소프트웨어 서비스(Software as a Service): 핫스팟 문제점을 해결하기 위한 통합시스템에 관한 연구

장수민, 최원혁, 김원영
한국전자통신연구원 SW 서비스연구팀
e-mail : {jsm, whchoi, wykim}@etri.re.kr

Software as a Service: A Study on Integration System for Mitigating Hotspot Problem

Su-Min Jang, Won-Hyuk Choi, Won-Young Kim
Software Service Research Team of Electronics and Telecommunications Research Institute

요 약

최근 컴퓨터 가상화 기술이 발전됨에 따라 필요한 소프트웨어를 서비스 형태로 사용하는 소프트웨어 서비스(Software as a Service SaaS)가 많은 응용분야에서 사용되고 있다. 그러나 소프트웨어 서비스를 제공하는 기존 시스템은 사용자의 증가에 서버 성능이 급속히 저하되는 문제점과 일시적인 사용자 폭증으로 생기는 핫스팟(Hotspot)에 안정적인 서비스를 제공하지 못하는 문제점을 가지고 있다. 이러한 문제점들을 해결하기 위하여 본 논문은 소프트웨어 실행을 위한 작업들이 모두 서버에서 실행되는 것이 아니라 데이터 집중적인 작업들은 서버에서 직접 실행하고 그래픽 집중적인 작업들은 네트워크 전송을 통하여 클라이언트에서 처리되는 분할 실행 방식과 개별적으로 운영되는 SaaS 서버들을 하나로 통합하는 시스템을 제안한다.

1. 서론

최근 컴퓨터 가상화 기술이 발전됨에 따라 클라우드 컴퓨팅을 사용하는 응용분야가 급속도로 확대되고 있으며 이러한 클라우드 컴퓨팅을 통하여 IT 관련된 소프트웨어를 서비스 형태로 제공되는 사용하는 소프트웨어 서비스(Software as a Service SaaS)가 활성화되고 있다. 소프트웨어 서비스의 경우 사용자들은 지원하는 기술에 대한 전문 지식이 없어도 또는 제어할 줄 몰라도 네트워크나 인터넷으로부터 서비스를 이용할 수 있다. 또한 정보가 인터넷 상의 서버에 영구적으로 저장되고 데스크톱이나 테이블 컴퓨터, 노트북, 벽걸이 컴퓨터, 휴대용 기기 등과 같은 클라이언트에서 일시적으로 사용하는 개념이다.

이러한 SaaS 의 가장 큰 장점은 인터넷에 접속할 수만 있다면 어떤 기기에서 다 사용 가능하다는 것이다. 운영체제에 상관없이 컴퓨터나 태블릿, 스마트폰 등과 같이 여러 기기에서 바로 접속할 수 있어 언제 어디서나 쉽게 사용하고 소프트웨어가 업데이트 될 때마다 새로운 버전을 설치할 필요도 없다. 또한 SaaS 는 필요한 소프트웨어를 사용한 만큼을 사용료를 내는 방식으로 비용 부담이 적은 것이 특징이다.

SaaS 의 실용 서비스로 MS 의 RemoteFX [1]에 기반한 터미널 서비스, Citrix 의 XenDesktop/HDX [2] 그리고 Vmware 의 View [3] 등이 있다. 이러한 데스크탑 가상화 기술은 서버에서 구동 되는 가상 머신(Virtual Machine)이나 서버의 세션 관리 기술 등을 통하여 클라이언트를 관리하고, 소프트웨어의 실행을 전적으로 서버나 서버의 가상 머신에서 실행하고, 실행된 결과 화면을 클라이언트의 터미널로 전송하는 방식이다 [4][5][6]. 클라이언트는 단순히 실행 화면이 스트리밍 된 데이터를 받아 디코딩하여 화면에 표시하는 터미널 역할 만을 수행한다. 그래서 서비스하는 클라이언트의 수가 늘어날수록 서버의 부하가 가중되기 때문에 다수 사용자에게 서비스하기 위한 서비스의 확장성에 문제점을 가지고 있다. 또한, 기존의 데스크탑 가상화 기술은 3D 그래픽을 이용하는 고사양의 소프트웨어를 서비스할 시에는 다수의 가상 머신이 서버의 그래픽 장치를 공유해야 하므로, 다수의 클라이언트에 대한 서비스를 제공할 때에 성능이 저하되는 단점이 있다. 이외에도 특정한 회사나 단체가 서버를 개별적으로 구축하여 사용하면서 일시적인 사용자 폭증으로 생기는 핫스팟(Hotspot)에 원활한 서비스를 제공하지 못하는 문제점을 가지고 있다.

