

# Energy-aware Dalvik Bytecode List Scheduling Technique for Mobile Applications

Ko Kwang Man<sup>†</sup>

## ABSTRACT

An energy of applications had consumed through the complexed inter-action with operating systems, run-time environments, compiler, and applications on various mobile devices. In these days, challenged researches are studying to reduce of energy consumptions that uses energy-oriented high-level and low-level compiler techniques on mobile devices. In this paper, we intended to reduce an energy consumption of Java mobile applications that applied a list instruction scheduling for energy dissipation from dalvik bytecode which extracted Android dex files. Through this works, we can construct the optimized power and energy environment on mobile devices with the limited power supply.

**Keywords :** Java Mobile Applications, Dalvik Virtual Machine, Energy Consumption Optimization, Control and Data Flow Graph, Instruction Scheduling

## 모바일 어플리케이션을 위한 에너지-인식 달빅 바이트코드 리스트 스케줄링 기술

고 광 만<sup>†</sup>

## 요 약

모바일 기기에서 어플리케이션의 에너지 소비는 운영체제, 실행시간 환경, 컴파일러, 어플리케이션 등이 복잡한 상호 작용을 통해 이루어진다. 최근까지 에너지-지향적인 고수준 및 저수준의 컴파일러 기술을 적용하여 모바일 기기에서 어플리케이션의 에너지 소비를 줄이기 위한 노력이 진행되고 있다. 본 논문에서는 안드로이드 달빅에서 실행되는 dex 파일로부터 달빅 바이트코드를 추출한 후 에너지 소비 최적화를 위한 리스트 인스트럭션 스케줄링을 적용하여 어플리케이션의 에너지 소비를 줄이고자 한다. 이러한 연구는 급속도로 확산되고 있는 안드로이드 기반 어플리케이션이 전력 공급이 제한적인 모바일 환경에서 최적화된 전력·에너지 소비할 수 있는 환경을 구축하는 데 활용할 수 있다.

**키워드 :** 자바 모바일 어플리케이션, 달빅 가상기계, 에너지 소비 최적화, 제어 및 자료 흐름 그래프, 인스트럭션 스케줄링

## 1. 서 론

인간들이 이동하는 횟수와 시간이 증가하면서 지난 몇 년간 모바일 기기의 사용이 폭발적으로 증가하고 있다. 특히, 최근에 등장한 스마트폰은 전통적인 휴대폰보다 고해상도 디스플레이, 향상된 처리 능력, 큰 저장 공간, 텍스트와 멀티미디어 메시징, 오디오 및 비디오 플레이어, 화상 회의, GPS 서비스, 향상된 근거리 무선 통신 등과 같은 유용한 특징을 가지고 있다.

안드로이드는 모바일 기기를 위한 운영체제, 리눅스 커널, 미들웨어 등을 포함하고 있는 소프트웨어 패키지로서, 안드로이드 어플리케이션은 레지스터 기반 가상머신인 달빅(dalvik) 인스턴스와 함께 자신의 프로세스에서 실행되며 달빅은 여러 인스턴스가 효율적으로 실행될 수 있도록 구현되어 있다. 달빅에서 Java 어플리케이션(\*.java)을 실행하기 위해서는 클래스 파일(\*.class)을 최소화된 메모리 영역을 차지하는 최적화된 Dalvik Executable(\*.dex) 형식으로 변환한다.

전력·에너지의 최적화 소비 관리는 모바일 기기를 위한 아키텍처 설계에서 중요한 문제이다. 따라서 고성능, 저전력 아키텍처의 설계를 위해 소프트웨어적인 실험, 전력·에너지 소비 최적화에 대한 다양한 기술 제안과 평가 방법이 필

※ 이 논문은 2012년도 상지대학교 교내 연구비 지원에 의한 것임.

