

제한된 리소스를 사용하는 임베디드 환경을 위한 가비지 컬렉션의 분석 및 설계

손필창⁰, 조문행, 김용희, 이철훈
충남대학교 컴퓨터공학과
(pcson⁰, mhcho, yonghee, chlee)⁰@ce.cnu.ac.kr

Analysis and Design of The Garbage Collection for Resource Constrained and Embedded Environment

Pil-Chang Son⁰, Moon-Hang Cho, Yong-Hee Kim and Cheol-Hoon Lee
Dept. of Computer Engineering, Chungnam National Univ.

요 약

급속도로 IT 산업이 발전하면서, 리소스가 제한된 소형 기기들의 사용이 비약적으로 증가하는 추세이다. 자바는 플랫폼 독립성(Platform Independency), 보안성(Security), 이동성(Mobility) 등의 장점을 가지고 있기 때문에 성능을 극대화하고 안정된 서비스를 제공해야 하는 소형기기들에게 중요한 핵심 소프트웨어 플랫폼이 되어가고 있다. 임베디드 장치나 모바일 시스템과 같은 제한된 리소스를 사용하는 기기들은 자바 어플리케이션 수행을 위해 자바의 소프트웨어 플랫폼중의 하나인 K 가상 머신(K Virtual Machine: KVM)을 탑재하여 사용한다. 본 논문에서는 K 가상 머신의 가비지 컬렉션이 임베디드 환경에서 핵심 리소스인 에너지와 메모리를 좀 더 효율적으로 소비하면서 동작하는 방법에 대하여 분석하고 설계한 내용을 기술한다.

1. 서론

자바의 플랫폼 독립성, 보안성, 네트워크 이동성, 실행 코드의 재사용성, 작은 실행 파일 크기, 동적 적응성, 이식성, 개발의 용이성 등과 같은 장점들은 소형 기기에 적용 시 개발 비용 감소 등 여러 가지 이득을 발생 시키기 때문에, 자바를 임베디드 장치나, 제한된 리소스를 사용하는 소형 기기에 적용하기 위한 연구들이 다방면에서 진행되어 왔다

임베디드 장치나, hand-held device와 같은 소형 기기에서는 리소스가 제한적이기 때문에 에너지와 메모리의 사용이 제한될 수 밖에 없다. 현재 어플리케이션 프로그램의 수행 효율을 높이기 위하여 제한된 에너지와 메모리를 효과적으로 관리하는 다양한 기법들이 소프트웨어적인 방법뿐만 아니라 하드웨어적인 방법으로도 제안되고 있다.

본 논문은 KVM의 힙 메모리를 대상으로 하는 다양한 에너지와 메모리 관리기법들 중 소프트웨어적 기법과 하드웨어적 기법을 병행하여 에너지와 메모리를 효율적으로 관리하는 가비지 컬렉션 기법에 대해 분석하고 설계한 내용을 기술하였다.

본 논문의 2 장 관련연구에서는 메모리에서 사용되는 에너지 및 효율적인 에너지 관리를 위한 하드웨어 구조인 Banked memory architecture의 소개와 다양한 가비지 컬렉션 기법에 대한 분석을, 3 장에서는 에너지와 메모리를 효율적으로 관리하는 가비지 컬렉션 기법에 대한 설계를, 4 장에서는 결론 및 향후 연구 과제를 기술한다.

2. 관련 연구

2.1 메모리에서 사용되는 에너지

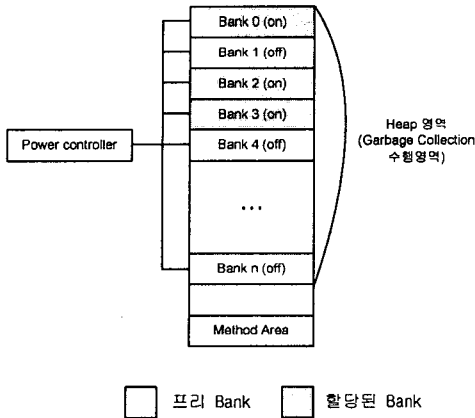
메모리에서 소모되는 에너지는 dynamic energy 와 leakage energy 로 크게 나눌 수가 있다. dynamic energy 는 메모리에 저장 되어있는 데이터를 read /write 할 때 소모되는 에너지로 메모리 관련 작업 수행 시 많이 소모되는 에너지이다. leakage energy 는 메모리에 데이터를 저장하기 위해 소모되는 에너지로서 read/write 를 수행하지 않아도 메모리에 전력이 공급되는 것만으로 소모되는 에너지이다.