2. 분할 실행 기법

본 논문은 SaaS 서버에서 서비스하는 클라이언트의 수가 늘어날수록 서버의 부하가 급격히 증가하는 문제점을 해결하기 위하여 클라이언트의 화면에 표시하

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

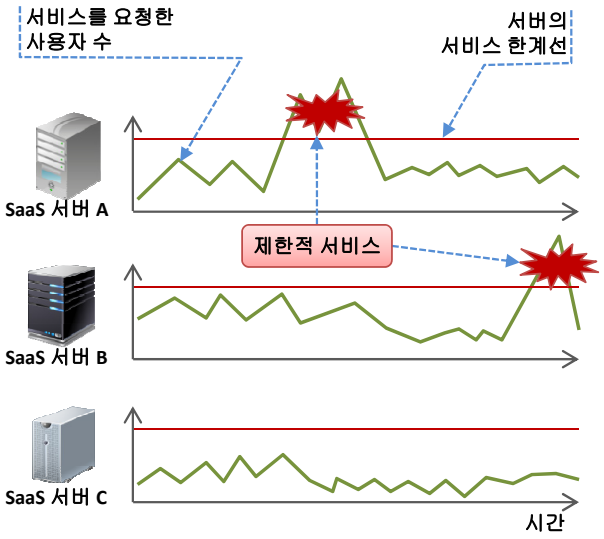
는 그래픽 명령어를 서버의 그래픽 장치가 아닌 클라이언트의 그래픽 장치에서 실행하는 분할 실행 기법을 제안한다. 그림 1 은 분할 실행 기법의 시스템 구조를 보여준다.

(그림 1) 분할 실행 기법의 시스템 구조

서버에서는 클라이언트의 요청을 받아서 분할 실행 소프트웨어의 실행 및 종료하는 기능과, 실행 소프트웨어로부터 그래픽 작업을 추출하는 그래픽 작업 추출 기능, 그 추출된 그래픽 작업들을 클라이언트에 전송하는 기능, 그리고 클라이언트에서 발생된 입력을 수신 및 처리하는 기능이 포함된다. 클라이언트는 서버의 분할 실행 소프트웨어 실행을 요청하는 분할 실행 소프트웨어 실행 기능, 클라이언트에서 발생한 키보드, 마우스 등의 입력 정보를 추출 및 서버로 송신하는 기능, 그리고 서버로부터 받은 그래픽 작업을 처리해서 화면에 표시하는 기능을 가지고 있다. 분할 실행 소프트웨어는 서버에서 일부 실행되지만 클라이언트의 화면에 표시하는 소프트웨어의 실행은 서버의 그래픽 장치가 사용되지 않고 클라이언트의 그래픽 장치를 사용한다. 이러한 분할 실행으로 서버의 부하가 클라이언트로 효과적으로 분산된다. 서버로부터 받은 그래픽 작업을 처리해서 화면에 표시하는 기능을 가지고 있다. 분할 실행 소프트웨어는 서버에서 일부 실행되지만 클라이언트의 화면에 표시하는 소프트웨어의 실행은 서버의 그래픽 장치가 사용되지 않고 클라이언트의 그래픽 장치를 사용한다. 이러한 분할 실행으로 서버의 부하가 클라이언트로 효과적으로 분산된다.

3. SaaS 통합시스템

그림 2 은 소프트웨어 서비스를 제공하는 서버들에서 일시적인 사용자의 폭증으로 인한 핫스팟이 생긴 예시를 보여준다. 서버 A, B, C 세개의 서버가 SaaS 형태로 서비스를 하고 있고, 이때 서버 A 와 B 에서 핫스팟 상태가 발생되어 일부 사용자가 제한적 서비스로 원활한 서비스를 제공받지 못하고 있다. 이를 해결하고자 보다 많은 시스템 증설이나 고성능을 서버를 설치하기에는 적절하지 않다.

