

특 집

저전력 소프트웨어 개발을 위한 전력 분석 도구

김 영 진* 백 용 기** 김 지 흥***

목 차

- 1. 서 론
- 2. 전력 분석 도구 요구사항
- 3. 사이클 수준 실측을 통한 전력 분석 도구
- 4. 인터벌 기반의 샘플링을 통한 전력 분석 도구
- 5. 결 론

1. 서 론

근래에 다양한 이동성 정보 단말기의 사용이 보편화됨에 따라 배터리 기반의 임베디드 시스템에 대한 저전력 요구가 크게 대두되고 있다. 기존의 저전력 설계 기법은 주로 회로 수준에서 이루어져 왔기 때문에 저전력 하드웨어 상에서 구현되어 왔으나 마이크로프로세서의 성능이 향상되고 네트워크 대역폭이 커짐에 따라 이동성 단말기 상에서의 동화상, 음악, 게임 등의 다양한 기능 및 고성능 요구로 인해 소프트웨어의 전력 소모가 커지고 있다. 따라서, 소프트웨어 수준에서의 저전력 기법의 요구가 되고 있으며 이를 위해서는 타겟 아키텍처 상에서 주어진 소프트웨어가 소비하는 전력을 정확히 측정하고 분석할 수 있는 도구가 필요하다.

기존에 연구, 개발되어 온 전력 도구는 주로 전력 측정에 초점이 맞추어져 개발되어 왔는데 전력 소모 모델을 포함한 시뮬레이터를 이용한 방법이 많이 연구되었다. 시뮬레이터를 이용한 방법은 기본적으로 시스템의 각 컴포넌트를 모델링하고 수

행 기능을 시뮬레이션하는 프로그램에 전력 소모 모델을 추가하여 사용한다. 시뮬레이션 수준에 따라 회로나 게이트 수준의 low-level 시뮬레이터, 아키텍처 시뮬레이터 및 명령어 수준 시뮬레이터로 구분된다. low-level 또는 아키텍처 시뮬레이터 기법들은 비교적 정확한 전력 소모값을 출력하지만 시뮬레이션에 많은 시간이 소모되며 하드웨어에 초점이 맞추어져 있으므로 소프트웨어와 연계하여 전력 소모 내용을 분석하는 것이 힘들다[1,2,3]. 명령어 수준 시뮬레이터는 프로세서의 명령어 수준의 시뮬레이터에 전력 소모 모델을 추가하여 전력 소모를 소프트웨어 연계하려는 노력을 하였으나 [4] 모델링 정도에 따라 정확성이 가변하게 되며 실제 측정을 바탕으로 모델링[5] 또한 명령어 피연산자에 대한 오차가 존재하고 명령어별 전력 수립 및 관리에 많은 비용이 소요된다는 단점이 있다. 그밖에 에너지 매크로 모델(Energy Macro-Model)을 사용하여 프로그램의 실행 경로에 따른 전력 값을 측정하여 프로그램의 각 기본 블록(Basic Block)별 전력 소모값을 추정하는 기법이 연구되었다[6].

이러한 시뮬레이션 기반 전력 측정 도구에서의

* 서울대학교 전기·컴퓨터공학부 박사 과정
 ** 서울대학교 전기·컴퓨터공학부 석사 과정
 *** 서울대학교 전기·컴퓨터공학부 부교수

<표 1> 기존 전력 측정 도구 분류

분류	연구기관 (대표 저자)	내 용	연도	
시뮬 레이터 사용	회로 수준	Synopsys Inc. PowerMill이라는 로직 및 회로 설 계 수준의 전력 소모 측정용 시뮬레 이터를 제시	2000	
	아키텍처 수준	Princeton University (D. Brooks)	Watch라는 아키텍처 수준에서의 전력 소모 측정용 시뮬레이터를 제시	2000
		Penn State University (N. Vijaykrishnan)	SimplePower라는 아키텍처 수 준에서의 전력 소모 측정용 시뮬레 이터를 제시	2000
	명령어 수준	Princeton University (V. Tiwari)	명령어 수준에서의 전력 분석 기법 및 최적화 기법 제시	1994
		Stanford University (T. Simunic)	ARM 프로세서에 프로그램 실행시 소모되는 에너지를 측정할 수 있는 전력 소모 시뮬레이터를 제시	1999
기본 블록 수준	Princeton (T. K. Tan)	Energy Macro 모델을 이용하여 기 본 블록별 전력 소모값을 추정하는 기법 제시	1999	
측정 장비 사용	Carnegie Mellon University (J. Flynn)	Powerscope라는 실제 측정에 기반 을 둔 소프트웨어 전력 측정기법 제시	1999	

