

# 혼합 예측에 기반한 프로세서의 동적 전압 변경 기법

최진욱<sup>○</sup> 최석원 차호정  
연세대학교 컴퓨터과학과  
[jwchoi, sukwon, hjcha}@cs.yonsei.ac.kr

## A DVS Technique based on Hybrid Prediction

Jinuk Choi<sup>○</sup> Sukwon Choi Hojung Cha  
Dept. of Computer Science, Yonsei University

### 요 약

본 논문은 내장형 시스템의 전력 감소를 위해 사용되는 과거 사용량 기반의 DVS의 단점인 응용 프로그램의 수행 성능 저하를 보상하기 위해, 운영체제의 스케줄러에서 제공하는 태스크의 미래정보를 이용하는 기법을 제안한다. 대표적 내장형 운영체제인 WinCE.net에서의 스케줄러는 제한된 자원의 효율적 관리를 위하여 동일 응용프로그램의 태스크들을 관리하면서 다음 태스크 시행시간 정보를 갖게 된다. 이러한 룩 어헤드(look ahead)정보와 과거사용량기법을 혼합한 혼합예측기법이 실제 내장형 시스템에서 전력소비를 감소시키며 응용프로그램의 수행 성능보상을 할 수 있음을 보인다.

## 1. 서론

PDA와 휴대폰과 같은 이동기기에 있어서 성능을 좌우하는 프로세서의 성능은 매년 급속도로 향상되고 있다. 따라서 프로세서 개발 업계에서는 저 전력 고성능 프로세서의 개발의 중요성을 인식하고 개발을 위한 노력을 기울이고 있다. 프로세서는 그 전기적 특성으로 인하여 성능이 높아질수록 전력소비가 높아진다. 하지만 수행하는 작업에 따라 프로세서의 사용량은 가변이다. 즉 항상 최대의 사용량이 요구되지는 않는다. 따라서 프로세서의 사용량에 따라서 프로세서에 공급되는 동작주파수와 전압을 조정함으로써 전력소비를 줄일 수 있다. 이러한 방법을 DVS(Dynamic Voltage Scaling)이라고 한다[4].

DVS기법중 대표적인 구간예측기법은 현재 태스크의 작업량은 대부분 바로 전 태스크의 작업량과 비슷한 경우일 수 있다는 경향을 가지며 구현이 용이해 DVS의 근간을 이루고 있다. Weiser[1]의 경우는 가장 최근의 프로세서 사용량을 그 이전의 프로세서 사용량 보다 가중치를 두어 DVS를 하여 주는 것이 효과적인 전력 소비 절감을 가져 올 수 있음을 보여 주었다. Govil[3]의 논문에서는 과거에 사용된 프로세서의 사용량의 평균을 구해 다음 단계에 사용될 프로세서의 사용량을 예측하고, 그 예측 량에 맞추어 프로세서의 공급전압을 정해주는 방법을 사용하여 전력 소비를 줄일 수 있음을 시뮬레이션을 통해 보여 주었다. Flautner[4]는 수행되는 태스크를 주기적인 것과 비주기적인 것으로 분류하고, 분류된 특성에 맞추어 성능을 예측하여 프로세서의 공급전압을 결정하는 방법을 채택하였다.

Weiser 가 사용한 과거 프로세서 사용량에 기반 기법들은 전력 소비를 감소시킬 수가 있으나 미래 부하에 대한 불확실성으로 예

측이 틀린 부하에 대하여 성능저하가 발생된다. 예를 들면, 바로 전 태스크의 적은 부하량을 기준으로 다음 프로세서의 성능을 낮게 책정하였으나 다음 태스크의 작업량이 실제로 많이 발생한 경우 성능 저하가 발생된다. 또한 Flautner의 접근 방식은 태스크의 수행정보를 미리 파악 하여 성능보상을 할 수가 있으나 커널의 수정과 지속적인 태스크 모니터링이 요구 된다. 또한 특정 시나리오에 기반한 응용프로그램에서만 적용 가능한 형태로 제시된다. 본 논문에서는 대표적인 내장형 운영체제 중 하나인 WinCE.Net의 스케줄러에서 미래 태스크와 관련된 정보를 이용하여 과거 정보로 만의 DVS에서 발생하는 성능저하를 보상하는 혼합예측기법을 구현한다. 논문의 구성은 다음과 같다. 2 장에서는 논문에서 제안하는 혼합예측방식에 기반을 둔 프로세서의 동적전압변경기법에 대해 기술한다. 3 장에서는 실험 방법 및 결과에 대해 기술하고 4 장에서 결론을 맺는다.

