배터리 잔량에 기반한 동적 전력 관리 기법

최석원° 차호정
연세대학교 컴퓨터과학과
{sukwon, hjcha}@cs.yonsei.ac.kr

A Dynamic Voltage Scaling based on Battery Residual

Sukwon Choi° Hojung Cha
Dept. of Computer Science, Yonsei University

요 약

본 논문은 기존 전력 소비 감소만을 고려한 DVS와 달리 배터리 특성에 기반하여 DVS를 수행하는 기법을 제시한다. 배터리는 잔량에 따라 다른 전력 소비 형태를 갖고 있으므로 잔량에 따라서 적합한 DVS정책을 수행해야 효율적이다. 본 논문에서는 실험을 통해서 배터리의 특성을 파악하고, 그 특성에 따라 배터리 구간을 설정한 후 그에 적합한 DVS 알고리즘을 적용하는 기법을 제시한다. 이를 통해 효율적인 DVS정책을 수립 할 수 있음을 IPAQ 5550 PDA에서 리눅스 운영 체제에서의 실험을 통해 보여준다.

1. 서론

최근 PDA나 스마트폰 등의 이동기기에 대한 관심이 증폭되고 있고 사용량 또한 증가하고 있다. 이러한 이동기기에서 소비전력 감소는 중요한 요소이며, 이를 위한 대표적 기법중 하나가 DVS(Dynamic Voltage Scaling)이다[1]. 기존의 DVS알고리즘은 프로세서가 소비하는 전력을 최대한 낮추면서 태스크의 데드라인을 보장하는 방법들을 제시하고 있다. 한편 이동형 기기의 경우, 주 전력의 공급원이 배터리이다. 배터리는 화학적 장치의 특성을 가지고 있기 때문에 배터리의 사용 방법에 따라 사용 시간이 차이가 난다[2]. 따라서 배터리의 특성을 고려해야만 효율적인 DVS가 가능하다. Luo[3]는 배터리의 소모 전력을 최적화 하고, 피크파워 없이 태스크를 수행 하는 방식으로 배터리의 사용시간을 연장하는 방법을 제안하였다. Benini[4]는 배터리의 상태를 실시간으로 모니터해서 배터리잔량에 따라 QoS를 조정하여 배터리의 사용시간을 늘리는 방법을 제안하였다. 기존 배터리를 고려한 DVS알고리즘의 경우, 시뮬레이션을 통한 제한된 환경에서 특수한 태스크를 기반으로 연구가 진행되었기 때문에 실제 시스템에 적용이 어렵고, 일반적인 태스크는 고려하지 않고 있다.

본 논문에서는 배터리 모니터와 태스크 모니터를 이용하여 배터리 및 태스크의 현재 상태를 모니터링하고, 모니터링된 정보에 따라 배터리의 잔량과 태스크의 상태 정보를 이용해 배터리의 구간을 설정한 후, 각 구간에 적합한 DVS정책을 사용하여 배터리 사용시간을 늘릴 수 있도록 한다. 제안된 시스템은 범용 운영 체제에서 실시간으로 배터리와 태스크를 모니터링 하여 배터리 잔량 및 현재 소모량 등의 배터리 정보와 태스크의 수행 시간 및 주기 등의 태스크 정보를 모니터링 한다. 모니터링된 정보를 기반으로 배터리 잔량에 따라 적절한 DVS기법을 적용하여 배터리 사용 시간을 연장 한다.

논문의 구성은 다음과 같다. 2장 에서는 논문의 배경에 대해서 설명하고, 3장에서는 제안된 DVS시스템의 구성과 기본적인 개념을 설명한다. 4장에서는 실험을 통해 제안된 시스템의 성능을 분석하며, 5장에서는 결론을 제시한다.

