

## Banked Memory System 에서 효율적인 전력관리를 위한 KVM 의 힙 메모리 관리체계 설계

최인범<sup>o</sup>, 강희성, 정명조, 이철훈

충남대학교 컴퓨터공학과

(ibchoi<sup>o</sup>, hskang, mjjung, chlee)<sup>o</sup>@ce.cnu.ac.kr

### A Design of KVM Heap Memory Management for Effective Power Management in Banked Memory System

In-Bum Choi<sup>o</sup>, Hui-Sung kang, Myung-Jo Jeong, Cheol-Hoon Lee  
Dept. of Computer Engineering, Chungnam National Univ.

#### 요 약

최근에는 자원이 제한적인 시스템을 위한 embedded JVM 환경에서 자바 응용 프로그램을 수행할 때 embedded JVM 의 성능에 따른 performance 뿐만 아니라 에너지 소비를 줄이는 일이 크게 대두되고 있다. 메모리에서 사용되는 에너지는 메모리에 접근할 때마다 소비되는 dynamic energy 와 메모리에 파워가 들어와 있을 때 항상 소비되는 leakage energy 로 구분할 수 있다. embedded 환경을 고려하지 않았던 이전에는 leakage energy 가 중요한 부분으로 인식되지 않았지만, 현재는 dynamic energy 못지 않게 중요한 부분으로 인식되고 있다.

본 논문에서는 Banked Memory System 을 사용하는 임베디드 JVM 의 환경하에서 leakage energy 를 효과적으로 줄일 수 있는 KVM 의 힙 메모리 관리체계를 설계하였다.

#### 1. 서론

임베디드 시스템의 소형화는 기능의 축소뿐만 아니라 배터리의 크기에도 영향을 주게 되었다. 따라서 최근에는 이러한 배터리의 사용시간이 임베디드 시스템의 성능을 결정하는 주요 요인으로 작용하고 있기 때문에, 배터리 사용시간을 늘이는 것이 매우 중요하다.

하드웨어 측면에서의 배터리 성능향상 못지 않게 소프트웨어 측면에서도 정해진 배터리의 성능을 극대화하여 사용할 수 있는 방법이 필요하며, 이를 위하여 다양한 분야에서 많은 방법들이 연구되고 있다.

자바는 플랫폼 독립성, 보안성, 네트워크 이동성, 실행코드의 재사용성, 작은 실행 파일 크기 등의 장점을 가지고 있기 때문에 제한된 자원을 사용하는 임베디드 시스템에서 보편적인 플랫폼으로 채택, 사용되고 있다. 이렇게 임베디드 시스템에서 자바 플랫폼을 채택하게 되면, 자바의 장점인 효율적인 메모리 관리를 통하여 메모리에서 사용되는 전력을 줄여 배터리 사용시간을 늘일 수 있게 된다.

본 논문은 여러 가지 임베디드 시스템들 중, 하드웨어적으로 Banked Memory 를 사용하는 시스템에서 KVM 의 효과적인 힙 메모리 관리를 통하여 메모리에서 사용하는 전력소모를 줄이고, 배터리의 수명을 늘일 수 있는 방법에 대하여 기술한다.

본 논문에서는 2 장에서 관련 연구로 메모리 사용시 소비되는 에너지의 종류와, Banked Memory

System, Supply-gating 알고리즘, KVM 에서 사용하는 기본 가비지 콜렉션에 대한 내용을 살펴보고, 3 장에서 Banked Memory 를 사용하는 시스템에서 효율적인 전력관리를 위한 KVM 의 힙 메모리 관리 방법에 대한 설계를, 4 장에서 결론 및 향후 연구 과제를 기술한다.

