

가상 3D 콘텐츠의 실시간 방송 시스템을 위한 효율적인 어플리케이션 가상화 기법

An Efficient Application Virtualization Method for a Real Time Broadcasting System of Virtual 3D Contents

장수민*, 김원영*, 유재수**

한국전자통신연구원 클라우드컴퓨팅연구부*, 충북대학교 정보통신공학부**

Sumin Jang(jsm@etri.re.kr)*, Wonyoung Kim(wykim@etri.re.kr)*,
Jaesoo Yoo(yjs@chungbuk.ac.kr)**

요약

최근 클라우드 컴퓨팅은 다양한 IT분야에서 온디맨드 방식으로 소프트웨어, 하드웨어, 데이터를 서비스처럼 제공하는 것을 가능하도록 만들었다. 본 논문은 가상 3D 콘텐츠의 실시간 방송 서비스를 위하여 이러한 클라우드 컴퓨팅 기술을 이용한다. 그러나 기존 관련 어플리케이션 가상화 솔루션들은 서비스 사용자가 증가할 때 서버의 부하가 급증하거나 유지비용이 매우 높은 문제점들을 갖고 있다. 그래서 본 논문은 이러한 문제점을 해결할 수 있는 새로운 어플리케이션 가상화 방식을 제안한다. 제안하는 방식은 그래픽 명령어들을 서버에서 실행되는 것이 아니라 클라이언트로 전송하여 실행하기 때문에 다수 사용자들에게 안정적인 서비스를 보장한다. 성능평가는 제안한 기법이 기존 기법들보다 우수함을 보여준다.

■ 중심어 : | 어플리케이션 가상화 | 클라우드 컴퓨팅 | 소프트웨어 서비스 |

Abstract

In recent, the cloud computing makes it possible to provide the on-demand provision of software, hardware, and data as a service in various IT fields. This paper uses the cloud computing techniques for the real time broadcasting service of virtual 3D contents. However, the existing related solutions have many problems that the load of the server is rapidly increased or the cost of the server is very high when the number of service users is increased. Therefore, we propose a new application virtualization method to solve these problems. It promises their stable operations in multi-user services because the proposed method does not execute the rendering commands of the application at the host server but delivers and executes them to clients via the Internet. Our performance experiments show that our proposed method outperforms the existing methods.

■ keyword : | Application Virtualization | Cloud Computing | Software as a Service |

* 본 연구는 미래창조과학부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음.
[10035185, 서버 기반 SW 서비스의 분할 실행 기술]

접수일자 : 2013년 05월 31일

심사완료일 : 2013년 08월 27일

수정일자 : 2013년 07월 09일

교신저자 : 장수민, e-mail : jsm@etri.re.kr

I. 서론

최근 클라우드 컴퓨팅 기술의 발전으로 사용자들은 다양한 단말기를 통하여 어디에서나 스토리지, 콘텐츠 사용, 소프트웨어를 서비스 형태로 제공받을 수 있게 되었다[1]. 또한 시스템의 하드웨어적인 문제들을 벗어나서 IT 리소스를 다양한 형태로 변경 및 설정해서 사용할 수 있게 되었다. IT 분야에서 클라우드 컴퓨팅 기술 만큼이나 최근에 다양한 응용분야에서 3D 그래픽 렌더링이 필요로 하는 가상 콘텐츠의 사용이 크게 증가하고 있다[2-6]. IT 관련 하드웨어 기술 발전으로 기존의 2D 로 형태로 개발된 소프트웨어를 3D 그래픽 소프트웨어로 변경하는 경우가 점점 증가하고 있으며 3D 가상 콘텐츠를 교육이나 스포츠에 접목하여 활용하는 경우가 많다.

[그림 1]는 가상 컴퓨터 그래픽 콘텐츠를 주로 사용하는 응용 사례들을 보여준다. [그림 1](a)는 골프연습장에 정보통신기술(ICT)을 결합해 새로운 개념의 오락 거리를 제공하는 스크린 골프이다. 골프선수들이 주로 사용하는 스윙 교정용 기계와 가상 컴퓨터 그래픽 콘텐츠를 결합하여 실제 필드에서 게임을 하는 것과 흡사하도록 만들어져 있다. 이러한 스크린 골프 시장은 2012년에 2763억 원의 매출을 올릴 정도로 규모가 확대되고 있는 실정이다. [그림 1](b)는 최신 온라인게임 리그전을 생방송하고 모습이다. 가상 3D 콘텐츠를 이용한 게임들을 e스포츠로 활성화하여 여러 개의 방송채널을 통하여 많은 사람들이 이를 시청하고 즐기도록 제공하고 있다.

