

가상데스크탑 제어를 위한 디바이스 가상화 프레임워크의 설계 및 구현

(Design and Implementation of a Device Virtualization Framework to control Virtual Desktop)

강 상 우 [†] 박 세 진 [†]
(Sangwoo Kang) (Sejin Park)

박 찬 익 ^{**}
(Chanik Park)

요 약 본 논문에서는 웹 브라우저를 통하여 로컬 디바이스를 쉽게 네트워크 디바이스로 바꾸어주는 프레임워크를 제안한다. 해당 네트워크 디바이스들은 가상데스크탑에 사용된다. 관리 소프트웨어는 디바이스풀을 유지하며 디바이스들을 관리한다. 사용자의 요청이 발생하면 데스크탑의 I/O를 관리하는 가상데스크탑 소프트웨어가 풀에 접근하여 디바이스를 선택, I/O를 데스크탑에 전달해준다. 본 논문에서는 로컬 컴퓨터에 설치가 필요하지 않은 RIA 솔루션을 채택함으로써 사용자는 쉽게 자신의 디바이스를 디바이스풀에 등록 또는 해지할 수 있으며 가상데스크탑 소프트웨어에 명령을 내릴 수 있게 된다. 본 논문의 후반부에서는 제안하는 프레임워크가 사용될 수 있는 시나리오를 설명하고 성능평가

· 본 연구는 지식경제부 및 정보통신 연구진흥원의 IT R&D 프로그램의 연구 결과로 수행되었음(2008-S034-01, Development of Collaborative Virtual Machine Technology for SoD (System on-Demand) Service)

· 이 논문은 제36회 추계학술발표회에서 '가상데스크탑 제어를 위한 디바이스 가상화 프레임워크의 설계 및 구현'의 제목으로 발표된 논문을 확장한 것임

[†] 학생회원 : POSTECH 컴퓨터공학과
kkangeva@postech.ac.kr
baksejin@postech.ac.kr

^{**} 종신회원 : POSTECH 컴퓨터공학과 교수
cipark@postech.ac.kr
논문접수 : 2009년 12월 28일
심사완료 : 2010년 3월 28일

Copyright©2010 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제16권 제6호(2010.6)

진행함으로써 프레임워크의 효율성 및 활용성을 보인다.

키워드 : 가상데스크탑, 디바이스 가상화, RIA

Abstract In this paper, we propose a framework to change local devices to networked devices by using web browser. Network devices will be used in the virtual desktop. Management software maintains a pool of devices to manage network devices. Whenever user's request is occurred, virtual desktop s/w which manages desktop's I/O access device pool. Then virtual desktop s/w selects proper device and forwards I/O to desktop. In this paper, we choose RIA solution for install-free environment. Consequently, user easily enrolls and revokes their devices. User easily commands virtual desktop s/w. In the second half of this paper, we explain usage scenario and evaluate our framework to show efficiency and flexibility of the framework.

Key words : Virtual Desktop, Device Virtualization, RIA

1. 서 론

초기의 가상화 연구는 비싼 하드웨어의 컴퓨팅 리소스를 공유하기 위해서 시작되었다. 시간이 흐르자 하드웨어의 가격이 떨어지게 되었고 공유 필요성이 없어지면서 가상화 연구들은 위축되게 되었다. 하지만 근래에는 높은 하드웨어 성능을 소프트웨어가 모두 이용할 수 없게 되었고 이로 인해 컴퓨팅 자원, 관리비용 낭비의 문제가 대두되었다. 따라서 소프트웨어가 사용 못하고 있는 가용한 컴퓨팅 리소스에 대한 이용을 최대화하고자 가상화에 대한 연구가 다시 활발히 진행되고 있다. 가상화 연구에도 많은 분야가 존재하며 최근 주요하게 연구되고 있는 가상화 분야는 여러 서버컴퓨터를 가상화하여 하나의 물리 컴퓨터 위에서 동작시키는 서버 가상화, 워크로드 마이그레이션, 보안, 실패복구 등이다. 특히 기존에는 대부분의 연구가 서버 가상화에 대해서 진행이 되었는데 반해 최근에는 서버 기반의 데스크탑 가상화 분야도 활발히 연구가 진행되고 있다.

