전하고 있다. 이에 따라 장치의 고급화는 빠르게 진행되고 있고 사용

자들의 기기 사용 시간은 계속 증가하고 있는 반면, 배터리 기술의 발전은 이를 따라가지 못하고 있는 상태이다. 휴대성이 강조된 스마트기기의 경우 배터리라는 에너지원의 한정적인 자원은 근본적으로 극복될 수 없으므로, 이를 최대한 효율적

으로 활용하는 것이 유일한 해결책이 될 수 있다 따라서 최근에 전력 소비 분석 및 예상에 관한 다양한 연구들이 활발히 수행되고 있다

본 논문에서는 이러한 연구들을 바탕으로 안드로이드 플랫폼에서 좀 더 정확한 전력 소비량을 측정하는 프레임워크를 만들기 위해 지금까지 연구된 기법들을 조사하고, 그들을 비교 분석하여 정리하였다.

II. 전력 소비 측정 프레임워크와 이를 위한 요구사항

본 연구에서 타겟으로 삼고 있는 전력 소비 측정 프레임워크는 안드로이드 플랫폼에서 여러 응용프로그램들이 동작 시에 응용별로 정확한 전력 소비량을 모니터링하고 예측하는 도구로써 궁극적으로 장비가 전력 소비를 효율적으로 할 수 있도록 기반 정보를 제공하는 역할을 한다

이러한 전력 소비 측정 프레임워크에서 가장 중요한 사항은 전력 소비에 대한 정확한 측정과 예측이 실시간으로 가능해야 한다는 점이다 따라서 안드로이드 플랫폼에서 보다 정확한 에너지 정보들을 산출하기 위해 수많은 연구들이 이루어지고 있다. 각 연구들은 서로 다른 방식으로 소비 전력을 측정하고 있는데, 각 방식에 따라 산출된 데이터의 값과 실제 에너지 사용량 간에 서로 다른 오차를 가진다. 본 논문에서는 각 방식의 특징을 비교하는데 목표를 둔다.

기존의 연구들은 주로 OS를 포함한 응용 프로그램들이 아닌 개별 하드웨어 구성 요소의 에너지 사용 분석에 초점을 두고 있었다 그러나 개별 하드웨어 구성 요소뿐만 아니라 각 소프트웨어 전력 소비에 대한 분석과 이를 통한 효율적인 시스템 제어 방법이 제시된다면 보다 유용하게 각종 스마트 디바이스에 사용될 수 있을 것이다

III. 전력 소비 측정 프레임워크 비교

3.1 프레임워크들의 비교 기준

안드로이드 기반 스마트 디바이스에서 보다 정확한 전력 소비량을 예측하기 위하여 지금까지 다양한 전력 소비 측정 프레임워크가 제안되었다

우선 전력 소비 모델과 관련하여 [1]의 연구에서는 모델이 구축되는 방식을 아래와 같이 구분하였다:

- **하드웨어를 사용하여 전력 소비 패턴을 수집하는 방법의 차이:** (1) 외부(External) 장비를 사용하여 측정/수집하는 방법, (2) 내부(Internal) 배터리 인터페이스를 사용하여 수집하는 방법
- **소비 전력 모델을 추론하기 위해 필요한 시스템 정보를 얻어오는 기법의 차이 :** (1) 시스템 사용률(Measurements of system utilization), (2) 시스템 호출(System-call), (3) 응용프로그램 인터페이스 호출(API-call)
- **이미 검증된 하드웨어 장치별 에너지 소비 모델에 따라 에너지 정보를 얻는 방법의 차이:** (1) 추정(Estimation), (2) 예측(Prediction)
- **시스템 정보를 얻는 모드의 차이:** (1) Supervised 모드, (2) Unsupervised 모드

표 1과 표 2는 각 전력 소비 프레임워크들의 특징을 설명하기에 앞서, 위의 구분법에 따라 이들을 비교한 것이다. 자세한 설명은 다음 절에 기술되어 있다.

