

사용자 입력 기반의 스마트폰 전력 감소 기법

우상정^{1,2}, 서원익¹, 김창대¹, 허재혁¹

¹KAIST 전자전산학과, ²삼성전자 DMC 연구소

{sjwoo, wiseo and cdkim}@calab.kaist.ac.kr, {jhuh}@kaist.ac.kr

User Input based Power Reduction Technique for Smartphones

Sangjung Woo^{1,2}, Wonik Seo¹, Changdae Kim¹, and Jaehyuk Huh¹

¹Computer Science Department, KAIST

²DMC R&D Center, Samsung Electronics Co. Ltd.

요 약

모바일 환경에서는 배터리 용량의 한계로 에너지 효율성이 매우 중요하나, CPU의 이용률에 기반한 DVFS로는 소비 전력 감소에 한계가 존재한다. 모바일 어플리케이션은 사용자와의 상호 동작에 따라 다른 성향을 보여주기 때문에, 이 특성을 이용하여 사용자가 체감하는 성능저하 없이 소비 전력을 줄일 수 있다. 본 논문에서는 사용자 입력과 어플리케이션 간의 상관 관계를 바탕으로 수행 단계를 분석하였으며, 이에 맞추어 CPU 동작 주파수 범위를 조절함으로써 사용자 반응성을 유지하면서 소비 전력을 줄이는 방법을 제안한다. 이를 안드로이드 플랫폼에 적용하여 실험한 결과, 기존 대비 휴대폰의 이용 시간을 최대 2시간 증가시켰다.

1. 서론

전세계적인 스마트폰의 확산에 따라 성능에 대한 사용자의 기대치가 데스크톱만큼 높아지게 되었으며, 이러한 이유로 모바일 기기에도 고성능 멀티 코어 CPU가 이미 탑재되고 있다. 하지만 내장배터리로부터 전력을 공급받는 모바일 기기는 이동성이 중요하므로 배터리의 용량을 늘리는데 한계가 있다. 그러므로 에너지 효율성을 높이는 저전력 기술이 매우 중요하다.

기존 연구[1]에 따르면 스마트폰에서 디스플레이 다음으로 많은 전력을 소모하는 것은 CPU이다. 따라서 CPU 전력 소모를 줄이는 방법으로 CPU 이용률에 따라 동작 주파수와 전압을 동적으로 조절하여 소비 전력을 최소화하는 기술인 DVFS(Dynamic Voltage and Frequency Scaling)가 널리 쓰이고 있다. 특히 모바일 어플리케이션은 사용자 입력에 대한 반응성이 가장 중요하므로, 사용자가 체감하는 반응성을 해치지 않는 범위 내에서 동작 주파수를 조절하면 성능저하 없이도 소비 전력을 줄일 수 있다.

본 논문에서는 모바일 어플리케이션의 사용자 입력에 따른 수행 단계를 분류하고, 이에 맞춘 동작 주파수를 제공함으로써 사용자 반응성을 유지하면서도 소비 전력은 줄일 수 있는 기법을 제안한다.

2. 관련 연구

CPU의 동적 소비 전력은 CPU의 동작 주파수와 인가 전압의 제곱에 비례한다. 하지만 동작 주파수를 낮추면

CPU가 단위 시간당 수행하는 명령어가 줄어들어 성능 감소가 발생한다. DVFS는 동작 주파수와 전압을 동적으로 조절하는 기법으로, 이를 이용하면 CPU의 소모 전력과 성능을 조절할 수 있다.

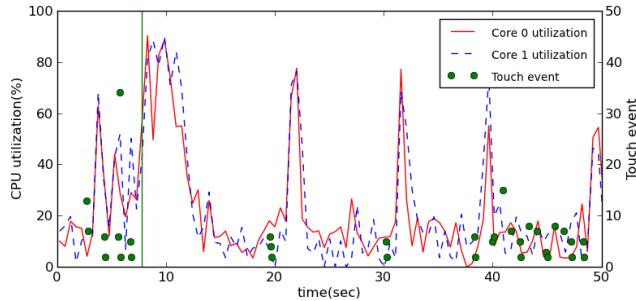
기존 연구에서는 CPU의 이용률에 맞추어 동작 주파수를 변경하였다. 하지만 이 방식으로는 모바일 환경에서 중요한 사용자 반응성과 저전력을 동시에 달성할 수 없다. 리눅스에 포함된 Conservative governor는 CPU 이용률에 따라 단계적으로 동작 주파수를 변경하는 방법으로 전력 소모는 줄어드나 반응성이 떨어지고, Ondemand governor는 CPU 이용률이 임계치를 넘는 경우 동작 주파수를 최대로 변경하는 방법으로 반응성은 향상되나 필요 이상의 전력 소모가 발생한다. Mike Chan은 반응성에 민감한 어플리케이션을 위한 interactive governor[2]를 안드로이드에 추가하였으나, 사용자의 입력에 따른 워크로드 특성 변화는 고려하지 못했다. 하나의 워크로드를 동적으로 다른 성향으로 구분하는 방법[3]이나, 사용자와의 상호 동작하는 워크로드를 위한 DVFS[4] 등도 연구되었지만, 모두 데스크탑 환경을 대상으로 하기 때문에, 입력 장치 등이 다른 모바일 환경에는 적용하기 어렵다.