DaaS(Desktop as a Service)라는 용어로도 불리우는 서버 기반의 데스크탑 가상화란 일반 사용자가 사용하는 개인용 컴퓨터를 가상화하는 것을 말한다. 이는 서버들에 여러 개의 가상데스크탑들을 설치하고 유저에게 서비스해주는 형태의 구조를 가진다. 이 기술을 일반기업에서 도입할 경우, 모든 데스크탑의 관리가 중앙에서 가능하기 때문에 최신의 데스크탑 상태를 쉽게 유지할 수 있으며 가상데스크탑들의 네트워크를 적절히 제어함으로써 인해 보안상 장점을 가진다. 또한 컴퓨팅 리소스를 좀 더 효율적으로 사용함으로써 새로운 컴퓨터를 구매하는데 사용되는 비용을 절감할 수 있게 된다. 이 같은

장점들로 인해서 데스크탑 가상화 분야의 연구가 활발하게 진행되고 있다.

데스크탑 가상화 기술이 점점 퍼지고 있음으로 인해 가까운 미래에 개인유저가 가상데스크탑을 하나 이상 가지고 있는 것을 예상하기란 어렵지 않다. 현재의 서버 기반의 데스크탑 환경에서는 사용자가 가상데스크탑을 접근하기 위해서 전용의 클라이언트 소프트웨어를 설치해야 한다. 그리고 해당 소프트웨어를 통해 자신의 가상 데스크탑을 사용하는 형태로 서비스가 진행된다. 즉, 가상데스크탑에서는 클라이언트 소프트웨어가 구동하는 컴퓨터의 일부 I/O 디바이스만을 사용할 수 있다. 하지만 많은 네트워크 디바이스가 등장하고 있는 상황에서 이는 가상 데스크탑의 융통성 및 확장성을 떨어트리는 단점이 된다. 따라서 하나의 컴퓨터내의 자원만을 사용하는 기존의 접근법에서 벗어나 네트워크 디바이스들을 자유롭게 사용할 수 있는 새로운 형태의 프레임워크가 필요하다. 그리고 이러한 프레임워크에서는 네트워크 디바이스가 아닌 레거시 디바이스들에 대한 지원도 포함되어야 한다. 이전의 연구에서는[1] 네트워크 디바이스가 아닌 레거시 디바이스들을 네트워크 디바이스 형태로 바꾸어주는 연구를 진행하였다. 결국 새로운 형태의 프레임워크 상에서는 모든 디바이스들이 디바이스 풀 형태로 관리되게 되며 요구에 따라서 동적으로 디바이스를 재구성하여 가상데스크탑의 I/O 디바이스들을 구성하게 된다. 이를 통해 일반 사용자에게 가상의 컴퓨터를 제공해 줄 수 있게 된다.

기존 연구에서는 레거시 디바이스를 네트워크 디바이스화하기 위해서 어댑터라고 하는 소형 하드웨어 장치가 추가로 필요하며 레거시 디바이스를 가상화하기 위한 소프트웨어를 구현 및 구현해야 하는 비용이 들게 된다. 또한 레거시 디바이스들은 어댑터를 이용해서만 네트워크 디바이스화가 될 수 있으므로 어댑터에 종속적인 이동성 제약 등을 승계하게 된다.

본 논문에서는 설치가 필요하지 않고 디바이스의 동적인 이동성을 지원하면서 레거시 디바이스를 네트워크 디바이스화 시킬 수 있는 새로운 형태의 프레임워크를 설계하고 구현하였다. 본 논문의 이후 구성은 두 번째 절에서는 우선 관련된 연구들을 소개하고, 설계 및 구현은 세 번째 절에서 설명한다. 그리고 네 번째 절에서는 개발된 프레임워크의 평가를 진행하고 마지막 절에서는 향후 연구 방향 제시 및 결론을 맺는 것으로 구성되어 있다.