• 본 연구는 정보통신연구진흥원에서 지원하는 정보통신기초기술연구지원사업으로 수행하였음 (과제번호 : C1-2003-A1-2000-0240)

2. Motivation

배터리는 충전에서 방전되기까지 소비되는 전력량이 다양한 요 인들에 의해서 변화한다[2]. 그 중에서 가장 크게 영향을 미칠 수 있는 특성은 전력 소비 비율유지 효과(rate capacity)와 회복 충전 효과 (recovery effect)이다. 본 논문에서는 사전 실험으로 위 두 가지 배터리의 특성 중 회복 충전 효과 (recovery effect)를 알아보기 위한 실험을 하였다. IPAQ 3600 PDA 하드웨어는 전압 강하에 의해 측정된 배터리의 현재 잔량을 중앙 ASIC으로 보내준다. 논문에서는 리눅스 커널을 수정하여 ASIC으로 넘어온 배터리 잔량 정보를 이용하여 배터리 모니터를 구현하였다. 그림 1은 배터리 모니터를 이용하여 IPAQ 3600 PDA에서 태스크의 수행과 유휴를 반복하였을 때, 배터리 잔량 변화를 측정한 실험 결과이다.

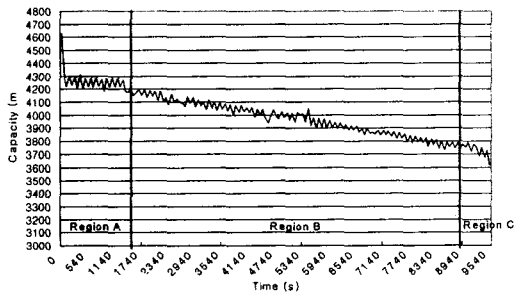

그림 1배터리의 Discharge에 따른 회복 충전 효과

태스크 수행시간은 30초로 하고 시스템의 유휴시간도 30초로 하여 2개의 상태를 반복적 수행하였다. 태스크를 수행할 때는 시스템의 소비전력이 4.2Watt이며 유휴상태의 소비전력은 0.3Watt로 고정시켰다. 사용된 배터리는 Li-on배터리이다. 그림에서 보듯이 배터리의 사용량에 따라 충전 회복 효과가 다르게 나타난다. Region A의 경우 충전회복 효과가 크며, Region B의 경우는 전력 소비 비율유지 효과와 비슷한 형태로 충전회복 효과가 나타났다. 즉, 충전회복 효과가 전력 소비 비율 유지 효과에 비해 크게 영향을 미치지 않는 것으로 나타났다. Region C의 경우, 충전 회복 효

과와 상관없이 배터리의 소비 곡선이 급격하게 떨어지는 것을 볼 수 있다. 이러한 결과는 배터리의 화학적 특징에 의해 충전량이 많을수록 배터리가 안정화되기 때문이다[2]. 그림 1과 같이, 각각의 영역에서 배터리 특성에 적합한 DVS기법을 사용하게 되면, 배터리의 사용시간을 연장 할 수 있다. 본 논문에서는 이러한 사실에 기초하여, 배터리 잔량에 따라 구간을 설정하고, 각각의 구간에 적합한 DVS기법을 제시한다.

3. 배터리 잔량을 고려한 DVS 시스템

다음은 배터리 잔량과 태스크의 상태를 실시간 모니터링 하여, 효율적인 전력관리를 하는 DVS 시스템에 대해 기술한다.

3.1. 시스템 개요

실시간으로 배터리 잔량을 모니터링하여 배터리 상태를 측정하고, 측정된 상태에 따라 적절한 DVS기법을 사용하기 위해서 배터리 및 태스크 모니터 시스템이 필요하다. 그림 2는 제안하는 DVS시스템의 개요이다. 제안하는 DVS시스템은 크게 모니터링 모듈, 정책 모듈, 프로세서 제어 모듈의 3개로 구성된다. 모니터링 모듈은 배터리와 태스크의 정보를 수집하는 부분으로 배터리 상태 모니터와 태스크 모니터로 구성된다. 정책 모듈은 배터리의 특성을 고려한 DVS알고리즘을 이용하여 프로세서의 전력 상태를 결정한다. 정책 모듈은 태스크 정보를 이용하여 태스크의 상태를 분류하는 태스크 분류기와 주기적 태스크의 QoS를 예측해 주는 QoS예측기, 그리고 배터리의 상태에 맞추어 적합한 DVS알고리즘을 선택하여 주는 DVS알고리즘 선택기로 구성된다. 프로세서 컨트롤 모듈은 DVS및 DFM(Dynamic Frequency Scaling), FCS(Frequency Change Sequence) 등의 메커니즘을 수행할 수 있는 프로세서 컨트롤 API와 프로세서 상태를 피드백 받을 수 있는 프로세서 상태 API로 구성되어 있다.