3. 제안 기법

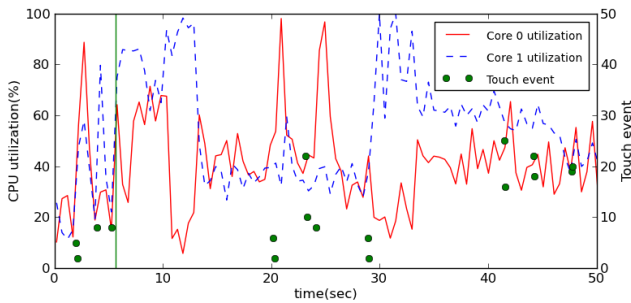
3.1 사용자 입력과 어플리케이션 간의 상관 관계 분석

모바일 환경에서는 제한된 화면 크기로 인하여 사용자와 상호 동작하는 어플리케이션은 대부분 1개만

존재하며 [5], 사용자의 입력에 따라서 어플리케이션의 특성이 동적으로 변한다. 다음은 갤럭시 넥서스 폰에서 CPU의 동작 주파수를 700Mhz로 고정하고 사용자의 입력에 따른 CPU의 이용률의 변화를 100ms마다 측정하여 Low pass filter를 적용한 결과이다.



[그림1 웹 브라우저의 상관 관계 분석]



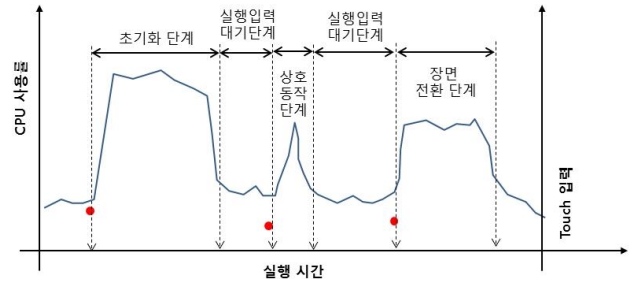
[그림2 Cut the rope 게임의 상관 관계 분석]

그림 1, 2에서 실선은 코어 0번, 점선은 코어 1번의 이용률 변화를 시간에 따라 측정한 결과이며, 점들은 사용자의 터치 입력을 커널의 input event를 통하여 수집한 횟수를, 왼쪽의 세로선은 각 어플리케이션이 시작된 시점을 나타낸다. 그림 1의 브라우저의 경우, 어플리케이션이 실행되는 시점(8초)과 사용자가 링크를 클릭(20초, 30초, 39초)하여 웹 페이지를 렌더링 하는 동안 CPU 이용률이 매우 높아지고, 40초 이후 주소창에 단순히 문자를 입력하는 경우에는 사용자의 입력과 상관없이 CPU의 이용률이 낮았다. 그림2의 게임에서는 실행 직후 수 초간 CPU 이용률이 매우 높다. 하지만 게임이 시작된 20초 이후에는 사용자의 입력에 따라서 CPU의 이용률이 변동되며, 다시 새로운 스테이지로 이동하는 30초 지점에서 수 초간 이용률이 증가하였다. 또한 게임 역시 목적에 부합하지 않는 입력을 하는 경우(40초 이후)에는 CPU 이용률에 큰 변화가 없었다.

3.2 모바일 어플리케이션의 수행 단계 분류

앞서 분석한 사용자 입력과 어플리케이션 간의 상관 관계 바탕으로 모바일 어플리케이션의 수행 단계를 그림3과 같이 4단계로 분류할 수 있다.

‘초기화 단계’는 어플리케이션이 실행되어 필요한 라이브러리 등을 로딩하고, 내부 자료구조를 초기화하는 단계이다. ‘실행 입력 대기 단계’는 사용자와의 상호 동작을 발생시키는 실행 입력을 기다리는 단계이다.