소프트웨어 연계 분석을 위한 정보 취득에 대한 하드웨어 측정 장치를 이용한 전력 분석 도구는, 전력 소모 모델링 과정없이 소프트웨어 수행에 따른 실시간 측정을 수행하고 소프트웨어 연동 분석을 하고자 하는 방법이다[7]. 이 방법은 멀티미터와 같은 외부 측정 장비를 사용하여 주기적으로 시스템의 평균 전류를 샘플링하여 전체 시스템의 전력 소모값을 측정하며 측정된 전력값을 프로그램의 소스 코드와 연계하기 위해 에너지 측정값과 프로그램의 프로그램 카운터(PC)를 동기시켜 추출 및 사용한다. [7]에서 사용된 기법은 커널의 적재 가능한 모듈과 데몬(daemon)을 이용하여 이식성을 높이고 있으나 실제적으로는 모듈에서 제공하는 함수들을 라이브러리화 함으로써 타겟 응용 프로그램에서의 디바이스 드라이버 호출 시에 함수 경로가 길어지고 시스템 수행 시 프로그램 카운터 샘플값과 전력 측정 샘플값과의 동기화를 위해 수

행되는 GPIO(General Purpose IO) 인터럽트 처리 시간의 가변으로 동기화 주기가 일정하지 않게 되어 정확한 에너지 산출이 불가하다는 단점이 존재한다. <표 1>은 지금까지 기술한 선행 전력 측정 및 분석 도구의 내용을 요약한 것이다.

본 고에서는, 전력 측정 및 분석을 위해 제안된 기법들 중 서울대학교 임베디드시스템 연구실에서 개발된 전력 분석 도구들을 중심으로 소개하고자 한다. 소개되는 도구들은 전력 소모값을 정확하면서도 신속히 측정하는 반면에 타겟 소프트웨어의 소스 코드를 연계하여 분석할 수 있어 실질적으로 소모 전력 최적화 작업과 연계하여 사용될 수 있는 장점이 있다.

2. 전력 분석 도구의 요구사항

근래에 배터리 기반의 이동성 임베디드 시스템의 광범위한 사용은 비예측적인 배터리 동작과 사용자의 실시간 응용, 고기능 응용 요구 등으로 인해 기존의 전력 분석 도구 연구 내용들에 대해 다음과 같이 새로운 요구사항을 추가로 요구하거나 보다 강화된 형태의 요구사항을 필요로 하게 되었다.

2.1 정확성 및 신속성

전력 측정 결과는 비교적 정확하면서도 빠르게 제공되어야 한다. 이것은 전력 측정 결과가 측정 과정동안 실행된 소프트웨어와 연계하여 신속하게 보여줌으로써 측정값에 대한 신뢰성을 높이고 즉각적인 분석이 가능하도록 하여 실용적인 도구로서의 유용성을 크게 하기 때문이다. 또한, 신속하게 최적화 작업으로 피드백되어 연계될 수 있기 때문이다. 정확성과 신속성을 얻기 위해서는, 타겟 시스템 상에서 수행되는 응용 S/W 및 시스템 S/W에 대한 전력 소모 및 시스템 동작에 대한 충분한 정보를 정확한 분석을 위해 수집할 수 있어야 하는

반면에, 타겟 시스템에 대한 전력 측정 도구의 부대비용(overhead)이 적어야 한다.

2.2 소프트웨어 연계

기존의 전력 도구는 주로 하드웨어 수준에서 정확성을 염두에 두고 개발됨으로써 측정의 목적은 충분히 달성할 수 있다고 여겨진다. 그러나, 앞서 기술한 바와 같이 이동 단말기에서의 멀티미디어 응용 프로그램 등의 사용이 보편화됨에 따라 소프트웨어의 소모 전력이 큰 비중을 차지하게 되어 소프트웨어의 구성요소들에 대해 상세한 소모 전력 프로파일링의 필요성이 크게 제기되었다. 소프트웨어를 연계하는 전력 분석은, 특히 타겟 시스템의 수행 정보와 전력 측정 내용을 연계하여 분석함으로써 타겟 소프트웨어의 모든 구성 요소들에 대해 소모 전력의 병목지점을 파악할 수 있도록 하여 전력 최적화 작업을 직관적이고 용이하게 할 수 있다.

