

가상화 환경에서 메모리 효율성 향상을 위한 메모리 중복제거 연구

조영중*, 이세호*, 엄영익*
*성균관대학교 정보통신대학
e-mail:{jyj8843, loadic, yieom}@skku.edu

A Study on Memory Deduplication for Improvement of Memory Efficiency in Virtualization Environment

Youngjoong Cho*, Seho Lee*, Young Ik Eom*
*College of Information and Communication Engineering, Sungkyunkwan University

요 약

가상화 기술은 하드웨어 위에서 여러 운영체제를 동작시키면서, 시스템의 활용률을 극대화 시키는 기술이므로 여러 분야에서 각광받고 있다. 가상화는 시스템 위험성 전파 등을 줄임으로써, 보안 노출을 막는 등 여러 장점들이 있다. 하지만, 게스트머신에서 하이퍼바이저로의 잦은 스위치는 가상화 성능을 떨어트린다. 또한, 다수의 가상머신에서 공유될 수 있는 페이지들에 대한 메모리 중복 문제도 존재한다. 우리는 가상화 환경에서 VMEXIT를 줄이고, 메모리를 절약할 수 있는 메모리 중복제거 기술을 제안하고, 이를 정성적으로 성능 평가를 진행하였다.

1. 서론

가상화 기술은 유한한 하드웨어 자원을 공유하는 다수의 데스크탑과 서버를 동작시킬 수 있기 때문에, 비용 및 공간 절약 측면에서 최근에 각광받고 있다. 또한, 가상화 기술을 사용하면 하나의 시스템에 여러 가상머신을 실행할 수 있으므로 시스템의 활용률을 극대화시킬 수 있다. 특히, 가상화의 가장 큰 장점은 여러개의 운영체제를 하나의 시스템에서 동작시킬 수 있다는 점이다. 각 운영체제는 각 가상머신 내에서 동작을 하고, 각 가상머신에서 동작하는 운영체제가 하드웨어에 대한 접근을 요구하거나 특권명령어를 실행할 때, 게스트에서 하이퍼바이저로 스위치가 일어나게 된다. 하이퍼바이저는 가상머신이 처리하지 못하는 하드웨어에 대한 명령어와 특권 명령어를 처리해줘야 한다. 하지만 이러한 스위치 동작은 가상화 환경에서 성능저하의 주요원인으로 작용한다 [1][2].

이러한 문제뿐만 아니라, 가상화 환경에서는 다수의 게스트머신이 동작하기 때문에 유한한 메모리를 효율적으로 관리하는 것이 중요하다. 다수의 게스트머신이 스토리지에 있는 동일한 파일을 각 게스트머신을 위한 페이지 형태로 관리하게 된다면, 이는 분명히 메모리를 낭비하는 것이다. 이러한 문제를 다룬 연구는 지금까지 활발히 연구되어 왔다 [3][4][5]. 하지만 기존의 연구는 제한조건을 가지고 있

어 효율적으로 가상머신간에 페이지를 공유할 수 있는 기회의 한계가 존재한다. 우리는 가상화 환경에서 성능저하의 원인인 VMEXIT를 줄이면서, 가상머신간에 효과적으로 페이지를 공유할 수 있는 모델을 제안하고, 이를 뒷받침할 수 있는 정성적 평가를 진행하였다.

2. 관련연구

2.1. SplitX

게스트가 하이퍼바이저의 간섭이 필요할 때, VMEXIT가 발생한다. 이때 게스트와 하이퍼바이저가 같은 코어에서 실행된다면, 이로 인해서 캐쉬가 오염된다. 또한, 하이퍼바이저로 스위치 되고 난 뒤에, 게스트는 실행이 중단되는 문제점이 있다. 스위치가 발생하고, 캐쉬는 게스트의 데이터에서 하이퍼바이저의 데이터로 채워져야 한다. SplitX는 앞에서 언급한 문제를 해결하기 위해서 게스트와 하이퍼바이저를 위해서 전용의 코어에서 실행되도록 하는 구조를 제안하였다. 이러한 접근은 캐쉬 오염과 스위치 이후에 게스트가 멈추는 것을 막을수 있다. 하지만 이러한 구조는 현재 하드웨어에서 구현될 수 없다는 단점이 있다.

2.2. ELVIS

가상화 환경에서 게스트가 IO 요청을 하면, 게스트에서 하이퍼바이저로 스위치가 발생한다. 이후에 하이퍼바이저는 게스트의 IO 요청을 에뮬레이션 프로세스에게 전달하여 대신하여 처리하게 된다. 하지만 이러한 과정에서도

“본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학 IT연구센터 육성지원 사업의 연구결과로 수행되었음(NIPA-2013-(H0301-13-4006))”

VMEXIT가 발생하게 된다. ELVIS는 이러한 VMEXIT를 줄이기 위하여 게스트와 하이퍼바이저가 공유할 수 있는 메모리영역을 만들고, 에물레이션 프로세스에 존재하는 IO 스레드가 공유 메모리영역을 폴링하도록 하여, 게스트의 IO 요청에 대한 오버헤드를 줄였다.