## 2. 혼합예측기법

DVS에 있어서 구간 기반의 예측기법은 시뮬레이션에 기반한 고전적 접근 Weiser[1] 으로부터 실제 시스템에서의 적용 실험이 되고 있다[5]. 하지만, 이 방법은 현재 작업량이 바로 다음 작업량과 유사하다는 태스크 패턴에서는 유용한 방법이나 특정 멀티미디어 응용프로그램 같이 작업량이 불규칙 한 경우에는 성능저하가 발생할 수 있다. 예를 들면 MPEG4 player같은 경우에 영상의 복잡도나, 디코딩 하는 프레임의 속성 등에 따라 과거 사용량만 가지고는 예상하기 힘든 태스크 부하들이 발생한다. 이러한 상황에서는 단순히 과거 태스크의 작업량만을 가지고 성능책정을 하기 보다는 다음 작업량을 좀더 정확히 예측할 수 있는 방법이 필요하다. 이러한 미래 태스크에 대한 정보를 보완하기 위하여 응용프로그램이 수행정보를 제공하는 경우 Lu[2]와 운영체제 상에서 이를 추정하는 방법 Flautner[4]등이 사용되고 있다. 하지만 전자의 방법은 모

<sup>○</sup> 본 연구는 정보통신연구진흥원에서 지원하는 정보통신기초기술연구지원사업으로 수행하였음 (과제번호 : C1-2003-A1-2000-0240)

든 수행하는 응용프로그램을 수정해야 된다는 부담을 갖고 후자의 방법은 오버헤드를 동반하기 때문에 소형이동기기의 제한된 자원 상에서는 또 다른 성능저하를 유발시킬 수 있다. 본 논문에서는 미래 태스크의 정보를 대표적인 내장형 운영체제중 하나인 WinCE.net에서의 스케줄러의 특성을 이용하여 추출한다. 이 방법은 응용 프로그램의 수정을 요구하지 않으며 커널의 수정으로 생기는 오버헤드가 발생하지 않는다는 장점이 있다. 최종적으로 이 정보를 과거프로세서 사용량기법과 혼합하여 프로세서 정책을 수행하는 혼합예측기법을 기술한다.

**2.1. 과거 프로세서 사용량을 이용한 예측방법**

프로세서 사용량에 기반을 둔 예측기법은 실행되는 응용프로그램의 종류에 상관없이 DVS를 수행할 수 있다. 이 기법은 최근 프로세서 사용량이 다음 프로세서 사용량과 가장 근접하다는 가정 하에 사용하는 방법이다. 본 논문에서 사용된 과거프로세서 사용량을 이용한 예측 방법에서는 다음 식(1)을 이용하여 다음 프로세서 사용량을 추정한다. 여기서  $U_{n+1}$  은 추정하는 사용량을 나타내고  $f_{n+1}$  은 책정하려고하는 프로세서의 성능이다.

$$f_{n+1} = \begin{cases} f_{normal}, & \text{if } U_{n+1} > U_{upper} \\ f_{low-power}, & \text{if } U_{n+1} < U_{lower} \end{cases} \quad (1)$$

where,  $U_{n+1} = aP_n + (1-a)U_n, \quad 0 < a < 1$

즉, 다음 프로세서의 성능 책정  $f_{n+1}$ 은 예측된 사용량  $U_{n+1}$ 값이 상한임계치  $U_{upper}$ 보다 크게 되면 프로세서의 성능 책정을 높은 수행능력  $f_{normal}$ 로 하게 되고  $U_{n+1}$ 값이 하한임계치  $U_{lower}$ 보다 작게 예측되면  $f_{n+1}$ 은 저전력모드  $f_{low-power}$ 로 책정되게 된다.  $a$ 는 현재 성능책정값을 다음 성능책정값에 반영정도를 나타내는 이득 값이다. 식 (1)에서 임계치  $U_{upper}$ 와  $U_{lower}$ 는 다음과 같은 방법으로 정해진다.

