

성능 모니터링 이벤트들의 통계적 분석에 기반한 모바일 프로세서의 전력 예측

(Power Prediction of Mobile Processors based on Statistical
Analysis of Performance Monitoring Events)

윤 희 성 [†] 이 상 정 ^{**}
(Hee-Sung Yun) (Sang-Jeong Lee)

요 약 제한된 용량의 배터리로 동작해야 하는 모바일 시스템에서는 소프트웨어 설계 시 성능뿐만 아니라 전력소모도 고려해야 한다. 따라서 소프트웨어의 실행 중에 전력소모를 정확하게 예측할 수 있으면 전력과 성능을 고려한 효율적인 소프트웨어의 설계가 가능해진다.

본 논문에서는 모바일 프로세서의 전력소모 예측을 위해 정량적으로 프로세서의 동작을 분석하고 모델링 하는 통계적인 분석 방법을 제안한다. 제안된 방식은 다양한 벤치마크 프로그램들을 실행하여 프로세서의 성능 모니터링 이벤트들과 전력소모 데이터를 수집한 후 계층적 클러스터링(hierarchical clustering) 분석 등을 적용하여 서로 중복되지 않으면서 전력소모에 크게 기여하는 대표적인 성능 모니터링 이벤트를 추출한다. 전력 예측 모델은 선택된 성능 모니터링 이벤트들이 독립변수가 되고 전력소모가 종속변수가 되는 회귀분석(regression analysis)을 수행하여 개발한다. 전력 예측 모델은 Intel XScale 아키텍처 기반의 PXA320 모바일 프로세서에 적용하여 평균 4% 이내의 에러율로 전력소모를 예측할 수 있음을 보인다.

키워드 : 전력 예측, 모바일 프로세서, 성능 모니터링 이벤트, 회귀분석, 계층적 클러스터링

Abstract In mobile systems, energy efficiency is critical to extend battery life. Therefore, power consumption should be taken into account to develop software in addition to performance. Efficient software design in power and performance is possible if accurate power prediction is accomplished during the execution of software.

In this paper, power estimation model is developed using statistical analysis. The proposed model analyzes processor behavior quantitatively using the data of performance monitoring events and power consumption collected by executing various benchmark programs. And then representative hardware events on power consumption are selected using hierarchical clustering. The power prediction model is established by regression analysis in which the selected events are independent variables and power is a response variable. The proposed model is applied to a PXA320 mobile processor based on Intel XScale architecture and shows average estimation error within 4% of the actual measured power consumption of the processor.

Key words : power prediction, mobile processor, performance monitoring events, regression analysis, hierarchical clustering

· 이 논문은 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원
을 받아 수행된 연구임 (KRF-2007-지역대학우수과학자지원사업-D00394)

[†] 학생회원 : 순천향대학교 컴퓨터학부
iloveyr2@hanmail.net

^{**} 정 회 원 : 순천향대학교 컴퓨터학부 교수
sjlee@sch.ac.kr

논문접수 : 2008년 12월 11일
심사완료 : 2009년 5월 11일

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제7호(2009.7)

1. 서론

최근 반도체 집적도의 증가, 고밀도 시스템 패키징 및 쿨링 비용의 증가로 인해 저전력을 소모하는 컴퓨팅 시스템의 설계가 주요 이슈 및 해결과제로 대두되고 있다. 특히 제한된 용량의 배터리로 동작해야하는 모바일 시스템에서 저전력을 소모하는 하드웨어와 소프트웨어 시스템 설계는 필수적이다. 프로세서의 전력소모는 실행 중인 프로그램의 워크로드(workload)에 크게 영향을 받기 때문에 모바일 디바이스에서 실행되는 소프트웨어 설계 시에는 성능뿐만 아니라 전력소모도 고려해야 한다. 따라서 소프트웨어의 실행 중에 전력소모를 정확하게 예측할 수 있으면 전력과 성능을 고려한 효율적인 소프트웨어의 설계가 가능해진다. 최근의 프로세서는 프로세서의 동작 중에 실행되는 성능 모니터링 이벤트들(클럭, 명령어, 캐시미스, 분기미스 등)을 저장하는 성능 카운터를 내장하고 있다[1]. 또한 클럭 주파수 및 공급 전압을 조정하여 전력소모를 줄이는 DVFS(Dynamic Voltage and Frequency Scaling) 기능이 탑재되고 있다. 따라서, 프로그램 실행 중에 성능 모니터링 이벤트들을 수집하여 각 클럭 주파수 변화에 따른 전력소모와 성능을 예측하고, 예측된 값을 사용하여 각 클럭 주파수에 대한 성능대비 전력소모 값 등을 비교하여 이 값이 가장 우수한 클럭 주파수로 변경하는 전력관리를 실시함으로써 성능감소는 최소화하면서 전력소모를 줄일 수 있다.

프로세서의 성능과 전력소모 등을 정확하게 예측하기 위해서는 프로세서의 동작을 이해하고 분석하여 모델링해야 한다. 그러나 최근의 프로세서들은 깊은 파이프라인 처리, 멀티-레벨 캐시 구조, 복잡한 분기 예측기 구조 등을 가지고, 단일 칩 상에서 여러 리소스 등을 공유하면서 다중 쓰레드를 실행하도록 설계되고 있어서 프로세서의 동작을 정성적 또는 정량적으로 분석하여 수학적 모델을 구축하는데 점점 많은 시간과 비용이 소모되고 있다[2-6].