본 논문에서는 dynamic energy 보다는 leakage energy 소비를 줄이는 방법에 초점을 맞추어 기술 한다[2].

2.2 Banked memory architecture

KVM의 힙 메모리 영역을 일정한 크기 단위인 Bank로 나누어 각 Bank에 전력 공급을 제어할 수 있는 메모리 구조로 본 논문에서 중요하게 다룰 leakage energy 소모를 줄이는데 유용한 하드웨어 구조이다.

[그림 1]은 KVM에서 힙 메모리 영역의 Banked memory architecture를 보여준다. 데이터가 저장되어있지 않은 메모리 Bank의 파워를 off 시킴으로써 불필요하게 소모되는 leakage energy를 줄일 수 있다.[2]

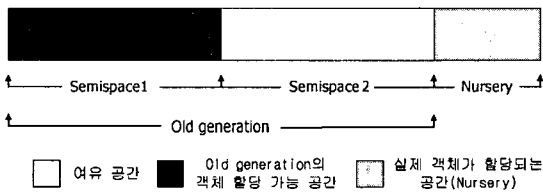


[그림 1] 힙 메모리의 Banked memory architecture

2.3 가비지 컬렉션

KVM 에는 마크-회수 방법을 이용하여 가비지 컬렉션이 구현되어 있다. 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. 이 방법은 간단하고 오버헤드가 적은 장점이 있으나, 시간이 지남에 따라 힙에 단편화 현상을 발생시켜 메모리의 효율성을 점점 감소시키는 단점이 있다[1].

KVM 에서 제공하는 마크-회수 방법 외에 여러 가지 가비지 컬렉션 기법들이 많이 연구되고 있다. 특히 힙 메모리 영역의 단편화와 가비지 컬렉션의 수행 횟수를 고려하여 좀 더 효율적인 방법들이 많이 나오고 있는데 그 대표적인 기법으로 Generational Copying 가비지 컬렉션 기법을 들 수 있다[3].



[그림 2] Generational Copying 가비지 컬렉션

[그림 2]는 Generational Copying 가비지 컬렉션을 위한 힙 메모리 영역의 구조를 나타내고 있다. 이 가비지 컬렉션은 힙 영역을 New generation 영역(Nursery)과 Old generation 영역으로 나누는 기법인 Generational 컬렉터에 Old generation 부분만 Copying 컬렉터를 적용시킨 기법이다. 메모리 단편화 현상과 수행의 효율성을 동시에 높일 수 있는 효과를 얻을 수 있지만 Copying 단계를 수행하기 위해 전체 Old generation 영역의 1/2 만 객체를 할당할 수 밖에 없다는 단점이 있다.

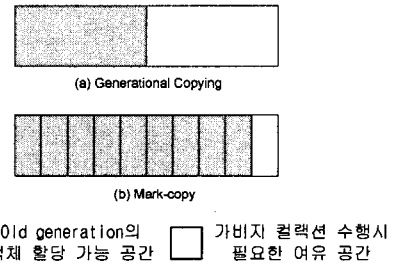
3. 효율적인 에너지와 메모리 사용을 위한 가비지 컬렉션 설계

본 논문에서는 앞 장에서 살펴본 하드웨어 구조와 여러 가지 가비지 컬렉션 기법들을 이용하여 에너지와 메모리 효율적인 가비지 컬렉션 기법을 설계하였다

3.1 에너지 효율적인 마크-카피 가비지 컬렉션 설계

본 논문에서는 KVM 의 마크-회수 가비지 컬렉션의 메모리 영역에 대한 단편화 문제와 Generational Copying 가비지 컬렉션의 메모리 사용의 비효율성 문제를 동시에 해결할 수 있는 마크-카피 가비지 컬렉션 기법을 설계하였다.