† 종신회원: 상지대학교 컴퓨터정보공학부 교수

Manuscript Received: May 26, 2014

First Revision: May 27, 2014

Accepted: May 28, 2014

\* Corresponding Author: Kwangman KO(kwangman.ko@gmail.com)

요하다. 모바일 기기에서 어플리케이션의 에너지 소비는 운영체제, 실행시간 환경, 컴파일러, 어플리케이션 등이 복잡한 상호 작용을 통해 이루어진다. 최근까지 에너지-지향적인 고수준 및 저수준의 컴파일러 기술을 적용하여 어플리케이션의 에너지 소비를 줄이기 위한 노력이 연구로서 에너지-지향 고수준, 저수준의 컴파일러 최적화[1, 2], 에너지 소비 효율성을 위한 코드 선택[3], 레지스터 할당[4], 에너지 소비를 고려하여 인스트럭션의 실행 순서를 재구성하는 스케줄링 연구[5]도 활발하게 진행되고 있다.

이 논문에서는 모바일 기기의 에너지 소비를 최적화하기 위한 소프트웨어적인 접근연구로서, 안드로이드 달빅에서 실행되는 dex로부터 달빅 바이트코드를 추출한 후 인스트럭션 스케줄링을 적용하여 어플리케이션의 에너지 소비를 줄이고자 한다. 이를 위해, 첫째, Java 어플리케이션에 대해 컴파일러, dx 도구를 활용하여 dex 파일을 생성한다. 둘째, dex 파일에 대해 기본 블록단위로 제어 및 자료 흐름 그래프를 구성한 후 [6]에서 제시한 에너지 인식 인스트럭션 스케줄링 알고리즘을 적용한다. 마지막으로, 에너지 스케줄링을 완료한 dex 파일을 undex 및 jad 도구를 활용하여 Java 파일을 생성한 후 실행을 통해 에너지 소비 효율성을 증명한다. 이러한 연구는 급속도로 확산되고 있는 안드로이드 기반 어플리케이션이 전력 공급이 제한적인 모바일 환경에서 최적화된 전력·에너지 소비할 수 있는 환경을 구축하는데 활용할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 기존의 연구들을 살피고, 3장에서는 본 논문에서 제안하는 전체 시스템 구성과 에너지 소비 효율성을 위한 시스템 동작과정에 대해 설명한다. 또한 실질적인 예제를 통해 에너지 소비 효율성을 위한 스케줄링을 위한 세부내용에 대해 설명한다. 마지막으로, 실험을 통해 제안하는 방법의 성능상의 이점을 보이고, 4장에서 결론 및 향후 연구내용에 대해 기술한다.

## 2. 관련 연구

소프트웨어 수준에서 전력·에너지 소비의 최적화와 모니터링을 위해서는 타깃 아키텍처에서 주어진 어플리케이션이 소비하는 전력 소비를 정확히 측정하고 분석할 수 있는 모델링 기술이 필요하다. 고수준 인스트럭션-수준에서 하나의 인스트럭션이 실행될 때 소비되는 에너지 소비량(base cost)과 하나 이상의 인스트럭션이 실행될 때 소비되는 에너지 소비량(inter-instruction cost)을 반복적인 시뮬레이션을 통해 측정된 전류(I)와 전압(Vcc)을 이용하여 전력( $P=I \times V_{cc}$ ) 소비량을 계산한다. 이를 기반으로, 어플리케이션 실행시간(T)에 의해 소비되는 에너지 소비량( $E=P \times T$ ,  $E=I \times V_{cc} \times N \times \tau$ )이 계산되며, 실행시간( $T=N \times \tau$ )은 클럭수(N)와 클럭주기( $\tau$ )를 이용하여 결정한다.