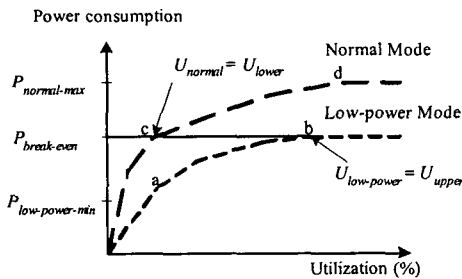


그림 1. 프로세서 사용량에 따른 임계값

그림 1은 프로세서 사용량에 따른 전력 소비량 추이를 저전력모드와 정상모드에 따라 표현한 것이다. X축은 프로세서의 사용량을 나타내고 Y축은 프로세서 코어에서 사용되는 전력소비를 나타낸다. 현재 프로세서가 저전력모드일때 프로세서의 사용량이 증가하게 되면 전력의 소비는 a지점에서 b 지점 쪽으로 이동한다. 저전력모드일때 프로세서의 사용량이  $U_{upper}$ 가 되면 b지점에 도달하게 되고, 전력 소비량은 저전력모드에서 최대 소비량인  $P_{break-even}$ 에 도달하게 된다. 이 상태에서의 프로세서는 저전력모드에서 제공 할 수

있는 최대의 프로세서 사용량을 초과 하게 되고, 부가적인 프로세서의 사용량은 태스크의 QoS에 영향을 미치게 된다. 즉 이지점에서는 부하가 증가하여도 사용량은 증가하지 않는다. 따라서 좀더 작업 처리량을 높일 수 있는 정상 모드로 바꾸어 준다. 그렇게 되면 c지점으로 이동 하게 되고, 프로세서 사용량이 증감에 따라서 c지점과 d지점 사이에서 작업량 및 전력 소모가 결정 될 것이다. 현재 전력 모드가 정상모드일 경우는 반대로 동작한다. 현재 d지점에 있던 작업량이 c지점(프로세서 사용량이  $U_{lower}$  되는 지점)까지 떨어지면 저전력모드의 최고 전력 소모량인  $P_{break-even}$ 에 도달하게 되고, 이때 저전력모드에서도 무리 없이 태스크를 수행 할 수 있게 된다. 따라서 프로세서의 전력모드를 저전력모드로 바꾸어주면 b지점에서 a 지점 사이에서 작업량 및 전력 모드가 결정되어 질 것이다. 본 논문에서는 이러한 방법으로 임계값( $U_{upper}$ 와  $U_{lower}$ )을 결정한다.

과거 프로세서 사용량에 기반을 둔 프로세서 성능 결정방법은 응용프로그램의 종류에 상관없이 사용할 수 있는 장점이 있으나 멀티미디어 응용프로그램 같은 갑작스러운 부하(bursty workload)에 대하여는 능동적으로 대처할 수 없다는 단점이 있다. 이를 보완하기 위하여 미래 태스크의 정보를 대표적인 내장형 운영체제인 WinCE.Net의 스케줄러로부터 추출하는 방법을 사용한다.

**2.2. 스케줄러 룩 어헤드 정보**

WindowsCE.Net의 스케줄러는 모든 태스크가 쓰레드 단위로 처리된다. 기본적으로 높은 우선 순위를 가진 쓰레드를 먼저 처리하는 선점형 방식이고, 같은 우선 순위를 가진 쓰레드에서는 라운드 로빈 방식으로 처리를 한다. 쓰레드는 현재 실행 상태에 따라서 런큐와 슬립큐 2가지 형태의 큐에서 존재한다. 또한, 현재 수행중인 쓰레드보다 우선순위가 낮을 경우 슬립큐에 존재한다. 이 쓰레드가 우선순위가 높아지면 런큐로 들어가게 된다. 런큐에는 현재 수행중인 런쓰레드와 우선 순위 및 기타 조건을 만족하여 수행중인 쓰레드가 끝날경우 바로 실행 상태로 될 수 있는 런어블 쓰레드가 존재한다. 쓰레드가 런어블 쓰레드가 되면 쓰레드의 타임 쿼텀 및 스케줄러의 타임 변수 및 우선순위에 따라서 웨이크업 타임을 할당받게 된다. 이 시간에 맞추어 러너블 쓰레드는 현재 수행중인 쓰레드와 순위청 하게 된다.

