그래픽 분할 실행 환경에서 가상 비트맵 관리 방법

양경아*, 최지훈**, 김원영*, 정문영*
*한국전자통신연구원 클라우드컴퓨팅연구부
**(주)유니인포
e-mail : *{kayang, wykim, mchung}@etri.re.kr, **autumn1026@gmail.com

A Method of Virtual Bitmap Management for Graphic Separated Software Execution Environment

Kyungah Yang*, Jihun Choi**, Wonyoung Kim*, Moonyoung Chung*
*Dept. of Cloud Computing, ETRI
**Uniinfo Co., Ltd

요 약

최근 데스크탑 가상화 서비스와 관련한 솔루션들이 많아지고 있다. 하지만, 이들 솔루션은 클라이언트의 수가 늘어날수록 서버의 부담이 커지며, 특히 3D 그래픽을 지원하는 고사양의 SW 를 지원하기 어려운 한계를 가지고 있다. 이러한 제약을 극복하기 위해 서버에서 데이터 집중적인 작업을 실행하고, 클라이언트 PC 에서는 그래픽 처리와 같은 사용자 인터페이스 작업을 분담하여 실행하는 서버 기반 그래픽 분할 실행 기술[4]이 제안되었다. 본 논문에서는 그래픽 분할 실행 기술을 개발하는데 있어 이미지 비트맵 관련 함수가 실행되었을 때 나타나는 비트맵 불일치 현상을 해결하기 위해 백업 비트맵을 이용한 가상 비트맵 관리 기법을 제안하여 궁극적으로 분할 실행 기술 개발을 지원하고자 한다.

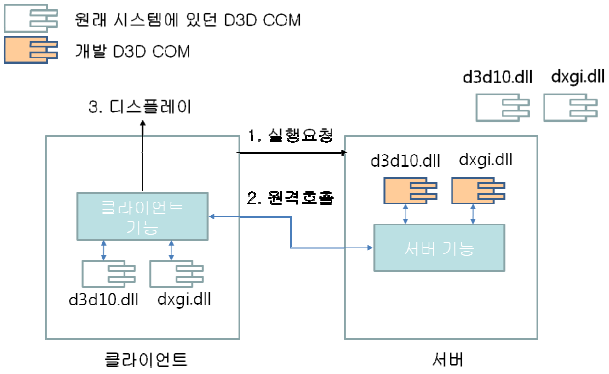
1. 서론

데스크탑 가상화 기술은 SBC(Server Based Computing, 서버기반 컴퓨팅)의 한 종류로 사용자가 씬 클라이언트와 같은 로컬 디바이스를 이용하여 데이터 센터 서버 내의 가상 데스크탑에 접속해 데이터, 운영체제 및 응용 프로그램 등을 활용하는 기술이다[1]. 대표적인 가상화 솔루션 업체인 VMWare[2], Ctrix[3] 등은 데스크탑 가상화 서비스와 관련한 솔루션들을 제공하고 있다. 하지만 이들 솔루션은 클라이언트의 수가 늘어날수록 서버의 부담이 커지고, 3D 그래픽을 지원하는 고사양의 SW 를 지원하기 어려운 한계를 가지고 있다.

이러한 한계를 극복하기 위해서 [4]에서는 서버 기반 SW 서비스의 분할 실행 기술을 제안하였다. [4]는 데스크탑 SW 를 서버에서 서비스로 제공하는 기술로서 고사양의 데스크탑 SW 들이 그래픽 데이터를 클라이언트 PC 의 자원을 활용하여 클라이언트에서 처리하는 방식을 지원한다. 또한 서버에서 가상 머신 단위의 데스크탑 서비스가 아니라 실행되는 SW 의 프로세스 단위의 사용자 서비스를 제공한다. 서버에서 데이터 집중적인 작업을 실행하고, 클라이언트 PC 에서는 그래픽 처리와 같은 사용자 인터페이스 작업을 분담하여 실행함으로써 데스크탑 가상화 기술에 비해 서버의 부하가 적고, 컨텐츠에 대한 보안을 제공한다.

본 논문에서는 서버 기반 그래픽 분할 실행 기술을 개발하는데 있어 이미지 비트맵 관련 함수가 실행되었을 때 나타나는 비트맵 불일치 현상을 해결하기 위해 가상 비트맵 관리 기법을 제안하여 궁극적으로 분할 실행 기술 개발을 지원하고자 한다.

