

# 실시간 임베디드 시스템을 위한 메모리 시스템 성능 최적화 기법

권용인, 조두산, 이종원, 김용주, 윤중희, 박상현, 백윤희  
 서울대학교 공과대학 전기컴퓨터 공학부  
 e-mail : {yikwon,dscho,jwlee,shpark}@optimizer.snu.ac.kr, ypaek@snu.ac.kr

## An Optimization Technique in Memory System Performance for RealTime Embedded Systems

Yongin Kwon, Doosan Cho, Jongwon Lee, Yongjoo Kim, Jonghee Youn, Sanghyun Park, Yunheung Paek  
 School of Electrical Engineering, Seoul National University

### 요 약

통상 하드웨어 캐시의 크기보다 수십에서 수백배 큰 크기의 데이터를 랜덤하게 접근하는 경우 낮은 메모리 접근 지역성 (locality)에 기인하여 캐시 메모리 성능이 급격히 저하되는 문제를 야기한다. 예를 들면, 현재 보편적으로 사용되고 있는 차량용 General Positioning System (GPS) 프로그램의 경우 최대 32 개의 위성으로부터 데이터를 받아 수신단의 위치를 계산하는 부분이 핵심 모듈 중의 하나이며, 이는 전체 성능의 50% 이상을 차지한다. 이러한 모듈에서는 위성 신호를 실시간으로 받아 버퍼 메모리에 저장하며, 이때 필요한 데이터가 순차적으로 저장되지 못하기 때문에 랜덤하게 데이터를 읽어 사용하게 된다. 결과적으로 낮은 지역성에 기인하여 실시간 (realtime)안에 데이터 처리를 하기 어려운 문제에 직면하게 된다. 통상의 통신 응용의 알고리즘 상에 내재된 (inherited) 낮은 메모리 접근 지역성을 개선하는 것은 알고리즘 상에서의 접근을 요구한다. 이는 높은 비용이 필요로함으로 본 연구에서는 사용되는 데이터 구조를 변환하여 지역성을 높이는 방향으로 접근하였다. 결과적으로 핵심 모듈에서 2 배, 전체 시스템 성능에서 14%를 개선할 수 있었다.

### 1. 서론

통상의 통신 어플리케이션을 위한 실시간 임베디드 시스템은 작은 크기의 하드웨어 캐시(2~16KB)를 갖고 있으며, 이보다 수백배 이상 크기의 데이터를 랜덤하게 접근하여 사용한다. 결과적으로 캐시 메모리 이용률(hit-ratio)가 낮아 성공적인 시스템 개발을 하기 어렵다. 본 연구에서는 GPS 어플리케이션을 대상으로 낮은 데이터 지역성 문제를 해결하여 시스템 성능을 높이고자 한다.

오늘날 많은 임베디드 시스템은 한정된 하드웨어 자원 (resource) 만을 가지고 설계 및 제작되고 있다. 가격 경쟁력을 높이기 위하여 시스템의 하드웨어 비용을 낮추는 대신 요구되는 성능기준을 맞추기 위한 부담은 소프트웨어에 지워지게 된다. 소프트웨어 최적화를 통하여 성능개선을 하는 경우 가장 중요한 부분 중에서 하나가 메모리 관련 모듈을 최적화하는 것이며, 이 중에서 하드웨어 캐시 이용률을 높이는 부분이 가장 효과적인 기법으로 사용되고 있다. 제안하는 기법은 새로운 접근으로 기존기법들과는 차이가 있다.

### 2. 실험 환경

시스템 플랫폼으로 ARM920T 를 이용하였다. ARM9 은 5-stage integer pipeline 이고, 폰노이만(von Neumann)

구조로 I-CACHE, D-CACHE 를 포함하고 있으며, 각각 16KB 이다. 그림 1 은 ARM920T 다이어그램이다. OS 는 WinCE 6.0, MS ARM9 compiler 를 사용하였다.

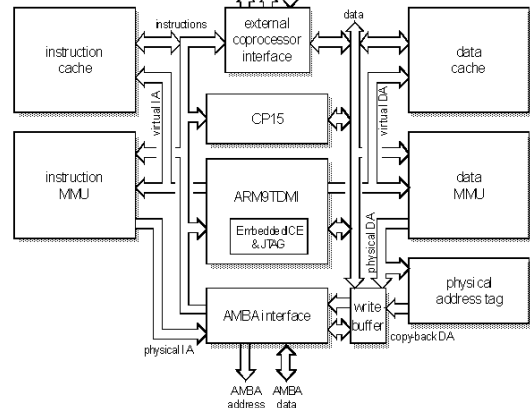


그림 1 ARM920T 의 구조

실험에 사용한 프로그램은 차량용 GPS 응용으로 256KB 크기의 데이터를 랜덤패턴으로 접근하여 처리한다. 한번에 사용하는 데이터는 캐시 크기에 비하여 16 배가 크다. 그림 2 는 메모리 접근에 사용되는 메

인 코드를 나타내고 있다.

```
foo(){
    for(i = 0 ; i < n ; i++)
        print map[rand*65536]
}
```

그림 2 핵심 모듈의 메모리 접근 코드

3. 데이터 접근 지역성 (locality) 분석

캐시성능 개선을 위해서는 우선 데이터 접근 지역성과 성능 사이의 상관관계를 파악해야 한다. 이것을 파악하기 위해서 우리는 (modulo operation) ‘%’을 사용하였다. 즉, 인위적으로 사용되는 데이터의 지역성을 높이고 이를 통하여 가능한 성능 개선 정도를 확인하였다.

```
foo(){
    for(i = 0 ; i < n ; i++)
        print map[(rand*65536)%m] }
    그림 3 %연산 적용 코드
```