2.3 이식성

사용자 입장에서의 저전력성에 대한 요구는 타겟 플랫폼이나 프로세서에 관계없이 요구되는 사항인데, 기존의 회로 수준의 전력 측정 도구는 특정 프로세서를 대상으로 해서 개발되므로 프로세서가 바뀌면 새로운 전력 측정 도구 개발이 요구된다. 이것은 많은 시간, 비용 및 노력을 요하는 일이므로 정확한 측정을 가능하게 하면서도 범용적으로 설치 및 사용의 편이가 있는 측정 장비를 사용하는 것이 필요하다. 또한, 측정 장비와 연계하여 사용하는 소프트웨어의 경우에 대해서도 적재 가능한(loadable) 커널 모듈 등을 이용하여 다른 플랫폼에 대한 전력 측정 도구 재구현을 용이하게 할 필요가 있다.

2.4 최적화 도구와 상호 연계

소프트웨어 전력 분석 도구는 기본적으로 전력

최적화 작업을 수행하기 위한 기반 도구로서 소모 전력이 큰 모듈, 루틴 또는 함수 등의 하위 컴포넌트를 발견해 내고 이에 대한 절대적 또는 상대적 전력 소모를 출력하는 기능을 가져야 한다. 지금까지 기존 연구들은 독립적으로 전력 측정에 주력하거나 소프트웨어 연계 분석을 수행하도록 발전되어 왔으나 향후의 전력 분석 도구는 소모 전력에 대한 정보를 체계화하고 의미화함으로써 최적화 작업에 요긴하고 쉽게 사용될 수 있도록 연구되어야 한다. 구현적인 측면에서는, 단위 임베디드 시스템에 대해서 분석된 소모 전력 내용 및 전력 최적화 내용을 상위 운영 프로그램에 연동하여 그 일부로서 처리해야 할 필요가 있다.

2.5 성능 분석 도구와 상호 연계

시스템에서 수행 시간과 같은 성능 지표는 지연(delay)에 반비례하므로 보통 소모 전력과 병치하여 나타난다. 실시간 멀티미디어나 네트워크 통신 등의 사용은 수행 시간에 대한 보장을 반드시 필요하므로 전력 최적화 작업은 성능 최적화와 연계하여 상관 관계를 고려하여 수행하는 것이 필요하다. 따라서, 소모 전력 분석 도구 역시 성능 분석 도구와 연계하여 개발함으로써 사용자가 요구하는 최적의 임베디드 소프트웨어 개발을 가능하게 하는 통합(integrated) 소프트웨어 분석 도구가 요구된다.

3. 사이클 수준 실측을 통한 전력 분석 도구

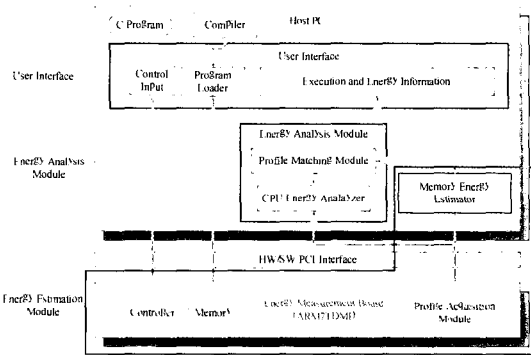
SES(SNU Energy Scanner)는 타겟 응용 프로그램을 실제 타겟 프로세서상에서 수행시키며 CPU의 전력 소모량을 사이클 단위로 직접 측정함으로써 정확하고 자세한 정보를 실시간에 제공해주는 장점을 가지도록 개발된 전력 분석 도구다 [8,9]. 또한, 기존의 성능 최적화와 디버깅을 위해서 사용된 소프트웨어 디버거에 전력 측정 정보를

함께 표시함으로써 성능, 정확성 및 저전력의 3가지 요소가 연관될 수 있도록 하고 있다. 본 도구는 사이클 단위로 전력을 측정하는 기법을 기반으로 매 사이클마다 프로세서의 전력 소모를 수집하고 모아진 결과를 C 프로그램이나 어셈블리 소스코드와 연계하여 사용자에게 전력 소모에 대한 정보를 제공한다. 소프트웨어와 연계된 전력 소모 결과는 GNU의 gdb와 유사한 GUI환경에서 제공되어, 프로그램 개발자가 전력 소모가 많은 지점을 편리하게 찾을 수 있도록 해준다.

모듈은 에너지 측정 보드와 메모리 에너지 측정기로 구성되어 있다. PCI 슬롯에 장착할 수 있는 형태로 개발된 측정 보드는 보드 내의 프로파일 획득 모듈을 이용하여 실시간으로 타겟 프로그램의 사이클 단위의 시스템 트레이스를 수집한다. 수집된 시스템 트레이스는 호스트 PC로 PCI 버스를 통하여 전달된다. 호스트 PC는 Linux 운영체제를 사용한다.