2. 관련 연구

가상화 솔루션은 게스트 운영체제의 소스 코드 수정 여부에 따라 크게 반가상화와 전가상화로 구분이 된다.

게스트 운영체제의 소스 코드 수정을 동반하는 반가상화의 경우, 게스트 운영체제의 도움으로 인하여 성능향상을 이끌어내게 된다. 반면 소스코드의 수정을 요구하지 않는 전가상화의 경우는 CPU와 메모리를 적절한 형태로 가상화해줘야 한다. 전가상화의 경우 특권명령에 대한 오버헤드 등이 추가적으로 요구됨으로 반가상화에 비해 성능이 떨어지는 것으로 알려져 있다.

젠 하이퍼바이저는 오픈소스로 개발된 유명한 가상화 솔루션이다[2]. 캠브리지 대학에서 개발되었으며 여러 반가상화 게스트 운영체제들을 하나의 물리 머신에서 동작시켜 단일 운영체제 환경에 근접하는 성능을 얻었다. 젠 또한 3.0 버전부터는 전가상화를 지원하고 있는데 이를 위해서는 CPU가 Intel VT 또는 AMD SVM 기술을 지원해야 한다.

로컬 컴퓨터가 아닌 원격지의 디바이스를 사용하는 연구는 기존에도 많이 진행되었다. SlideShow Commander program에서는[3] 모바일 디바이스로 파워포인트 슬라이드쇼를 제어하는 것과 관련된 연구였다. MobiUS는[4] 작은 화면을 가진 모바일 디바이스의 제약사항을 극복하기 위해서 다른 모바일 디바이스의 화면을 통합하여 좀 더 큰 화면을 제공하는 연구이며, PointRight [5]는 하나의 입력장치를 여러 컴퓨터에서 공유하는 방식을 사용하였다. 마찬가지로 Remote Commander Program은 SlideShow Commander program의 개념을 확장하여 모바일 디바이스 자체를 범용 입력 장치로 사용할 수 있는 방식을 제안하였다. 하지만 위에 언급된 연구들은 모두 특정 하드웨어에 종속적이거나 융통성이 결여된 형태의 프레임워크를 가지고 있는 단점이 있다. 본 논문의 프레임워크를 적용하게 되면 쉽게 모바일 디바이스를 인풋 디바이스로 사용할 수 있으며 다른 디스플레이에 자신의 화면을 쉽게 출력할 수 있게 된다. 그러므로 위에서 언급된 연구들이 해결하고자 하는 문제를 쉽게 해결할 수 있을 것이다.

VNC[6]는 원격지의 마우스와 키보드, 디스플레이 장치를 사용하기 위해서 프로토콜을 정의하여 컴퓨팅 자원을 사용할 수 있다. 하지만 VNC 자체는 원격지의 컴퓨터를 고정되어 있는 다른 컴퓨터에서 사용하는 형태로서, 본 논문에서 해결하고자하는 로컬 컴퓨터의 자원만을 사용할 수 있는 단점을 가지고 있다. 하지만 VNC 프로토콜은 범용적이며 많은 소프트웨어들에서 채택하고 있으므로 본 연구에서는 화면출력을 위한 프로토콜로 VNC를 사용하고 있다.

Stanford 대학에서 연구된 iRoom은 회의장 내의 모든 디바이스를 통합함으로써 사용자나 응용 소프트웨어가 투명하게 그들을 접근할 수 있는 프레임워크를 제안하였다[7]. 마이크로소프트의 EasyLiving의 경우 iRoom

과는 달리 하나의 홈 네트워크를 목표로 설정하고 있다 [8]. iRoom에서는 사용자의 움직임을 카메라로 모니터링 함으로써, 사용자의 이동을 바탕으로 서비스를 계속 이동시키게 된다. iRoom과 EasyLiving은 특정 공간을 디바이스존 형태로 개발하여 사용하는 예를 보여주고 있다. 이러한 형태의 존은 본 논문을 통해서도 충분히 구성이 가능한 형태이다. 그리고 추가적으로 본 논문에서는 존 내에 새로운 디바이스가 나타나더라도 동적으로 이를 처리함으로써 하드웨어나 소프트웨어적인 구성을 새롭게 할 필요가 없는 장점을 가지고 있다.

