

2차 전지의 효율적 사용을 위한 운영체제 수준에서 순간최대 전력 분산기법

이병호⁰, 김형필, 서효중
가톨릭대학교 컴퓨터정보공학부
{canine⁰, pili, hjsuh}@catholic.ac.kr

Operating System Driven Peak Power Distribution for Secondary Battery Optimization

Byong-Ho Lee⁰, Hyo-Joong Suh, Hyong-Pil Kim
Department of Computer Science Engineering, The Catholic University of Korea

요 약

최근 배터리를 에너지원으로 작동하는 모바일 디바이스의 사용이 늘어남에 따라 배터리를 효율적으로 사용하기 위한 연구에 많은 관심이 증대되고 있다. 한편 배터리는 각각의 방전 패턴에 따라 사용가능한 시간이 결정이 되며 따라서 효율적인 배터리 관리 정책에 따라 배터리 사용 시간을 연장 할 수 있다. 이에 본 논문에서는 기존 평균 전력 소비 감소만을 고려한 방법들과 달리 배터리의 방전 특성을 고려한 순간 최대전력 분산 기법을 적용하여 운영체제 수준에서 효율적인 전력사용 기법을 제시하고 이를 통해 전체 시스템의 효율적인 에너지 관리를 할 수 있음을 보이고자 한다.

1. 서 론

최근 배터리를 에너지원으로 작동하는 모바일 디바이스들이 많은 응용에서 사용되어짐에 따라 배터리 사용시간을 효율적으로 관리하기 위한 연구에 많은 관심이 증대되고 있다. 이중 시스템 수준에서 전력사용 최적화에 대해 많은 논문이 발표 되고 있으며, 대표적으로 동적 전압 조절기법 DVS(Dynamic Voltage Scaling)[1]과 동적 전력 관리기법 DPM(Dynamic Power Management)이 있다. 동적 전압 조절기법은 프로세서의 동작 주파수에 따라 공급 전압을 온라인 상태에서 바꾸어 전력사용을 줄이는 기법이며, 동적 전력관리 기법은 시스템의 구성요소별 사용 패턴에 따라 해당 장치가 비활성화 상태에 놓인 경우 해당 장치가 제공하는 파워 상태 중 수행의 요구 조건에 맞추어 가장 적절한 파워 상태로 전이함으로써 전력의 소모를 줄이는 기법이다. 또한 이와 달리 단순히 전력 소모의 감소뿐만 아니라 배터리 방전 특성에 따른 배터리 성능변화에 기반 한 전력사용 최적화 연구에 많은 관심을 보이고 있다[2].

본 논문에서는 2 차전지의 의해 작동하는 임의의 멀티미디어 응용을 수행하는 모바일 디바이스 모델을 정의하고, 에너지원인 배터리의 효율적 사용을 위한 운영체제 수준에서 지원하는 순간 최대전력(Peak Power) 분산 기법을 제시한다. 여러 디바이스가 복합적으로 사용되는

응용의 경우 짧은 시간 구간동안 디바이스의 사용의 집중으로 인한 순간 최대전력 사용이 발생 할 수 있다. 이러한 순간 최대전력이 발생할 경우 배터리의 성능은 악화되며 따라서 전체 시스템의 사용시간의 감소를 가져온다. 따라서 배터리 성능을 보장하기 위해 임의의 시간구간 동안 사용가능한 에너지의 양을 제한하고, 사용하는 에너지의 양을 초과할 경우 다음 시간구간까지 프로세서의 상태를 유휴(idle) 상태로 놓이게 함으로써 순간최대 전력 사용의 분산을 이룰 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 배터리의 특성에 대해서 설명을 하고, 3장에서는 시스템에서 순간 최대전력을 유발하는 태스크들과 이러한 태스크들의 에너지 소모의 측정에 대해 설명하고 4장에서는 제안된 시스템의 기법 및 실 시스템에서 구현 방법 및 수행 과정을 설명하며, 마지막으로 5장에서는 결론을 제시한다.

