

# 가상화 환경에서 GPU 성능의 향상을 위한 장치 가상화 프레임워크

주영현, 이동우, 엄영익  
성균관대학교 정보통신대학  
e-mail : {dudgus0627, lightof, yieom}@skku.edu

## Device Virtualization Frameworks for Accelerating GPU Performance on Virtual Environments

Younghyun Joo, Dongwoo Lee and Young Ik Eom  
College of Information & Communication Engineering, Sungkyunkwan University

### 요 약

최근 가상화 기술에 대한 많은 관심과 연구들로 인해 가상 머신은 물리(Native) 머신에 가까운 성능을 보이며 프로세서 및 메모리 자원을 제공하고 있다. 하지만 GPU 와 같은 그래픽 하드웨어에 대한 장치 가상화는 다른 가상화 기법에 비해 연구가 미흡한 상태로 가상화 환경에서의 영상처리에 걸림돌이 되고 있다. 가상화 환경에서의 영상처리는 기존의 X 윈도우 시스템을 이용하여 영상을 처리하는데, 이는 2D 영상처리에 최적화 되어 있어서 3D 영상을 처리하는데 성능의 한계 보일 뿐만 아니라 가상 머신에서 메모리가 중복으로 복사되면서 낮은 성능 보여주고 있다. 제안하는 장치 가상화 프레임워크는 기존의 메모리의 중복 복사를 제거하면서 성능을 향상 시킬 수 있다. 본 논문에서는 가상화 환경에서 GPU 성능 향상을 위한 장치 가상화 프레임워크를 제안하고 평가를 통해 본 기법의 타당성을 입증한다.

### 1. 서론

최근의 가상화 기술은 소규모의 모바일 장치에서부터 대규모의 클라우드 컴퓨팅을 위한 가상 머신에 이르기까지 폭 넓게 이용되고 있다. 가상화 기술을 통하여 물리적인 컴퓨팅 자원을 다른 시스템이나 응용 프로그램 등이 상호작용하며 활용할 수 있도록 하는 등의 다양한 장점을 제공한다. 이에 따른 가상화 시스템 소프트웨어 및 운영체제 기술이 많은 주목을 받으며 연구가 활발히 진행되고 있다[1][2][3]. 하지만 많은 연구가 진행된 프로세서 가상화 기술이나 메모리 가상화 기술[4]과는 다르게 장치 가상화 기술에 있어서는 연구가 미흡한 상태이다. 특히 가상화 환경에서의 Graphic Processor Unit (GPU)를 이용한 영상처리의 성능은 큰 문제점으로 제기되고 있다.

초창기의 GPU 는 영상처리의 기능을 수행하기 위해 존재하였다. 그래픽의 발전에 따라 효율적인 영상처리를 위해 다중 스레드 처리가 가능하도록 CPU 보다 수 배 많은 멀티코어를 이용하여 GPU 를 구성하고, 각 코어의 공유를 통해 병렬 프로그래밍이 가능하도록 하였다. 이에 따라 연산 능력은 고속화되어 CPU 가 전통적으로 취급하였던 응용 프로그램의 연

산을 효율적으로 처리하는 GPGPU 의 역할 또한 수행하며 앞으로의 필요성이 주목되고 있다.

기본적으로 GPU 를 이용한 리눅스의 영상처리는 X 윈도우[5] 시스템이 담당하였다. X 윈도우는 서버-클라이언트 구조로 2D 영상에 최적화되어 처리를 진행한다. 하지만 2D 영상처리에 최적화된 X 윈도우 시스템은 성능의 한계가 있었다. 이를 개선하기 위해 Direct Rendering Infrastructure (DRI)[6]가 도입되었고, 현재 DRI 를 통한 3D 가속 영상처리를 하고 있다[7]. 그러나 가상화 환경에서의 가상 머신의 영상은 여전히 X 윈도우 시스템을 이용한 영상처리에 의존하고 있다. 이 때문에 3D 와 같은 영상처리에 있어 성능의 문제점이 발생한다. 또한 가상 머신에서의 영상 처리 동작에서 발생하는 메모리의 중복 복사가 영상처리의 성능을 감소시킨다.

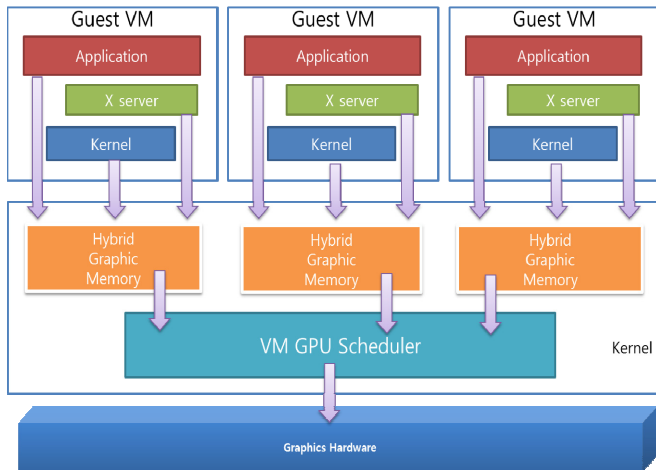
본 논문에서는 가상화 환경에서 GPU 성능을 향상시킬 수 있는 장치 가상화 프레임워크를 제안하며 제안한 프레임워크에 대한 정성적 평가를 진행하여 가상화 환경에서의 영상처리 성능의 향상을 입증한다.

### 2. 제안 기법

가상화 환경에서의 가상 머신의 영상은 여전히 X 윈도우 시스템을 이용한 영상처리에 의존하고 있다. KVM-QEMU 환경에서의 가상 머신은 QEMU 상의 가상 그래픽 하드웨어에 X 윈도우 방식과 DRI 방식의

† 이 논문은 2012 년도 정부(교육과학기술부)의 재원으로 한국연구재단-차세대정보컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No. 2012-0006423)."