본 논문에서는 모바일 프로세서의 전력소모 예측을 위해 정량적으로 프로세서의 동작을 분석하고 모델링하는 통계적인 분석 방법을 제안한다. 먼저, 다양한 벤치마크 프로그램들을 실행하여 프로세서 관련 다수의 성능 모니터링 이벤트들과 전력소모 데이터를 수집한다. 프로세서가 동시에 측정할 수 있는 성능 모니터링 이벤트들의 수는 한정되어 있기 때문에 계층적 클러스터링(hierarchical clustering) 분석 등을 적용하여 서로 중복되지 않으면서 전력소모에 크게 기여하는 대표적인 성능 모니터링 이벤트들을 추출한다. 전력 예측 모델은 선택된 성능 모니터링 이벤트들이 독립변수가 되고 전

력소모가 종속변수가 되는 회귀분석(regression analysis)을 수행하여 개발한다. 제안된 통계적 분석의 전력 예측 모델은 Intel Xscale 마이크로아키텍처[7] 기반의 Marvell PXA320 모바일 프로세서[8]에 적용하여 타당성을 검증한다. MiBench 벤치마크 프로그램[9,10]과 SPEC 벤치마크 프로그램[11] 등을 실행하여 학습하고 적용한 후 실제 측정된 전력 값과 비교한 결과 평균 4% 이내의 에러율을 갖고 정확하게 프로세서의 전력소모를 예측할 수 있음을 보인다.

2. 관련 연구

전력 예측에 관한 기존의 논문들은 대부분 프로세서 동작을 정성적 또는 정량적으로 분석한 수학적 모델을 기반으로 구축되었다[2-4]. Contreras[2] 등은 PXA255 프로세서 상에서 성능 모니터링 이벤트들과 전력소모 간의 선형적 관계를 가정하고 OLS(Ordinary Least Square) 등 선형대수를 적용한 전력소모 예측 모델을 제안하였다. Rajamani[3] 등은 Pentium-M 프로세서에서 디코딩된 명령어 수를 나타내는 성능 모니터링 이벤트만을 고려하여 전력을 예측하는 단순 선형 방정식 모델을 정의하였다. 이 모델에 단순 배열을 참조하는 마이크로 벤치마크들을 적용하고 학습하여 선형 방정식의 계수를 구함으로써 전력소모를 예측하였다. Wu[4] 등은 Pentium-M 프로세서에서 메모리 액세스 정도에 따라 DVFS적용 후 성능 손실 여부를 정성적으로 분석하여 메모리 버스 및 실행 완료된 명령어 수를 나타내는 성능 모니터링 이벤트를 적용한 수학적 모델을 구축하였다. 이들 수학적 모델의 전력 예측 방식들은 프로세서의 구조가 복잡해짐에 따라 점점 많은 시간과 비용이 요구되고, 타겟 프로세서가 변경될 경우 수리적인 예측 모델의 구축과정이 변경되고 재구성된다는 문제점이 있다[12].

Sasaki[12] 등은 Pentium-M과 AMD Opteron 프로세서 상에서 통계적인 분석에 기반하여 DVFS를 적용하는 기법을 제안하였다. 즉, 성능과 성능 모니터링 이벤트들 간의 관계를 회귀분석 모델을 적용하여 미리 통계적으로 학습한 후 성능을 예측하는 모델을 개발하고, 컴파일러에 이 예측 모델과 주파수-전압을 세팅하는 런-타임 코드를 삽입하였다. 이 방식은 정량적으로 프로세서의 동작을 분석하는 통계적인 방법을 제안하였지만, 컴파일러가 프로그램의 동작을 분석하고 라이브러리에 코드를 삽입하는 정적 최적화 방식이라는 한계점과 전력소모 예측이 아닌 성능 예측을 위해 선형 회귀분석 모델을 적용하였지만 본 논문은 프로그램 실행 중에 동적으로 성능 모니터링 이벤트를 측정하고 전력소모를 예측하였다는 점에서 차이가 있다. S.LEE[13] 등은 Pentium-M 프로세서에서 실행된 클럭 수와 메모리 사

용의 성능 모니터링 이벤트들을 사용하여 성능을 예측하는 회귀분석 모델을 제안하였고, 이를 이용하여 명시된 성능 손실 범위 안에 DVFS를 적용하여 성능 손실은 최소로 하면서 전력소모를 줄이는 방법을 연구하였다. 그러나 이 연구는 2개의 성능 모니터링 이벤트를 사용하여 성능을 예측하였고, 본 논문과 같이 다수의 성능 모니터링 이벤트들 중에서 회귀분석에 적용될 이벤트들을 추출하는 과정 등을 제시하지는 못하였다.

B.Lee[14,15] 등은 마이크로아키텍처 설계의 최적화와 평가를 위해 시뮬레이션을 수행 할 때 설계 영역 및 파라미터의 증대로 인해 지수 함수적으로 증가하는 시뮬레이션 비용을 줄이기 위해 통계적으로 샘플링된 시뮬레이션 데이터 상에서 타겟 프로세서의 성능과 전력소모를 추정하는 모델을 제안하였다. 즉, 샘플링 데이터 상에서 계층적 클러스터링, 연관분석, 상관분석 등을 적용하여 아키텍처 설계 파라미터들을 추출하고, 이들 파라미터에 회귀분석 모델을 적용하여 성능 및 전력소모를 추정하였다. 이들 연구는 마이크로아키텍처 설계 초기 단계에서 시뮬레이션 샘플링 데이터를 기반으로 설계된 마이크로아키텍처의 평가와 최적화를 위하여 제안된 모델이다. 그러나 본 연구는 실제 프로세서 상에서 프로그램 실행 중에 동적으로 파라미터를 측정하여 회귀분석에 적용하였고, 동적 전력관리를 위해 전력소모를 예측하는 모델이라는 점에서 근본적인 차이가 있다.

Mathur[16] 등은 동적으로 성능 모니터링 이벤트를 변경하는 멀티플렉싱 기법을 제안하였다. 프로세서 상에서 샘플링 타임을 분할하여 제한된 성능 모니터링 카운터의 동시측정 개수의 한계를 극복하는 방법을 제안하였다. 멀티플렉싱 기법은 샘플링되는 인터벌을 분할하여 카운터를 세트, 리셋하는 인터럽트를 발생시켜야 하는 오버헤드가 추가되고, 비교적 높은 에러율을 보여 모바일 프로세서에 적용하기 어렵다.