2. 배터리 특성

모바일 디바이스에서 사용되는 배터리는 일반적으로 리튬이온(Li-ion)과 같이 재사용 가능한 2차 전지가 사용되며 충전을 통해 전기 에너지는 화학 에너지로 방전을 통해 화학에너지는 전기에너지로 전환 된다.

배터리의 총 전력량은 다양한 요인들에 의해 영향을 받을 수 있는데 그 중에서 가장 크게 영향을 미칠 수 있는 특성은 전력 소비 비율유지 효과(rate capacity)와 회

복 충전 효과 (recovery effect)이다[3]. 전력소비 비율 유지효과는 큰 전류가 흐를 경우 배터리 전압이 차단 전압으로 일찍 감소하여 발생 가능한 전하의 양이 줄어들어 제공할 수 있는 전력량이 줄어드는 것을 말하며, 회복 충전 효과는 방전 전류의 크기가 줄어들 경우 전지 전압이 증가되어 배터리가 제공할 수 있는 용량이 늘어나는 것을 말한다. 이러한 화학적 특성으로 인해 배터리는 전력소모 패턴에 따라 제공할 수 있는 총 용량이 일정하지 않으며 방전전류의 증가에 따라 그 효율은 더욱 감소되는 비선형적인 특성을 보인다[3]. 따라서 이러한 배터리의 화학적 특성을 고려할 때 평균전력 사용량 감소만을 고려한 배터리 관리 정책은 효율적이지 못하며 배터리 사용시간은 순간 최대전력의 분산을 통해 최대화 될 수 있다.

그림 1은 방전 전류의 비율에 따라 배터리의 효율이 선형적이지 않음을 보여주며 따라서 순간최대 전력의 분산을 통해 배터리의 사용시간을 증대 시킬 수 있음을 보여주고 있다.

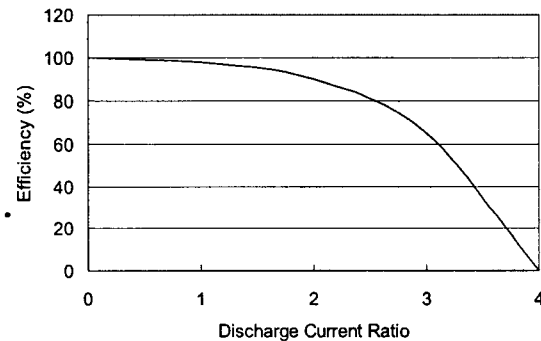


그림1 배터리 효율

3. 시스템에서 에너지를 소모하는 태스크 분류

모바일 디바이스에서 소모되는 에너지는 해당 휴대용 임베디드 기기에 장착되어 사용되는 컴포넌트 및 그러한 컴포넌트를 사용하는 응용 소프트웨어에 따라 크게 좌우된다. 따라서 순간 최대전력을 유발하는 장치 또는 컴포넌트에 대한 전력 소모를 분석하고 이를 기반으로 한 전력 관리가 필요하다.

본 논문에서는 하드디스크에 저장된 mp3 포맷의 음악 파일을 재생하고 동시에 무선 랜을 통해 스트리밍 서비스를 제공하는 멀티미디어 시스템 응용모델을 정의한다. 이러한 응용에서 주기적으로 전력을 소모하고 순간 최대 전력 소모를 유발하는 태스크를 정의하면 다음과 같다.

- 프로세서를 통한 mp3 디코딩
- 하드디스크를 통한 읽기 수행
- 무선 랜을 통한 데이터 전송 및 수신

그 외의 디바이스 들은 작동 유무에 따른 전력 소모의 변화가 전체 시스템에 미치는 영향이 미비하기 때문에 정의하는 시스템에서 고려대상에 포함하지 않는다.