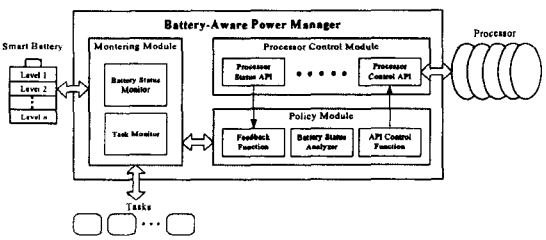

그림 2 Battery-aware DVS System

3.2 DVS 기법

배터리및 태스크 모니터링과 함께 모니터링 정보에 따라 각 구간에서 효과적으로 동작하는 DVS기법이 필요하다. 일반적으로 범용 운영 체제의 경우 사용자 응답시간 혹은 태스크의 특징에 따라 QoS를 가지게 된다[5]. 특히 MPEG과 같은 멀티미디어 데이터의 경우, 그림 3와 같이 주기적인 QoS 보장 시간 Q_i를 가진다. 본 논문에서는 QoS가 보장되어야 하는 주기적 태스크에서 배터리의 사용 시간을 연장하는 방법을 제시한다.

프로세서가 n개의 전력 상태를 가지고 있다고 가정하고 이를 $P_i(1 \le i \le n)$로 표현 한다. DVS를 수행하지 않았을 때 성능이 최고로 되는 전력 상태를 P_{max}, 최저 성능일때 전력 상태를 P_{min}으로 정의한다. 태스크는 P_{max} 와 P_{min} 범위에서 수행 될 수 있으며, P_i에 따라 수행시간 W_i와 유휴시간 $Idle_i$의 길이가 달라진다. 배터리의 특성과 W_i, $Idle_i$의 관계를 보면 W_i의 길이는 전력 소비 비율유지 효과와 관계 있고, $Idle_i$의 길이는 충전 회복 효과와 관계가 있다.

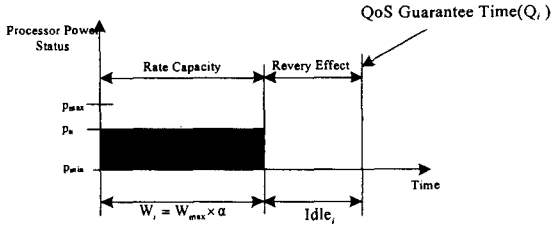

그림 3 배터리의 특성에 따른 태스크의 수행

현재 태스크에서의 QoS를 Q_i이라고 하면

$$W_i = W_{max} \cdot \alpha \; (\alpha \propto P_i,\ P_{min} \le P_i \le P_{max}) \quad (1)$$

$$Q_i = W_i + Idle_i \; (\text{이때 태스크 } T_i\text{는 } Q_i > W_i \text{인 태스크}) \quad (2)$$

으로 표현 할 수 있다. 여기서 α는 0에서 1사이의 값으로, 프로세서의 전력상태 i에서의 태스크 실행 시간을 결정하는 상수값이고, W_{max}는 P_{max}에서의 태스크 수행시간이다. α는 태스크 모니터에 의해서 구해진다.

본 논문에서는 식(1), (2)를 기반으로 각각의 배터리상태 구간에서 W_i, $Idle_i$을 조정함으로써 배터리 특성에 따른 배터리 사용시간을 극대화 하는 P_i의 설정방법을 제시한다. 이를 위해 첫째로 영역 설정을 위한 임계값 설정, 두 번째로 회복 충전 효과 우선 영역을 위한 정책, 마지막으로 전력 소비 비율 유지 효과 우선 영역을 위한 정책의 3가지가 필요하다.

임계값(T_R)은 일반적으로 회복 충전 효과가 배터리에 미치는 영향이 소비 비율 유지 효과에 미치지 못할 때까지 잔량을 측정함으로써 얻어진다. 이 값은 배터리의 종류 및 용량에 따라서 달라진다. 본 논문에서는 태스크를 회복 충전 효과와 소비 비율 유지 효과를 나타 낼 수 있도록 각각의 태스크를 수행하여 그 교차점을 찾아 임계값을 구하였다.

