

# 서버 기반 SW 서비스의 분할실행을 위한 효율적인 스레드 관리 방법

정문영\*, 최지훈\*\*, 최원혁\*, 김원영\*  
한국전자통신연구원 실시간분석인프라연구팀\*  
유니인포\*\*

e-mail : mchung@etri.re.kr, autumn1026@gmail.com, whchoi@etri.re.kr, wykim@etri.re.kr

## An Efficient Thread Management for the Client-side Graphic Rendering on a Server-based Software Service

Moonyoung Chung\*, Jihoon Choi\*\*, Won-Hyuk Choi\*, Won-Young Kim\*  
Electronics and Telecommunications Research Institute\*  
Uniinfo\*\*

### 요 약

본 논문에서는 서버 기반 SW 서비스의 분할 실행 기술에서 API 리모팅 방법을 통해 서버에서 실행되는 SW 의 그래픽 렌더링 작업을 클라이언트에 전송할 때, 효율적인 스레드 및 버퍼 관리 방법을 제안하였다. 또한, 기존의 API 리모팅 방법과 비교하여 우수한 성능을 제공함을 실험을 통해 보여주었다.

### 1. 서론

PC 기반 컴퓨팅 환경에서 발생하는 데이터 보안, 다수의 사용자 PC 관리 비용 증가 등의 문제를 해결하기 위해 서버 기반의 컴퓨팅 환경이 보급되고 있다. 서버 기반 컴퓨팅 환경을 지원하는 기술로는 Citrix 의 XenDesktop, VMWare 의 VDI 그리고 Microsoft 의 RDP(Remote Desktop Protocol)을 기반한 터미널 서비스 등이 있다. 이러한 서버 기반 컴퓨팅 기술을 이용하면 서버에서 SW 를 실행하고 클라이언트 PC 는 단순히 터미널 역할을 수행하기 때문에 클라이언트 수가 늘어날수록 서버의 부하가 커지게 된다. 서버에서 SW 를 실행하고 결과 이미지를 클라이언트에 전송하는 방식을 사용함으로써 특히 3D 렌더링과 같은 고성능 그래픽 작업을 처리하는 데는 한계점을 가진다. 따라서 고사양 클라이언트 PC 자원의 낭비, 서버 부하에 따른 서비스 한계, 막대한 서버 구입 비용, 느린 서비스 성능 등의 단점을 가진다.

서버 기반 SW 서비스의 분할 실행 기술은 데스크탑 SW 를 서버와 클라이언트에서 분할하여 실행하는 사용자 인터페이스 가상화에 대한 기술로, 서버 기반 SW 분할 실행 기술을 활용하면 CPU 활용률이 높아짐으로써 여러 대의 서버를 추가로 설치한 효과를 얻을 수 있어, 가상화에 의한 비용 삭감 및 소프트웨어

라이선스 비용 삭감 면에서도 매우 중요한 기술이다.

즉, SW 분할 실행 기술을 활용하면 3D 렌더링과 같은 그래픽 집중한 작업들은 클라이언트 PC 에서 실행하고, 데이터 집중한 작업들은 서버에서 처리해서 서버에 집중되는 부하를 줄이고, 클라이언트에서 처리 및 관리되는 사용자 데이터에 대한 보안을 제공할 수 있다.

서버 기반 SW 서비스의 분할 실행 기술에서 3D 그래픽 렌더링과 같은 작업을 클라이언트 PC 에 분할 실행하는 방법으로 서버에서 구동되는 SW 의 3D 그래픽 API 를 클라이언트에서 처리하기 위한 API 리모팅[1] 방법이 있다.

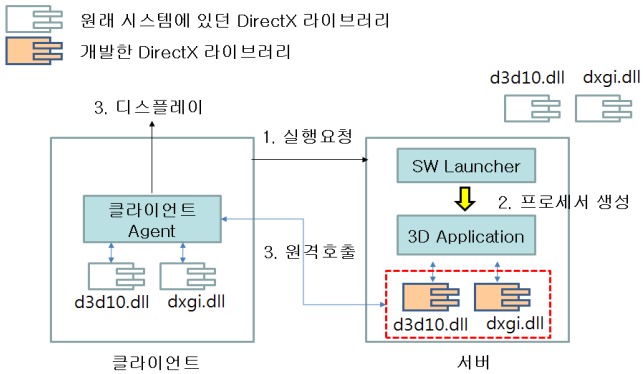
본 논문에서는 API 리모팅 방법을 효율적으로 제공하기 위한 스레드 및 버퍼 관리 방법을 제안하고, 기존의 방법에 비해 우수한 성능을 제공함을 보여준다.