이들 각각의 태스크들이 소모하는 에너지는 각 태스크가 수행하는 단위 시간당 소비전력과 수행시간의 곱으로 나타낼 수 있으며, 프로세서의 경우 프로세서의 프로세서 사용시간과 단위 시간당 프로세서의 전력소모로 나타낼 수 있으며 무선 랜의 경우 데이터 송신 및 수신 데이터 크기와 데이터 송신 및 수신에 따른 단위 시간당 무선 랜의 소모 전력과의 곱으로 나타낼 수 있으며, 하드디스크의 경우 입출력에 따른 데이터 크기와 데이터 입출력을 수행함에 있어 단위 시간당 하드디스크의 소모 전력과의 곱으로 나타낼 수 있다. 여기서 데이터의 크기에 따라 각 디바이스에서 입출력에 소모되는 시간은 비례하다고 가정한다.

4. 제안하는 시스템

태스크들의 수행 시 발생하는 순간 최대전력 분산을 위해 이를 관리하기 위한 운영체제 수준에서 에너지 소비를 관리하는 정책이 필요하며 이를 만족하기 위해 본 논문에서 다음과 같은 정책을 제안한다.

제안하는 기법에서 사용되는 요소들을 다음과 같이 정의한다.

- T_e : 에너지 소모의 기준이 되는 시간 구간
- E_t : T_e 시간 동안 사용할 수 있는 총 에너지 양
- C_e : T_e 시간구간에서 현재 사용 가능한 에너지의 양
- $Task_i$: 순간 최대전력 분산을 위해 운영체제에서 관리되어지는 태스크
- P_i : $Task_i$ 의 수행함으로써 소비되는 에너지

제안하는 기법의 구체적인 절차는 다음과 같다.

- 시스템이 수행되는 동안 에너지 소모의 기준시간 구간 T_e 는 주기적으로 생성된다.
- T_e 구간동안 $Task_i$ 수행 시 마다 P_i 와 C_e 를 비교하여 $P_i \leq C_e$ 인 경우 $Task_i$ 는 수행이 되고, $P_i > C_e$ 인 경우 수행을 중단하고 유휴 상태로 대기한다.
- C_e 의 값은 새로운 T_e 이 시작할 때마다 E_t 값으로 초기화 되고 $Task_i$ 의 수행 시 마다 $C_e = C_e - P_i$ 으로 변경된다.

그림 2는 제안하는 시스템의 수행과정을 보여주고 있다.

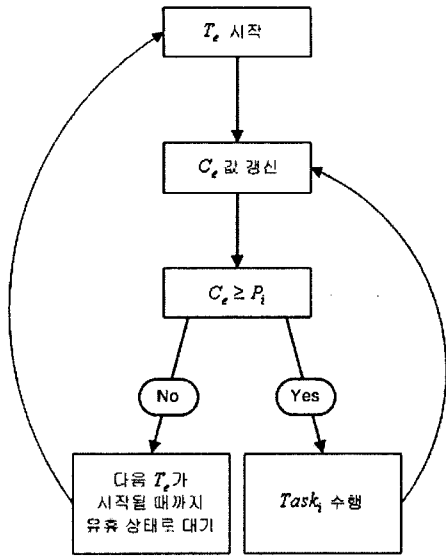


그림2 제안하는 시스템의 수행과정

제안하는 시스템의 실 시스템에서 구현은 리눅스 커널의 수정을 통해 구현될 수 있다. T_e 는 리눅스 커널에서 1틱(tick) 시간의 임의의 상수배로 나타낼 수 있으며, 커널 2.4버전에서 1틱은 10msec 이다. 커널에서는 jiffies 값을 갱신하기 위해 매 틱마다 타이머 인터럽트 발생하고[4] 이 시점을 기준으로 새로운 T_e 는 시작과 종료를 수행 할 수 있다. T_e 시간구간에서 현재 사용가능한 에너지의 양을 나타내는 C_e 는 커널의 전역변수로 등록을 통해 새로운 T_e 의 시작 시 또는 $Task_i$ 의 수행 시 마다 갱신을 할 수 있다. $Task_i$ 가 소모하는 전력 P_i 는 입출력 디바이스의 디바이스 드라이버의 수정을 통해 구현될 수 있으며, 데이터 입출력이 발생하는 ISR(Interrupt Service Routine) 수행 시 각 동작이 소모하는 전력과 수행하는 데이터의 양의 곱으로 자신의 소비할 에너지를 구할 수 있으며, 이 값을 통해 전역변수 C_e 와 비교하고 $Task_i$ 를 수행할지 여부를 결정할 여부를 판단 할 수 있다.