요약하면, 제시한 프레임워크는 아래와 같은 점에서 기존의 연구들과 차이점을 가진다고 할 수 있다. 첫째, 기존 연구들이 특정 디바이스를 가정하는 형태라면 본 논문의 프레임워크는 좀 더 일반적인 형태의 구조를 취함으로써 확장성을 제공한다. 둘째, 설치가 필요 없는 클라이언트 소프트웨어를 제공함으로써 대상 디바이스의 이동성을 제공하고 유저가 쉽게 디바이스를 사용할 수 있도록 하고 있다.

3. 설계 및 구현

제안하는 시스템은 크게 세 부분으로 나눌 수 있다. 레거시 디바이스를 가상화해 줄 브라우저 내의 클라이언트 소프트웨어, 디바이스 풀과 맵핑정보 등을 관리 할 디바이스 맵핑 관리 소프트웨어, 데스크탑의 실제 I/O를 처리해줄 가상데스크탑 소프트웨어가 그 것들이다. 이들 간의 전체 구조 및 상호 관계는 그림 1과 같은 형태를 가진다.

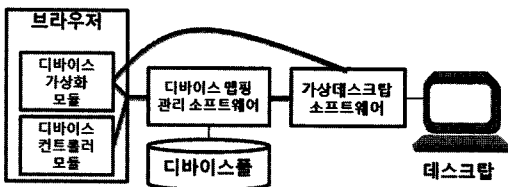


그림 1 시스템 구조도

3.1 RIA 가상화 소프트웨어

RIA 가상화 소프트웨어의 내부 형태는 그림 2와 같다. 해당 모듈은 다시 디바이스의 목록확인 및 할당해제를 수행하는 디바이스 컨트롤러 모듈과 로컬 컴퓨터에 연결된 디바이스를 네트워크 디바이스화 해주는 디바이스 가상화 모듈로 나누어지게 된다. 디바이스 가상화 모듈은 초기에 자신이 가상화하고 있는 디바이스의 목록을 맵핑 관리 소프트웨어에 전송하고 추가 명령이 올 때까지 대기하게 된다. 만약 디바이스 할당 명령이 내려지면 가상데스크탑 소프트웨어와의 I/O 채널이 형성되

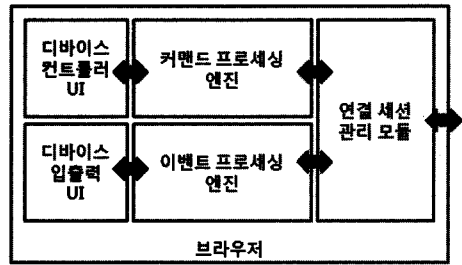


그림 2 RIA 가상화 소프트웨어

고 이후 I/O데이터는 이 채널을 통해 전송된다. 현재 프로토타입으로 모니터, 키보드, 마우스가 구현이 되어있으며 모니터의 경우는 VNC 프로토콜을 선택하였다. 키보드, 마우스는 QEMU가 받아들일 수 있는 형태의 패킷으로 변환이 된 이후 전송이 된다. 이를 통해서 사용자는 자신의 가상데스크탑을 로컬 컴퓨터내의 브라우저에서 접속할 수 있고 풀에 등록된 다른 모니터 디바이스로 화면을 이동시키거나 자신의 컴퓨터내의 입력 장치로 해당 화면을 제어할 수 있게 된다.