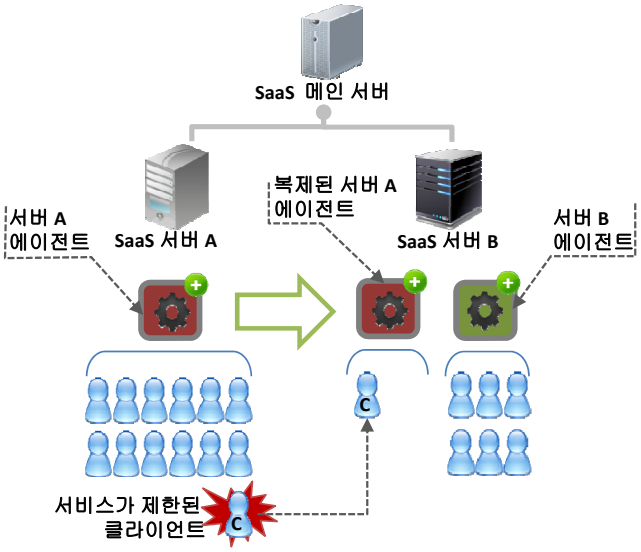


(그림 2) SaaS 서비스에서 핫스팟 발생

이러한 문제점을 해결하기 위해서 제안하는 통합 시스템에는 각각 개별적으로 서비스하던 서버들을 하나로 통합하여 서버들 간에 서로 보조하여 핫스팟으로 인한 제한된 서비스를 해결하도록 설계하였다. 이를 위하여 기존 서버 에이전트에 새로운 기능들을 추가할 필요가 있다. 추가될 새로운 기능에는 핫스팟이 발생되더라도 다른 서버의 여유 자원을 이용하여 원활한 서비스가 되도록 다른 서버로 서비스를 복제하는 기능이 추가되어야 한다. 예를 들면 서버 A 의 핫스팟이 발생한 시점에 서버 B, C 가 서버 A 의 서비스를 보조함으로써 보다 원활한 서비스를 제공하는 것이다.

SaaS 통합시스템을 사용할 경우 개별적으로 서비스할 경우와 다르게 핫스팟을 발생시키는 서비스를 통합된 서버들과 협조하여 처리할 수 있다. 그림 3 는 SaaS 통합시스템에서 핫스팟이 발생하고 이를 처리하는 과정을 보여준다. 서버 A 에서 핫스팟이 발생되었다. 이에 서버 A 는 서버 A 의 소프트웨어 서비스가 제한된 클라이언트 C 를 서버 B 에서 서비스를 받도록하기 위하여 서버 B 에 관련 데이터들을 복제하고 있다. 이때 서비스 복제를 위해서 전송되는 파일들은 서비스 소프트웨어 실행 파일, 서비스 데이터 파일, 서버 에이전트 실행 파일 등이다. 여기서 추가된 SaaS 메인 서버는 각각 서버의 여유 자원에 관한 정보 및 접속을 위한 정보 등을 관리하는 기능을 한다.

대부분의 소프트웨어 서비스는 서버/클라이언트 방식으로 네트워크를 통하여 서비스가 이루어지고 있다. 서버에는 사용자들의 계정관련 데이터베이스, 서비스를 위한 소프트웨어, 서비스를 위한 관련 파일들, 그리고 클라이언트의 서비스 요청을 받아 처리하는 서버 에이전트 등이 있다. 그리고 클라이언트 쪽에는 클라이언트 에이전트와 서비스에 관련된 개인적인 자료 등이 있다.

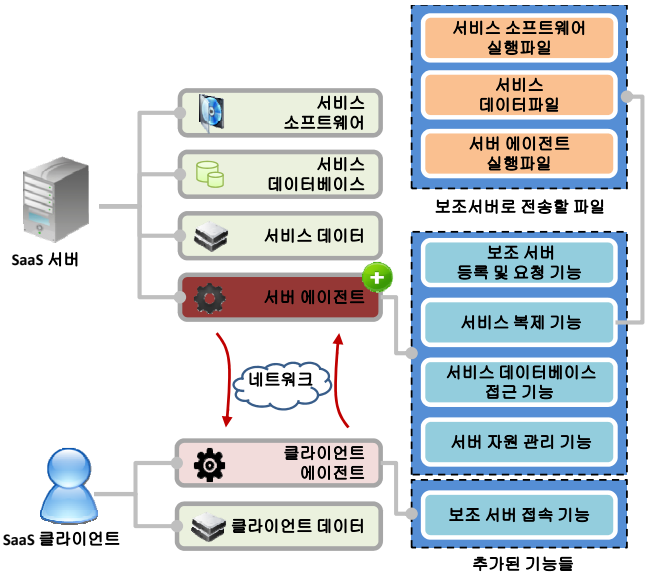


(그림 3) SaaS 통합시스템의 핫스팟 처리

이러한 기능들로 구성된 서버가 독자적으로 처리하던 서비스를 다른 서버들과 통합적으로 처리하기 위해서는 기존의 서버 에이전트 기능에 추가적으로 여러 가지 기능들이 필요하다. 그 추가적인 기능은 아래와 같다.