3. PXA320 프로세서의 성능 모니터링 이벤트

본 논문의 전력 예측 모델은 스마트폰, PMP, 네비게이터 용의 모바일 단말기를 위해 개발된 Marvell사의 PXA320 프로세서[7,8]에 적용하여 생성하였다. PXA320 프로세서(모델명 Monahans)는 ARM-11 코어와 Intel XScale 아키텍처를 기반으로 동적 전압/주파수 스케일링(DVFS)을 지원한다. 최대 클럭은 806MHz로

동작하며 최소 104MHz까지 7개의 DVFS 포인트를 갖는다. PXA320 프로세서는 36개의 성능 모니터링 이벤트를 지정할 수 있고, 4개의 32비트 성능 카운터를 제공한다. 따라서 프로그램 실행 중에 동시에 측정할 수 있는 성능 모니터링 이벤트는 4개로 제한된다. 36개의 성능 모니터링 이벤트들 중에서 표 1과 같이 중복된 기능의 성능 모니터링 이벤트들은 배제하고, 모바일 프로세서에서 수행되는 응용 프로그램 등을 반영하여 프로세서의 성능과 전력소모에 가장 영향을 미치는 10개의 성능 모니터링 이벤트들을 초기의 측정 성능 모니터링 이벤트들로 선정하였다. 표 1에서 처음 5개 이벤트는 Contreras[2] 등이 PXA255 프로세서의 전력 예측 모델에 사용된 성능 모니터링 이벤트와 일치한다. Contreras는 전력소모와 각 성능 모니터링 이벤트들 간의 상관관계를 기반으로 이들 성능 모니터링 이벤트를 선정하였다. 본 연구에서는 이들 파라미터에 중복된 기능의 파라미터를 배제한 3개의 파라미터(L\$_{2}\$_MISS, BR\$_{2}\$_MISS, BUS)와 모바일 프로세서의 응용 프로그램 중 멀티미디어 응용을 반영한 2개의 파라미터(MULT, CO\$_{2}\$_STALL)를 더 추가하여 초기 측정 성능 모니터링 이벤트로 선정하였다.

표 1 초기 측정 성능 모니터링 이벤트들

| 이벤트 | 내용 |
|------------------|--------------------------------|
| INST | 실행 완료된(retired instruction) 명령 |
| STALL | 데이터 종속관계에 기인한 실행 중지 |
| L_TLB_MISS | 명령 TLB 미스 |
| D_TLB_MISS | 데이터 TLB 미스 |
| I\$_{2}\$_MISS | 명령어 캐시 미스 |
| L2\$_{2}\$_MISS | 통합 L2 캐시 미스 |
| BR\$_{2}\$_MISS | 분기 예측 미스 |
| MULT | 곱셈기 사용 |
| CO\$_{2}\$_STALL | 보조 프로세서 실행 중지 |
| BUS | 데이터 버스 트랜잭션 |

4. 통계적 분석 기반 전력 예측 모델 생성

본 논문의 전력 예측 모델은 그림 1과 같이 통계 분석을 위한 데이터를 준비하고, 이 데이터들에 계층적 클러스터링 분석을 실시하여 회귀분석에 적용할 성능 모니터링 이벤트들을 선택한 후 회귀분석을 수행하여 생성된다.

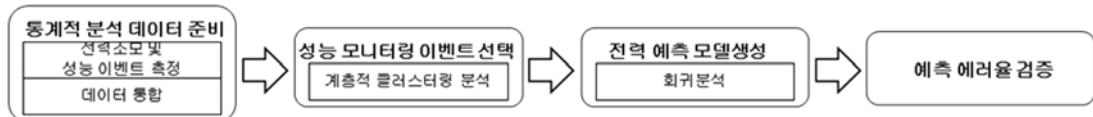


그림 1 통계적 분석 기반 전력 예측 모델 생성 과정

4.1 통계적 분석 데이터 준비

통계 분석을 위한 데이터는 다양한 벤치마크 프로그램들을 실행하여 10ms 단위로 샘플링하여 초기 성능 모니터링 이벤트들과 전력소모를 측정하였다(벤치마크 프로그램과 전력측정 환경은 5.1절 참조). PXA320 프로세서는 동시에 4개의 성능 모니터링 이벤트만을 측정할 수 있기 때문에 10개의 초기 성능 모니터링 이벤트들을 아래와 같이 3개의 그룹으로 나누었다.

{INST, STALL, L2\$_MISS, BR\$_MISS},

{INST, I\$_MISS, I_TLB\$_MISS, D_TLB\$_MISS},

{INST, MULT, CO_STALL, BUS}

통합적인 통계분석을 위해서는 3개의 그룹으로 각기 독립적으로 실행되어 측정된 성능 모니터링 이벤트들과 전력소모를 하나의 데이터로 결합해야 한다. 3개의 독립된 그룹의 데이터 통합을 위해 각 그룹별로 공통 성능 모니터링 이벤트인 INST를 측정하였다. 즉, 같은 프로그램에 대해 서로 다른 시간에 측정된 성능 모니터링 이벤트들 값을 실행 순서대로 통합하였다. 통합 시 프로그램 외부 환경 요인(운영체제 및 I/O 간섭 등)에 의해 영향을 받은 샘플링 데이터의 영향을 최대한 배제하기 위해 공통의 INST 하드웨어 이벤트들 값을 비교하고 그 차이가 50% 이상 벗어나는 샘플링 데이터는 배제하고 통합하였다(5.1절 참조).

4.2 성능 모니터링 이벤트 선택

PXA320 프로세서는 동시에 4개의 성능 모니터링 이벤트 만을 측정할 수 있다. 따라서 프로그램 실행 중에 측정된 성능 모니터링 이벤트를 사용하여 전력을 예측하기 위해선 초기 10개의 성능 모니터링 이벤트들 중에서 4개의 성능 모니터링 이벤트들을 선택하여야 한다. 따라서 전력소모에 비슷한 영향을 미치는 성능 모니터링 이벤트들을 그룹들을 찾아서 대표적인 하나의 이벤트만을 선택하도록 통합된 데이터에 계층적 클러스터링(hierarchical clustering) 통계분석[15,17]을 적용하였다.