2. 관련연구 및 연구배경

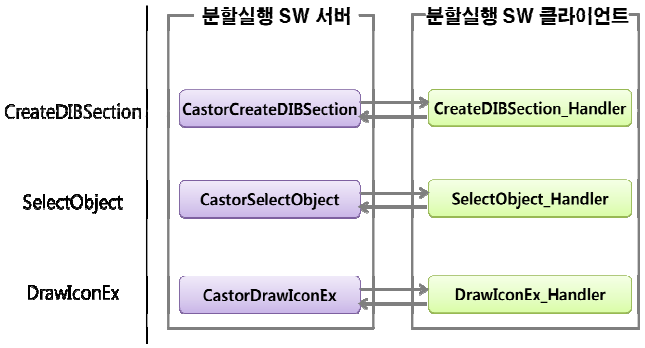


(그림 1) 서버 기반 그래픽 분할 실행 기술의 예

(그림 1)은 서버 기반 그래픽 분할실행 기술의 예로서 서버에서 구동되는 SW 의 3D 그래픽 API 를 클라이언트에서 처리하기 위한 방식을 보인 것이다. 클라이언트가 임의의 3D SW 를 실행 요청하면, 서버에서는 실행 요청된 SW 를 자체 개발한 DirectX 라이브러리로 바인딩하여 실행한다. 개발된 DirectX 라이브러리는 자체적으로 개발된 미들웨어를 이용하여 3D SW

가 생성하는 라이브러리 호출을 클라이언트로 전송한다. 라이브러리 호출을 수신한 클라이언트는 클라이언트 시스템의 DirectX 라이브러리를 호출하고 이에 대한 처리 결과는 클라이언트 디스플레이 장치에서 렌더링된다. 렌더링 된 후 처리에 대한 반환 값들은 서버로 전송된다[5].

(그림 1)에서와 같이 분할 실행 기술을 적용할 경우, 클라이언트에서 아이콘과 같은 비트맵 이미지가 제대로 그려지지 않은 경우가 발생하는 것을 확인하였다. 이러한 현상은 다음과 같은 함수 호출 과정에서 보여질 수 있다.



(그림 2) 분할실행 환경에서 함수 호출 예

SW 에서 비트맵 이미지를 그리기 위해 (그림 2)와 같은 순서대로 CreateDIBSection, SelectObject, DrawIconEx 등의 함수가 호출된다고 가정해보자.

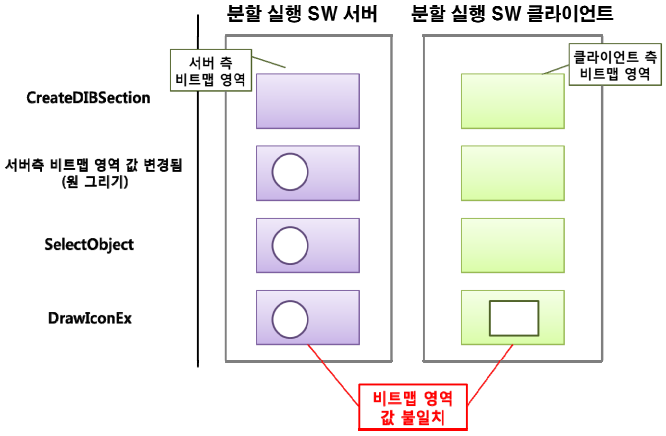
해당 함수들은 Win32 API 함수들로 비트맵을 처리하는데 주로 사용된다. CreateDIBSection 는 비트맵 영역을 화면에 보이기 위해 필요한 데이터를 모두 포함하고 있는 DIBSection 을 생성하는 함수이다. SelectObject 는 디바이스 컨텍스트에 펜이나 브러시, 폰트, 비트맵 등의 객체를 선택할 수 있는 기능을 수행한다. 그리고 DrawIconEx 는 아이콘 이미지를 지정된 영역에 출력하는 함수이다.

분할 실행 환경에서는 분할 실행 SW 서버에서 분할 실행 SW 를 실행하고, 실행할 때 호출되는 함수들의 정보를 클라이언트로 전송한다. 클라이언트에서는 그래픽 관련 함수 정보를 받아 자신의 GPU 를 이용해 렌더링하여 화면에 표시한다.

이와 같은 순서대로 서버 측에서 CreateDIBSection 함수가 호출이 되면 자체 개발된 라이브러리인 CastorCreateDIBSection 함수가 바인딩되어 호출된다. CastorCreateDIBSection 함수는 CreateDIBSection 을 수행하기 위한 매개변수 값들을 프로토콜 메시지에 저장하여 클라이언트 측으로 전송하는 역할을 수행한다.