인스트럭션 수준에서 전력·에너지 소비를 모델링한 후 이를 기반으로 어플리케이션의 에너지 소비 최적화를 위한

스케줄링 연구는 명령어의 전력 소모량을 우선순위로 고려하는 Cold 스케줄링, VLIW 인스트럭션에서 저전력 소비를 위한 스케줄링, 파이프라인 프로세서를 위한 에너지 소비 최적화 스케줄링, 어플리케이션에 대한 DAG를 구성한 후 Top-down, Bottom-up, Look-ahead 방식으로 순회하면서 우선순위에 따른 리스트를 구성한 후 스케줄링[2]하는 기법 등이 제안되었다. 또한 임베디드 시스템의 전력 소비를 줄이기 위해 어플리케이션의 DAG로부터 자료 및 제어 흐름 그래프와 인스트럭션 수준의 전력 소비 정보를 참고하여 전력 소비 테이블(PDT)을 구성한 후 이를 활용하여 명령어 실행 순서를 재조정하는 연구가 수행되었다[6].

최근에는 어셈블리코드 또는 목적코드를 역어셈블하여 최적화를 수행하는 Post-pass 최적화 기법과 Scratch-pad 메모리를 이용하여 전력 최소화를 위한 연구가 진행되었으며 사용하지 않는 장치·요소를 휴면 상태로 전환하고 컴파일러가 프로그램 분석을 통해 휴면 장치·요소를 운영체제에 정보로 전달하는 기법과 임베디드 또는 모바일 장치가 근거리 통신을 이용하여 지역적으로 가까이 있는 서버의 자원을 빌려 쓸 수 있도록 하여 CPU, 메모리 등의 전력·에너지에 관한 소모 제한을 극복하려는 연구(offloading)가 시도되고 있다. 국내에서의 연구시도는 임베디드 프로세서와 프로그램의 저전력 및 에너지 관리를 위한 최적화 기법, 저전력 관리 및 분석 시스템 개발에 관한 연구 결과, 문서가 인용되고 있지만 전반적인 연구 환경은 매우 열악한 상태이다.

## 3. 에너지-인식 달빅 바이트코드 리스트 스케줄링

### 3.1 달빅 바이트코드 스케줄링을 위한 시스템 구성

본 논문에서는 Java 어플리케이션(\*.java)의 에너지 소비를 최적화하기 위해 그림 1과 같다.

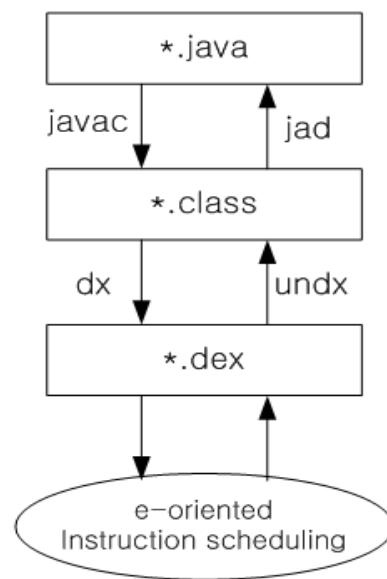


Fig. 1. System for Energy-aware Scheduling

Java 컴파일러를 통해 달빅 클래스 파일을 생성한 후 dxx 도구를 활용하여 dex 파일의 기본블록에 대해 에너지 소비 효율적인 인스트럭션 스케줄링을 적용한다. Java 어플리케이션의 에너지 소비 효율성을 증명하기 위해 dex 파일을 기본블록 단위로 구성한 후 제어 흐름 그래프, 자료 흐름 그래프를 구성한 후 기본블록 내에서 [6] 알고리즘을 응용하여 에너지 소비 효율적인 인스트럭션 스케줄링을 적용한다.