충전회복 효과 우선 영역에서는 충전 회복 효과를 극대화 시키는 알고리즘을 사용해야 한다. 충전 회복효과는 $Idle_i$의 길이가 길수록 효과가 크게 나타나기 때문에 $Idle_i$을 늘여 줄 수 있는 방법으로 태스크를 수행하여야 한다. 따라서 $\alpha = 1$일때의 P_i를 구하면 된다. 일반적인 운영체제에서는 $P_i = P_{max}$ 설정하면 되는데, 이때 $Idle_i$도 같이 짧아지기 때문에 $Idle_i = Q_i - W_{max}$로 태스크 수행시간을 유지한다.

소비 비율 유지 우선 영역에서는 배터리에 걸리는 부하를 줄이는 방향으로 태스크가 수행 되어야 한다. 부하를 줄이기 위해서 $Idle_i$를 최소화 하면서 QoS를 보장 할 수 있는 $P_{optimal}$을 찾아야 한다. $P_{optimal}$을 구하는 방법은 다음과 같다.

$$P_{optimal} = \frac{W_{max} - Idle_i}{Q_i - Idle_i} \quad (3)$$

4. 실험

실험은 리눅스 운영체제를 수행하는 IPAQ 5550 PDA에서 배터리 모니터와 태스크 모니터를 구현하여 수행 하였다. 배터리 모니터는 IPAQ 5550 PDA내부의 DS2760 표준 스마트 배터리 모니터 칩의 정보를 리눅스 커널의 HAL영역을 수정하여 배터리의 잔량 및 현재 소비량을 체크 할 수 있도록 하였다. 태스크 모니터는

스케줄러와 시스템콜에서 태스크의 실행 정보를 수집하여 QoS 및 수행 시간과 유휴 시간을 측정할 수 있도록 커널을 수정하였다. 태스크는 200Kbps, 25 frames를 갖는 MPEG데이터를 버클리 MPEG플레이어를 통해 수행되는 태스크를 이용하였다. MPEG 데이터를 10frames단위로 주기성을 부여하여 실행하고, 태스크 모니터를 이용하여 QoS 보장 시간 Q_i와 α값을 실시간으로 모니터링 하였다. 성능을 비교하기 위해서 Vertigo[5]에서 사용한 최근 사용량에 기반한 DVS기법을 사용한 방법(DVS로 표기)과, DVS를 사용하지 않은 방법(Non-DVS로 표기), 그리고 제안된 방법(Proposed로 표시)을 이용하여 각각 같은 태스크를 수행하였다. QoS를 측정하기 위해서 재생된 프레임수와 수행 시간을 측정하였다.

먼저 임계값을 구하기 위해서 충전 회복 우선 효과를 이용한 방법 과 전력 소비 비율 유지 효과를 이용한 방법으로 각각 DVS 한 결과, 임계값(T_R)은 3810mV이 최적값이였다. 이렇게 구해진 임계값을 이용하여 배터리 잔량이 3810mV이상인 영역을 충전 회복 효과 구간으로 지정하고 나머지 영역을 소비비율 유지 효과 구간으로 지정하였다. 그림 4에서 볼 수 있듯이 충전 회복 효과 구간에서는 약 18%정도의 배터리 시간의 연장이 있었다.

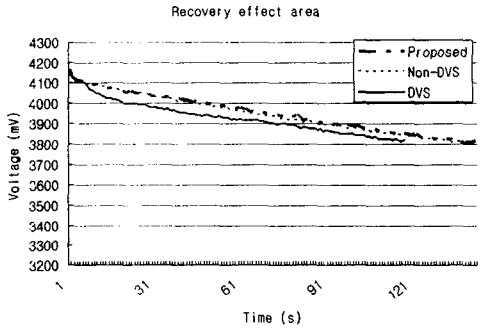

그림 4 200Kbps, 25 frames MPEG1에서 충전 회복 효과 구간

그림 5는 3810mV이하의 영역을 전력 소비 비율 유지 효과 구간에서 배터리 잔량이 Cut-off Voltage 이하로 떨어질 때 까지 태스크를 수행한 결과를 보여준다. 그림에서 볼 수 있듯이 충전 회복 효과구간에서는 약 12%정도의 배터리 시간이 연장되었다