[그림3 모바일 어플리케이션의 수행 단계]

여기서 실행 입력이란 수행할 연산을 급격하게 증가시키는 사용자의 입력으로, 앞서 설명한 브라우저의 링크 클릭이나 게임에서의 유효한 입력이 이에 해당한다. ‘상호 동작 단계’는 사용자의 실행 입력에 반응하기 위한 연산이 수행되는 구간으로, 이 구간이 길어질수록 반응성은 나빠지게 된다. 마지막으로 ‘장면 전환 단계’는 어플리케이션이 명시적으로 사용자에게 알리고 장시간의 작업을 수행하는 단계이다. 예를 들면, 게임에서 다음 스테이지로 이동하기 위해서 새로운 맵과 데이터를 로딩하는 경우가 해당된다.

‘초기화 단계’와 ‘장면 전환 단계’에서는 CPU의 이용률은 높지만, 사용자는 특별한 입력 없이 대기하기 때문에 이 단계들이 빨리 끝날수록 반응성에 좋다. 한편 기존 연구[6]에 따르면, 모바일 환경에서 사용자는 입력한 후 100ms 이내로 반응이 발생하면 자연스럽게 느낀다. 그러므로 ‘실행 입력 대기 단계’와 ‘상호 동작 단계’에서는 사용자 입력과 수행할 연산량에 따라 동작 주파수를 조절하여 사용자의 체감 성능 저하 없이 소비 전력을 절약할 수 있다.

모든 모바일 어플리케이션에서 위 4단계가 명확하게 나타나는 것은 아니다. 예를 들어 사용자와 상호 동작이 많은 Angry Bird Space와 같은 게임의 경우 실행 입력 단계와 상호 동작 단계가 명확하게 구분되나, 그렇지 않은 메신저 프로그램 등에서는 이를 구분하기 어렵다.

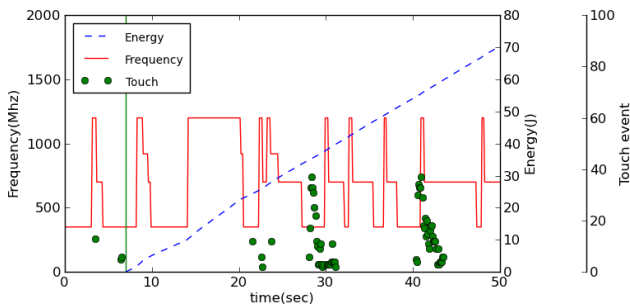
3.3 어플리케이션의 수행 단계를 고려한 DVFS 방안

지금까지의 연구에서는 CPU의 사용량만을 고려하여 동작 주파수를 변경하였다. 하지만 이번 연구에서 확인한 결과, 대부분의 어플리케이션에서 상호 동작 단계에서 처리할 연산이 많지 않아 동작 주파수는 즉시 낮은 단계로 떨어졌다.

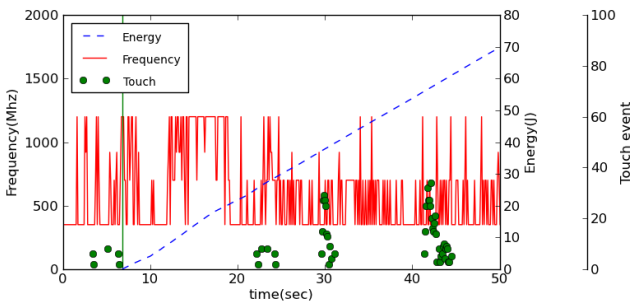
본 연구에서는 사용자의 입력과 CPU 이용률을 기반으로 어플리케이션의 수행 단계를 미리 예측하여, 실행 입력 대기 단계와 상호 동작 단계에서는 연산량이 많지 않기 때문에, 동작 주파수 변경 폭을 상대적으로 낮은 단계로 설정하였다. 반면에 초기화 단계와 장면 전환 단계에서는 높은 동작 주파수도 사용 가능토록 하여, 사용자의 대기 시간을 최소화 하였다. 제한한 기법은 CPU의 소비 전력을 최소화하면서도 반응성을 기존과 동일하게 유지할 수 있다.