마크-카피 가비지 컬렉션은 힙 메모리 영역을 일정한 크기의 Bank 단위로 나누어 메모리의 효율성을 높이면서 Generational Copying 가비지 컬렉션을 수행할 수 있는 기법이다[3].



[그림 3] Old generation 영역의 비교

[그림 3]은 Generational Copying 컬렉터와 마크-카피 컬렉터 각각의 Old generation 영역에서 새로운 객체나 배열을 위한 메모리 할당이 가능한 영역의 크기를 비교하여 나타내고 있다.

새로운 객체나 배열을 위한 메모리 할당 시 Generational Copying 컬렉터는 Old generation 영역의 1/2 에 해당되는 부분만 할당 가능한 반면에 마크-카피 컬렉터는 하나의 Bank 만 남겨두고 모든 영역에 할당이 가능하다. 마크-카피 컬렉터는 힙 메모리 영역을 Bank 라는 단위로 나누어 메모리 효율성을 높이고 있는데, 이 점을 이용하여 마크-카피 컬렉터 기법에 Banked memory architecture 를 적용한다면 에너지와 메모리 면에서 더 효과적인 가비지 컬렉터를 설계할 수 있다.

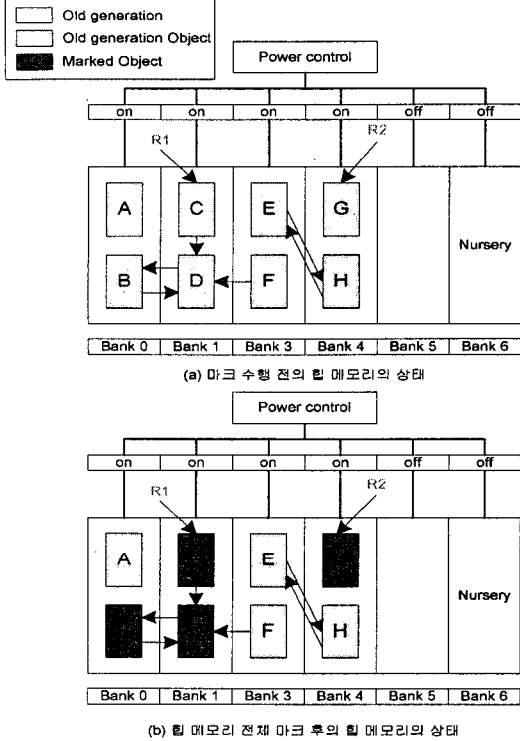
마크-카피 컬렉터는 마크 단계와 카피 단계로 나누어서 수행된다. 각각 단계에서 어떻게 Banked memory architecture 와 병행해서 동작할 것인지에 대해 좀 더 자세히 살펴 보도록 하겠다.

3.2 마크 단계 (Mark phase)

마크-카피 컬렉터는 Old generation 영역에 더 이상 새로운 객체나 배열을 할당할 공간이 없을 시에 Full collection을 수행하게 된다. Full collection이 시작되면 첫 번째 단계인 마크 단계를 수행하게 되는데, 이 단계에서는 현재 사용되고 있는 객체나 배열들의 Mark bit만을 셋팅하여 참조되고 있는 객체와 참조되

지 않는 객체들을 분리하게 된다.

[그림 4]는 힙 메모리 영역이 6 개의 Bank로 나누어져 있고 각 Bank마다 2 개의 객체가 할당 가능하다는 가정 하에서 마크 단계를 나타낸다.