N개의 데이터 요소에 대한 클러스터링은 두 데이터 요소 i, j 의 유사성(Similarity)을 $N \times N$ 행렬 S 상에서 $S(i,j)$ 로 표시하여 클러스터링을 시도한다. 계층적 클러스터링은 이전에 구해진 클러스터들에 근거하여 다음 클러스터들을 결정한다. 즉, N개의 단일 요소 클러스터들로 시작하여 비슷한 클러스터들의 쌍을 하나의 클러스터로 결합하고, N개의 요소를 갖는 하나의 클러스터로 통합될 때까지 반복한다. Spearman의 상관관계 계수 ρ^2 을 사용하여 각 클러스터들 사이의 유사성을 측정하였다.

그림 2는 10개의 초기 성능 모니터링 이벤트들에 대한 전력소모에 미치는 영향을 계층적 클러스터링을 적용한 결과이다. 클러스터를 연결한 각 레벨은 유사성의

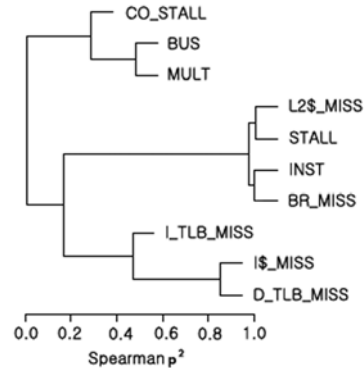


그림 2 초기 성능 모니터링 이벤트들의 계층적 클러스터링 결과

정도를 표시한 Spearman의 상관관계 계수 ρ^2 를 나타내며 이 값이 클수록 전력소모에 큰 상관관계를 갖고 두 클러스터 간의 유사성이 크다는 것을 의미한다. 그림 2의 결과를 기준으로 상관관계를 고려하여 4개의 클러스터로 분리하면 아래와 같이 된다.

{{STALL, L2\$_MISS}, {INST, BR\$_MISS}}

{{I\$_MISS, D_TLB\$_MISS}, I_TLB\$_MISS}

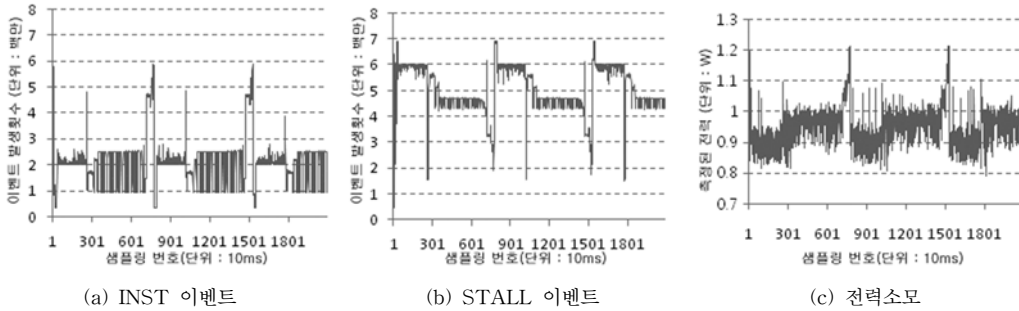
{{BUS, MULT}, CO_STALL}

데이터 종속관계에 의한 실행 중지(STALL)는 메모리 액세스와 깊은 관련이 있기 때문에 L2 캐시미스(L2\$_MISS)와 큰 상관관계를 갖고 클러스터링 되었다. 또한 실행되는 명령의 수가 많을수록(INST) 많은 분기 명령을 실행하므로 분기예측 미스(BR\$_MISS)가 많이 발생하므로 높은 상관관계를 보였다. I\$_MISS, D_TLB\$_MISS, I_TLB\$_MISS도 메모리 참조와 페이지 참조라는 공통성을 가져서 서로 상관관계를 갖고 같은 클러스터링에 결합되었다. 나머지 BUS, MULT, CO_STALL은 각기 개별적인 하드웨어 리소스에 관련된 이벤트들로 그다지 높은 상관관계를 갖지 못하였다.

그림 2와 같은 계층적 클러스터링 결과를 고려하고, 같은 클러스터에 속한 성능 모니터링 이벤트들 중에서 상대적으로 더 빈번하게 발생하는 이벤트를 고려하여 STALL, I\$_MISS, BUS 등의 성능 카운터 이벤트들을 각 클러스터에서 대표적인 성능 모니터링 이벤트들로 선정하였다. 여기에 성능 및 데이터 샘플링을 위한 지표로 사용되는 INST 성능 카운터 이벤트를 포함하여 아래와 같이 4개의 성능 카운터 이벤트들을 전력 예측 모델을 위한 성능 카운터 이벤트로 선정하였다.

{INST, STALL, I\$_MISS, BUS}

그림 3은 bzip2 벤치마크에 대해 선정된 4개의 성능 모니터링 이벤트들 중에서 INST, STALL 성능 모니터링 이벤트 변화와 전력소모 변화를 측정하여 보여주는



(a) INST 이벤트 (b) STALL 이벤트 (c) 전력소모
 그림 3 Bzip2 벤치마크에 대한 INST, STALL 성능 모니터링 이벤트와 전력소모 추정값

그림이다. 그림 3에서 나타난 바와 INST가 커지면 프로세서의 작업량이 증가하여 전력소모가 커지고, STALL이 커지면 프로세서의 동작이 중지되어 전력소모가 작아짐을 알 수 있다.

4.3 전력 예측 모델 생성

4.3.1 회귀분석 이론

회귀분석(regression analysis)은 주어진 설계영역에서 종속변수와 독립변수 간의 관계를 수학적으로 표현하는 모델로써 설계 영역의 임의의 지점에서 종속변수를 예측하기 위해 사용된다. 최근 컴퓨터 시스템 연구에서도 성능 및 전력소모를 예측하기 위해 선형 회귀분석 모델을 널리 사용하고 있다[12-15].

