

실시간 시스템에서 태스크별 평균 실행 시간을 활용한 동적 전압 조절 방법

*방철원, 김용석

강원대학교 컴퓨터 정보통신공학과

핸드폰/011-9792-2614

Dynamic Voltage Scaling Using Average Execution Time in Real Time Systems

*Chul-Won Baang, Yong-Seok Kim

Dept. of Computer Information and Communications Engineering,

Kangwon National University

E-mail : gunbaang@hanmail.net, yskim@kangwon.ac.kr

Abstract

Recently, mobile embedded systems used widely in various applications. Managing power consumption is becoming a matter of primary concern because those systems use limited power supply.

As an approach reduce power consumption, voltage can be scaled down. according to the execution time and deadline. By reducing the supplying voltage to $1/N$, power consumption can be reduced to $1/N$.

DPM-S is a well known method for dynamic voltage scaling.

In this paper, we enhanced DPM-S by using average execution time aggressively. The frequency of processor is calculated based in average execution time instead of worst case execution time.

Simulation results show that our method achieve up to 5% energy savings than DPM-S.

1. 서론

최근에 휴대용 기기들이 급격히 발달하여 널리 보급

이 논문은 2003년도 강원대학교 두뇌한국21사업에 의하여 지원되었음.

되고 있다. 이러한 이동성이 요구되는 내장형 시스템들은 대부분 제한된 전력 공급 장치를 사용하기 때문에 주어진 전원 용량을 가지고 얼마나 오랫동안 사용할 수 있는지가 중요한 관심사가 되고 있다. 고성능 프로세서들을 사용함으로써 인해서 전력 소모가 계속 증가하는데 비해 휴대용 배터리의 성능은 그 기대치에 못 미치고 있다. 또한 인공위성과 같이 전력 소비량이 많은 시스템에서는 전력 소모로 인하여 열의 발산이 증가함으로써 인해서 시스템의 신뢰도가 감소하는 문제가 있다. 그로 인해 냉각 기술이 필요하게 되고 더 비싼 패키지가 필요하게 되었다. 따라서 이동 내장형 시스템에서 저전력 구현 기법에 대한 연구가 중요한 문제로 대두되고 있다.

실시간 시스템에서 동적 전압 조절 기법은 프로세서의 속도를 조절하여 전력 소모량을 줄이는 기법이다. 실시간 시스템에서 주어진 태스크들의 연산량이 최악의 경우의 연산량보다 작을 경우 주어진 제한 시간 내에 끝낼 수 있는 한도 내에서 프로세서의 동작 속도를 늦추어주면 이에 비례하여 공급 전압을 낮출 수 있고 따라서 전력 소모량을 감소시킨다. 공급 전압이 V_{dd} 이고 회로의 커패시턴스 부하는 C_{ef} 일 때 전력 소모량은 $P_d = C_{ef} \times V_{dd}^2 \times f$ 이다[2]. 여기서 f 는 클럭 주파수를 나타내는데 프로세서의 클럭 속도는 대

체로 공급 전압에 비례하고 소비 전력은 공급 전압의 제곱에 비례하기 때문에 동작 속도를 $1/N$ 로 낮추면 전체 실행 시간은 N 배로 늘어나지만 단위 시간 당 소비 전력은 $1/N^2$ 로 감소하게 된다. 전체 작업을 완료하는데 소모되는 전력은 $1/N$ 로 감소하게 된다.

이러한 원리를 이용하는 동적 전압 조절 기법으로서 여러 가지 연구가 이루어졌는데 현재까지는 DPM-S가 가장 좋은 결과를 보여주고 있다[1].

본 논문에서는 태스크별 평균 실행 시간을 활용함으로써 DPM-S보다 개선된 동적 전압 조절 기법을 제안하였다.