[그림 4] 마크-카피 컬렉터의 마크 단계

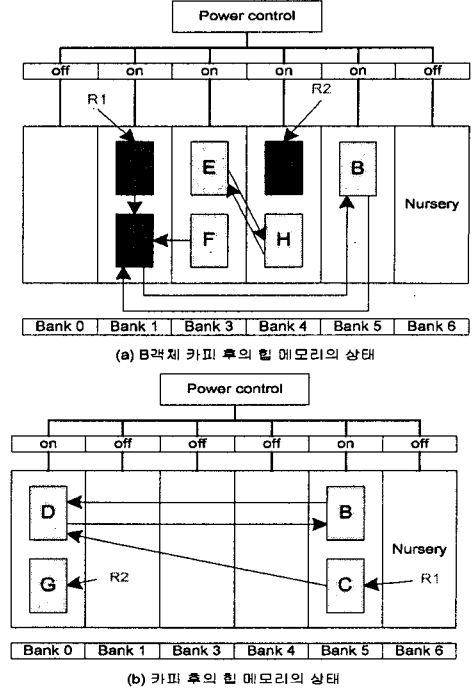
Old generation 영역에 빈 공간이 Bank 5 하나이기 때문에 Full collection이 수행되면서 마크가 이루어진다. (a)에서 보는 바와 같이 C와 G는 root 객체이고, C와 G가 참조하고 있는 객체들을 따라가다 보면 B와 D도 root로부터 도달 가능한 살아있는 객체라는 것을 알 수 있다. 따라서 (b)에서 객체 B, C, D, G를 마크하고 있다. 또한 쓰이지 않는 Bank 메모리들의 파워를 off 시켜 불필요한 에너지 소모를 방지하고 있다.

3.3 카피 단계 (Copy phase)

이 단계에서 마크-카피 컬렉터는 카피 수행 시 Generational Copying 컬렉터와 다르게 Old generation 영역의 하나의 빈 Bank 메모리 영역만 필요하게 된다. 카피는 한번에 하나의 Bank만 이루어지며 순서는 객체를 오래 저장하고 있었던 Bank부터 카피된다. Bank를 카피하게 되면 다시 하나의 빈 Bank 메모리 영역이 나오게 되기 때문에 그 다음 순서의 Bank를 카피할 수 있다.

[그림 5]는 [그림 4]의 마크 단계에 이어서 수행되는 카피 단계를 나타낸다. (a)는 B객체 하나만 카피되는 모습을 보여주고 있고 (b)는 카피가 모두 이루어진 힙의 상태를 나타내고 있다. 마크 단계와 마찬가지로

쓰이지 않는 Bank 메모리들의 파워를 off 시켜 불필요한 에너지 소모를 방지하고 있다.



[그림 5] 마크-카피 컬렉터의 카피 단계

4. 결론 및 향후 연구 과제

본 논문에서는 KVM에서 제한된 에너지와 메모리를 효율적으로 사용하기 위해 Banked memory architecture와 마크-카피 가비지 컬렉션 기법을 이용해서 에너지와 메모리의 관리를 효과적으로 하는 가비지 컬렉션을 분석 및 설계하였다.

본 논문에서 제시한 마크-카피 가비지 컬렉션 기법은 수행 시에 Pause되는 시간이 길기 때문에 실시간성을 만족하지 못한다. KVM은 소형기기뿐만 아니라 실시간 운영체제 상에서도 동작이 가능하다. KVM의 확장 측면에서 생각한다면 에너지와 메모리의 효율적인 사용뿐만 아니라 실시간성까지 만족하는 좀 더 발전된 가비지 컬렉션 기법이 필요하다.

5. 참고문헌

- [1] Sun Microsystems, *JSR-30 J2ME Connected, Limited Device Configuration*
- [2] G.Chen, R. Shetty, M. Kandemir, N. Vijaykrishna, M. J. Irwin, and M. Wolczko. *Tuning garbage collection in an embedded Java environment*. In Proc. The 8th ISHPCA, Cambridge, MA, 2002
- [3] Narendran Sachindran and J. Elliot B. Moss. *Mark-Copy: Fast copying GC with less space overhead*. In OOPSLA' 03 ACM, Anaheim, CA, 2003
- [4] Bill Venners, *Inside the Java Virtual Machine*