표 1. Modulation 크기에 따른 데이터처리량 증감

Mod operation with 3	2.66second(데이터처리량)
Mod operation with 10	2.68
Mod operation with 100	2.62
Mod operation with 1000	2.38
Mod operation with 5000	2.02
No Mod operation	1.67

그림 3 에 %연산을 적용한 코드를 나타내었다. 해당 실험에서 m 의 크기를 3,10,100,1000,5000 까지 변경하여 각각의 효과를 측정하였다. %연산을 적용하지 않은 경우 1.67 초를 나타내었다. 이 수치 의미는 현재 구현된 GPS 시스템이 1 초 동안 1.67 초 분량의 데이터를 처리할 수 있다는 것을 나타낸다. 구현된 GPS 시스템이 정상 동작하기 위해서는 3 초 이상의 성능을 얻어야 한다. 결론적으로 목표 성능을 얻기 위해서는 “modulo 3”이상 수준의 지역성을 얻을 수 있도록 해야하는데 이는 단일기법으로는 불가능하다.

4. 데이터 압축을 통한 지역성 개선

GPS 응용의 알고리즘 수정보다 적은 비용으로 필요한 성능을 얻기 위하여 사용되는 데이터를 압축하여 사용하도록 하였다. 데이터 크기를 줄임으로서 지역성을 높일 수 있고 이로 인하여 캐시 성능이 개선됨으로 시스템 성능 개선을 유도할 수 있었다.

데이터 압축시에 무손실 압축을 해야 한다. 손실 압축의 경우 복원비용이 크기 때문에 전체 성능이 복원 연산량 증가로 인하여 떨어질 수 있기 때문이다. 제안하는 기법은 다음과 같다.

통상 통신 응용에서 사용되는 데이터는 테이블로 (ex, routing table, ip table, location table, etc.) 구성되어 있다. 테이블은 두개 필드로 분류할 수 있는데 특정한

데이터를 담은 주소 필드와 값 필드로 구분할 수 있다. GPS 코드의 경우도 위성으로부터 받은 데이터를 통하여 이를 바탕으로 위치를 검색하는 테이블을 사용하게 되고 위치를 찾을 때 인코딩된 주소를 사용한다. 따라서 주소 필드와 데이터 필드를 분류하여 값을 확인하여 보면 중복된 부분이 많음을 확인할 수 있다. 다양한 색인 값 (index)을 여러 위성으로부터 받아 한 위치를 검색하는 특성에 기인하여 발생하는 현상이다. 이러한 현상을 값의 지역성 (value locality) [2] 이라 부른다. 이러한 경우 주소와 값 테이블을 분리하여 전체 데이터 크기를 압축하는 것이 가능하다.

예를 들어, GPS 코드의 경우 사용하는 테이블은 65535 개의 데이터 중 실제 구분되는 값은 617 개로 구성됨을 확인하였다. 따라서 데이터 테이블을 16bit 주소 테이블 (Address Table: 65536 개의 2byte array, 128kbyte)와 32bit data table(617 개의 4byte array, 2.5kbyte) 두 개로 분리할 수 있는 것이다. 두 데이터의 총 용량은 130.5kbyte 로 약 1/2 로 size 를 줄일 수 있다.

```
foo(){
    for(i = 0 ; i < n ; i++)
        print data[address[rand*65536]] }
    그림 4 수정된 코드 (130KB 테이블사용)
```

그림 4 와 같이 메모리 접근은, 주소 테이블에 1 회, 값 테이블에 1 회, 총 2 회 증가하였지만 총 데이터의 크기가 반으로 줄어들어 지역성이 증가하였기에 캐시 성능증가에 따른 시스템 성능 개선효과로 데이터 처리량이 1.67 초에서 1.87 초로 평균 14% 증가하였다. 적용된 기법 중 단일 기법으로 가장 큰 개선 효과를 얻었다.

5. 결론

실시간 임베디드 시스템 개발시 많은 경우 시스템의 성능을 결정하는 핵심 부분은 메모리 접근 지연에 관련되어 있다. 설계 단계에 있어서 이러한 문제는 하드웨어 변경 및 추가로 가능하나 하드웨어 부분 개발이 완료되어 있는 시점에서 메모리 시스템 성능 개선은 용이한 해결방안이 제시되어 있지 못한 실정이다. 결과적으로 소프트웨어 성능 향상을 위하여 다양한 컴파일러 최적화 기법이 동원 될 수 있으나 알고리즘 단계의 지원 없이는 성능개선에 한계가 있다. 본 연구에서는 이러한 경우 타겟 응용의 입력으로 사용되는 데이터의 특성을 분석하고, 지역성 (locality)를 높일 수 있도록 데이터를 재구성하여 전체 시스템 성능을 개선할 수 있음을 실험적으로 보여주고 있다.

참고문헌

[1] Cache memory design for network processors, Tzi-cher Chiueh, Prashant Pradhan, IEEE HPCA, 2000  
 [2] Frequent value locality and its applications, Jun Yang, Rajiv Gupta, ACM TECS, 2002

### Acknowledgement

본 연구는 교육과학기술부/한국과학재단 우수연구센터육성사업 (R11-2008-007-01001-0), 지식경제부 출연금으로 ETRI, SoC 산업진흥센터에서 수행한 ITSoC 핵심설계인력양성사업, 서울시 산학연 협력사업, 2008 년도 정부(교육과학기술부)의 재원으로 한국과학재단의 국가지정연구실사업(R0A-2008-000-20110-0), 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업(IITA-2008-C1090-0801-0020), 지식경제부 및 정보통신연구진흥원의 IT 원천기술개발사업[과제관리번호: 2006-S-006-02, 과제명: 유비쿼터스 단말용 부품/모듈]의 지원을 받아 수행되었습니다.