2. 전압 조절 기법

실시간 시스템에서 동적 전압 조절 기법은 프로세서의 공급 전원을 조절하여 에너지 소모량을 줄이는 기법으로서 태스크들의 여유시간 부분을 이용한다. 여유시간은 태스크 집합이 주어질 때 생기는 자연적인 여유시간 즉 정적인 여유시간과 태스크의 실제 실행시간과 최악의 경우의 실행 시간과의 차이에 따른 동적인 여유시간으로 나눌 수 있다.

이러한 여유시간을 태스크의 실행시간에 적용하여 프로세서의 클럭 속도를 정적으로나 동적으로 낮춤으로써 소모 전력을 줄일 수 있게 된다.

실시간 시스템에서는 다른 시스템과 달리 주어진 작업에 대한 마감 시간을 가지므로 클럭 속도의 조절은 마감시간을 넘기지 않는 한도 내에서 제한된다.

태스크 집합의 여유시간을 어떻게 활용하는가에 따라 다음과 같이 여러 가지 전압 조절 기법들이 연구되었다.

여기에서 주어진 태스크 집합의 마감시간은 d 이고 각 태스크의 최악의 경우의 실행시간과 평균적인 실행 시간은 각각 c_i 와 c_i^{avg} 로 표시된다.

NPM(No Power Management)은 전압 조절을 하지 않는 경우이다. 클럭 속도는 주어진 최고 속도로 실행을 하고 남아있는 여유 시간은 활용하지 않고 그 시간 동안 프로세서는 쉬게 된다. 클럭 최고 속도가 S 라면 속도의 변화가 없이 계속 S 의 속도로 실행한다 [1].

SPM(Static Power Management)은 작업이 시작되기

전에 알 수 있는 정적인 여유시간을 이용한다. 정적인 여유시간은 태스크 집합의 마감시간에서 모든 태스크의 최악의 경우의 실행 시간의 합을 뺀 것이다. 마감시간과 최악의 경우의 실행 시간에서 생기는 정적인

$$\text{여유시간을 사용하여 클럭 속도는 } S^{spm} = \frac{\sum_{i=1}^n c_i}{d}$$

로 낮추어서 실행을 하게 된다. 따라서, 최악의 경우의 실행 시간의 합이 10 이고 마감시간이 20 이라면 정적인 여유시간은 10이 되고 클럭 속도는 S 에서 $\frac{S}{2}$ 로 되어 단위시간당 실행시간은 2배로 늘어나고 전력은 $1/4$ 로 감소하게 되며 전체 소비 전력은 $1/2$ 로 감소한다[5].

DPM-P(Dynamic Power Management Proportional)는 정적인 여유시간과 태스크들이 최악의 실행 시간보다 일찍 완료됨으로서 남은 여유시간을 사용한다. 클럭 속도는 마감시간까지의 현재 남아있는 태스크들의 실행시간의 합을 이용하여 결정하게 된다.

시간 t 에서 태스크 i 가 완료되었다면 태스크 $j+1$ 을

$$\text{실행하기 위한 클럭 속도는 } S_{j+1}^P = \frac{\sum_{i=j+1}^n c_i}{d-t}$$

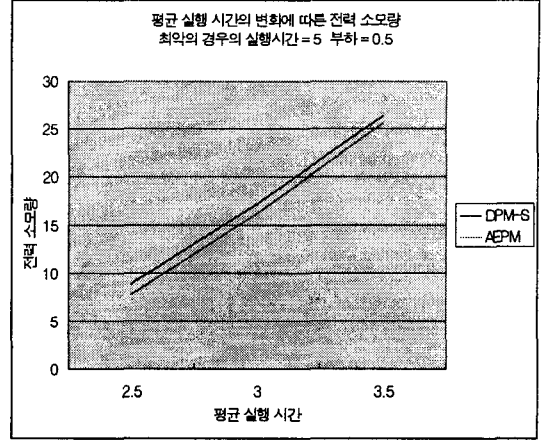
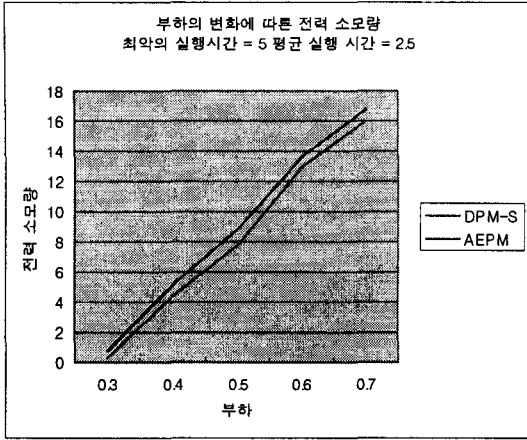
로 낮추어서 실행을 하게 된다.