#### 2. 관련 연구

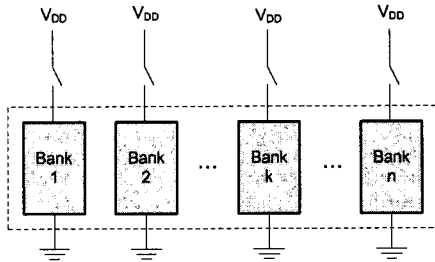
##### 2.1 메모리 사용시 소비되는 에너지

임베디드 시스템에서 메모리를 사용하는 경우, 메모리에 데이터가 존재하게 되면 해당 데이터를 유지하기 위해 전력을 소모하게 된다. 이렇게 메모리를 사용할 때 소비되는 에너지는 두 가지 종류가 있는데, 직접적으로 메모리 상의 데이터에 읽기, 쓰기 등의 접근을 할 때 소비되는 에너지를 Dynamic Energy, 메모리에 데이터가 사용되지 않는 상태로 저장되어 있을 때 소비되는 에너지를 Leakage Energy 라고 한다. Dynamic Energy 는 메모리에 저장된 데이터를 사용하기 위해 접근할 때마다 항상 소비되는 에너지로 전력 소비를 줄이기 위해 고려할 수 없는 에너지인 반면에, Leakage Energy 는 더 이상 사용되지 않는 데이터를 유지하는데 소비되는 경우도 있으므로, 불필요한 데이터를 메모리 상에 유지하기 위해 소요되는 Leakage Energy 를 줄이게 된다면 전체적인 전력 소비 감소에 큰 영향을 줄 수 있게 된다.

<sup>o</sup> 본 연구는 산업자원부 중기거점과제 연구비 지원에 의한 것임

2.2 Banked Memory System 구조

Banked Memory System 은 사용하는 메모리를 Bank 단위로 나누어 각 Bank 별로 소모하는 전력을 관리하는 구조이다. Memory Bank 의 구조는 [그림 1] 에서 보는 바와 같이 전체 메모리를 각각의 Bank 로 나누고 각 Bank 별로 사용하는 전력을 관리할 수 있도록 하는 방식이다.



[그림 1] Banked Memory System 구조

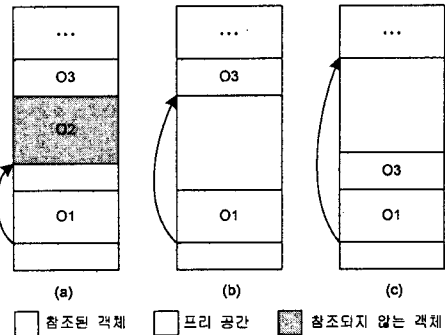
2.3 Supply-gating 알고리즘

Supply-gating 알고리즘은 Memory Bank 를 사용하는 시스템에서 임의의 Memory Bank 에 들어있는 데이터가 더 이상 필요하지 않을 경우 해당하는 Memory Bank 의 전력 공급을 차단하는 알고리즘이다. 이 알고리즘은 메모리 사용시 소비되는 두 가지 에너지 중 Leakage Energy 를 줄이기 위한 방법이다. Supply-gating 알고리즘을 사용하게 되면 메모리의 Leakage Energy 를 줄일 수 있는 장점이 있는 반면에, 임의의 Memory Bank 에 전력 공급을 차단하게 되면 해당 Memory Bank 의 상태정보를 모두 잃어버리기 때문에 다시 Memory Bank 를 활성화 하기 위해서는 수백 번의 Processor Cycle 이 필요하게 된다. 이는 임베디드 시스템의 전체적인 성능에 매우 큰 오버헤드를 발생시킬 수 있기 때문에 Memory Bank 의 제어를 위하여 Supply-gating 알고리즘을 적용하는데 신중을 기해야 한다.

2.4 KVM 의 기본 가비지 콜렉션

SUN 사의 CLDC1.0.4 를 기본으로 하는 KVM 에는 마크-회수(압축) 방법을 이용하여 가비지 콜렉션이 구현되어 있다. [그림 2]의 (a)와 (b)에서 보는 바와 같이 새로운 객체나 배열을 위한 메모리 할당 시 더 이상 힙에 메모리를 할당하지 못할 경우 마크-회수 방법을 이용하여 참조되지 않는 오브젝트의 Mark bit 을 셋팅한 후 이를 수거하여 프리 리스트에 추가하게 된다. 이렇게 추가된 프리 리스트에서 할당할 적당한 메모리 공간을 받아 사용하게 되는데, 만약 프리 리스트에 적당한 크기의 빈 공간이 없을 경우에는 [그림 2]의 (c)에서 보는 바와 같이 압축을 실행하여 빈 메모리 공간을 모아 전체적인 큰 하나의 공간으로 만들어 할당하게 된다. 이 방법은 간단하고 오버헤드가 적은 장점이 있으나, 시간이 지남에 따라 힙에 단편화 현상

을 발생시켜 메모리의 효율성을 점점 감소시키는 단점이 있다.