선형 회귀분석은 다음과 같은 식 (1)로 표현된다. 여기서, y_i 는 독립변수 x_1, x_2, \dots, x_k 값에 의해 결정되는 i^{th} 종속변수의 값이다. β_i 는 변수 x_i 의 계수이고, e_i 는 i^{th} 예측 값의 에러를 나타내는 값이다. 선형 회귀분석 모델의 계수 β_i 는 식 (2)와 같이 관측된 y_i 값과 회귀분석 예측 값 사이의 에러인 MSE(Mean Square Error)를 최소화하는 최소 자승법(least square method)을 사용하여 결정된다.

$$f(y_i) = \beta_0 + \sum_{j=1}^k \beta_j X_{ji} + e_i \quad (1)$$

$$MSE = \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^k \beta_j X_{ji} \right)^2 \quad (2)$$

회귀분석으로 구해진 모델의 적합성은 기여도(multiple correlation statistic) R^2 에 의해 검증된다. 이 값은 식 (5)와 같이 회귀분석 에러(SSE) 값과 전체 에러(SST)로부터 구해진다. SSE는 식 (3)과 같이 관측된 값 y_i 와 회귀분석 모델에 의해 예측된 값 \hat{y}_i 사이의 에러를 나타내고, SST는 식 (4)와 같이 관측된 값 y_i 와

단순 평균값의 에러를 나타낸다. 만약 회귀분석 모델의 에러가 단순 평균값으로 구한 값과 같은 큰 에러를 보인다면 R^2 은 0이 된다. R^2 값이 클수록 회귀분석 모델이 관측된 값에 가까운 값을 정확하게 예측할 수 있음을 나타낸다.

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3)$$

$$SST = \sum_{i=1}^n \left(y_i - \frac{1}{n} \sum_{i=1}^n y_i \right)^2 \quad (4)$$

$$R^2 = 1 - \frac{SSE}{SST} \quad (5)$$

4.3.2 전력 예측 회귀분석 모델

PXA320 프로세서의 전력 예측을 위한 회귀분석 모델은 4개의 성능 모니터링 이벤트들이 독립변수가 되고, 전력소모가 종속변수가 되어 구성된다.

$$Power = \beta_0 + \beta_1 INST + \beta_2 STALL + \beta_3 I_MISS + \beta_4 BUS \quad (6)$$

회귀분석 결과 구해진 계수 및 R^2 값은 표 2와 같다. 표 2는 PXA320 프로세서가 클럭 주파수 806MHz, 624MHz, 403MHz에서 동작 시 각각 회귀분석을 적용하여 구해진 회귀계수들과 R^2 값을 보여주는 표이다. 표 2에 나타난 바와 같이 모든 클럭 주파수의 전력 예측 회귀분석 모델에 대해서 충분히 큰 R^2 값을 가지고 있어서 생성된 모델이 전력 예측 시 큰 오차 없이 정확하게 예측할 수 있음을 보여준다.

그림 4는 본 논문의 전력 예측 회귀분석 식 (6)에 대해 잔차분석(residual analysis)을 실시한 결과를 보여주는 그림이다. 잔차분석은 잔차 -관측된 값과 예측된 값 사이의 편차가 예측된 값과 상관관계가 없으면서 모든

표 2 PXA320 프로세서의 회귀계수 및 R^2 값

| 클럭 주파수 | β_0 | β_1 | β_2 | β_3 | β_4 | R_2 |
|--------|-------------------|-------------------|--------------------|--------------------|-------------------|-------|
| 806MHz | $9.561 * 10^{-1}$ | $1.657 * 10^{-8}$ | $-7.324 * 10^{-7}$ | $-1.778 * 10^{-8}$ | $2.402 * 10^{-6}$ | 0.967 |
| 624MHz | $6.316 * 10^{-1}$ | $3.486 * 10^{-8}$ | $-5.839 * 10^{-9}$ | $3.086 * 10^{-7}$ | $1.677 * 10^{-6}$ | 0.973 |
| 403MHz | $2.781 * 10^{-1}$ | $1.511 * 10^{-8}$ | $-3.566 * 10^{-9}$ | $1.603 * 10^{-7}$ | $1.290 * 10^{-6}$ | 0.935 |

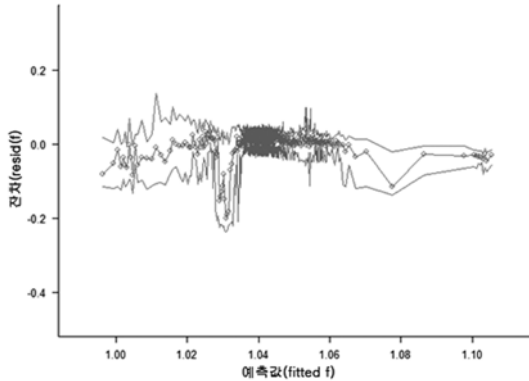


그림 4 잔차분석 결과 분포

예측된 값의 잔차가 랜덤하게 분포되어 있는가를 분석한다. 그림 4는 x 축의 각 예측값(fitted f)들에 대한 잔차(resid(f))들의 중간(median), 상위 및 하위 사분위(quartile) 분포를 보여주는 그림이다. 그림 4에 나타난 바와 같이 제안된 전력 예측의 잔차가 편향(bias)되지 않고 서로 독립적으로 랜덤하게 0 근처에 분포해 있음을 알 수 있다.

5. 실험 및 결과

5.1 실험 환경

본 논문의 전력 예측 모델을 위한 실험은 PXA320 프로세서가 장착된 마이크로비전의 MV320-LCD 개발 보드[18] 상에서 리눅스 커널 2.6.14를 포팅하고, 벤치마크 프로그램들을 실행하여 성능 모니터링 이벤트들과 전력소모를 측정함으로써 수행되었다. 벤치마크 프로그램으로는 표 3과 같이 임베디드 프로세서의 벤치마크로 널리 사용되는 MiBench 벤치마크 프로그램들[9,10]과 SPEC CPU2006 벤치마크[11]들 중에서 10개를 선택하였다.