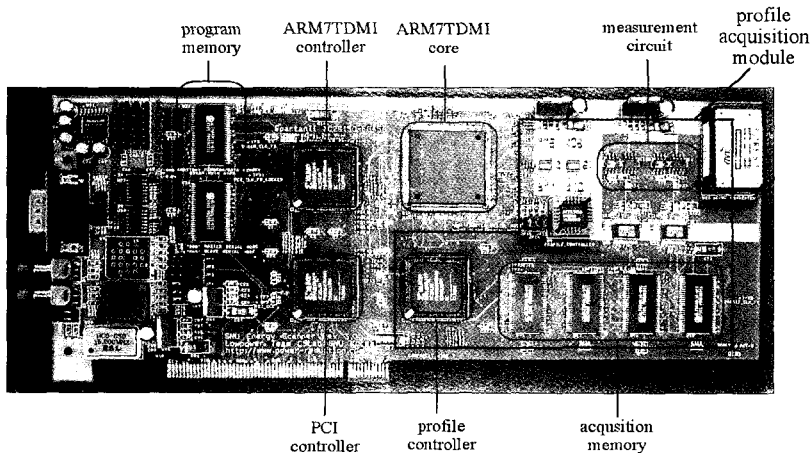
(그림 2)에서 보는 바와 같이 에너지 측정 보드는 ARM7TDMI 프로세서 코어와 컨트롤러, 프로파일 획득 모듈, 프로그램 모듈, 그리고 PCI 컨트롤러로 구성되어 있다. 에너지 측정 보드는 ARM7TDMI의 에뮬레이터로서 동작한다. 측정 보드로부터의 시스템 트레이스는 프로세서의 사이클 수준의 에너지 트레이스와 사이클 수준의 메모리 트레이스로 구성되어 있다.

호스트 PC에서 동작하는 메모리 에너지 측정기는 다양한 캐쉬와 버스, 메모리의 에너지 모델을 가진 소프트웨어 메모리 시뮬레이터다. 메모리 에너지 시뮬레이터는 입력 값으로 측정 보드로부터의 메모리 트레이스를 받아서 캐쉬와 오프칩 메모리에서의 사이클 수준의 에너지를 출력한다. 에너지 분석 모듈은 프로세서와 메모리 시스템의 사이