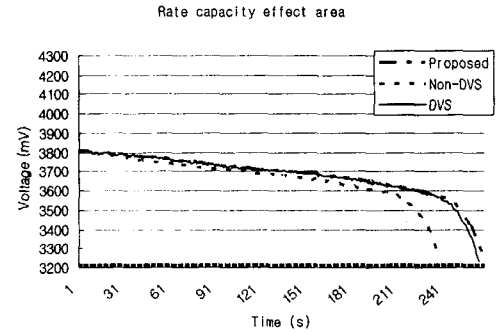

그림 5 200Kbps, 25 frames MPEG1에서 소비 비율 유지 효과 구간

이와 같은 결과에 따라서 3810mV를 임계값으로 하고 각각 구간에 적합한 DVS알고리즘을 사용하여 실험한 결과 그림 6와 같은 결과를 얻을 수 있었다.

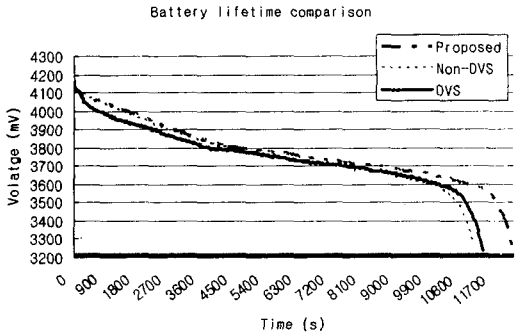

그림 6 정책에 따른 배터리 감소 곡선 그래프

표 1은 이상의 결과를 QoS를 측정한 결과와 함께 정리한 것이다. 제시된 기법을 사용하면 DVS사용하지 않은 시스템에서 보다 QoS를 0.07%만 저하 시키면서 약 11%의 배터리 사용시간 연장을 얻을 수 있다.

표 1 배터리 사용시간 연장 및 QoS 측정

	Battery lifetime	Frame rate	Lifetime extension	QoS drop
Non DVS	11070 Sec.	9.992 f/s	—	—
General DVS	11340 Sec.	9.944 f/s	3%	0.05%
Proposed DVS	12180 Sec.	9.925 f/s	11%	0.07%

4. 결론

본 논문에서는 배터리의 잔량에 따라 구간을 설정하고 각각에 적합한 DVS기법을 사용함으로써 효율적인 DVS기법을 구현하였다. 제시된 기법은 IPAQ 5550PDA에서 MPEG데이터를 플레이한 실험을 통해 다른 DVS기법에 비하여 범용 운영체제에서 실제 태스크가 실행될 때 효과적임을 보여 주었다. 향후 연구과제로는 그림 2에서 보여준 Region 3영역에서 수행될 기법을 보강하여, 배터리의 사용 시간을 더욱 연장 하는 것이다.

참고문헌

[1] K. Govil, E. Chan, H. Wasserman,"Comparing algorithms for dynamic speed-setting of a low-power CPU, " *International Conference on Mobile Computing and Networking*, pp. 13-25, Nov. 1995.

[2] D. Linden, T.Reddy, *Handbook of Batteries*, McGraw-Hill, NY. 2001.

[3] J. Luo, J. K. Niraj,"Battery-aware static scheduling for distributed real-time embedded systems Design," *Proceedings of ACM/IEEE conference on Design automation*, Las vegas, Nevada, United States, pp. 444 -449, Dec. 2001.

[4] L. Benini, G. Castelli, A. Macii, E. Macii, R. Scarsi,"Battery-Driven Dynamic Power Management of Portable Systems," *International Symposium on System Synthesis(ISSS)*, Madrid, Spain, pp. 25-35, Sep. 2000.

[5] K. Flautner, T. Mudge,"Vertigo:Automatic Performance-Setting for Dynamic Voltage Scaling," *Proceedings of the Operating Systems Design and Implementation(OSDI)*, pp. 105-116 Dec. 2001.