### 2. API 리모팅 기반의 클라이언트 3D 렌더링

아래 그림 1 은 API 리모팅 기반의 3D 그래픽 클라이언트 렌더링 구조이다[1]. 클라이언트가 임의의 3D SW 를 실행 요청하면, 서버의 SW Launcher 는 실행 요청된 SW 를 윈도우 OS 내부의 DirectX 라이브러리가 아닌 자체 개발된 DirectX 라이브러리를 바인딩하여 실행한다. 실행된 3D SW 의 프로세스가 DirectX API 를 호출할 경우 윈도우 OS 시스템 내부의 DirectX 라이브러리가 호출되는 것이 아니라, 이를 대체하는 개발된 DirectX 라이브러리가 호출된다. 개발된 DirectX 라이브러리는 자체적으로 개발된 미들웨어

<sup>1</sup> 본 연구는 지식경제부 산업원천기술개발사업의 일환으로 수행하였음. [10035185, 서버 기반 SW 서비스의 분할 실행 기술 개발]

어를 이용하여 3D SW 가 생성하는 라이브러리 호출을 클라이언트로 전송한다. 라이브러리 호출을 수신한 클라이언트 Agent 는 이를 클라이언트 시스템의 DirectX 라이브러리를 호출하고, 이에 대한 처리 결과는 클라이언트 디스플레이 장치에서 렌더링 된다. 또한, 처리에 대한 반환 값들은 서버로 전송된다[4].



(그림 1) API 리모팅 기반 3D 그래픽 클라이언트 렌더링

이와 같이 API 리모팅을 이용한 클라이언트 렌더링은 윈도우 OS 에서 DirectX COM 객체의 가상화를 통하여 수행된다. 성능 향상을 위하여 라이브러리 호출을 전송하기 위한 미들웨어 레벨에서 전송 캐시 기능을 추가하여 전송할 API 명령이나 데이터를 전송 캐시에 모았다가 한번에 전송하거나, 클라이언트 상에서 실제 렌더링 관련 API 들을 선별하여 전송한다.

### 3. 효율적인 API 리모팅을 위한 스레드 관리 방법

본 논문에서는 API 리모팅 방법에서 서버에서 클라이언트 PC 로 라이브러리를 호출할 때, 이를 효율적으로 하기 위한 스레드 및 버퍼 관리 방법을 제안한다.

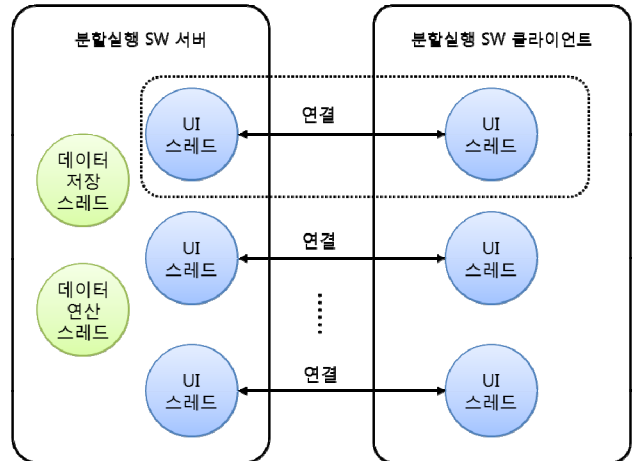
분할 실행 서버에서 SW 를 실행하면 데이터 처리 관련 작업과 그래픽 렌더링, 사운드 재생, 사용자 입력 처리와 같은 UI 관련 작업들을 처리하여야 한다. 이때 데이터 저장이나 연산과 같은 작업들은 데이터 보안을 위해 서버에서만 실행되고, 그래픽 렌더링이나 사운드 재생 작업은 서버로부터 클라이언트로 API 리모팅 방법으로 전송되게 된다. 사용자 입력 처리와 같은 UI 작업은 클라이언트에서 발생하여 서버로 전달되게 된다.

따라서, 아래 그림 2 와 같이 데이터 처리 관련 스레드들은 서버에만 동작하며, UI 관련 스레드들은 서버와 클라이언트에 재현된다. 이때 스레드들의 동시 실행에 의해 발생할 수 있는 간섭 및 데드락 등의 문제를 최소화하기 위해 서버와 클라이언트의 한 쌍의 스레드들은 독립적인 연결 및 송수신 버퍼를 가지게 한다.

네트워크 전송량을 최적화하기 위해서 서버에서는 클라이언트로 보내는 API 들을 버퍼에 쌓아 두었다가 버퍼가 가득 차면 한꺼번에 보내는 서버 버퍼를 두며,

클라이언트에서도 받은 API 를 버퍼에 저장해 두었다가 적절한 실행 시점에 순차적으로 실행하기 위해 클라이언트 버퍼를 둔다.