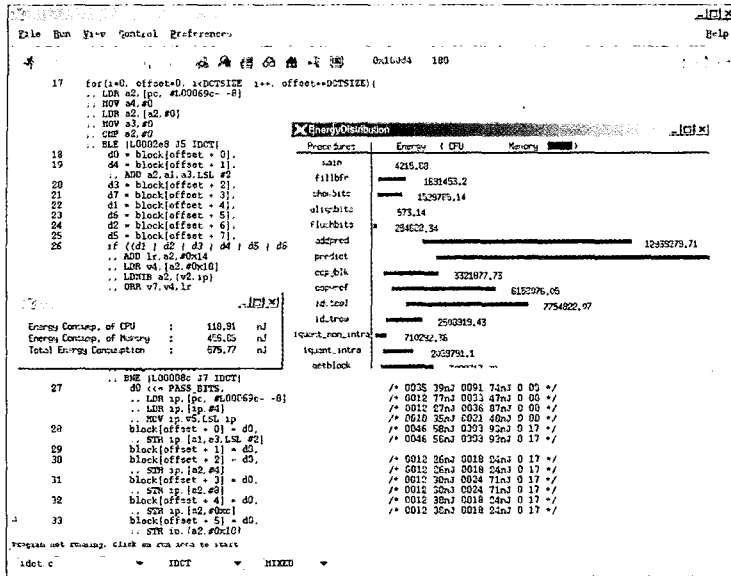


(그림 1) SES의 전체 구조

SES는 에너지 측정 모듈, 에너지 분석 모듈, GUI의 3개 모듈로 구성되어 있으며 에너지 측정



(그림 2) SES의 하드웨어 구성



(그림 3) SES 사용자 인터페이스의 예

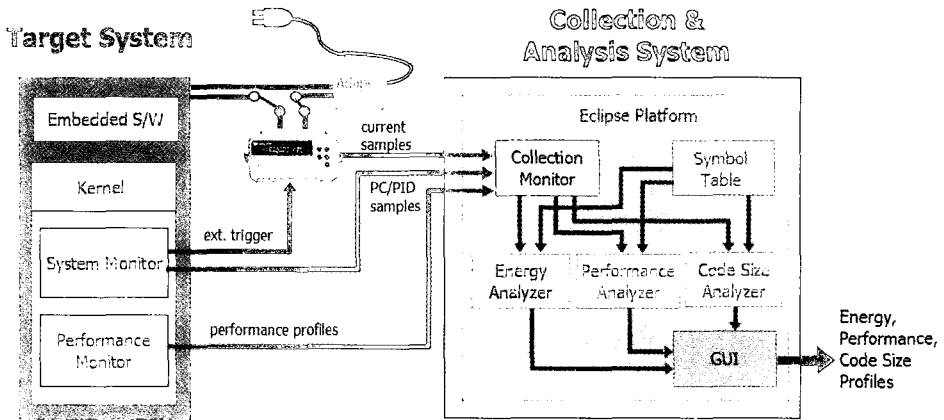
클 수준의 에너지 프로파일을 소스 코드와 연계시킨다. 에너지 프로파일은 소스코드와 C 소스, 어셈블리어, C 함수의 3가지 단계로 연계가 된다. 사용자는 (그림 3)에서 보여지듯이 gdb와 유사한 GUI를 이용하여 SES를 사용한다. (그림 3)에서 주 디스플레이 윈도우는 관찰되는 프로그램의 소스를 보여주며, 메뉴 바는 프로그램의 에너지 프로파일을 얻기 위한 다양한 기능을 제공하고 있다. 메뉴 바의 기능을 통해서 소스코드를 컴파일하고 프로파일할 영역을 설정하고 브레이크 포인트를 설정하고 실행 바이너리 코드를 에너지 측정 보드로 다운로드 할 수 있다. (그림 3)에서는 27행부터 32행까지가 프로파일 영역으로 설정되어 있고 해당 영역에 에너지 프로파일 결과가 보여주고 있는데 함수 수준에서의 에너지 모니터링 결과는 에너지의 분포 그래프로 나타난다.

4. 인터벌 기반의 샘플링을 통한 전력 분석 도구

SES는 하드웨어 보드 내에 전력 측정을 위한 회

로가 내장되어 있기 때문에 전력 측정을 위한 멀티미터와 같은 추가적인 측정 장비는 필요하지 않으며 사이클 수준의 정확성 및 실시간성을 제공한다. 또한, 기존의 소프트웨어 디버거인 gdb 기반의 도구를 사용함으로써 성능 최적화 도구와의 연계 가능성을 취하고 있다. 그러나, 다른 프로세서에 대한 전력 측정 시에는 완전히 새로운 하드웨어 및 프로그램 개발이 요구되므로 시간, 노력 및 비용이 추가로 많이 요구된다는 문제점을 가진다. 또한, 성능 지표에 대한 정보는 따로 측정하지 않고 있어 정확한 의미의 성능 및 전력 연계 분석 도구라고 말할 수 없다.

ePRO는 SES를 포함한 기존 전력 분석 도구들과 앞서 제시한 전력 분석 도구의 확장된 요구사항을 충족하도록 개발된 통합 소프트웨어 분석 도구로서 소모 전력 측정값과 성능 모니터링 내용을 타겟 소프트웨어의 함수 수준에서 연계하여 출력한다[10, 11]. ePRO는 [7]에 대해서 데몬에서의 주기적 파일 시스템 작업의 부대비용을 없앴으며 디바이스 드라이버의 함수 호출 및 수행 경로 단순화에 따른 부대비용을 줄였다. 또한, 오픈 루프 트리



(그림 4) ePRO의 전체 구조

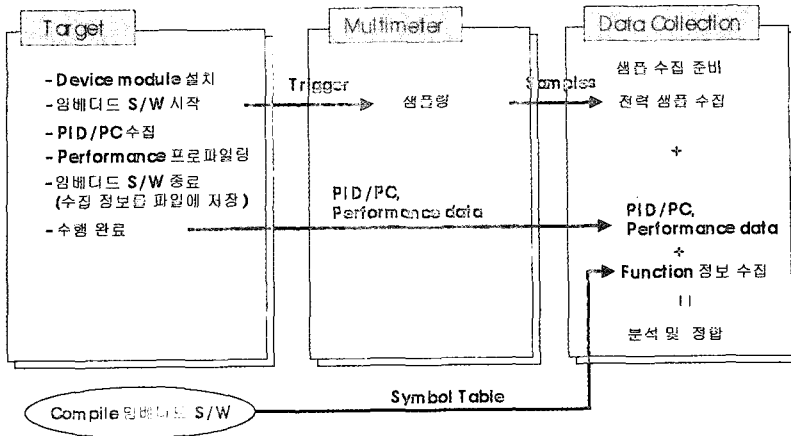
거링만을 취함으로써 GPIO 인터럽트 처리시간의 가변에 의한 멀티미터에 대한 피드백 트리거링 주기의 불규칙성으로 인해 발생했던 소모 에너지 측정값의 부정확성을 제거하였다. 한편, ePRO는 SES에 비해 이식성 및 전력 최적화 도구와 성능 분석 도구에 대한 연계성을 높이고 있어 통합 소프트웨어 분석 도구로서의 개발 방향을 취하고 있다.