만약 수행되는 태스크가 우선순위가 높고 수행량이 많다면, 런어블 쓰레드가 많이 존재하고, 리스케줄 시간도 짧아 지게 될 것이다. 하지만 수행되는 태스크가 적다면 다음 스케줄 시간은 쓰레드의 타임 쿼텀내에 스케줄 시간이 할당되지 않을 것이다. 실제 시스템에서 측정된 결과 MPEG player와 같은 멀티미디어 응용 프로그램 수행 상에서는 다음 스케줄시간이 상대적으로 짧게 나타나는 것을 알 수 있었다. 하지만 I/O 블로킹 등에 의한 우선 순위 역전 등으로 실제 스케줄 시간과 정확히 일치 하지 않기 때문에 과거의 프로세서 사용량 측정과 함께 사용되어 단점이 보완 되어야 한다. 본 논문에서 제시된 혼합예측 기법에서는 이 값을 과거사용량방법이 예측하지 못한 부하를 예측하여 프로세서 성능책정에 반영하는 방법에 사용하게 된다.

### 2.3. 혼합 예측 기법 알고리즘

과거 프로세서 사용량을 이용한 예측 방법 식(1)을 보완하고자 스케줄러를 통해 얻어진 다음 태스크의 리스케줄 타임( $T_{reschedule}$ )을 이용한 혼합예측 방법은 다음과 같다.

$$f_{n+1} = f_{normal}, \text{ if } U_{n+1} > U_{upper} \\ \text{ or } T_{reschedule} < THD_{LookAhead} \\ = f_{low-power}, \text{ if } U_{n+1} < U_{lower} \\ \text{ and } T_{reschedule} > THD_{LookAhead}$$

where,  $U_{n+1} = aP_n + (1-a)U_n, 0 < a < 1$

과거의 프로세서 사용량  $U_n$ 을 통해  $U_{n+1}$ 을 계산한 후 이를 이용하여 다음 프로세서의 성능  $f_{n+1}$ 을 설정하고, 다음 태스크의 스케줄 시간  $T_{reschedule}$ 을 이용하여 최종 프로세서의 상태를 책정한다. 임계값  $THD_{LookAhead}$  값은 실제 멀티미디어 응용 프로그램의 수행상태를 모니터링을 통하여 구한다.

즉, 과거 사용량이 일정 임계값  $U_{upper}$  이상 되는 경우에 정상상태로 성능 책정하는 것을 기본으로 하지만 과거 사용량이 일정 임계값 이하라 하더라도 리스케줄 시간이 록 어헤드 임계값보다 작을 경우(다음 태스크의 순위가 높아 스케줄러에 의하여 짧은 리스케줄 시간이 책정된)에는 정상 상태로 가게 된다. 또한 프로세서 과거 사용량이 적었던 경우  $U_{n+1} < U_{lower}$  에도 다음 태스크의 수행정도가 일정시간 이상 되어야지만 프로세서 정책을 저 전력 모드로 설정하게 된다. 이러한 메커니즘을 통하여 응용프로그램의 성능 보상을 하게 된다.

### 3. 실험

혼합 예측 기법의 성능을 실험하기 위한 환경은 WinCE.Net 4.1 과 Intel Xscale DBPXA250 개발보드로 구성 되었다.  $f_{normal}$  은 공급 전압이 1.65V, core clock 400Mhz에 해당하고,  $f_{low-power}$  는 공급 전압이 1.3V core clock 200Mhz 에 해당한다. 또한 Fluke사의 Scope meter를 통하여 프로세서에 공급되는 전력을 측정하였다. 부하는 MP3 (128kbps)와 66kbps (20frames encoded)/300kbps (20frames encoded)/1Mbps (25frames encoded)의 MPEG4 clip을 재생하였다. Bit rate에 따른 MPEG4 부하에 따른 전력소비량은 표 1 과 같고 혼합예측정책을 사용 했을 때의 전력 감소 비율을 마지막 열에서 보여준다. 표 2에는 부하에 따 재생 프레임 수를 나타내고 정책 열에는 혼합예측기법과 과거 프로세서 사용량 기법, 정책을 사용하지 않은 경우의 결과를 비교하였다. 표 2 의 마지막 열은 정책을 사용함으로써 발생하는 성능 저하를 정책을 사용하지 않았을 때와의 비율로 나타낸다.