그림 3과 그림 4는 기존의 시스템과 제안하는 시스템의 태스크의 수행 과정을 나타낸 것이다. 그림 3은 임의의 시간 단위 Δ_t 에 대해 $8 \cdot \Delta_t$ 동안 $Task_1$, $Task_2$, $Task_3$, $Task_4$ 가 반복적으로 수행되는 예로써 각 태스크는 임의의 에너지 단위 Δ_e 에 대해 $4 \cdot \Delta_e$, $3 \cdot \Delta_e$, $4 \cdot \Delta_e$, $2 \cdot \Delta_e$ 의 에너지를 소모한다. 그림 4는 임의의 단위 시간 Δ_t 에 대해 에너지 소모의 기준이 되는 시간구간 T_e 를 $4 \cdot \Delta_t$ 로 정의하고 T_e 구간 동안 사용가능한 에너지 양 E_t 는 $10 \cdot \Delta_e$ 정의하여 순간 최대전력 소모 분산 기법을 적용한 시스템을 수행한 결과이다.

그림 3의 경우 각각의 태스크 들의 연속된 수행으로

인해 순간 최대 전력소모가 발생을 하였다. 반면에 순간 최대전력 소모 분산 기법을 적용한 그림 4의 경우 $Task_1$ 과 $Task_2$ 의 수행 후 현재 사용가능한 에너지의 양 C_e 는 $1 \Delta_e$ 가 되어 다음 $Task_3$ 은 다음 T_e 의 시작 때 까지 유휴 상태로 대기하게 된다. 이를 통해 적절한 유휴시간의 분배를 통한 순간최대전력의 분산이 이루어 짐을 볼 수 있다.

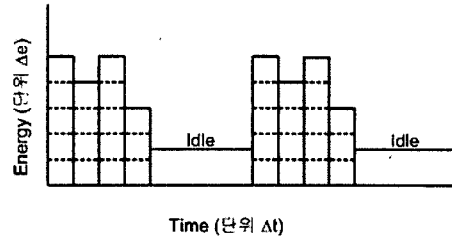


그림 3 기존 시스템

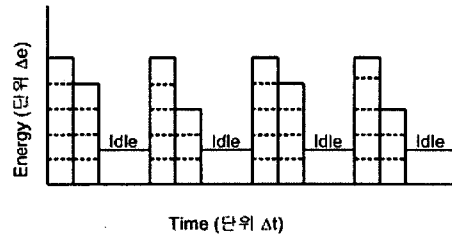


그림 4 제안하는 시스템

5. 결론

본 논문은 모바일 디바이스에서 전력소모 형태를 모델화 하고 이를 기반으로 배터리를 효율적으로 사용하기 위한 효율적사용을 운영체제 수준에서 지원하기 위한 기법과 구체적인 구현 방법에 대해 제시하였다. 향후 연구 과제로는 제안한 기법을 리눅스 커널 상에서 구현 하고 이를 통한 에너지 효율성을 측정 및 성능 평가를 제시하는 것이다.

6. 참고 문헌

- [1] T. Mudge, "Power: A First Class Architectural Design Constraint," IEEE Computer, Vol. 34, No. 4, pp. 52-58, April 2001.
- [2] H. Zeng, C. Ellis, A. Leveck, and A. Vahdat. Ecosystem: Managing energy as a first class operating system resource. In Proceedings of Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 2002.
- [3] D. Linden, T.Reddy, Handbook of Batteries, McGraw-Hill, NY. 2001.
- [4] R. Love. Linux Kernel Development. Sams Publishing, 2005