본 논문은 이러한 가상 3D 콘텐츠를 사용하는 실시간 방송시스템의 온라인 서비스를 위한 새로운 가상화 기술을 제안한다. 가상 3D 콘텐츠의 실시간 방송을 위한 최근의 기술로는 서버에 관련 어플리케이션을 실행하고 그 결과 화면을 클라이언트에 전송하는 어플리케이션 가상화 방식이 있다. 이러한 방송 서비스에 적용하기 적절한 가상화 솔루션은 Citrix 사의 XenDesktop[7], MS사의 RemoteFX[8]와 Vmware의 View/Mirage[9]가 있다. 이 기존 솔루션들은 3D 그래픽을 사용하는 어플리케이션을 소프트웨어 서비스 형

태로 지원하기 위하여 서버에 장착된 그래픽 카드를 전용으로 사용하거나 공유하여 사용하기 때문에 사용자가 증가함에 따라 서버의 부하가 급속히 증가하거나 서버 유지비용이 매우 큰 문제점을 가지고 있다[10-12]. 그래서 본 논문은 이러한 문제점들을 해결하기 위한 방안으로 새로운 어플리케이션 가상화 방식인 분할실행 기법을 제안한다.



(a) 스크린 골프



(b) 온라인게임 생중계

그림 1. 가상 컴퓨터 그래픽 콘텐츠를 사용하는 응용사례

II. 관련연구

가상 3D 콘텐츠를 어플리케이션 가상화 방식으로 방송에 적용할 수 있는 Citrix의 XenDesktop, MS의 RemoteFX와 같은 기존 솔루션들은 서버에서 전적으로 소프트웨어를 실행하고, 그 실행 화면을 클라이언트로 전송하는 방식으로 서비스가 이루어진다. 이때, 클라이언트는 단순히 실행 화면의 스트리밍된 데이터를 받아 디코딩하여 화면에 표시하는 터미널 역할만을 수행한다.

Citrix사는 캠브리지 대학의 연구 프로젝트로 개발된 Xen 하이퍼바이저를 이용한 가상화 시스템을 제공한다. Citrix는 서버 기반 데스크탑 가상 기술인 XenDesktop 및 XenServer, 클라이언트 가상화 기술인 XenClient, 어플리케이션 가상화 솔루션인 XenApp와 같은 기술들을 보유하고 있다. XenDesktop 기술은 Hypervisor XenServer에 XenDesktop이라는 Desktop Delivery Controller 를 구성하여 사용자에게 데스크탑(OS)를 제공해 주는 솔루션으로 다양한 디바이스에서 웹을 통해 가상 데스크탑에 접속하여 사용할 수 있도록 되어 있다. XenApp는 어플리케이션 가상화 영역으로 특정 어플리케이션만 사용자에게 전달해주는 방식이며, 중앙 서버에 배포된 어플리케이션을 다수의 사용자가 동시에 서비스 받는 구조로 되어 있어 인프라 구축 비용은 XenDesktop에 비해 상대적으로 적게 들어간다.

말기를 통하여 서버쪽에서 실행된 어플리케이션의 결과 화면을 가상 데스크탑 에이전트를 통하여 전송받고 서버와 인터페이스는 클라이언트 입력디바이스를 통하여 처리한다. 이때 서버쪽 입력 디바이스는 가상화하여 처리된다.

이러한 Citrix는 3D 렌더링이 필요한 실시간 방송 프로그램을 서비스하기 위해서는 GPU 하나당 시청자 한 명을 지원함으로써 서버의 비용이 매우 높을 뿐만 아니라 서비스 확장성이 매우 제한적인 문제점이 있다. 그래서 방송을 시청하는 사람이 증가하게 되면 그 만큼의 GPU를 서버에 장착해야 서비스가 가능하기 때문에 Citrix를 이용한 가상 3D 콘텐츠의 포함한 실시간 방송 시스템의 구축은 매우 고비용을 요구하는 문제점을 가지고 있다.