표 1 에서 보는바와 같이 부하정도에 따라 최고 25.9%에서 15.6% 까지 전력을 절감할 수 있었다. 직관적으로 MPEG4 인 경우보다는 부하량이 적을 수 있는 MP3 디코딩의 경우 더 많은 전력 절감을 보여주고 있다. QoS측면에서 1Mbps MPEG4 부하인 경우에도 혼합예측기법사용으로 3%이내의 손실만 보여주고 있다. 표 2 는 혼합예측방식이 단순 과거사용량 기법보다 명확하게 성능을 개선함을 보여준다. 이러한 효과는 부하량이 증가함에 따라 두드러진다. 이 현상은 부하량이 증가하면 단위 시간 내에 처리해야하는 데드라인을

가진 태스크들이 증가하고 이것들을 처리하는데 있어 혼합예측이 효과적으로 작용한다는 것을 볼 수 있다. 특별히, 1Mbps clip에서는 과거 프로세서 사용량 기법만 사용 시에서는 성능이 47.2% 감소하고 반면에 혼합 예측 기법에서는 2.3%만 성능감소가 일어남을 볼 수 있다.

| Power<br>Workload | Policies                 |            | Reduction<br>Ratio |
|-------------------|--------------------------|------------|--------------------|
|                   | Hybrid DVS<br>Power@1.47 | No DVS@1.6 |                    |
| MP3 @128kb        | 215 mW                   | 290 mW     | 25.9 %             |
| MPEG4 @66kb       | 231 mW                   | 277 mW     | 16.6 %             |
| MPEG4 @300kb      | 336 mW                   | 402 mW     | 16.4 %             |
| MPEG4 @1Mb        | 342 mW                   | 405 mW     | 15.6 %             |

표 1. 전력 소비량

| Frames/Sec<br>Workload | Policies |              |           | Reduction<br>penalty ratio<br>Hybrid/PAST |
|------------------------|----------|--------------|-----------|---|
|                        | Hybrid   | PAST<br>only | No<br>DVS |   |
| MPEG4 @66kb            | 19.76    | 19.31        | 19.83     | 0.35%/2.62%                               |
| MPEG4 @300kb           | 18.82    | 15.01        | 18.96     | 0.74%/20.8%                               |
| MPEG4 @1Mb             | 21.75    | 11.74        | 22.27     | 2.33%/47.2%                               |

표 2. 재생 프레임수

### 4. 결론

본 논문에서는 기존의 사용량기반의 DVS의 응용프로그램 성능 보안을 위하여 대표적인 내장형 운영체제인 WinCE.net 스케줄러의 미래 태스크 정보를 이용한 혼합예측 기법을 제시하고 실제 상용 내장형 시스템에서의 실험을 통하여 소비전력을 측정 하여 보았다. 실험결과에서 볼 수 있듯이 3% 이내의 성능저하 내에서 최대 26%의 전력 소비 감소를 시킬 수 있었다. 또한 부하의 로드가 증가 하면서는 더욱 제시한 방법이 성능 보상을 할 수 있음을 보여주었다. 내장형 운영체제는 제한된 자원을 효율적으로 사용하기 위한 구조를 가지고 있으며, 제시한 방법은 저전력 설계가 중요한 소형 이동기기에서 대표적인 WinCE.net의 스케줄러의 특성을 이용했기 때문에 다른 방법들이 접근했던 응용프로그램의 수정이나 커널 수정등의 많은 오버헤드를 갖지 않는다. 향후 과제로는 Linux 및 ECOS등 다른 내장형 운영체제에서, 제시된 혼합예측 기법을 사용하여 더욱 표준화된 방법을 제시하는 것이다.

### 참고자료

- [1] Weiser, M, et al., "Scheduling for reduced CPU energy", *Proc. of the first USENIX Symposium on OSDI*, pp. 13-23, November 1994.
- [2] Y. Lu, L. Benini, G. Micheli, "Power-Aware Operating Systems for Interactive Systems", *IEEE Trans. On VLSI Systems*, Vol.10, No.2, April 2002, pp.119-2002.
- [3] K. Govil, E. Chan, and H. Wasserman, "Comparing algorithms for dynamic speed-setting of a low-power CPU", *Proc. ACM Int'l Conf. on Mobile Computing and Networking*, pp. 13-25, Nov. 1995.
- [4] K. Flautner, S. Reinhardt, and T. Mudge, "Automatic performance Setting for Dynamic Voltage Scaling", *MobiCom'01, July 2001*.
- [5] Dirk Grunwald, et al., "Policies for Dynamic Clock Scheduling", *Proc. of the 4th USENIX Symposium on OSDI'00*, Oct. 2000.