MiBench 벤치마크들 중에서 각 응용 프로그램들 그룹 중에서 하나씩 선택하였고, SPEC 벤치마크에서는

가능하면 모바일 프로세서 응용에 가까운 벤치마크들을 선택하였다. 10개의 벤치마크들 중에서 6개의 벤치마크는 통계분석용 학습 데이터 생성과 전력예측 에러 검증에 모두 적용되었고, 4개의 벤치마크는 전력예측의 에러 검증에만 적용하였다. MiBench 벤치마크의 입력 데이터 세트로는 MiDataSets[19,20]을 사용하였다. MiDataSets은 MiBench 벤치마크 프로그램의 비교적 짧은 실행시간을 보완하기 위해 각각 20개의 추가적인 입력 데이터 세트를 제공한다. jpeg, gsm, dijkstra, ispell 벤치마크에 대해서는 홀수 번호의 10개 데이터가 학습 데이터 생성에 사용되었고, 짝수 번호의 10개의 데이터는 전력예측 에러 검증 시 사용하였다. SPEC 벤치마크인 경우 test 입력 데이터를 적용하였다. 벤치마크 프로그램을 실행하여 생성된 학습 데이터에 오픈소스 통계 소프트웨어 R[17]을 사용하여 계층적 클러스터링, 회귀분석 등 통계적 분석을 실시하였다.

4.1절에서 기술한 바와 같이 각 벤치마크들은 3개의 그룹으로 자기 독립적으로 실행되어 성능 모니터링 이벤트들과 전력소모가 측정되었다(실험의 정확도를 위해 동일 벤치마크 프로그램에 대해 4번 반복 수행하여 측정하였다). 독립적으로 측정된 각 그룹들의 데이터는 각 그룹의 공통의 INST 성능 이벤트 값을 비교하여 그 차이가 50% 이상 벗어나는 샘플링 데이터는 배제하고 통합하였다. 표 4는 INST 차이에 따른 데이터 필터링(제외된 데이터 비율)과 R^2 값을 보여주는 표이다. 표 4에 나타난 바와 같이 INST 차이를 너무 작게 하면 필터링되는 데이터의 비율이 높아지고, R^2 값은 낮아지게 된다. 반대로 INST 차이를 크게 하면 외부 환경 요인에 영향을 받을 샘플링 데이터의 필터링이 정상적으로 이루어지지 못하여 R^2 값이 낮아진다. 따라서 본 논문에서는 이를 고려하여 INST 값 차이가 50%를 기준으로 샘플링 데이터를 통합하였다.

MV320-LCD 개발보드 상에서 PXA320 프로세서의 전력소모를 측정하기 위해 그림 5와 같이 프로세서로 진입하는 전류와 프로세서의 전압을 디지털 멀티미터

표 3 벤치마크 프로그램

| 이름 | 타입 | 내용 | 학습 데이터 | 검증 데이터 |
|-----------|---------|--------------|---------------|---------------|
| jpeg | MiBench | JPEG 디코더 | MiDataSets 홀수 | MiDataSets 짝수 |
| gsm | MiBench | GSM 디코더 | MiDataSets 홀수 | MiDataSets 짝수 |
| dijkstra | MiBench | 그래프 최단거리 계산 | MiDataSets 홀수 | MiDataSets 짝수 |
| ispell | MiBench | 스펠링 체커 | MiDataSets 홀수 | MiDataSets 짝수 |
| bzip2 | SPEC | 파일 압축 | dryer.jpg | input.program |
| gobmk | SPEC | Go 게임 | dniwog.tst | connect.tst |
| blowfish | MiBench | blowfish 암호화 | X | MiDataSets |
| adpcm | MiBench | ADPCM 디코더 | X | MiDataSets |
| perlbench | SPEC | Perl 인터프리터 | X | splitmail.pl |
| mcf | SPEC | 차량 운행 스케줄링 | X | inp.in |

표 4 INST 차이에 따른 데이터 필터링과 R² 값

| INST 차이(%) | 데이터 필터링(%) | R ² |
|------------|------------|----------------|
| 5 | 79.3 | 0.470 |
| 20 | 54.1 | 0.680 |
| 50 | 8.16 | 0.981 |
| 75 | 4.84 | 0.713 |

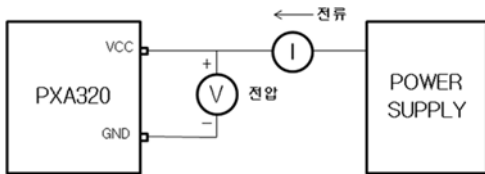


그림 5 PXA320 프로세서의 전류 및 전압 측정

(NI PXI-4017 DMM)로 연결하여 측정하고 샘플링 한 후, 전력 $P = I \times V$ 를 계산하였다. 전류의 측정을 위해 개발보드 상에서 프로세서의 V_{CC}에 직렬 연결된 인덕터를 분리하고 디지털 멀티미터를 직렬로 연결하여 측정하였다. NI PXI-4071 디지털 멀티미터는 고성능 전류, 전압 측정 장비로 7디지트까지 측정이 가능하며, 10nV에서 1000V까지의 전압과 1uA에서 3A까지의 6가지 실효 교류범위 전류를 측정할 수 있다.

그림 6은 본 논문에서 적용한 성능 모니터링 이벤트 및 전력 측정 데이터 수집 구성도이다. 그림 6에서 제어 프로그램은 벤치마크 프로그램 실행 및 성능 모니터링 모듈의 동기화를 제어하는 프로그램이다. 성능 모니터링 모듈은 매 10ms마다 인터럽트를 발생하며, 성능 모니터링 이벤트를 저장하고 다음 샘플링 구간의 값을 위해 초기화된다. 트리거 제어 모듈은 동일한 샘플링 시점에서 프로세서 전력소모를 측정하도록 성능 모니터링 모듈에서 매 10ms마다 인터럽트가 발생될 때 GPIO 인터페이스를 통하여 트리거 신호를 디지털 멀티미터(NI PXI-4071 DMM)로 5V의 동기 신호를 전송한다. 디지털 멀티미터는 이 신호에 동기되어 전압, 전류를 측정하고 컨트롤러(NI PXI-8106)로 송신한다. 벤치마크 프로그램이 종료되면 성능 모니터링 모듈은 성능 모니터링