디바이스 컨트롤러 모듈을 통해 사용자는 현재 디바이스 맵핑 관리 소프트웨어에 등록된 디바이스의 목록과 가상데스크탑 목록을 확인할 수 있다. 동시에 여러 디바이스 컨트롤러가 동일한 디바이스를 사용하기 위해서 요청을 보낼 경우 디바이스 맵핑 관리 소프트웨어는 가장 먼저 요청을 보낸 디바이스 컨트롤러 모듈의 요청을 우선적으로 처리하게 된다.

클라이언트 소프트웨어는 앞에서 언급했듯이 개인의 이동성과 하드웨어 구매 비용을 발생시키지 않도록 하기 위해서 임의의 시간에 등록 및 해지가 자연스러운 형태로 구현이 되어야 한다. 따라서 소프트웨어를 실제로 로컬 컴퓨터에 설치하지 않고 웹 브라우저를 통해서 수행을 시키는 RIA 솔루션을 채택하였다. RIA를 구현하는 많은 기술들이 존재하지만 본 논문에서는 실버라이트를 사용하였다.

3.2 디바이스 맵핑 관리 소프트웨어

디바이스 맵핑 관리 소프트웨어는 RIA 가상화 소프트웨어를 제공해주는 웹 서버에서 동작하게 된다. 내부적으로 그림 3과 같이 세 개의 매니저를 가지고 있으며 각 매니저들은 자신과 연결된 클라이언트들과의 데이터 전송을 담당하게 된다. 이들 세 개의 매니저는 상호연동을 통하여 유저의 요청을 처리하게 된다. 예를 들면 가상데스크탑풀 관리 매니저는 새로운 가상데스크탑 소프트웨어가 등장하게 되면 자신의 데스크탑풀을 업데이트 하고 이를 디바이스 컨트롤러풀 관리 매니저에게 알려 연결된 모든 컨트롤러들이 해당 정보를 업데이트하도록 돕는다. 그 모든 연산이 정상적으로 수행이 되면 업데이

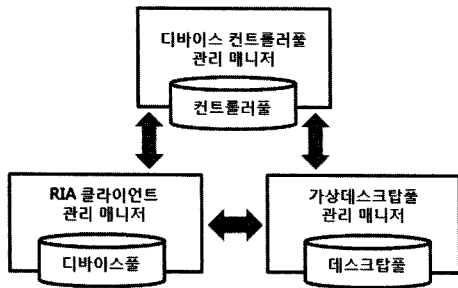


그림 3 디바이스 맵핑 관리 프트웨어

트된 디바이스 목록을 연결된 컨트롤러들에게 보내게 되고 실패했을 경우에는 아무런 연산이 수행되지 않았던 초기상태로 돌아감을 보장하게 된다. 다른 연산들도 이와 같은 과정을 통해서 진행된다.

맵핑 관리 소프트웨어와의 통신은 정해진 프로토콜을 따라서 진행된다. 해당 프로토콜은 디바이스 프로파일과 디바이스 명령에 대한 패킷들을 정의한다. 따라서 네트워크 디바이스가 이 프로토콜을 따를 경우 맵핑 관리 소프트웨어와 상호작용을 할 수 있게 된다. 디바이스 맵핑 관리 소프트웨어는 리눅스 기반의 유저레벨 어플리케이션으로 작성이 되었다.

3.3 가상데스크탑 소프트웨어

가상데스크탑 소프트웨어는 가상화된 데스크탑의 I/O를 처리해주는 역할을 수행한다. 즉, 데스크탑이 사용하고 있다고 믿는 오디오, 마우스, 키보드, 모니터 등 다양한 형태의 자원을 디바이스플에서 가져와서 동적으로 구성, 사용할 수 있도록 해준다.

해당 모듈은 젠 상에서 관리도메인의 응용으로 존재하는 QEMU-DM을 수정하여 작성 되었다[9]. 이는 그림 4에서 나타나 있는데, QEMU-DM의 역할이 게스트 운영체제에게 가상 하드웨어를 제공해주는 것이므로 이 부분의 수정을 통해 네트워크 디바이스의 I/O연산을 가상데스크탑에 전달해줄 수 있게 된다.