- 보조 서버 등록 및 요청 기능
- 서비스 복제 기능
- 서비스 데이터베이스 접근기능
- 서버 자원 관리 기능

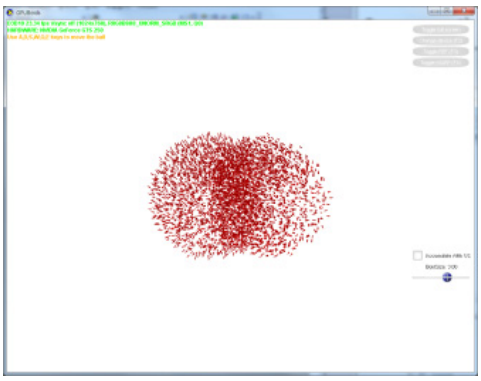
그림 4 은 제안하는 SaaS 통합시스템의 구성도를 보여준다. 보조 서버 등록 및 요청 기능은 각각의 서버가 IP 주소, 포트번호와 같은 네트워크를 통하여 접근에 필요한 정보들, 그리고 서비스 가능한 소프트웨어 목록 등을 메인 서버에 등록을 위한 것이다. 또한 등록된 서버들의 정보를 통하여 접속 가능한 서버의 정보를 요청하는 기능도 포함된다. 서비스 복제 기능에는 보조 서버로 전송할 파일들을 송수신하고 복제된 서버 에이전트를 기동시키는 기능이 있다. 여기서, 미리 서비스를 위한 파일들을 통합한 모든 서버에 미리 관련 파일들을 복사하지 않는 이유는 보조 서비스가 필요한 시점에 가장 최신의 에이전트로 서비스하기 위한 기능이다. 서비스 데이터베이스 접근 기능은 물리적 데이터베이스의 위치에 상관 없이 데이터베이스에 데이터를 접근하기 위한 기능이다. 서버 자원 관리 기능은 메인 서버에 각 서버의 자원 상태를 보고하는 기능, 보조 서버로써 사용 가능한 자원 관리 기능, 그리고 사용 후, 불필요한 자원을 회수하는 기능 등이 포함된다. 클라이언트 에이전트 경우에는 기본 SaaS 서버에 접속하는 기능에 추가적으로 보조 서버 접속 기능하는 기능이 포함된다.



(그림 4) 제안하는 SaaS 통합시스템

4. 성능평가

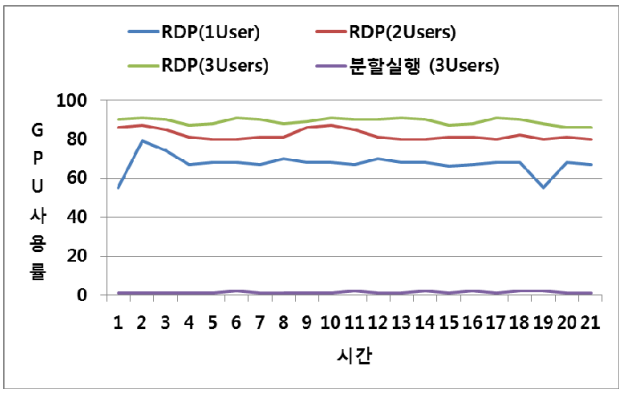
MS 의 RDP 와 분할 실행 기법의 다수 사용자를 위한 서버 및 클라이언트의 성능을 평가하기 위하여 그림 5 와 같은 Direct3D 의 3D 버텍스로 구성된 많은 객체들이 군무형태로 이동하는 튜토리얼 GPUBoids [7] 프로그램을 사용하였다.



(그림 5) Direct3D 의 튜토리얼 GPUBoids

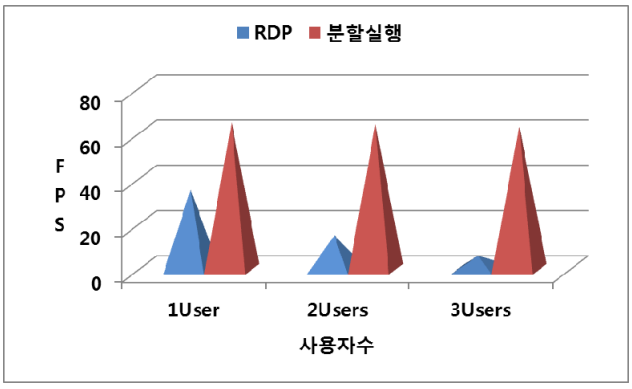
그림 6 은 RDP 와 분할 실행 기법의 그래픽 카드 (GPU) 사용률을 성능 평가한 것이다. 서버는 Intel Core i7 2.8GHz, 메모리 4G, 그래픽 카드 NVIDIA GeForce GTS 25 이고 클라이언트는 Intel Core i5 2.53GHz, 메모리 4G, 그래픽 카드 NVIDIA GeForce GT 330 를 사용하였다. RDP 의 경우 접속한 클라이언트가 서버의 그래픽카드를 공유하기 때문에 클라이언트 수가 증가하면 GPU 사용률이 증가하게 된다. 한 서버에 동시 접속한 클라이언트 수가 1 인 경우 RDP 의 GPU 사용률이 약 60%~70% 정도를 차지하고 동시 접속한 클라이언트 수가 3 인 경우 거의 90% 이상을 사용하는 것을 보여준다. 하지만 분할 실행 기법의 경우는 그래픽에 관련 API 은 서버가 아닌 클라이언트 그래픽카드를 이용하여 수행되어 화면에 표시되기 때문에