클라이언트에서는 이에 대응하는 가상 라이브러리인 CreateDIBSection_Handler 함수가 호출되고 전송된 프로토콜 메시지를 해석하여 매개변수 값들을 추출하여 저장한다. 저장된 매개변수 값들은 클라이언트 측에서 실질적인 CreateDIBSection 함수를 수행하는데 사용된다. 클라이언트에서 CreateDIBSection 함수가 정상적으로 수행된 후, 성공적으로 수행되었다는 리턴 값을 서버 측에 보낸다. SelectObject 와 DrawIconEx 함수 역시 상기에 기술한 방식과 유사한 방식으로 서버와 클라이언트 간 통신이 이루어지면서 분할 실행 처리된다.

비트맵을 그리는 과정에서 상기 기술한 방식으로 분할 실행 SW 가 처리되었을 때, 서버에서 생성되는 비트맵과 클라이언트에서 만들어지는 비트맵 간 불일치가 일어나 수행 결과의 신뢰성이 떨어지는 경우가 발생할 수 있다.



(그림 3)서버-클라이언트 간 비트맵 값 불일치

(그림 3)은 서버와 클라이언트 간 비트맵 정보가 불일치 되는 상황을 보인 것이다. 분할 실행 서버에서 CastorCreateDIBSection 이 호출되면 서버 측에서는 매개변수 값들을 비롯한 관련 정보들을 프로토콜 메시지에 저장시켜 클라이언트로 보낸다. 클라이언트 측에서는 메시지를 받아 매개변수 값을 저장하여 실질적인 CreateDIBSection 함수를 수행하고 그 결과로 빈 비트맵 영역을 생성한다. 클라이언트에서는 성공적으로 비트맵 영역이 생성되었다는 리턴 값을 서버에 전송한다.

서버에서는 성공한 리턴 값을 받고 클라이언트 측과 마찬가지로 빈 비트맵을 만든다. 이때 서버 측에서 개발자가 빈 비트맵을 조작하는 작업(그림에서와 같이 빈 비트맵에 원형 그리기)을 수행하였을 때, 그 작업이 void 형 함수로 수행되었다면 서버 측 비트맵에만 원 그리기가 반영되고 클라이언트의 비트맵에는 반영되지 않는다. 즉, 서버에서는 원이 그려진 비트맵을 가지게 되고 클라이언트는 변경된 비트맵 정보가 반영이 되지 않고 계속 빈 비트맵을 가지게 된다. 이러한 상태에서 서버에서는 SelectObject 함수를 비롯한 DrawIconEx 함수가 호출될 수 있다.

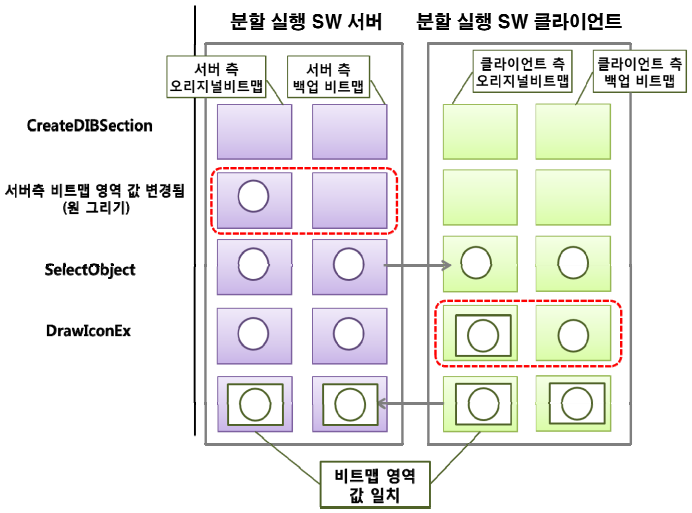
서버에서 DrawIconEx 함수가 호출되었을 때 CastorDrawIconEx 함수가 바인딩 되어 호출되고 클라이언트 측에 프로토콜 메시지를 전송한다. 클라이언트에서는 대응되는 DrawIconEx_Handler 함수가 호출되고 클라이언트의 실질적인 DrawIconEx 함수가 수행된다. 수행 결과로 클라이언트의 빈 비트맵 영역에 아이콘이 생성되지만 서버 측 비트맵 영역에는 원 이미지만 그려진 형태로 남게 된다.

이같이 비트맵 관련 함수를 실행하면서 서버 측과

클라이언트의 비트맵이 불일치 되는 경우를 보완하기 위해 본 논문에서는 백업 비트맵을 이용해 차이를 보정해 나가는 방법을 제안한다.

3. 개선 방안

분할 실행 SW 를 수행하는데 서버와 클라이언트 간 비트맵 영역 불일치를 해소하기 위해 백업 비트맵을 이용함으로써 차이를 보정할 수 있다.



(그림 4) 백업 비트맵을 이용한 가상 비트맵 관리

(그림 4)는 (그림 3)에서 보인 순서대로 함수를 수행하는 화면으로 백업 비트맵이 추가된 것을 볼 수 있다. 백업 비트맵을 이용해 관련 함수를 처리하는 과정은 다음과 같다.