2.4. satori

가상화 환경에서 다수의 가상머신에서 공유 페이지를 페이지 스캐닝 기법을 이용하여 찾게 된다면, 많은 공유 페이지를 찾을 수 있었지만, 감당할 수 없는 성능저하가 발생하게 된다. satori는 다수의 가상머신이 하나의 가상 디스크 이미지를 공유하여 물리메모리를 절약할 수 있도록 한다. 하지만 satori는 처음 제공되는 디스크 이미지만을 공유하기 때문에, 추가적인 파일에 대해서는 페이지 공유를 할 수 없는 단점을 가지고 있다.

3. 제안기법

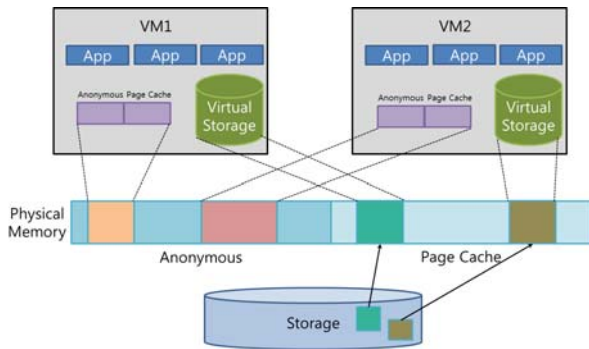


그림 1. 제안된 메모리 중복제거 기법

페이지 공유 기법에서 중요한 요소는 가상머신 간에 중복되는 페이지를 빠르게 찾는 것이다. 기존 연구에서는 공유 페이지를 스캐닝하는 도중에 짧은 시간동안 공유되는 페이지를 찾기 어려웠고, 찾더라도 그 비용이 상당하였다. 또한, 공유 페이지를 찾기 위해서 가상머신에서 호스트로 전달하는 정보는 파일의 섹터 넘버뿐이라서, 호스트에서 이 정보를 이용하여 가상머신 간에 페이지 공유를 제공하는데 많은 제약사항이 있었다.

그림1은 우리가 제안하는 페이지 공유 기법을 나타낸다. 호스트 영역의 익명메모리 영역을 일정 주기로 스캐닝하는 기존 연구와는 달리 우리가 제안한 방식은 파일 오퍼레이션을 직접 호스트로 직접 전달하며, 파일의 섹터 넘버가 아닌 파일의 이름을 전달하여 가상머신 간에 페이지가 효과적으로 공유될 수 있도록 한다.

4. 평가

본 논문은 기존 페이지 공유 방법과 제안 방법에 대한 정성적 평가를 표1과 같이 진행하였다. 우리가 제안한 방법은 기존 중복기법과 달리 주기적으로 호스트의 익명 메

모리 영역을 스캔하지 않는다. 이로써 주기적으로 익명 메모리 영역을 스캔하는 기존 기법들과 차별성이 있고, 시스템에 주는 오버헤드가 적다.

표 1. 제안된 페이지 공유 기법 평가

	기존 방법	제안 방법
페이지 공유 기회	드물다	빈번하다
게스트 운영체제 수정	필요하다	불필요하다

5. 결론

본 논문에서는 가상화 환경에서 메모리 중복제거를 효과적으로 할 수 있는 구조를 제안하였고, 이에 대한 정성적 평가를 진행하였다. 제안된 구조는 가상화 환경에서 가장 큰 성능저해 요소인 VMEXIT를 제거하는 방법을 제안하였고, 또한 가상머신간에 공유되는 페이지를 하이퍼바이저에서 파일의 이름으로 검색될 수 있는 방법을 제안하였다. 우리가 제안한 방법을 사용하면 기존 메모리중복 기법들에서 제기되어 왔던 주기적인 스캐닝 방법을 회피할 수 있을 뿐 아니라, 페이지 공유기회도 증가된다.

앞으로 우리는 제안된 기법을 리눅스 KVM에서 실제 구현하여, 제안기법에 대한 정성적 평가를 정량적으로 검증 하겠다.

참고문헌

- [1] A. Landau, M. Ben-Yehuda, and A. Gordon. SplitX: Split guest/hypervisor execution on multi-core. In USENIX Workshop on I/O Virtualization (WIOV), 2011.
- [2] A. Gordon, N. Har'El, A. Landau, M. Ben-Yehuda, and A. Traeger. Towards exitless and efficient paravirtual I/O. In Proceedings of the 5th Annual International Systems and Storage Conference, 2012.
- [3] G. Miłos, D. G. Murray, S. Hand, and M. A. Fetterman. "Satori: Enlightened Page Sharing," Proceedings of the 2009 USENIX Annual Technical conference, 2009.
- [4] C. Waldspurger, "Memory Resource Management in VMware ESX Server," ACM SIGOPS Operating Systems Rev. 2002.
- [5] M. Konrad, "XLH: More effective memory deduplication scanners through cross-layer hints," Proceedings of the 2013 USENIX Annual Technical conference, 2013.