ePRO는 (그림 4)에서 보는 바와 같이 크게 시스템 수행 내용을 수집하는 타겟 시스템과 멀티미터로부터의 전력 샘플을 넘겨받아 시스템 수행 내용과 심볼 테이블 및 전력 수집 내용을 비교 및 분석하고 이들을 연계하는 전력 수집 및 분석 시스템과 멀티미터로 구성된다. 타겟 시스템은 임베디드 S/W, 시스템 모니터 및 성능 모니터로 구성되며 시스템 모니터 블록은 디바이스 드라이버 모듈 및 인터럽트 처리 루틴으로 이루어지고 성능 모니터 블록은 성능 지표 수집을 위해 코그 생성기에 의해 삽입되는 시스템 함수 호출을 포함한다. 한편, 수집 및 분석 시스템은 멀티미터로부터 넘겨받은 전류값을 수집하여 소모 전력을 구하고 타겟 시스템으로부터 전송된 시스템 수행 정보 및 성능 정보를, 심볼 테이블을 이용하여 소스 코드와 소모 전력을 비교 및 분석하여 연계시킨다. 또한, 성능 수집 및

분석 시스템은 수집 모니터, 에너지 분석기 및 성능 분석기로 구성되며 수집 모니터는 통신 모듈, 수집 모듈 및 사용자 인터페이스로 이루어진다.

ePRO는 기본적으로 리눅스를 탑재한 타겟 시스템을 가정하며 호스트인 전력 수집 및 분석 시스템에 대해서는 타겟 시스템과 통신 가능한 장치를 가져야 한다는 것 이외에 아무런 제한이 없으며 동작을 요약하면 다음과 같은 수행 흐름을 가진다.(그림 5 참조)

1. 타겟 시스템은 타겟 임베디드 S/W를 수행하면서 일정하게 규칙적인 인터벌로 타이머 인터럽트를 발생시키고 인터럽트 처리 루틴에서 GPIO를 통해 멀티미터에 트리거 입력을 주어 현재 전류를 샘플링하고 전력 수집 및 정합 시스템에 전류 샘플을 송신한다.
2. 타겟 시스템은 또한 인터럽트 처리 루틴에서 현재 PID(Process Identifier)와 PC(Program Counter) 및 성능 지수를 수집한다.
3. 임베디드 S/W가 수행 완료되면 메모리 버퍼에 수집된 S/W 수행 정보들을 파일 시스템에 저장하고 이 파일을 전력 수집 및 분석 시스템으로 송신한다.
4. 수집 및 분석 시스템은 타겟 S/W의 컴파일 시에

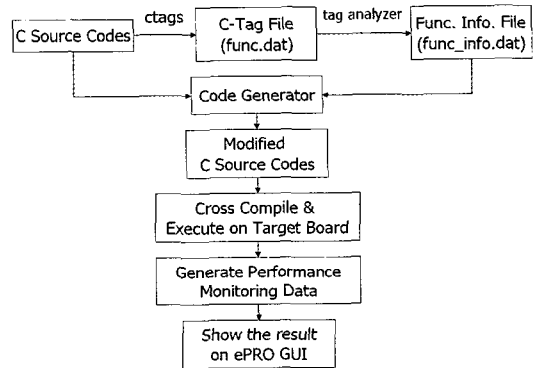


(그림 5) ePRO의 동작도

생성된 심볼 테이블과 S/W 수행 정보, 전류 샘플에서 계산된 전력 값들 및 성능 모니터에 의해 수집된 성능 지표 데이터를 비교 분석하여 연계하고 코드 수행에 따른 전력 소모 정보, 성능 지표 정보를 출력하고 간단한 전력 최적화 방안을 도출한다.

ePRO의 성능 분석 부분은 크게 태그 분석기, 코드 자동 생성기, 실행 경로 분석기로 구성되어 있으며 (그림 6)에서 보는 바와 같이 태그 분석기 및 코드 자동 생성기에 의한 성능 모니터링 과정을 거쳐 ePRO의 성능 분석기에 의해 태그 소프트웨어와 연계, 분석된다. 코드 자동 생성기는 태그 분석기가 생성한 데이터 파일과 타겟 프로그램의 원본 소스 코드(C 언어 수준)들을 입력으로 받아, 성능 모니터링 코드가 삽입된 수정된 코드(C 언어 수준)를 자동으로 생성하여 준다. 코드 자동 생성기는 타겟 프로그램의 원본 소스 코드를 분석하여, 타겟 프로그램에 구현되어 있는 모든 함수의 시작부와 종료부에 일련의 성능 모니터링 코드들을 자동으로 삽입하여 준다. 이를 통해 타겟 프로그램의 각 함수별로 성능 모니터링을 가능케 해준다.