3.2 각 프레임워크의 특징

EnTracked[2]는 내부 배터리 인터페이스를 사용하여 소비전력을 모니터링한다. 응용프로그램 인터페이스 호출을 사용하여 시스템 정보를 얻어오고, 이 정보를 바탕으로 현재 시스템의 상태와 변화될 상태에 따라 추후 전력 소비에 대한 추정과 예측 분석이 가능한 전력 모델을 사용한다. 예측을 증명하기 위해 노키아 휴대폰의 내부 스마트 배터리 인터페이스를 사용하여 소비 전력을 모니

표 1. 전력 소비 측정 프레임워크 비교 1

	EnTracked [2]	PowerTutor [3]	Sesame [4]	Pathank [5]	PowerProf [1]
External		v		v	
Internal	v	v	v		v
Utilization		v	v		
System-call				v	
API-call	v				v
Estimation	v	v	v	v	v
Prediction	v				v
Supervised	v				
Unsupervised		v	v	v	v

표 2. 전력 소비 측정 프레임워크 비교 2

	AppScope [6]	Eprof [7]	PowerScope [8]	Quanto [9]	PowerBooter [3]
External			v		
Internal	v			v	v
Utilization					
System-call					
API-call		v	v		
Estimation	v	v	v	v	
Prediction					v
Supervised					
Unsupervised	v				

터링 하였다.

PowerTutor[3]는 각 장치의 전력 소모량을 외부 장비로 측정 분석한 후 그 데이터를 배터리 소모량과 매칭시켜 별도의 외부 장비 없이 내부 배터리 인터페이스만을 사용해 전력 측정이 가능하다. PowerBooter[3]라는 전력 측정 도구를 사용하여 전력 소비량을 추정한다. 하지만 PowerTutor는 각 하드웨어 구성 요소에 대한 에너지 소비의 정확성을 보장하지는 못한다.

Sesame[4]는 FSM (Finite State Machine) 기반의 내장된 전류 센서를 사용하는 스마트 배터리 인터페이스를 사용하여 자동으로 스마트폰 전원을 모델링 한다. 하지만 Sesame은 각 하드웨어 구성 요소의 전력 분석이 아닌 전체 시스템의 전반적인 전력 소비를 측정하기 때문에 개별 응용프로그램의 전력 소모를 추정하기는 어렵다.

Pathank의 연구[5]는 외부 장비의 도움을 받아 자세한 전력 소모를 추정하는 기법을 제안하고 있다. 시스템 호출 기반으로 여러 장치를 직접 FSM 방식으로 측정하여 전력을 모델링 한다. 이 기법은 각 응용프로그램의 에너지 측정에도 사용될 수 있지만, 다양한 대상 장치의 세부적이고 자세한 전원 상태의 정보를 얻기 위해서 주로 사용된다.

PowerProf[1] 역시 내부 배터리 인터페이스를 사용한다. 또한, 응용프로그램 인터페이스 호출을 통하여 전력을 측정하므로 장치의 업그레이드나 새로운 장치로 변경되더라도 그대로 사용할 수 있다. 하지만 응용프로그램 인터페이스 호출에 의존적인 특징 때문에 전력 측정 주기에 의존적이라는 단점이 있다. PowerProf는 genetic 알고리즘 기반으로 모델링되어 있고 이 알고리즘은 각 하드웨어의 전원 상태를 자동으로 확인할 수 있을 뿐 아니라, 실시간 전력 추정 및 예측을 할 수 있다. 하지만 전력 모델링 보다는 응용프로그램의 에너지 소모 추정에 중점을 두고 있어 전력 모니터링의 관점에서는 사용에 제한적인 측면이 있다.

AppScope[6]는 실시간 프로세스별 전력 분석을 제공하는 안드로이드 기반의 에너지 측정 시

스템이다. 각 프로세스의 전력 소비 예측은 linear 모델을 사용하며, 리눅스의 하드웨어 사용 정보에 접근하기 위해 Procfs와 sysfs를 포함하는 BatteryStats라는 안드로이드 응용프로그램 인터페이스를 사용한다. 특별히, 이 시스템은 커널 레벨에서 구현되고, 이벤트 대응 모니터링 방식으로 낮은 오버헤드와 높은 정확성을 제공한다.

Eprof[7]는 전술한 Pathank의 연구[5]에서 제안된 모델을 기반으로 구현되어진 전력 프레임워크로, 스마트폰 응용프로그램에 대한 세부 전력사항 프로파일러이다. FSM 전력 모델을 기반으로 응용프로그램의 비동기 에너지 상태를 분석할 수 있고, 전력 레벨의 세분화 및 하드웨어 구성 요소의 마지막 상태(tail-state)까지 에너지 특성을 모델링 한다. 또한, 안드로이드 프레임워크의 수정은 응용프로그램 인터페이스 호출을 추적해야 한다 따라서 만약 안드로이드 NDK가 사용되는 경우에는 응용프로그램 코드가 수정되어야 한다.

PowerScope[8]는 모바일 장치에 대한 프로세스별 에너지 추정을 자세하게 제공한다. 전력 측정 시 추가 외부 장비 자원을 필요로 하며 프로그래머는 전력 소비를 추정하기 위해 응용프로그램 인터페이스 호출을 사용한다.