[그림 2] 마크-회수(압축) 알고리즘

3. Banked Memory 를 사용하는 시스템에서 효율적인 전력관리를 위한 KVM의 힙 메모리 관리 방법 설계

본 논문에서는 2 장에서 살펴본 내용을 토대로 KVM 을 사용하는 임베디드 자바환경에서 효율적인 전력관리를 위한 KVM 의 힙 메모리 관리 방법을 설계하였다. Banked Memory 를 사용하는 시스템에서는 전체 메모리를 각각의 Memory Bank 단위로 나누어 전력을 관리할 수 있기 때문에, 전체적인 시스템의 전력소모를 줄일 수 있는 장점을 가지고 있다. 본 논문에서는 Memory Bank 에 대한 제어를 위해 Supply-gating 알고리즘을 사용하여 더 이상 필요하지 않는 데이터를 포함하고 있는 Memory Bank 의 전력 공급을 차단하는 방법을 사용하여 전력관리 방법을 설계하였다.

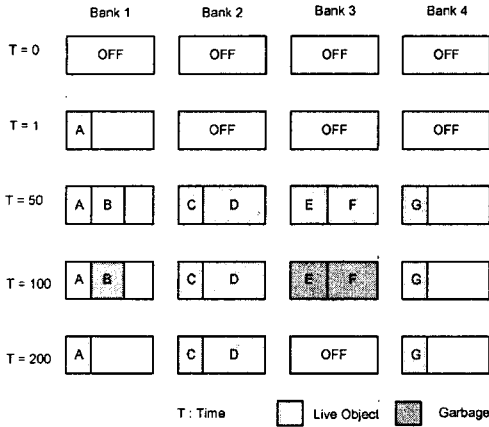
일반적인 임베디드 시스템에서 전력관리를 위하여 Supply-gating 알고리즘을 사용할 경우 메모리에 저장되어 있던 상태정보를 잃어버릴 위험이 있기 때문에, Supply-gating 방식을 적용하기 매우 어렵다. 하지만 임베디드 자바환경에서는 더 이상 필요 없는 메모리상의 데이터를 수거하는 방식을 사용하기 때문에 Supply-gating 알고리즘을 적용하기에 적합하다.

3.1 KVM 의 가비지 콜렉션을 통한 전력관리 설계

KVM 에서는 기본 가비지 콜렉션 기법으로 마크-회수(압축) 알고리즘을 사용하고 있다. 이 방법은 2 장에서 설명하였으며, 가비지 콜렉션을 위한 오버헤드가 적고, 수행속도가 빠른 장점이 있다. 하지만 반면에 메모리에 단편화 현상을 발생시킬 수 있는 단점을 가지고 있다. Memory Bank 를 사용하는 임베디드 시스템에서 자바환경을 사용할 때, 전력관리를 위해서 KVM 의 가비지 콜렉션이 실행 될 때 전력관리를 위한 Supply-gating 알고리즘을 적용하면 된다.

[그림 3]에서 보는 바와 같이 초기 T=0 시점에서는 메모리를 사용하지 않기 때문에 모든 Memory Bank 가 OFF 상태였다가 시간이 지남에 따라 메모리 사용량이 증가하게 되면 OFF 된 Memory Bank 를 ON 시켜 데이터를 사용하게 된다. T=100 인 시점에서 더 이

상 메모리를 할당할 공간이 없을 경우 가비지 콜렉션이 수행되어, B,E,F 는 가비지가 되고 Bank3 에 저장된 데이터들은 더 이상 사용되지 않는 것이므로 메모리 낭비가 발생하게 된다. 이때 Bank3 의 전력공급을 차단한다. 이렇게 Bank3 의 전력을 차단하게 되면 메모리 상태정보가 초기화 되기 때문에 가비지 콜렉션의 메모리 해제 단계를 거치지 않아도 되는 장점이 있다.