코드 자동 생성기가 생성한 수정된 C 소스 코드를 리눅스 호스트인 수집 및 분석 시스템에서 크로스 컴파일한 후, 타겟 보드에 다운로드하여 타겟



(그림 6) 성능 모니터링 과정

보드 상에서 수행하면 원하는 성능 측정 결과를 얻을 수 있으며 비슷한 과정에 의해 실행 분석기에 의해 얻은 함수 호출 경로와 성능 모니터링 결과를 다시 수집 및 분석 시스템에 전송하여 함수별로 연계된 성능 분석 결과를 얻을 수 있다.

(그림 7)은 타겟 소프트웨어로서 선택된 FMFT (Frequency Modified Fourier Transform) 프로그램에 대해 ePRO를 사용하여 전력 측정 및 성능 모니터링을 수행하고 소스 코드의 함수 수준에 연계하여 분석하는 내용을 보인 것이다. FMFT는 주파수를 변환하여 푸리에 변환을 수행하는 공학용 프로그램이다. 그림에서 보는 바와 같이, phifun 함수가 전체 소모 에너지의 70% 이상을 차지하면서 수

Collecting	Analysis	Configuration	Execution Time(s)			
Source	Function	Energy (mJ)	Energy Ratio	Pf_11	Pf Ratio Cr.	Optimizatio
N/A	..init	63.07		N/A		
fmt.c:72	fmt	5,519.941		0.00070416		l-cache an...
fmt.c:431	window	488.337		0.55593234		
fmt.c:452	power	567.72		0.12534841		
fmt.c:479	four1	3,599.596		0.81034512		
fmt.c:531	bracket	79.256		0.01441366		
fmt.c:579	golden	47.264		0.00087842		
fmt.c:652	phifun	24,710.569		12.75582...		

(그림 7) 타겟 프로그램의 ePRO 수행 결과

행 시간의 80%이상을 차지함을 쉽게 알 수 있으며 이러한 정보를 바탕으로 phifun 함수가 매우 자주 불리는 함수이므로 많은 에너지를 소모함을 추론할 수 있다. 실제로 phifun 함수는 내부에서 불필요한 malloc 함수 및 free 함수를 교대로 호출하며 math 라이브러리 함수인 sin 함수와 cos 함수를 번갈아 호출한다. 최적화 과정으로서 불필요한 함수 호출을 제거하고 sin 함수와 cos 함수에 대해서는 고정 소수점을 사용하도록 lookup table을 사용함으로써 전체 성능과 에너지 면에서 약 4.4%와 5.4%의 개선을 가져올 수 있었다. 이러한 예는 바로 전력 측정 및 분석뿐만 아니라 성능 측정 및 분석이 최적화된 소프트웨어 개발에 유기적으로 필요함을 보여주는 좋은 예며 통합 분석 도구의 유용성이 확인되는 예다.

5. 결론

배터리로 동작하는 이동성 임베디드 시스템의 사용이 확산됨에 따라 멀티미디어 응용과 같이 실시간성을 가지는 다기능 및 고성능의 응용 소프트웨어의 사용 요구가 증가되고 있어 소프트웨어 수준의 저전력 기법의 개발이 매우 중요한 문제로 부각되고 있다. 이를 위해서는 소모 전력을 정확히 측정하고 소프트웨어에 연계하여 분석할 수 있는

반면에 다양한 이동 단말기 플랫폼 간에 이식성이 높고 분석 도구 재구현시 비용이 적게 드는 도구의 개발이 절실히 요구된다. 또한, 수행 시간과 같은 성능 지표와의 상호 연관 분석을 통해 성능 및 전력 최적화된 소프트웨어 개발에 대한 필요성이 제기되고 있다.