### 3.2 에너지-인식 달빅 바이트코드 리스트 스케줄링

이 논문에서 에너지 소비 효율성을 목적으로 인스트럭션 스케줄링을 실험하고 효율성을 증명하기 위해 그림 2와 같은 Java 어플리케이션에 대해 dex 파일을 생성한 후 기본블록(BB1)에 대한 제어 및 자료 흐름 그래프를 구성한다.

```
public static long sumArray(int[] arr) {
    long sum = 0;
    for(int i : arr) {
        sum += i;
    }
    return sum;
}
0007: if-ge v0, v2, 0010
0009: aget v1, v8, v0
000b: int-to-long v5, v1
000c: add-long/2addr v3, v5
000d: add-int/lit8 v0, v0, #int 1
000f: goto 0007
```

Fig. 2. Dalvik DEX File

본 논문에서 적용하는 에너지-지향 인스트럭션 스케줄링 기법은 스케줄링이 가능한 인스트럭션 i를 중심으로 스케줄링을 결정한 후에 인스트럭션 i로부터 선택 가능한 인스트럭션 중에서 최소의 비용을 갖는 노드 j를 결정한다. 다음 단계에서 인스트럭션 j는 스케줄링이 선택 가능한 인스트럭션 후보 중에서 최소의 비용을 갖는 인스트럭션을 결정한다[1].

```
//Energy-oriented instruction scheduling[6]
schld = ∅
while can-sch = ∅ do
    if schld = ∅ then
        n = can-sch[1]
    else
        min-energy = +∞
        for each j ∈ can-sch do
            if energylast,j < min-energy then
                min-energy = energylast,j
                n=j
        last = n
        schld ⊕= [n]
        can-sch -= {n}
        for each i ∈ succ(n) do
            if ∃m ∈ pred(i) ∃j such that schld[j] = m then
                can-sch ∪= {i}
    return (schld)
```

Fig. 3. Energy-aware Instruction Scheduling Algorithm

### 3.3 실험 및 분석

Quick.java에 대해 dex 파일을 생성한 후 기본블록에 내에서 에너지 소비 효율성을 고려한 자료 흐름 그래프를 구성한 후 그림 4와 같은 에너지-인식 스케줄링 달빅 코드를 생성하였다.

```
public class Quick {
    public static void main(String[] args) { //BB1
        int[1000] array = new int();
        quickSort(array, 0, array.length - 1);
        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i] + " ");
        }
    }
}
```

```
.class QuickSort
.super java/lang/Object
.method public static quickSort(III)V
.limit registers 7
.line 3 sub-int v0,v6,v5
    if-gtz v0,l356
...
l356:
.line 6 aget v2,v4,v6
.line 7 add-int/lit8 v0,v5,255
    move v1,v0
    move v0,v6
l362:
.line 10 add-int/lit8 v1,v1,1
    aget v3,v4,v1
    if-lt v3,v2,l362
l36e:
.line 11 if-le v0,v5,l37e
    add-int/lit8 v0,v0,255
    aget v3,v4,v0
    if-gt v3,v2,l36e
l37e:
.line 12 if-lt v1,v0,l39e
.line 17
.line 18
    add-int/lit8 v0,v1,255
...
.end method
.method static swap(III)V
.limit registers 5
parameter[2] : v4 (I)
.line 24 aget v0,v2,v3
.line 25 aget v1,v2,v4
    aput v1,v2,v3
.line 26 aput v0,v2,v4
.line 27 return-void
.end method
```

Fig. 4. Energy-aware Dalvik Bytecode List Scheduling

전체 Java 어플리케이션의 에너지 소비 효율성을 실험하고 분석하기 위해 Swap.java와 Quicksort.java에 대해 입력 데이터를 10,000개, 100,000개, 1,000,000개에 대해 반복 실험하여 그림 5 (a), (b)와 같은 결과를 산출하였다. 실제 에너지 소비 효율성을 고려한 스케줄링을 통해 5~11% 정도의 에너지 소비 절감을 확인하였다.