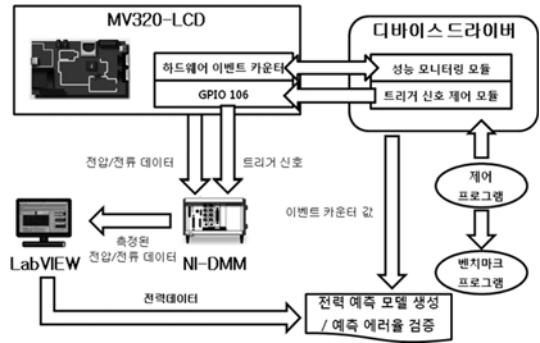


그림 6 성능 모니터링 이벤트 및 전력 측정 데이터 수집 구성도

이벤트 카운터 값을 출력하고, 디지털 멀티미터는 컨트롤러에 설치된 LABVIEW[21] 프로그램을 통해 전력 값을 출력한다. 출력된 전력과 성능 모니터링 이벤트 카운터 값은 하나의 파일의 학습 데이터로 통합되어 회귀분석을 실시하여 전력 예측 모델을 생성한다. 동일한 형태의 실험을 다른 검증 데이터에 대해 반복하여 실시하면서 예측 모델에 의한 예측 전력값과 실측 전력값을 비교하여 예측 에러율을 검증한다. 실험 시 리눅스에서 실행되는 다른 데몬들에 의해 샘플링 데이터의 값이 영향 받는 것을 최소화하기 위해 OS 구동에 필수적인 데몬들을 제외한 불필요한 데몬(LCD 드라이버, AC97, 시스템 로그)들을 정지시킨 상태에서 실험을 진행하였다.

그림 7은 PXA320 프로세서의 전력을 측정하기 위해 사용된 측정장비와 보드와의 연결 상태를 보여주는 그림이다. 좌측 그림은 NI 컨트롤러와 디지털 멀티미터의 연결 그림이고, 중앙은 멀티미터와 MV320-LCD 사이의 연결을 보여주는 그림이다. 우측은 MV320-LCD 보드 상에서 전압 및 전류 측정을 보여주는 그림으로 위쪽의 프로세서의 V_{CC}에 직렬 연결된 인덕터를 단절한 전류 측정, 아래 쪽은 병렬로 연결된 전압 측정, 오른쪽은 트리거를 위한 GPIO 연결을 보여준다.



그림 7 전력 측정 실험 환경과 측정 장비

5.2 실험 결과

제안된 통계적 분석 기반 전력 예측 모델의 정확도를 검증하기 위해 측정된 전력 값과 예측된 전력의 값의 차이로부터 예측 에러율을 계산하였다. 그림 8은 각 벤치마크 수행 시의 실측전력과 예측전력의 절대값을 보여준다. 그림 8에서 5.1절에서 언급된 바와 같이 jpeg에서부터 gobmk까지 6개의 벤치마크는 통계 분석용 학습 데이터 생성에도 사용된 벤치마크이고(단, 학습 데이터 생성과 예측 에러율 측정에는 서로 다른 입력 데이터를 사용하였다), blowfish에서 mcf까지 4개의 벤치마크는 에러율 검증에만 적용된 벤치마크들이다. 그림 8에서 실제 측정된 전력은 평균 0.98W(최소 0.84W, 최대 1.05W)이고, 예측된 전력은 평균 0.95W(최소 0.89W, 최대 1.04W)임을 알 수 있다. 그림 9는 3개의 클럭 주파수에서 10개 벤치마크에 대해 측정된 에러율을 보여준다. 그림 9에 나타난 바와 같이 클럭 주파수 806MHz, 624MHz, 403MHz에서 평균 예측 에러율이 3.31%, 2.39%, 3.49%로 제안된 전력 예측 모델이 정확하게 전력소모를 예측함을 알 수 있다. 또한 벤치마크 프로그램 MiBench와 SPEC, 그리고 학습 데이터 생성에 포함 여부에 관계없이 일관되게 정확한 예측을 하였다. 예외적으로 클럭 주파수 624MHz에서, MiBench 벤치마크 프로그램들 중 adpcm의 경우 평균 에러율보다 상당히 높은 에러율을 나타냈다. 높은 에러율을 보이는 대부분의 벤치마크에서는 높은 메모리 액세스로 인해 실행 중지(STALL)가 크게 증가되어 수행되는 명령어(INST)가 크게 감소하는 예외적인 구간들로 인하여 높은 평균 에러율을 보였다. 즉, 메모리 액세스율이 크게 높아져서 성능 모니터링 이벤트 값들이 정상적인 통계적인 예측 범위를 벗어나서 다소 부정확한 전력 예측률을 보였다. 향후 연구에서는 이와 같은 예외적인 상황도 고려한 전력 예측 모델 개발도 모색할 예정이다.

제안된 전력 예측 모델은 10ms 단위로 성능 모니터링 이벤트 값과 전력 측정 값을 샘플링하여 생성하였다. 그림 10은 806MHz의 주파수에서 각기 다른 샘플링 주기에 제안된 예측 모델을 적용하여 측정된 평균 예측

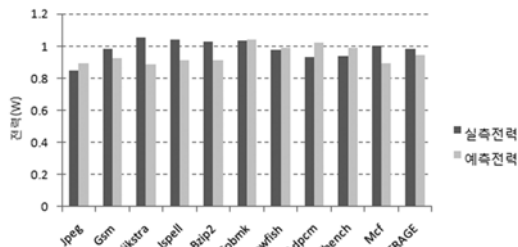


그림 8 벤치마크 별 실측전력과 예측전력

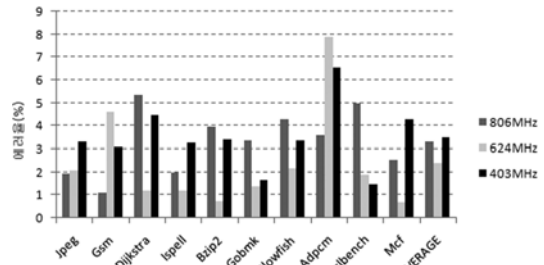


그림 9 전력 예측 모델의 예측 에러율

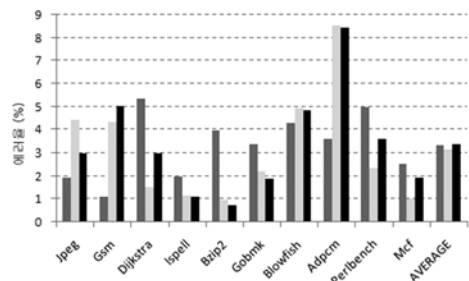


그림 10 샘플링 주기 변화에 따른 평균 전력 예측 에러율

에러율을 보여준다. 샘플링 주기 10ms, 50ms, 100ms에서 예측 에러율은 각각 3.31%, 3.15%, 3.36%로 제안된 모델은 샘플링 주기에 영향을 받지 않고 선택된 성능 모니터링 이벤트로부터 정확하게 프로세서의 전력소모를 예측할 수 있음을 확인하였다.