DPM-G(Dynamic Power Management Greedy)는 마감시간에서 현재 남아있는 태스크들의 최악의 경우의 실행 시간을 뺀 여유시간 전부를 그 다음 태스크에서 사용하도록 한다. 시간 t 에서 태스크 i 가 완료되었다면 태스크 $j+1$ 을 실행하기 위한 클럭 속도는

$$S_{j+1}^G = \frac{c_{i+1}}{d-t - \sum_{i=j+2}^n c_i}$$

이 된다. 태스크 i 의 최악의 경우의 실행 시간이 5이고 실제 실행한 시간이 3이라면 남은 동적인 여유시간 2는 태스크 $i+1$ 에서 사용하는 것이다[1].

DPM-S(Dynamic Power Management Statistical)는 실제 실행시간의 평균 시간을 사용하여 클럭 속도를 결정하게 된다. DPM-P에서는 최악의 실행 시간을 사용하는데 반해서 여기서는 평균 실행 시간을 사용하여 클럭 속도를 낮춰준다. 실제 실행 시간이 평균 실행 시간을 초과할 경우에는 마감시간을 초과하는 경우가 발생할 수 있으므로 최소한 DPM-G로 계산한 속도 이 <그림 1> 부하의 증가에 따른 전력 소모량의 변화



상으로 실행하도록 한다. 따라서 태스크 $j+1$ 을 위

한 클럭 속도는 $S_{j+1}^s = \frac{\sum_{i=j+1}^n (c_i^{avg})}{d-t}$ 와

DPM-G의 속도 S_{j+1}^G 중 큰 값을 사용한다.

Mosse와 Aydin의 연구 결과에 의하면 DPM-S가 다른 방법들에 비해서 평균적으로 더 나은 전력 소모의 감소를 보여준다[1].

여기서 비교하는 DPM-G의 속도를 개선된 기법으로 대체하면 기존의 기법보다 좋은 결과를 얻을 수 있을 것이다.

3. 제안하는 기법

본 논문에서는 DPM-S를 개선한 AEPM(Adaptive Execution Power Management)을 제안한다. 시간 t 에서 태스크 j 가 완료되고 태스크 $j+1$ 을 실행하기 위한 클럭 속도를 계산하는데 있어서 태스크 $j+1$ 도 c_{j+1}^{avg} 에 실제 실행을 종료할 것이라는 낙관적인 예측을 사용한다. DPM-S에서는 태스크 $j+1$ 의 최악의 실행시간인 c_{j+1} 에 종료되는 것을 전제로 하고 있다.

즉, DPM-S에서는 $S_{j+1}^S = \max(\frac{\sum_{i=j+1}^n (c_i^{avg})}{d-t}, \frac{c_{j+1}}{d-t - \sum_{i=j+2}^n c_i})$ 인데 반해서 AEPM에서는

$$S_{j+1}^{AEPM} = \max(\frac{\sum_{i=j+1}^n (c_i^{avg})}{d-t}, \frac{c_{j+1}^{avg}}{d-t - \sum_{i=j+2}^n c_i})$$