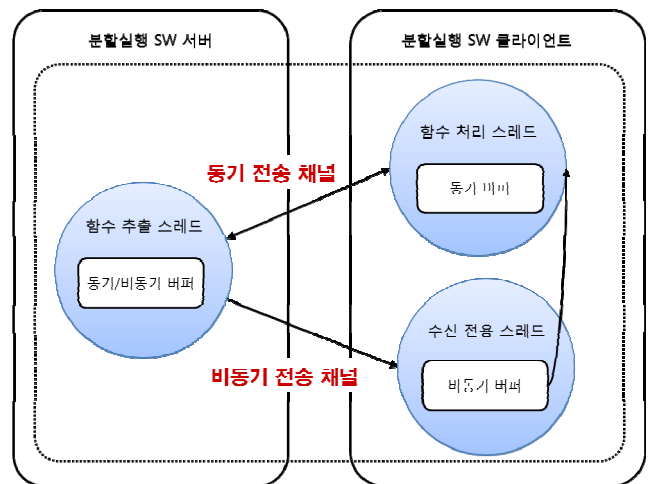
서버와 클라이언트 한 쌍의 스레드들은 API 호출의 종류에 따라 양방향 요청을 처리하기 위한 동기 연결과 한 방향 요청을 처리하기 위한 비동기 연결로 서로 통신한다.



(그림 2) 스레드 연결 방법

즉, API 호출이 반환 값이 있는 경우에는 동기 연결을 통해 바로 결과값을 받아서 처리해야 하며, 반환 값이 없는 경우에는 버퍼에 저장해 두었다가 비동기 연결을 통해 한꺼번에 처리한다.

따라서, 그림 2 의 스레드를 자세히 보면 아래 그림 3 과 같이 클라이언트에는 서버와 동기 연결을 하는 함수 처리 스레드와, 비동기 연결을 하는 수신 전용 스레드로 나뉘게 된다.



(그림 3) API 의 연결 관리 방법

또한, API 리모팅 방법의 성능 개선을 위해 서버에서 클라이언트로 API 를 보내지 않고 내부적으로 처리할 수 있는 함수는 서버에서 단독 처리 하는 방법으로 함수별 처리 로직을 개선하였다.

#### 4. 성능 분석

본 장에서는 API 리모팅 방법에서 서버 클라이언트 간의 서비스 구조 개선을 통한 성능 분석 결과를 보여준다.

본 논문에서는 기존 방법 [1] 에서 서버 클라이언트 간의 반환 값 없는 API 처리를 위한 비동기 수신 전용 스레드를 추가하고, 클라이언트의 동기/비동기 버퍼 기능을 추가함으로써 효율적인 연결관리 방법을 제안하였다.

기존의 API 리모팅 방법[1]과 본 논문에서 제안하는 스레드 및 연결 관리 방법을 적용하였을 때의 성능을 비교하였다. 시험 환경은 윈도우 7 기반의 PC 두 대를 사용하였으며, CPU는 Intel Core i7 2.8GHz, 메모리 4G, 그래픽카드는 NVIDIA GeForce GTS250를 사용하고, LAN 환경에서 시험하였다.

아래 표 1 은 성능 비교 결과를 보여주며, 기존의 방법(v1.0), 본 논문에서 제안하는 방법(v2.0) 및 SW를 로컬 실행 했을 때의 FPS (Frame per Second)를 비교하였다. 성능 측정에 사용된 SW는 아래 표 2 와 같다.

<표 1> 서비스 구조 개선을 통한 성능 비교

SW	v1.0	v2.0	로컬 실행
GPUBoids	30~33	46~48	80~82
ArchiSpace LT	6~7	11~12	38~39
Mass Effect 2	4~5	13~18	60

<표 2> 성능 측정 SW

이름	설명
GPUBoids	새떼와 같은 군집동물의 움직임을 시뮬레이션
ArchiSpace LT	VirtualBuilder사의 상용 설계 SW
Mass Effect 2	DirectX 9 기반 상용 게임 SW

#### 5. 결론

본 논문에서는 SW 서비스의 분할 실행 기술에서 API 리모팅 방법을 통해 서버에서 실행되는 SW의 그래픽 렌더링 작업을 클라이언트에 전송할 때, 효율적인 스레드 및 버퍼 관리 방법을 제안하였다. 서버와 클라이언트 간의 한 쌍의 스레드들은 독립적인 연결 및 송수신 버퍼를 가지며 API 유형에 따라 효율적인 처리를 하기 위한 동기 연결과 비동기 연결을 통해 서로 통신하도록 설계 하였다. 또한, 실험을 통해 기존 방법보다 효율적임을 보여주었다.

향후 연구 과제로는 API 리모팅 방법에서 API 유형 및 유형별 분포에 따른 성능 개선 방법에 대한 연구가 필요하다.

#### 참고문헌

- [1] 최원혁, 김원영, “서버 기반 SW 서비스에서 API 리모팅 기반의 GPU 가상화를 이용한 그래픽 분할 실행의 구현”, 인터넷정보학회 논문지, 12 권, 6 호, pp.53-62, 2011 .
- [4] 김기현, 최원혁, 김원영, "GPU 가상화 기반 3D 원격 렌더링", 제 34 회 한국정보처리학회 추계학술대회 논문집 제 17 권 2 호(2010.11), pp.1651~1654
- [2] B. C. Cumberland, G. Carius and A. Muir, “Microsoft Windows NT Server 4.0, Terminal Server Edition: Technical Reference.” Microsoft Press, Redmond, Wash, 1999.
- [3] Citrix Metafram(<http://www.citrix.com>)
- [4] Virtual Network Computing (<http://www.realvnc.com>)