MS사의 RemoteFX[6]는 RDP(Remote Desktop Protocol)의 확장버전인 RDP 7.1에 포함된 기술로서 서버 상의 GPU를 가상화 자원으로 활용해 썬 클라이언트를 비롯한 거의 모든 디바이스에 로컬 PC 환경과 동일한 강력한 그래픽 성능을 제공할 수 있다. RemoteFX의 가장 큰 특징은 원격 호스트(원격 DVI의 가상 머신 또는 원격 터미널 서버)의 GPU를 이용해서 그래픽을 렌더링하고 그 결과를 압축된 비트맵으로 클라이언트로 전송하는 호스트 렌더링 기술이다. RemoteFX를 통해 3D나 HD 동영상들은 원격 호스트의 GPU를 이용해서 렌더링이 되고 그 결과가 클라이언트로 전송되어 출력되기 때문에 클라이언트에서도 다양한 그래픽 효과를 즐길 수 있다. RemoteFX는 원격 호스트의 전체적인 윈도우 환경을 클라이언트 디바이스에 전송하는 것을 목적으로 한다. 즉 사용자가 개인용 PC를 사용해 경험할 수 있는 모든 윈도우 환경(멀티 모니터, 에어로 글래스(Aero Glass), 3D 게임, 멀티미디어, 플래쉬, 실버라이트, 비디오 편집 등)을 RemoteFX를 통해 호스트 컴퓨팅 환경에서 유사하게 경험할 수 있도록 한다. 그러나 MS사의 RemoteFX도 3D 렌더링이 필요한 어플리케이션의 서비스를 위해서는 사용자 마다 각각의 Guest OS를 두기 때문에 서버의 부하가 적지 않고 여러 사용자가 하나의 GPU를 공유해서 그래픽 렌더링 작업들을 처리해야하기 때문에 성능이 떨어지는 단점을

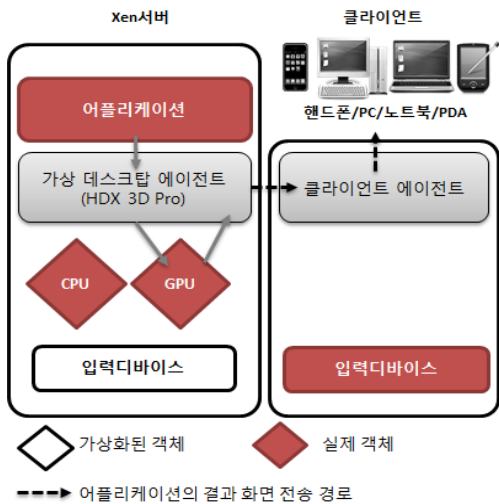


그림 2. Citrix의 XenApp의 개념도

[그림 2]은 Citrix의 XenApp의 개념도를 보여준다. XenApp는 소프트웨어를 제공하는 XenApp서버, Citrix Licensing서버, 웹 인터페이스로 구성되어 있으며, 클라이언트 PC에서는 커널 기반의 소프트웨어 가상화 기술을 이용하여 소프트웨어를 가상적으로 설치하며, XenApp 서버에서 사용자에게 소프트웨어를 온디맨드로 스트리밍하여 제공한다. 사용자는 다양한 사용자 단

가지고 있다.

마지막으로 VMWare는 SDDC(Software-Defined Data Center)인프라를 통하여 가상화를 이용한 클라우드 컴퓨팅의 솔루션들을 제공하고 있다. 또한, 다양한 단말들을 위한 가상화 업무 환경을 제공하기 위하여 VMware View/Mirage를 제공하고 있다. 이를 통하여 중앙 집중형 윈도우 관리와 보안 및 모바일 데스크탑을 위하여 가상화된 또는 물리적인 이미지를 제공한다. VMware View는 사용자가 사용하는 데스크탑의 이미지를 온라인상에서 서버의 가상화 환경을 통하여 제공하여 다양한 형태의 단말에 데스크탑 업무 환경을 제공한다. VMware Mirage는 View에서 사용하는 데스크탑 이미지를 오프라인 상황에서 사용할 수 있도록 사용자의 단말에 설치 없이 구동하고, 온라인으로 연결되었을 때, 동기화하여 사용자에게 일관성 있는 업무환경을 제공하고 있다. 특히, VMware의 vDGA 방식은 서버에 장착된 GPU를 사용자에게 대응하는 가상 머신이 전용하여 서비스하는 방식이다. 가상 머신에 설치된 GPU의 실제 그래픽 드라이브를 이용하여 PCI Pass-through 방식 통하여 GPU를 이용한 렌더링을 수행한 후 그 결과를 클라이언트로 전송하는 방식이다. 이는 3D 그래픽 렌더링 어플리케이션을 사용하는 3D 디자이너 등을

위한 방식이다. 그러나 이 방식은 하나의 GPU를 한 사용자가 전용할 수 있기 때문에 고성능의 그래픽 처리를 지원하지만, 다수의 사용자에 대한 고가의 GPU가 필요하기 때문에 서비스 확장에 따른 인프라 구축비용이 고 비용이라는 단점이 있다.