가상데스크탑이 아닌 일반 데스크탑 유저가 본 논문

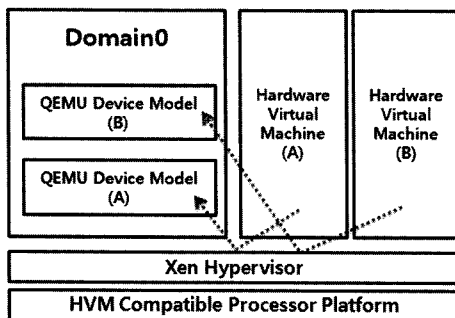


그림 4 가상데스크탑 소프트웨어

에서 제안하는 네트워크 디바이스를 사용하고자하는 시나리오를 생각해 볼 수도 있다. 이 같은 경우에는 디바이스 맵핑 관리 소프트웨어와 연동하기 위해서 정해진 프로토콜을 사용하는 형태로 추가의 소프트웨어를 구현한다면 해결할 수 있다.

4. 성능평가

본 시스템의 소프트웨어 구성은 다음과 같다. 우선 서버 기반 데스크탑 가상화 솔루션으로는 젠 3.3.1버전을 사용하였다. 관리도메인의 커널 버전은 기본으로 제공되는 2.6.18.8 버전이 사용되었다. 그 외 RIA에 사용된 실버버라이트의 경우는 3버전을 사용하였고 VNC프로토콜은 해당 버전의 젠 QEMU-DM에서 기본으로 채택된 프로토콜을 사용하고 있다.

가상데스크탑을 동작시키는 서버의 경우 Dell Power-Edge 6850이 사용되었다. 해당 서버의 경우 Xeon 3.00 GHz CPU, 8G RAM, 200GB SAS 디스크 재원을 가지고 있다.

젠 상에서의 소스코드 수정은 기존의 기능을 변경하는 것이 아닌 추가하는 형태로 진행이 되었다. VNC프로토콜은 기본적으로 내장되어 있는 프로토콜임으로 수정사항이 발생하지 않았으며 다른 장비, 현재로써는 마우스, 키보드 장비를 추가적으로 가상머신에 넣어주는 부분과 관리 소프트웨어와의 명령을 주고받는 부분 등이 추가되었다.

4.1 디바이스 가상화 소요시간

일반유저가 자신의 디바이스를 가상화하기로 결정했을 때 소요되는 시간을 측정하였다. 시간측정은 사용자가 실버라이트 어플리케이션의 디바이스 가상화 버튼을 클릭한 시간부터 해당 디바이스들이 디바이스플에 등록된 이후 사용자가 보는 화면에 갱신되기까지의 시간을 측정한 결과이다.

실험결과 늦어도 0.3초 이내에는 자신의 디바이스를 서버에 등록하고 사용할 수 있는 형태로 서비스됨을 확인할 수 있었다. 그러므로 레거시 디바이스의 가상화 소요시간은 매우 짧아서 프레임워크를 사용하는데 큰 문제가 되지 않음을 알 수 있다.

4.2 디바이스 할당시간

디바이스를 실제로 할당하는 데 걸리는 시간을 측정함으로써 사용자가 실제 디바이스를 할당할 때 어느 정도의 딜레이가 발생하는지 조사해보았다. 시간측정은 사용자가 디바이스 컨트롤러를 통해 모니터 디바이스 할당 명령을 요청하였을 때, 해당하는 모니터의 최초 패킷이 도착할 때까지의 시간을 측정하였다. 간단한 테스트를 위해 동시에 다른 디바이스 컨트롤러가 요청을 보냄으로 인해서 발생하는 딜레이나 요청실패시의 문제는

고려하지 않았다.

실험결과는 디바이스 가상화 소요시간과 비슷하게 늦어도 0.3초의 시간이 소요되었다.