을 사용한다. 따라서 DPM-S에 비해서 클럭 속도를 더 낮출 수 있게 된다. 태스크 $j+1$ 의 실제 실행 시간이 c_{j+1}^{avg} 보다 길어지게 되면 마감시간을 넘길 수 있으므로 일정한 시간 이상 태스크 $j+1$ 이 완료되지 않으면 클럭 속도를 최대로 높여준다. 즉 이 시간은 최대 클럭 속도로 남은 태스크들을 완료할 수 있는 시간으로서 다음과 같이 결정할 수 있다.

$$t_{j+1}^{lim} \leq \frac{\sum_{i=j+1}^n c_i - (t+d)}{2}$$

즉 태스크 $j+1$ 은 c_{j+1}^{avg} 시간에 완료될 것을 전제로 실행하다가 이 시간을 초과하게 되면 클럭 속도를 최대한으로 높여서 남은 태스크들을 실행한다.

4. 성능평가

성능 평가는 두 가지 기법으로 모의 실험하였다. 부하를 변화시키는 기법과 평균 실행시간을 변화시키는 방법으로 각각 모의 실험을 하였다. 먼저 태스크의 집합은 최악의 경우의 실행 시간이 5이고 실제 실행 시간은 0에서 5 사이의 균등 분포를 사용하여 30개의 태스크를 생성하였다.

부하의 변화는 태스크 집합의 마감시간과 최악의 경우의 실행시간 사이의 정적인 여유시간의 변화이다. 로드가 커질수록 정적인 여유시간이 더 많아지는 것을 의미한다.

부하의 변화는 0.3에서 0.7 사이에서 부하를 변화시키면서 전력 소모량을 계산하였다. 그림 1에서와 같이

<그림 2> 평균 실행 시간의 변화에 따른 전력 소모량

부하의 변화에 따라 전력 소모량이 증가하는 것을 볼 수 있다. 부하의 증가에 따라 AEPM도 전력 소모량이 비례해서 증가하지만 전체적으로 DPM-S보다 전력 소모량이 5 % 가량 적은 것을 보여 주고 있다.

평균 실행 시간의 변화는 태스크들의 실제 실행 시간의 변화를 나타낸다. 실제 실행 시간이 최악의 경우의 실행시간에 근접할수록 그만큼 동적인 여유 시간이 줄어들게 된다. 그림 2에서 평균 실행 시간이 증가하면서 점점 전력 소모량이 증가하는 것을 볼 수 있다. AEPM도 평균 실행 시간이 증가함에 따라 전력 소모량이 증가하지만 전체적으로 DPM-S보다 5 % 가량 적은 것을 보여 주고 있다.

5. 결론

최근 내장형 시스템들의 소형화와 이동성이 확대되고 있는 추세에 따라 전력을 줄이는 문제의 중요성이 커지고 있다. 본 논문에서는 저전력 구현 기법 중에 태스크들의 작업량에 따라 프로세서의 공급 전압을 조절하여 전력 소모량을 줄이는 기법으로서 기존의 DPM-S를 개선한 동적 전압 조절 기법을 제시하였다. 모의 실험을 통하여 성능을 평가를 해본 결과 약 5 % 정도 성능이 개선된 것을 확인하였다.

6. 참고문헌

- [1] Daniel Mosse, Hakan Aydin. "Compiler-Assisted Dynamic Power-Aware Scheduling for Real-Time Application" COLP, Oct 2000.
- [2] Y.H.Lee, C.M.Krishna. "Voltage-Clock Scaling for Low Energy Consumption in Real-time Embedded Systems" RTSS '2000.
- [3] 김운석. "가변 전압 프로세서를 위한 저전력 소프트웨어 설계 기법" 정보과학회 2002.
- [4] Dakai Zhu, Rami Melhem. "Scheduling with Dynamci Voltage/Speed Adjustment Using Slack Reclamation in Multi-Processor Real-Time System" RARTS 2001.
- [5] Padmanabhan Pillai, Kang G. Shin. "Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systmes" SOSP, Oct 2001.