본 논문은 이러한 문제점을 해결하기 위하여 그래픽 명령어들을 서버에서 실행되는 것이 아니라 클라이언트로 전송하여 실행하는 새로운 어플리케이션 가상화 방식을 제안한다.

III. 가상 3D콘텐츠의 실시간 방송 서비스

현재 방송에서 사용되는 콘텐츠 생성 및 제작에 관련된 시스템은 기존의 2D(Dimension) 방송을 위해 개발되었다. 많이 사용되고 있는 콘텐츠 생성 시스템 중의 하나인 가상 스튜디오(Virtual Studio)는 블루 스크린에서 촬영된 배우의 영상과 미리 준비된 2차원 배경이나 3차원 컴퓨터 그래픽을 크로마키(Chroma-Key)에 의해 합성하는 방식을 사용한다. 따라서 배우는 파란색 옷을 입고 촬영할 수 없는 제약을 가지고 있고 단순한 색상의 치환이기 때문에 깊이에 따른 장면 연출에 많은



그림 3. 제안하는 시스템의 개념도

제약들이 따른다. 또한 배경이 3차원 컴퓨터 그래픽일 지라도 단순히 파란색 대신에 배경을 삽입하는 방식이므로 복수의 배우와 복수의 컴퓨터 그래픽 모델이 중첩되는 상황은 연출하기가 매우 어려운 것이 현실이다. 본 논문에서 제안하는 시스템은 이러한 문제점들을 해결하는 다시점 입체 영상 서비스를 지원한다.

[그림 3]은 제안하는 시스템의 전체 개념도를 보여준다. 실시간 방송을 위한 가상 3D 객체들은 여러 개의 3D 모델들, 이를 조작하는 인터페이스, 그리고 배경음악이나 효과 사운드들로 구성되고 이러한 가상 3D 객체들이 모여 서버에는 농구장과 같은 가상 월드가 구성된다. 사용자는 클라이언트 에이전트 프로그램을 통하여 가상 월드 서버에 접속이 가능하고 이를 통하여 농구시합대회의 방송 시청이 가능하다. 기존 방송 방식은 모든 시청자가 방송 시스템에서 생성한 획일적인 영상만을 볼 수 있다. 그러나 제안된 시스템에서는 사용자마다 제어 가능한 자신만의 가상 카메라를 가질 수 있기 때문에 사용자별 카메라를 통하여 사용자는 스스로 자신이 원하는 영상부분을 선택하고 감상할 수 있다. 기존 소프트웨어 가상화 방식도 사용자가 원하는 다시점 서비스를 제공할 수 있지만 서버에서 사용자가 원하는 부분에 해당하는 3D객체들을 모두 렌더링하고 그 결과들을 전송하기 때문에 서버의 부하가 가중되는 문제점이 있다.

VI. 분할실행기법

앞에서 언급한 상용제품들은 서버에서 소프트웨어를 실행하고 그 결과 화면을 클라이언트에 전송하는 방식이다. 이러한 VDI 방식은 사용자마다 GPU를 독점적으로 사용하거나 한 개의 GPU를 여러 명이 공유하기 때문에 사용자 수가 증가하면 서버의 부하가 가중되는 방식으로 서비스 확장성에 한계점이 있다. 결론적으로, 기존 방식들은 가상 콘텐츠의 3D 렌더링을 필요로 하는 실시간 방송 서비스에 적합하지 않다. 본 논문은 이러한 한계점을 극복하기 위하여 3D 렌더링을 필요로 하는 어플리케이션 실행하기 위해 필요한 작업들의 일부

를 클라이언트로 분산시키는 분할실행기법을 제안한다. 제안하는 기법은 서버에서 어플리케이션을 실행시키고 그 결과화면만을 클라이언트로 보내는 것이 아니라 화면을 생성하기 위해 사용되는 그래픽 명령어를 서버에서 실행하지 않고 클라이언트로 전송하여 실행하도록 하는 방식이다. 이 방식은 전송 데이터양이 매우 적고 각각 클라이언트의 그래픽 장치를 사용함으로써 서버에 부하를 주지 않기 때문에 네트워크 부하가 많은 실시간 방송 서비스에 매우 적합하다.