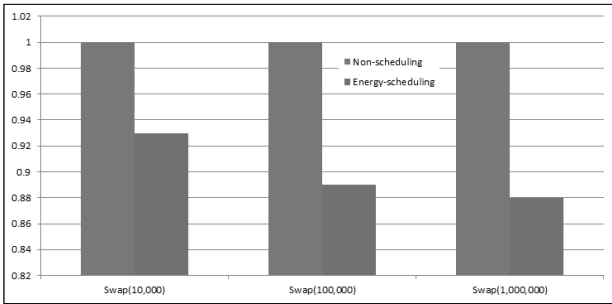


Fig. 5(a). Swap: Experimental Results of Energy Consumption

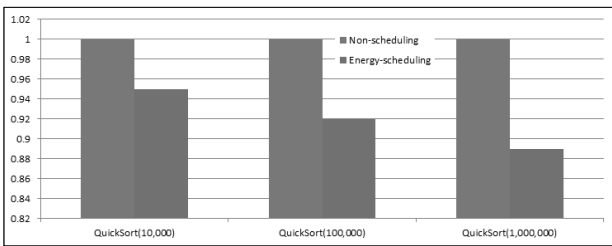


Fig. 5(b). Quicksort: Experimental Results of Energy Consumption

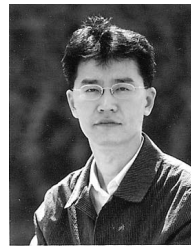
#### 4. 결론

전력·에너지의 최적화 소비 관리는 모바일 기기를 위한 아키텍처 설계에서 중요한 문제이다. 따라서 고성능, 저전력 아키텍처의 설계를 위해 소프트웨어적인 실험, 전력·에너지 소비 최적화에 대한 다양한 기술 제안과 평가 방법이 필요하다. 최근까지 에너지-지향적인 고수준 및 저수준의 컴파일러 기술을 적용하여 어플리케이션의 에너지 소비를 줄이기 위한 연구로서 에너지-지향 고수준, 저수준의 컴파일러 최적화 연구가 진행되고 있다.

이 논문은 모바일 기기의 에너지 소비를 최적화하기 위한 소프트웨어적인 접근연구로서, 안드로이드 달빅에서 실행되는 dex로부터 달빅 바이트코드를 추출한 후 리스트 인스턴스 스케줄링을 적용하여 어플리케이션의 에너지 소비를 줄이는 연구결과로서 Swap.java와 Quicksort.java에 대해 실제 에너지 소비 효율성을 고려한 스케줄링을 통해 5~11% 정도의 에너지 소비 절감을 확인하였다. 이러한 연구는 급속도로 확산되고 있는 안드로이드 기반 어플리케이션이 전력 공급이 제한적인 모바일 환경에서 최적화된 전력·에너지 소비할 수 있는 환경을 구축하는 데 활용할 수 있다.

#### Reference

- [1] A. Parikh, Soontae Kim, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, "Instruction Scheduling for Low Power", Journal of VLSI Signal Processing, Vol.37, pp.129-149, 2004.
- [2] Kyuwon Choi, Abhijit Chatterjee, "Efficient Instruction-Level Optimization Methodology for Low Power Embedded Systems", ISSS 2001, 2001.
- [3] R. Leapers, "Compiler Design Issues for Embedded Processors", IEEE Design & Test of Computers, pp.2-9, 2002.
- [4] Timothy M. Jones et al., "Energy-Efficient Register Caching with Compiler Assistance", ACM Transactions on Architecture and Code Optimization, Vol.6, No.4, Article 13, 2009.
- [5] Meikang Qiu et al., "Energy-aware Loop Scheduling and Assignment for Multi-core, Multi-functional Unit Architecture", Journal of Signal Processing System, Vol.57, pp.363-379, 2009.



#### 고 광 만

e-mail : kwangman.ko@gmail.com

1991년 2월 원광대학교 컴퓨터공학과 (공학사)

1993년 2월 동국대학교 컴퓨터공학과 (공학석사)

1998년 2월 동국대학교 컴퓨터공학과 (공학박사)

1998년 3월~2001년 8월 광주여자대학교 컴퓨터학과 전임강사

2001년 9월~현재 상지대학교 컴퓨터정보공학부 교수

관심분야: Energy-aware Compiler Technology, Eco-Mobile Computing