Quanto[9]는 안드로이드가 아닌 TinyOS 플랫폼에서 동작하는데, AppScope와 유사한 방식의 시스템 접근 방식을 사용하여 시스템의 각 하드웨어 전력 사용 현황을 알 수 있다 또한 이벤트 기반 방법에 따라 빠른 에너지 측정을 위한 네트워크 전체의 에너지 프로파일러로 개발되었다.

마지막으로, PowerBooter[3]는 내부 배터리 인터페이스를 사용한 스마트폰의 전력 모델이며서 버에서 전력 센서를 사용하여 가상 컴퓨터의 전원 모델을 구축할 수 있다. 또한, 사용별 통계에서 최적의 상태를 선택해 모델에 맞게 적용할 수 있다. 그러나 PowerBooter는 모델 구축 비용이 많이 들기 때문에 비용 대비 성능은 높지 못하다는 단점이 있다.

IV. 결 론

본 논문에서는 안드로이드 플랫폼을 위한 소프트웨어 기반의 정확한 전력 소비 측정 프레임워크를 설계하기 위해 현재까지 수행되어온 연구들의 기법들을 조사하였다.

이 기법들이 궁극적으로 목표로 하는 것은 프로세스 단위 별로 전력 소모량을 프로파일링하여 소모 데이터에 기반하여 프로세스들을 관리함으로써 잔존 에너지를 효율적으로 사용하는 것이고 더 나아가 응용프로그램이 소모하는 에너지 사용 트레이스를 프로그램 설계 단계에서 분석하고 고려함으로써 저전력 프로그램 설계도 가능케 하는 것이다.

요컨대, 이들의 장단점을 종합 해보면 전력 소비 측정 도구 설계 시 가장 중요한 포인트들은 (1) 추가적인 하드웨어의 도움 없이 소프트웨어적으로 보다 정확한 에너지 소모 모델을 세우고 (2) 시스템 오버헤드를 최소화하면서 각 응용의 장치 접근을 트레이스하여 (3) 오차가 최소화된 프로세스별 에너지 소모량을 계산 예측하는 것이라 할 수 있을 것이다.

참고문헌

[1] Mikkel B. Kjergaard and Henrik Blunck, "Unsupervised Power Profiling for Mobile Devices," 8th International ICST Conference on Mobile and Ubiquitous Systems(MobiQuitous'11), Copenhagen, Denmark, December 2011

[2] M. B. Kjergaard, J. Langdal, T. Godsk, and T. Toftkjær, "Entracked: energy-Efficient Robust Position Tracking for Mobile Devices," 7th international conference on Mobile systems(MobiSys'09), Kraków, Poland, 2009

[3] Tiwana. B., Dick. R.P., Zhiyun Qian, Mao. Z.M., Zhaoguang Wang, Lei Yang, "Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones," Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis(CODES/ISSS'10), Scottsdale, AZ, USA, October 2010

[4] Mian Dong, Lin Zhong, "Self-Constructive High-Rate System Energy Modeling for Battery-Powered Mobile Systems," Proceedings of the 9th international conference on Mobile systems, applications, and services(MobiSYS'11), Bethesda, MD,

USA, 2011

[5] Abhinav Pathak, Y. Charlie Hu, Ming Zhang, Paramvir Bahl, Yi-Min Wang, "Fine-Grained Power Modeling for Smartphones Using System Call Tracing," Proceedings of the sixth conference on Computer systems(EuroSYS'11), Salzburg, Austria, 2011

[6] C. Yoon, D. Kim, W. Jung, C. Kang, H. Cha, "AppScope: Application Energy Metering Framework for Android Smartphone using Kernel Activity Monitoring," 2012 USENIX Annual Technical Conference (USENIX ATC'12), Boston, MA, USA, June 2012

[7] Abhinav Pathak, Y. Charlie Hu, Ming Zhang, "Fine Grained Energy Accounting on Smartphones with Eprof," Proceedings of the 7th ACM european conference on Computer Systems(EuroSYS'12), Bern, Switzerland, 2012

[8] Jason Flinn, M. Satyanarayanan, "PowerScope: A Tool for Profiling the Energy Usage of Mobile Applications," Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications(WMCSA'99), Washington, DC, USA, 1999

[9] Rodrigo Fonseca , Prabal Dutta , Philip Levis , Ion Stoica, "Quanto: Tracking Energy in Networked Embedded Systems," Proceedings of the 8th USENIX conference on Operating systems design and implementation(OSDI'08), Berkeley, CA, USA, 2008