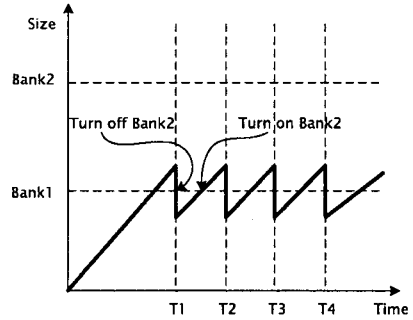


[그림 3] 가비지 콜렉션을 통한 전력관리

### 3.2 Thrashing 문제

Supply-gating 알고리즘의 기본원리는 가비지 콜렉터가 동작을 수행할 때 가비지를 가지고 있는 Memory Bank 를 turn on 할 것인지 turn off 할 것인지를 결정하는 것이다. 그런데 만약 가비지 콜렉션에 의해 너무 적은 양의 가비지가 회수된다면, 하나의 Bank 에 너무 많은 turn on/off 가 발생하여 Thrashing 이 생기게 된다. [그림 4]에서 보는 바와 같이 시간 T1, T2, T3, T4 에서 가비지 콜렉션이 일어난게 되면 매우 작은 크기의 가비지만이 회수되게 되어 Bank2 에 너무 많은 turn on/off 가 발생하게 되는 것을 볼 수 있다. Thrashing 은 가비지 콜렉션이 빈번하게 일어나면 발생하는 문제로 특정한 두 개의 Memory Bank 사이에서만 일어나는 문제가 아니라 전체 Memory Bank 에서 모두 나타날 수 있는 문제이다. 이러한 Thrashing 을 피하기 위해서는 가비지 콜렉션의 수행횟수를 조절하여, 너무 적은 양의 가비지가 회수되는 것을 막는 방법을 이용하여야 한다. 또한 가비지 콜렉션의 수행횟수를 조절하게 되면 가비지 콜렉션의 수행에 드는 오버헤드가 상대적으로 적어지기 때문에 KVM 전체의 성능향상에도 도움이 된다. 본 논문에서는 새로운 오브젝트나 배열을 위한 메모리 할당 시 마다 가비지 콜렉션을 수행하지 않고, 적절한 시기를 예측하여 가비지 콜렉션을 수행시키기 위하여, K-allocation 가비지 콜렉터를 사용한다. K-allocation 가비지 콜렉터는 기본적으로 마크-회수(압축) 알고리즘을 사용하며, 가비지 콜렉션의 수행 시점을 조절하기 위하여, K 개의 Object allocation 이

발생할 때 가비지 콜렉션을 수행하는 가비지 콜렉터이다. K 값은 임의로 조절이 가능하며, 수행하는 애플리케이션의 성격에 따라서 적당한 값으로 조절할 수 있다.



[그림 4] 인접한 두 Memory Bank 의 상태

K-allocation 콜렉터를 사용하게 되면, 전체적으로 KVM에서 사용하는 메모리 공간의 여유가 많을 경우에는 가비지 콜렉션을 수행하지 않아도 되므로, 가비지 콜렉션 수행에 드는 오버헤드를 줄일 수 있고, Thrashing 문제도 해결이 가능하다.

### 4. 결론 및 향후 연구과제

본 논문에서는 Thrashing 문제를 해결하기 위하여 K-allocation 가비지 콜렉터를 사용하고 Memory Bank의 전력을 관리하기 위하여 Supply-gating 알고리즘을 적용하여, 전체적인 KVM의 성능을 저해하지 않으면서 효과적으로 메모리의 전력을 관리할 수 있는 기법을 설계하였다.

향후 자바환경을 사용하는 다양한 임베디드 시스템 환경에서 적용이 가능하도록 특화된 전력 관리 기법에 대한 연구를 지속하여, KVM의 전력관리 성능을 향상시키기 위한 다양한 방법이 연구되어야 한다.

### 5. 참고문헌

- [1] G. Chen, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, M. Wolczko; " Adaptive Garbage Collection for Battery-Operated Environments "; Proceedings of the 2nd Java™ Virtual Machine Research and Technology Symposium, 2002, pp.1-12
- [2] Sun Microsystems, " JSR-30 J2ME Connected, Limited Device Configuration "
- [3] Bill Venner, " Inside the Java Virtual Machine ", 1999
- [4] Tim Lindholm, " The Java™ Virtual Machine Specification Second Edition ", 1999