1. 서버 측에 CreateDIBSection 함수가 호출되면 클라이언트 측에 대응하는 함수가 호출되고, 빈 오리지널 비트맵 비트가 생성된다. 이때, 백업용 비트맵 비트가 같이 클라이언트 측에 생성된다. 클라이언트 측에서 성공적인 리턴 값을 서버 쪽으로 반환하면 서버 역시 가상으로 오리지널 비트맵 비트와 백업용 비트맵 영역을 생성한다.
2. 서버 측에서 CreateDIBSection 함수를 수행하면 그 결과로 비트 배열 포인터 값을 넘겨받을 수 있다. 서버 측에서는 그 포인터 값을 이용해 원을 그리는 것과 같은 비트맵 값을 변경하는 작업을 수행할 수 있다.
3. 서버 측에서 그 다음 함수로 SelectObject 를 호출할 때, 오리지널 비트맵과 백업 비트맵을 비교해서 값이 다른지 확인한다. 오리지널 비트맵과 백업 비트맵의 값이 다르면 오리지널 비트맵의 값을 백업 비트맵에 반영한다. 즉, 백업 비트맵에도 원을 그려준다. 서버 측에서 오리지널 비트맵과 백업 비트맵의 값을 비교했을 때 값이 달랐다면, 값이 다르다는 사실을 클라이언트 측에 알려주고 변경된 값의 정보(원그리기)도 같이 전송한다. 클라이언트 측에서는 서버 측에서 변경된 값을 이용해 클라이언트 측의 오리지널 비트맵과 백업 비트맵에 각각 반영한다.
4. 서버 측에서 DrawIconEx 를 호출할 때, 함수 수행 정보는 클라이언트 측으로 전송되며 클라이언트의 DrawIconEx 함수를 통해 실질적 수행이 이루어진다. 클라이언트 측에서는 비트맵에 아이콘을 그리기 위해 오리지널 비트맵에 대한 포인터 정보를 넘겨받고 아이콘을 오리지널 비트맵에 그린다. 그 후 오리지널 비트맵 값과 백업 비트맵 값을 비교해서 이 둘의 값이 다른지 확인한다. 값이 다르면 오리지널 비트맵의 값을 백업 비트맵에 반영한다. 그리고 이 비트맵 간의 차이가 발생한 사실을 서버 측에 알려준다.
5. 서버 측에서 클라이언트가 전송한 메시지 정보를 이용해 오리지널 비트맵과 백업 비트맵에 변경된 값(아이콘 그리기)을 반영한다.

이와 같이 분할 실행 과정에서 비트맵을 출력할 때, 서버 측과 클라이언트 측에 각각 오리지널 비트맵과 백업 비트맵을 두고 전송 과정 중 이 두 비트맵의 값을 비교하면서 정보의 정확성을 체크해볼 수 있다.

4. 결론

본 논문에서는 서버 기반 분할 실행 서비스에 대해 살펴보고 분할 실행 SW 가 비트맵 관련 함수를 실행하는데 있어 서버-클라이언트 간 비트맵 정보가 불일치하는 현상에 대해 알아보았다. 이를 보완하기 위해 백업 비트맵을 이용하여 오리지널 비트맵 값과의 비교함으로써 정보 불일치성을 확인할 수 있는 가상 비트맵 관리 기법을 제안하였다.

본 논문에서 제안한 기법을 통해 사용자에게 보다 안정화된 분할 실행 서비스를 제공할 수 있게 되었다. 향후에는 서버 기반 분할 실행 SW 가 최적화되어 수행될 수 있도록 GDI 함수들을 중심으로 성능 개선을 위한 연구가 필요하다.

참고문헌

[1] 박영민, “업무용 PC 가상화”, 한국지역정보개발원 지역정보화 동향분석, 제 4 호, 2010
[2] http://www.vmware.com/products/view/overview.html
[3] http://www.citrix.com/xendesktop
[4] 김기헌, 최원혁, 김원영, “GPU 가상화 기반 3D 원격 렌더링”, 제 34 회 한국정보처리학회 추계학술대회 논문집 제 17 권 2 호, pp.1652~1654, 2010.11.
[5] 최원혁, 김원영, “서버 기반 SW 서비스에서 API 리모팅 기반의 GPU 가상화를 이용한 그래픽 분할 실행의 구현”, 한국인터넷정보학회 논문지 제 12 권 6 호, pp.53~62, 2011.12.

본 연구는 지식경제부 산업원천기술개발사업의 일환으로 수행하였음. [10035185, 서버 기반 SW 서비스의 분할 실행 기술 개발]