[그림 4]는 본 논문에서 제안하는 소프트웨어 서비스의 분할실행기법의 개념도를 보여준다. 분할실행 SW 서버에는 사용자 정보 데이터베이스와 3D 소프트웨어가 있고, 사용자로부터 소프트웨어의 사용 요청이 들어오면 그에 해당하는 소프트웨어를 서버에서 실질적으로 실행한다. 기존 방식과의 차이점은 소프트웨어를 실행하는 과정에서 렌더링에 관련된 그래픽 명령어를 서버에서 처리하지 않고 클라이언트로 전송하여 클라이언트의 그래픽 장치를 이용하여 처리하는 것이다. 즉, 실시간 방송화면은 클라이언트에서 가상 월드와 캐릭터들의 3D 렌더링 작업을 통하여 완성된다.

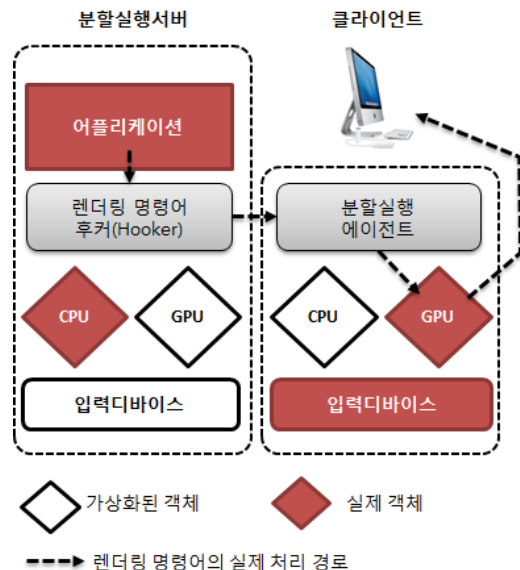


그림 4. 분할실행기법의 개념도

제안하는 방식은 이러한 분할된 실행을 위하여 실행되는 어플리케이션으로부터 그래픽 명령어를 후킹하는 과정이 필요하다. 윈도우 OS (Operating System) 환경에서 일반적인 어플리케이션들은 디스플레이를 위해 DirectX, OpenGL, GDI와 Win32 그래픽 라이브러리를 사용한다. 제안하는 방식에서는 이러한 그래픽 라이브러리와 같은 API를 후킹하는 사용자레벨 API 후킹 방식을 사용한다. 이 방식은 서버의 어플리케이션에 의하여 호출되는 라이브러리를 사용자 라이브러리로 대체하는 방식이나 직접 API를 후킹하는 방식을 사용한다. 예를 들면, 클라이언트가 소프트웨어의 실행을 요청하면, 서버에서는 소프트웨어를 기존 시스템에 있는 DirectX 라이브러리가 아닌 가상 라이브러리를 바인딩하여 실행하여 인터페이스 메소드나 함수들을 후킹해서 클라이언트 측으로 함수 호출을 전달하여 실행함으로써 분할 실행한다. 서버쪽은 그래픽카드(GPU)와 입력디바이스가 가상화되어 처리되고 클라이언트쪽은 CPU와 어플리케이션이 가상화되어 처리된다.

예를 들어 서버의 응용프로그램이 그래픽 관련 함수 CreateWindowEx()을 호출 했을 경우 아래와 같은 절차를 통하여 실행 및 처리된다.

1. 서버는 응용 프로그램의 그래픽 관련 함수 CreateWindowEx()를 후킹한다.
2. 서버는 그래픽 관련 함수 CreateWindowEx()의 호출 정보들을 네트워크를 통하여 클라이언트에 전달한다.
3. 클라이언트는 전달된 그래픽 관련 함수 CreateWindowEx()를 실행하여 실제 윈도우즈 창을 생성한다.
4. 클라이언트는 서버에 그래픽 관련 함수 CreateWindowEx()의 리턴 값 그리고 사용자의 마우스 및 키보드 관련 정보를 전송한다.
5. 서버는 클라이언트의 함수 리턴 값 및 관련 변수들을 저장 및 관리한다.