4. 평가 및 분석

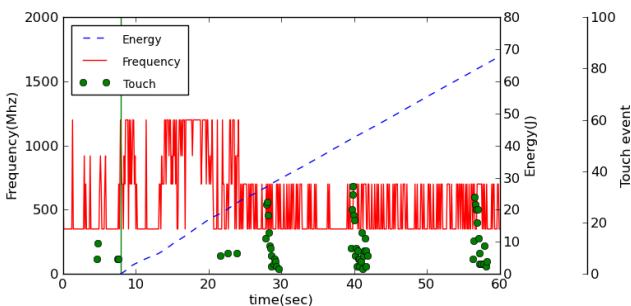
제안한 기법을 안드로이드 4.0.3에 포함된 리눅스 커널 부분에 구현하였다. 실험 장비는 갤럭시 넥서스 폰을 사용하였으며, 소비 전력 측정을 위해 Monsoon Power meter를 사용하였다. 스마트폰 사용자들이 일반적으로 많은 시간 사용하는 것 중에 하나가 게임[7]이고, 앞서 분석한 수행 단계들도 명확히 들어가기 때문에, 실험에는 대표적인 게임인 Angry Bird Space를 사용하였다. 기존 연구에서 제시된 ondemand, interactive governor와 비교하였으며, 사용자 반응성과 소비 전력을 기준으로 평가하였다. (그림4,5,6)



[그림4 ondemand governor시 시스템 trace]



[그림5 interactive governor시 시스템 trace]



[그림6 제안한 DVFS 사용시 시스템 trace]

우선 모든 경우에서 게임의 반응성에는 문제가 없었다. 하지만 ondemand와 interactive를 사용했을 때는 게임을 진행하는 동안 수시로 필요 이상으로 높은 동작 주파수로 변경되었다. (그림 4의 30초 이후, 그림 5의 25초 이후) 특히 ondemand의 경우에는 동작 주파수가 다시 낮아지는데 상대적으로 많은 시간이 소요되었다. 반면에 제안한 기법은 초기화 단계와 장면 전환 단계에서는 반응성을 위하여 높은 동작 주파수로

변경되는 반면에, 게임이 진행되는 단계에서는 낮은 동작 주파수 대역을 유지하는 것을 확인할 수 있다. 또한 에너지 측면에서도 제안한 기법이 더 적은 에너지를 사용하는 것을 확인할 수 있다.

표1은 Angry Bind Space 게임을 2시간 동안 진행하였을 때 소모되는 에너지를 계산한 결과이다. 실험에 사용된 스마트폰은 3G를 활성화한 대기 상태에서 시간당 150J이 소모된다. 따라서 제안한 기법을 적용한 상태에서 Angry Bind Space 게임을 2시간 동안 진행한다면, 통화 대기 시간이 ondemand 대비 2시간, interactive 대비 45분 증가하게 된다.

Governor	에너지(kJ)
Ondemand	11674.6
Interactive	11409.2
제안한 DVFS 방식	11293.4

[표1 각 governor 별 소모 에너지]

5. 결론 및 향후 연구

본 논문에서는 모바일 환경에서 사용자의 반응성을 유지하면서도 소비 전력을 줄일 수 있는 방법을 제안하였다. 사용자의 입력과 CPU 이용률을 기반으로 어플리케이션의 수행 단계에 맞추어 변경 가능한 실행 주파수를 제한함으로써 사용자의 반응성을 유지하면서 소비 전력을 줄이는 방법을 제안하였다. 추후 이를 다양한 어플리케이션에 적용해 볼 것이며, CPU나 memory의 일부를 끄는 hotplug 기법을 사용하여 스마트폰의 소비 전력을 줄이는 연구도 진행하고자 한다.

6. 참고 문헌

- [1] A. Carroll and G. Heiser, "An Analysis of Power Consumption in a Smartphone" USENIX ATC 2010
- [2] Linux CPUFreq Governors, <http://androidxref.com/source/xref/kernel/common/Documentation/cpu-freq/governors.txt>
- [3] K. Choi, R. Soma et al, "Dynamic Voltage and Frequency Scaling based on Workload Decomposition" ISLPED 2004
- [4] H. Zheng and J. Nieh, "RSIO: Automatic User Interaction Detection and Scheduling", ACM SIGMETRICS 2010
- [5] H. Falaki, R. Mahajan, S. Kandula et al, "Diversity in Smartphone Usage" MobiSys 2010
- [6] X. Zhao, Y. Guo and X. Chen, "Transaction-based adaptive dynamic voltage scaling for interactive applications" ISLPED 2009
- [7] Q. Xu, J. Erman et al, "Identifying Diverse Usage Behaviors of Smartphone Apps", ACM SIGCOMM IMC 2011