본 고에서는 기존 전력 분석 도구들을 살펴보고 이들을 확장하는 요구사항들에 대해 기술하였으며 확장된 전력 분석 도구에서의 요구 사항을 바탕으로 하여 서울대학교 임베디드시스템 연구실에서 개발한 임베디드 소프트웨어 전력 분석 도구들을 살펴보았다. 이 중에서도, ePRO는 전력뿐만 아니라 성능 지표를 측정하고 타겟 소프트웨어의 소스 코드와 연계 분석하는 통합 분석 도구로서, 연관 관계를 맺기 힘들었던 성능과 전력과의 관계를 효율적으로 파악할 수 있도록 해주며 최적화된 소프트웨어를 개발할 수 있도록 해주는 강력한 도구다. 향후 연구에서는 최적화 도구와의 연계가 쉽도록 하는 도구로서의 확장성을 보완할 예정이며 성능과 전력의 상호 연계 분석을 위해 많은 응용 프로그램들에 대한 분석 및 최적화 작업을 수행할 예정이다. 최종적으로는, 이러한 작업들을 기반으로 ePRO가 해당 소프트웨어에 대해 전력 또는 성능 면에서 필요한 최적화 방향을 효과적으로 제시할 수 있도록 할 예정이다.

〈Acknowledgements〉

본 연구는 대학 IT연구센터 육성지원사업의 연구결과로 수행되었으며 한국전자통신연구원지원 (# 1010-2004-0098)에 의해 수행되었다. 현재, ePRO는 한국전자통신연구원의 위탁과제로서, 코

드 크기 분석과 같이 또 다른 시스템 측정 지표에 대한 분석 및 연계 작업을 수행하고 있으며 최적화 도구와의 유연한 연계를 위해 Java 기반의 Eclipse 플랫폼 상에서 개발을 진행하고 있다.

참고문헌

[1] PowerMill, Synopsys Inc., http://www.synopsys.com/products/etg/powermill_ds.html.

[2] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations", Proc. of the 27th International Symposium on Computer Architecture, pp.83-94, 2000.

[3] N. Vijaykrishnan, M. Kandemir, M. J. Irwin, H. S. Kim, and W. Ye, "Energy-Driven Integrated Hardware-Software Optimizations Using SimplePower", Proc. of the 27th International Symposium on Computer Architecture, pp.95-106, 2000.

[4] T. Simunic, L. Benini, and G. De Micheli, "Cycle-Accurate Simulation of Energy Consumption in Embedded Systems", Proc. of the 36th Design Automation Conference, pp.867-872, 1999.

[5] V. Tiwari, S. Malik, and A. Wolfe, "Power Analysis of Embedded Software: A First Step Towards Software Power Minimization", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 2, No. 4, pp.437-445, Dec. 1994.

[6] T. K. Tan, A. Raghunathan, G. Lakshminarayana, and N. K. Jha, "High-level Software Energy Macro-modeling", Proc. of the 38th Design Automation Conference, pp.605-610, 2001.

[7] J. Flinn and M. Satyanarayanan, "Power Scope: A Tool for Profiling The Energy Usage of Mobile Applications", Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp.2-10, 1999.

[8] 한국전자통신연구원, 내장형 시스템의 파워 소모량 측정에 관한 연구 - 최종연구보고서, 2002. 11.

[9] D. Shin, H. Shim, Y. Joo, H.-S. Yun, J. Kim and Naehyuck Chang, "Energy Monitoring Tool for Low-Power Embedded Programs", IEEE Design and Test of Computers, Vol. 19, No. 4, July 2002.

[10] 한국전자통신연구원, 임베디드 S/W 소모 전력 분석 및 최적화 도구에 관한 연구 - 최종연구보고서, 2003. 11.

[11] W. Baek, Y.-J. Kim, and J. Kim, "ePRO: A Tool for Energy and Performance Profiling for Embedded Applications", Proc. of International SoC Design Conference (ISOCC), Seoul, Korea, October 25-26, 2004.

저자역력



김영진

1997년 서울대학교 전기공학부(학사)
1999년 서울대학교 전기공학부(석사)
1999년~2003년 ETRI 연구원
2003년 - 현재 서울대학교 전기·컴퓨터공학부(박사 과정)
관심분야 : 임베디드 시스템, 저전력 소프트웨어 기법, 성능
및 전력 분석 도구
이 메 일 : youngjk@davinci.snu.ac.kr



김지흥

1986년 서울대학교 계산통계학과(학사)
1988년 University of Washington 컴퓨터과학과(석사)
1995년 University of Washington 컴퓨터과학 및
공학과(박사)
1995년~1997년 미국 Texas Instruments 선임연구원
1997년 - 현재 서울대학교 전기·컴퓨터공학부 부교수
관심분야 : 임베디드 소프트웨어, 저전력 시스템,
멀티미디어 시스템, 컴퓨터 구조
이 메 일 : jihong@davinci.snu.ac.kr



백웅기

2003년 서울대학교 재료공학부/전기공학부(학사)
2003년 - 현재 서울대학교 전기·컴퓨터공학부(석사 과정)
관심분야 : 임베디드 시스템, 저전력 시스템, 컴파일러
최적화
이 메 일 : wkb@davinci.snu.ac.kr