6. 결론

본 논문에서는 모바일 프로세서의 전력소모 예측을 위해 정량적으로 프로세서의 동작을 분석하고 모델링하는 통계적인 분석 방법을 개발 제안하였다. 제안된 전력 예측 방식은 다양한 벤치마크 프로그램들을 실행하여 프로세서 관련 다수의 성능 모니터링 이벤트들과 전력소모 데이터를 수집하고, 계층적 클러스터링 분석 등을 적용하여 서로 중복되지 않으면서 전력소모에 크게 기여하는 대표적인 성능 이벤트들을 선정하였다. 전력 예측 모델은 선택된 성능 모니터링 이벤트들이 독립변수가 되고 전력소모가 종속변수가 되는 회귀분석을 수행하여 개발하였고, R² 값과 잔차분석을 통하여 통계적으로도 유의함을 확인하였다. 제안된 통계적 분석의 전력 예측 모델은 PXA320 프로세서 상에서 MiBench 벤치마크 프로그램과 SPEC 벤치마크 프로그램 등을 실행하여 학습하고 적용한 후 실제 측정된 전력 값과 비교한 결과 평균 4% 이내의 에러율을 갖고 정확하게 프로세서의 전력소모를 예측할 수 있음을 보였다.

향후 제안된 통계적 전력예측 모델을 PXA320 프로세서 외에 Core2Duo 등 다른 프로세서에도 적용하고, 성능 모니터링 이벤트 값들로 성능과 전력소모를 함께 예측하여 성능 손실은 최소로 하면서 전력소모는 최대한 줄이는 전력관리 기법을 개발할 예정이다.

참 고 문 헌

[1] B. Sprunt, "Basics of Performance-Monitoring Hardware," *IEEE Micro*, vol.22, Issue 4, pp.64-71, July-August, 2002.

[2] G. Contreras and M. Martonosi, "Power Prediction for Intel XScale Processors Using Performance Monitoring Unit Events," *Proc. Int'l Symp. Low Power Electronics and Design (ISLPED'05)*, Aug. 2005.

[3] K. Rajamani et al., "Application-Aware Power Management," *Proc. Int'l Symp. Workload Characterization (IISWC '06)*, Oct. 2006.

[4] Q. Wu et al., "Dynamic Compilation Framework for Controlling Microprocessor Energy and Performance," *Proc. Int'l Symp. Microarchitecture (MICRO '05)*, Nov. 2005.

[5] C. Isci and M. Martonosi, "Phase Characterization for Power: Evaluating Control-Flow-Based and Event-Counter-Based Techniques," *Proc. Int'l Symp. High-Performance Computer Architecture (HPCA '06)*, Feb. 2006.

[6] C. Isci, G. Contreras, and M. Martonosi, "Live, Runtime Phase Monitoring and Prediction on Real Systems with Application to Dynamic Power Management," *Proc. Int'l Symp. Microarchitecture (MICRO '06)*, Dec. 2006.

[7] 3rd Generation Intel XScale® Microarchitecture Developer's Manual, Intel, 2007.

[8] Monahans P Processor System and Timer Configuration Developers Manual, Marvell, 2006.

[9] M. Guthaus et al., "MiBench: A Free, Commercially Representative Embedded Benchmark Suite," *Proc. Annual Workshop on Workload Characterization*, Dec. 2001.

[10] MiBench Version 1.0, <http://www.eecs.umich.edu/mibench/>

[11] Standard Performance Evaluation Corporation (SPEC), <http://spec.org/>

[12] H. Sasaki, Y. Ikeda, M. Kondo and H. Nakamura, "An Intra-Task DVFS Technique based on Statistical Analysis of Hardware Events," *Proc. Int'l Conf. Computing Frontiers*, May 2007.

[13] S. Lee, H. Lee and P. Yew, "Runtime Performance Projection Model for Dynamic Power Management," *Proc. Asia-Pacific Computer Systems Architecture Conference (ACSAC '07)*, pp.186-197, Aug. 2007.

[14] B. Lee and D. Brooks, "Accurate and Efficient

Regression Modeling for Microarchitectural Performance and Power Prediction," *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS '06)*, Oct. 2006.

[15] B. Lee, and David M. Brooks, "Spatial Sampling and Regression," *IEEE Micro*, vol.7, Issue 3, pp.74-93, May-June 2007.

[16] W. Mathur and J. Cook, "Improved Estimation for Software Multiplexing of Performance Counters," *Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, 2005.

[17] R Project for Statistical Computing, <http://www.r-project.org/>

[18] MICROVISION, <http://mvtool.co.kr>

[19] G. Fursin et al., "MiDataSets: Creating The Conditions For A More Realistic Evaluation of Iterative Optimization," *Proc. Int'l Conf. High Performance Embedded Architectures & Compilers (HiPEAC '07)*, Jan. 2007.

[20] MiDataSets for MiBench, <http://midatasets.sourceforge.net/>

[21] National Instruments LabVIEW, <http://www.ni.com/labview>



윤희성
순천대학교 컴퓨터학과 졸업(학사). 2008년~현재 순천대학교 컴퓨터학부 석사과정. 관심분야는 임베디드 시스템 및 저전력 시스템, 모바일 플랫폼 설계 등



이상정
한양대학교 전자공학과 졸업(학사, 석사, 박사). 1999년 미국 University of Minnesota 방문교수. 1988년~현재 순천향대학교 컴퓨터학부 교수. 관심분야는 멀티코어 프로세서와 모바일 시스템의 성능 및 전력 동적 최적화 등