윈도우 OS에서 실행되는 소프트웨어들의 대부분은 2D 렌더링과 3D 렌더링이 혼재되어 있는 것이 일반적

이다. 3D와 2D 화면을 따로 따로 처리하면 둘 간의 동기화 문제가 발생하기 때문에 3D 렌더링을 클라이언트 상에서 분할 실행하는 경우와 같이 2D 렌더링도 동일한 방식으로 처리해야 한다. 기본적인 2D 그래픽 라이브러리는 비트맵, 폰트와 텍스트, 도형, 드로잉, 펜, 브러쉬, 변환, 프린팅 등과 같은 기능을 제공하는 매우 방대한 함수들로 구성되어 있다.

제안하는 실시간 방송 서비스는 사용자마다 카메라 제어를 통하여 선택적인 방송을 제공하기 때문에 동종 또는 이종의 소프트웨어가 서버에서 동시에 실행될 수 있어야 한다. 제안하는 방식은 이를 위하여 서버에서 실행되는 프로세스들 간에 서로 영향을 받지 않고 독립적으로 해당 소프트웨어를 실행하도록 윈도우 가상화 기술을 제공한다. 여러 개의 분할실행 어플리케이션을 위한 윈도우들이 동시에 생성되면 윈도우들은 서로 다른 우선순위로 선정됨에 따라 다른 윈도우에 영향을 주거나 받을 수 있는 간섭 효과가 생긴다. 그래서 윈도우가 서버 상에 가상으로 생성되게 하여 여러 개의 윈도우들 간의 간섭 효과를 피한다.

분할실행기법의 시스템 구성은 크게 서버와 클라이언트로 나누어진다. 서버의 주요기능에는 클라이언트의 요청을 받아서 분할실행 소프트웨어의 실행 및 종료하는 기능과, 실행소프트웨어로부터 그래픽작업을 추출하는 그래픽작업 추출기능, 그 추출된 그래픽 명령어들을 클라이언트에 전송하는 기능, 그리고 클라이언트로부터 전송되는 메시지 및 이벤트 정보와 사용자의 입력 정보를 서버로 수신하는 기능이 포함된다. 클라이언트는 서버의 분할실행 소프트웨어 실행을 요청하는 분할실행 소프트웨어 실행기능, 클라이언트에서 발생한 키보드, 마우스 등의 입력 정보를 추출 및 서버로 송신하는 기능, 그리고 서버로부터 받은 그래픽 작업을 화면에 표시하는 기능을 가지고 있다. 제안하는 기법의 가장 중요한 장점은 기존 솔루션이 서버의 GPU 자원을 독점하거나 공유하는 방식에 비하여 서버의 GPU 자원을 거의 사용하지 않기 때문에 다사용자 서비스를 위한 서버의 확장이 매우 용이하게 한다. 분할실행기법은 이러한 서버의 확장성이 좋기 때문에 많은 사용자들이 동시에 서비스를 요청하는 가상 3D 콘텐츠의 실시간 방

송 서비스에 기존의 방식들보다 적합하다.

VI. 성능평가

제안하는 분할실행기법의 성능평가는 MS의 RemoteFX(RD Session), Citrix사의 ZenDesktop와 비교하였다. 서버에 하나의 클라이언트를 연결하는 1:1 성능평가에는 3D 객체들을 이용하여 건축물을 설계하는 Archispace LT[13]를 사용하여 서버의 GPU 사용량과 CPU 사용량을 측정하였다. 그리고 서버에 여러 대의 클라이언트를 연결하는 1:N 성능평가는 실행소프트웨어로 Direct3D의 튜토리얼 GPUBoids[14]을 사용하여 클라이언트의 FPS를 측정하였다.

[표 2]는 성능평가에서 사용되는 서버 및 클라이언트의 사양을 보여준다. 제시된 Citrix 서버와 MS RemoteFX 서버의 사양들은 다사용자를 지원하기 위한 최소의 사양이다. 이에 비하여 분할실행서버는 일반 PC용 그래픽카드를 사용하여 성능평가를 하였다.

표 2. 성능평가를 위한 시스템 세부 사양

종류	세부사양
Client PC	Intel Core i7 3.5GHz, Ram 8G, NVIDIA GeForce GTX 570
Citrix HDX 서버	Intel Core i7 3.4GHz, Ram 16G, NVIDIA Quadro FX3800 4GB x 4 (480만원)*
MS RemoteFX 서버	Intel Xeon X5650 2.66GHz, Ram 20G, NVIDIA Quadro 6000 8GB (500만원)*
분할실행서버	Intel Core i7 2.8GHz, Ram 4G, NVIDIA GeForce GTS 480 (40만원)*

* 2013년 04월로 산정된 가격입니다.