로 영상을 처리하게 된다. 하지만 QEMU 에서 GPU 드라이버로 영상을 처리하는 과정에서 X 윈도우 시스템을 거치게 되면서 3D 와 같은 영상처리에 있어서 성능의 한계가 발생하게 된다. 또한 기존의 방법은 가상 머신의 프레임버퍼에서 QEMU 가 가상 그래픽 메모리로의 영상이 복사되고, 영상이 커널의 프레임버퍼로 전달되고, 마지막으로 그래픽 하드웨어로 전달되면서 불필요한 영상 메모리의 복사가 진행된다.



(그림 1) GPU 성능을 위한 장치가상화 프레임워크

제안하는 가상화 환경에서 GPU 성능의 향상을 위한 장치가상화 프레임워크의 구조는 그림 1 과 같다. 각 가상 머신은 기존과 같이 X 윈도우와 DRI 방식을 이용하여 하이브리드 그래픽 메모리에 영상을 매핑하게 된다. 이렇게 매핑된 영상은 X 서버를 거치지 않고, 바로 DRI 방식으로 GPU 에 전달하여 영상을 처리하게 된다. 가상 머신의 영상은 DRI 를 통해 바로 하이브리드 그래픽 메모리에 매핑되고, 이는 바로 그래픽 하드웨어로 전달되면서 불필요한 영상의 복사를 줄이게 된다. 다수의 가상 머신이 작동 시에 각각의 그래픽 메모리를 효율적으로 처리할 수 있도록 가상 머신 GPU 스케줄러를 이용하여, 스케줄링을 통해 GPU 에 전달한다

3. 제안 기법 평가

본 장에서는 기존의 영상처리를 위한 장치 가상화 시스템 구조와 제안한 장치 가상화 프레임워크와의 비교, 분석을 위해서 정성적 평가를 진행하였다. 한 개의 영상에 대해서 3000 번의 프레임을 처리한다고 가정하였고 이 때의 프레임버퍼의 용량을 1000 byte 로 기준을 정하였다. 영상 처리 과정에서 발생하는 클리핑 등으로 처리되는 메모리 경우는 제외하고, 기본 영상에 대한 메모리 처리만으로 평가하였다.

분석 결과, 제안한 기법은 주어진 기준에서 프레임의 복사를 2/3 으로 줄일 수 있다. 기존에는 가상 머신의 프레임 버퍼에서 하이퍼바이저의 가상 그래픽 메모리로, 가상 그래픽 메모리에서 프레임 버퍼로, 프레임 버퍼에서 그래픽 하드웨어로 크게 3 번의 매핑이 이루어지는 반면에, 제안한 기법은 가상 머신의

프레임 버퍼에서 하이퍼바이저의 하이브리드 그래픽 메모리로, 하이브리드 그래픽 메모리에서 그래픽 하드웨어로 크게 2 번의 매핑이 이루어 지지 않기 때문에 불필요한 메모리 매핑 혹은 복사를 줄일 수 있다. 또한 DRI 를 이용하여 3D 처리의 성능 향상을 가져올 수 있다.

<표 1> GPU 장치 가상화 기법 평가

특 성	기존 기법	제안 기법
메모리 복사 횟수 (회)	9000	6000
처리되는 메모리 양 (byte)	3000	2000
3D 처리 성능	↓	↑

4. 결론 및 향후 연구 방향

본 논문에서는 가상화 환경에서 GPU 성능을 향상시킬 수 있는 장치가상화 프레임워크를 제안하고, 이에 대한 평가를 진행하였다.

가상화 환경에서 가상 머신에 대한 영상처리가 X Server 를 거치게 되면서 영상처리의 성능에 대한 제한이 생기게 되고, 원활한 영상처리에 대한 문제가 발생하였다. 하지만 제안한 GPU 장치가상화 프레임워크를 통해 DRI 방식으로 영상을 처리하고, 스케줄링 함으로써 3D 영상처리와 원활한 처리속도를 보장할 수 있다.

최근의 가상화 기술과 GPU 의 발전에 따라 가상화 환경에서의 영상처리의 중요성이 증대되고 있다. 향후 프로토타입 구현과 평가를 통해 연구를 입증할 계획이다.

참고문헌

- [1] P. Barham, B. Dragovic, K.Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization", In Proc. 19<sup>th</sup> ACM Symposium on Operating Systems Principles (SOSP'03), pp.164-177. 2003.
- [2] A. Kivity, Y. Kamay, D. Laor, U. Lublin and A. Liguori, "kvm: the Linux Virtual Machine Monitor", In Proc. Linux Symposium, pp.225-230, 2007.
- [3] S. Devine, E. Bugnion, and M. Rosenblum, "Virtualization system including a virtual machine monitor for a computer with a segmented architecture", US Patent, 6397242, 1998.
- [4] K. Adams and O. Agesen, "A comparison of software and hardware techniques for x86 virtualization", In Proc. 12<sup>th</sup> International conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'06), pp.2-13, 2006.
- [5] R. W. Scheifler and J. Gettys, "The X window system", In Proc. ACM Transaction on Graphics (TOG), Volume 5, Issue 2, pp.79-109, 1986.
- [6] K. E. Martin, R. E. Faith, J. Owen, and A. Akin, "Direct Rendering Infrastructure, Low-Level Design Document", Precision Insight, Inc., 1999.
- [7] 최승한, 설동명, 안성호, 이경희, "리눅스 3D 그래픽 기술 동향", ETRI 전자통신동향분석 제 21 권 제 1 호, 2006.