동시 접속한 클라이언트 수가 3 인 경우에도 GPU 사용률이 약 3% 이하로 매우 적다. 이처럼 본 논문에서 제안하는 분할 실행 기법은 서버의 부하가 클라이언트로 효과적으로 분산되기 때문에 사용자 증가하더라도 안정적인 서비스를 제공하는데 매우 적합하다.



(그림 6) 사용자 수에 따른 GPU 사용률

그림 7 은 RDP 와 분할 실행 기법의 사용자 증가에 따른 클라이언트의 최소 FPS 을 측정한 것이다. 역시 사용된 프로그램은 GPUBoids 이다. 이때, 서버는 Intel Core i7 2.8GHz, 메모리 4G, 그래픽 카드 NVIDIA GeForce GTS 25 이고 모든 클라이언트는 Intel Core i7 3.20GHz, 메모리 12G, 그래픽 카드 NVIDIA GeForce GTX 480 을 사용하였다. 한 서버에 접속한 클라이언트 하나인 경우 RDP 의 최소 FPS 가 35 이었으나 동시 접속한 클라이언트 수가 3 인 경우 최소 FPS 가 7 로 급격이 떨어지는 것을 보여준다. 이에 비해 그래픽 명령어를 각 클라이언트에서 처리하는 분할 실행 기법은 동시 접속한 클라이언트 수의 증가에 영향을 받지 않는 것을 보여준다.



(그림 7) 사용자 증가에 따른 최소 FPS

5. 결론

본 논문은 한 SaaS 서버에서 다수 사용자에게 소프트웨어 서비스를 제공하는데 발생되는 서버의 과부하를 획기적으로 감소시키기 위하여 클라이언트의 화면에 표시하는 그래픽 명령어를 서버의 그래픽 장치가 아닌 클라이언트의 그래픽 장치에서 실행하는 분할 실행 기법을 사용하였다. 그리고 기존의 개별적으로 구축되어 서비스하는 SaaS 서버들을 하나로 통합하는 시스템을 제안하였다. 우리는 이러한 통합 시스템을 통하여 일시적으로 사용자의 폭증으로 인한 핫스팟(Hotspot)에 의한 사용자들에게 원활한 서비스를 제공하고 다수의 사용자에게 적은 시스템 유지 비용으로 보다 효율적이고 안정적인 서비스를 제공할 것이라 확신한다. 향후 연구는 제안하는 통합 시스템의 기능들을 개선 및 구현하여 실제 응용 분야에 적용하고 이를 통하여 보다 최적화된 통합 시스템을 개발하는데 있다.

참고문헌

[1] Brian Craig Cumberland, Gavin Carius and Andrew Muir, “Microsoft Windows NT Server 4.0, Terminal Server Edition: Technical Reference.” Microsoft Press, Redmond, Wash, 1999.

[2] Citrix XenDesktop, http://www.citrix.com/

[3] Vmware VDI, http://www.vmware.com/Roger S. Pressman. “Software Engineering, A

[4] Fabrizio Lamberti and Andrea Sanna, “Streaming-based solution for remote visualization of 3D graphics on mobile devices,” IEEE Transactions on Visualization Computer Graphics, 13 (2) (2007), pp. 247–260

[5] David Koller, Michael Turitzin, Marc Levoy, Marco Tarini, Giuseppe Croccia, Paolo Cignoni, Roberto Scopigno, “Protected interactive 3D graphics via remote rendering,” ACM Transactions on Graphics (TOG), v.23 n.3, August 2004

[6] Lars Ailo Bongo, Grant Wallace, Tore Larsen, Kai Li, and Olga Troyanskaya. “Systems support for remote visualization of genomics applications over wide area networks,” In Proc. of GCCB’06. LNBI 4360, 2006.

[7] GPUBoids, http://msdn.microsoft.com