[그림 5]는 Archispace LT 프로그램을 사용하여 3D 건축물 객체를 360도 회전하면서 렌더링된 건물에 화면에 연속적으로 보여줄 때 서버의 GPU 사용량을 측정 한 것이다. 실험 결과는 제안하는 분할실행기법이 Citrix HDX 서버와 MS RemoteFX 서버에 비하여 매우 적은 GPU 자원을 사용하는 것을 보여준다. 이러한 이유는 분할실행기법이 Archispace LT 프로그램에서 3D 렌더링처리를 할 때 발생하는 그래픽 명령어를 서버에서 클라이언트로 전송하여 클라이언트 GPU를 사

용하여 실행하기 때문에 분할실행서버의 GPU 자원을 사용하지 않기 때문이다.

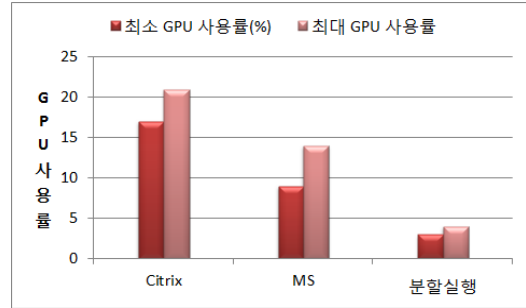


그림 5. 서버의 GPU 사용률 측정

[그림 6]은 역시 3D 건축물 객체를 360도 회전하면서 서버의 CPU 사용률을 측정 한 것이다. 제안하는 분할실행기법이 다른 서버에 비하여 매우 적은 CPU 자원을 사용하는 것을 보여준다. 이러한 이유는 어플리케이션의 모든 연산을 서버에서 처리하는 기존의 방식보다 서버의 부하가 클라이언트로 분산되었기 때문이다. 그래서 분할실행서버에는 서버의 CPU 자원의 사용률이 적다.

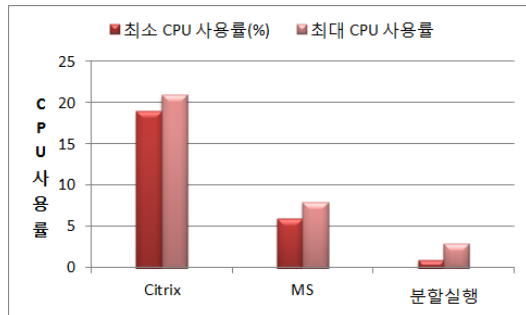


그림 6. 서버의 CPU 사용률 측정

[그림 7]은 각 서버에 같은 클라이언트의 수를 증가시키면서 클라이언트에서 실행되는 GPUBoids 프로그램의 FPS를 측정 한 것이다. 사용자가 한명일 때는 제안하는 기법과 RemoteFX의 FPS가 거의 비슷하다. 그러나 사용자가 증가할수록 RemoteFX는 서버 GPU의 여러 클라이언트가 공유하기 때문에 클라이언트의 FPS가 급격히 감소하는 것을 알 수 있다. 이에 비하여

제안하는 기법은 클라이언트의 수가 증가함에 따라 FPS의 감소폭이 적은 편이다.

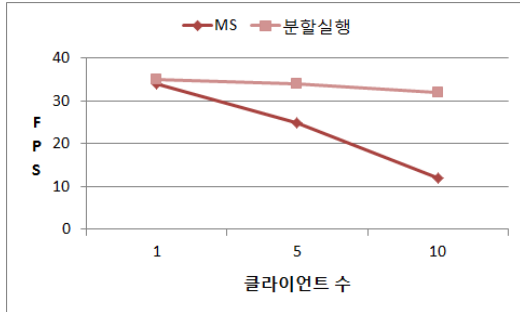


그림 7. 클라이언트 수 증가에 따른 FPS 측정

VI. 결론

본 연구는 클라우드 컴퓨팅 기술을 이용하여 가상 3D 콘텐츠의 실시간 방송 서비스하기 위한 새로운 어플리케이션 가상화 기법인 분할실행기법을 제안하였다. 제안하는 분할실행기법은 방송을 시청하는 사용자의 증가에 따른 기존 방식의 서버 과부하 문제점을 해결하고 시청자에게 제어 가능한 카메라를 제공함으로써 방송을 시청하는 과정에서 사용자의 편의성을 극대화시켰다. 그리고 다양한 성능평가를 통하여 가상 3D 콘텐츠의 실시간 방송 서비스를 위한 서버구축비용을 획기적으로 줄이는 우수성을 입증하였다. 향후 연구는 실 사용자의 요구사항을 수렴 및 반영하여 실 서비스를 구축하기 위해 성능 향상과 사용자 편의성을 추가 보완하는 것이다.

참고 문헌

[1] 장수민, 양경아, 최원혁, 정문영, 김원영, "고화질 소프트웨어 서비스의 기술 동향 및 전망", 한국통신학회지, 제30권, 제4호, pp.38-44, 2013.
 [2] 송진외, "클라우드 컴퓨팅 기반의 3D GIS 구축 사례에 관한 연구", 한국정보과학회 논문지, pp.29-34,

2012.
 [3] 이병엽, 박준호, 유제수, "고가용성 클라우드 컴퓨팅 구축을 위한 그리드 소프트웨어 아키텍처", 한국콘텐츠학회논문지, 제12권, 제2호, pp.19-29, 2012.
 [4] 이명선, 오형용, 민병원, 오용선, "모바일 클라우드 컴퓨팅에 최적화된 모바일 웹 사용성 개선", 한국콘텐츠학회논문지, 제11권, 제9호, pp.85-95, 2011.
 [5] 지수미, 이정중, 김성국, 우경덕, 백성욱, "효율적인 3D 게임 및 애니메이션 콘텐츠 제작을 위한 직관적인 저작 기술 개발", 멀티미디어학회논문지, 제13권, 제5호, pp.780-791, 2010.
 [6] 이형묵, 우운택. "증강현실 콘텐츠 제작을 위한 감각형 상호작용 기반 3D 모델 조립 시스템", 신호처리합동학술대회, pp.1-4, 2008.
 [7] Citrix XenDesktop, <http://www.citrix.com/>
 [8] RemoteFX of Microsoft, <http://www.microsoft.com/en-us/server-cloud/windows-server/>
 [9] Mirage of Vmware, <http://www.vmware.com/>
 [10] F. Lamberti and A. Sanna, "Streaming-based solution for remote visualization of 3D graphics on mobile devices," IEEE Transactions on Visualization Computer Graphics, pp.247-260, 2007.
 [11] D. Koller, M. Turitzin, M. Levoy, M. Tarini, G. Croccia, P. Cignoni, and R. Scopigno, "Protected interactive 3D graphics via remote rendering," ACM Transactions on Graphics, pp.695-703, 2004.
 [12] Lars Ailo Bongo, Grant Wallace, Tore Larsen, Kai Li, and Olga Troyanskaya, "Systems support for remote visualization of genomics applications over wide area networks," GCCB2006, pp.157-174, 2007.
 [13] Archispace LT, <http://archispace.gongtools.com>
 [14] GPUBoids, <http://msdn.microsoft.com>

저 자 소 개

장 수 민(Su-Min Jang)

정회원



- 1997년 2월 : 목포대학교 전산통계학과(이학사)
- 1999년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2007년 8월 : 충북대학교 정보통신공학과(공학박사)

▪ 2011년 5월 ~ 현재 : 한국전자통신연구원 소프트웨어연구부문 선임연구원

<관심분야> : 온라인게임서버, 어플리케이션 가상화, 센서네트워크 및 RFID, 데이터마이닝, 데이터베이스

김 원 영(Won-Young Kim)

정회원



- 1989년 2월 : 이화여자대학교 전산학과(이학사)
- 1991년 2월 : 한국과학기술원 전산학과(공학석사)
- 1998년 2월 : 한국과학기술원 전산학과(공학박사)

▪ 1999년 1월 ~ 현재 : 한국전자통신연구원 소프트웨어연구부문 책임연구원

<관심분야> : 클라우드 컴퓨팅, 어플리케이션 가상화, 데이터베이스

유 재 수(Jaesoo Yoo)

종신회원



- 1989년 2월 : 전북대학교 컴퓨터공학과(공학사)
- 1991년 2월 : 한국과학기술원 전산학과(공학석사)
- 1995년 2월 : 한국과학기술원 전산학과(공학박사)

▪ 1996년 8월 ~ 현재 : 충북대학교 정보통신공학부 및 컴퓨터정보통신연구소 교수

<관심분야> : 데이터베이스시스템, 빅데이터, 센서네트워크 및 RFID, 소셜네트워크서비스, 분산객체컴퓨팅, 바이오인포매틱스 등