4.3 디바이스 성능

네트워크 디바이스가 가상데스크탑 소프트웨어와 연결이 되면 둘 사이에 I/O 채널이 구성되게 된다. 디바이스의 성능은 결국 이 I/O 채널에서의 성능과 직결된다. 마우스의 경우 15ms, 키보드의 경우 20ms의 시간이 소요되었다. 이는 Tight VNC의 14ms, 21ms와 비슷한 성능을 나타냄을 알 수 있다[10]. 일반적으로 키보드나 마우스 같은 입력 장치의 수용 가능한 응답시간은 50~150ms 정도로 알려져 있다[11].

원격지의 가상데스크탑을 접근한다는 것 자체가 네트워크 성능에 종속적인 접근법이다. 본 시스템에서는 VNC프로토콜을 사용하여 화면을 출력하기 때문에 근본적으로는 VNC프로토콜과 동등한 성능을 나타내게 된다. 하지만 VNC의 경우는 마우스와 키보드 이벤트 또한 같은 채널을 통해서 전송이 되지만 본 프레임워크에서는 각각의 채널이 따로 구성이 되므로 이벤트의 알림은 기존 방식에 비해 빠르다고 볼 수 있다.

반면 실버라이트로 구현되어 브라우저 내에서 동작하는 어플리케이션들의 경우 추가적인 오버헤드를 가지게 되는데 이는 브라우저 내에서 시스템의 서비스를 접근하지 못하도록 그 권한을 막은 데서 비롯된다. 해당 문제는 실버라이트의 자체의 성능개선, 다른 RIA 솔루션의 선택 등으로 우회할 수 있는 여지가 남아있다.

5. 결론 및 향후 연구

본 논문에서는 가상데스크탑을 제어하기 위한 네트워크 디바이스를 만들어내는 것이 목적이다. 이를 위해 RIA를 적용하여 쉽게 디바이스를 가상화하는 프레임워크를 소개하였다. 그리고 해당 프레임워크가 기존 연구들에 비해 높은 확장성을 가지고 있음을 보였다.

현재 프로토타입에서는 키보드, 마우스, 모니터 같은 단순한 I/O 디바이스만이 디바이스플에 등록이 되어 사용되고 있다. 하지만 쉽게 확장될 수 있는 형태의 프레임워크로 구성이 되어있으므로 다른 디바이스들로 쉽게 확장될 수 있을 것이다. RIA 솔루션의 태생적 한계 때문에 로컬 컴퓨터의 스토리지 디바이스를 네트워크화된 가상 스토리지 디바이스로 변환하는 것은 어려움이 따르지만 오디오 디바이스의 경우 현재 디바이스플에 등록하고 사용하고자 하는 시도가 진행 중에 있다. 이 같은 접근법이 RIA 솔루션에 종속적인 접근법일 수도 있으나 RIA 솔루션이 가지는 설치로부터의 해방, 쉬운 실행 및 플랫폼 독립적인 성격은 프레임워크의 확장성을 높여주는 주요한 요소일 것이다.

참고 문헌

- [1] 박세진, 강상우, 박찬익, "레거시 디바이스의 네트워크 디바이스화를 지원하는 프레임워크의 설계 및 구현", 한국정보과학회 춘계학술발표회 2009.
- [2] 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 *proceedings of the 19th ACM symposium on Operating Systems Principles*, pp.164-177, Bolton Landing, NY, 2003.
- [3] Myers, B.A., Miller, R.C., Bostwick, B., and Evankovich, C., "Extending the Windows Desktop Interface With Connected Handheld Computers," in *4th USENIX Windows Systems Symposium*, Seattle, WA: pp.79-88, 2000.
- [4] Shen, G., Li, Y., Zhang, Y., "MobiUS: enable together together-viewing video experience across two mobile devices," *ACM Mobile Systems and Applications Conference, MobiSys'07*, June 11-14th, 2007, Puerto Rico, USA, pp.30-42.
- [5] Johanson, B., Hutchins, G., and Winograd, T. (2000): PointRight: A System for Pointer/Keyboard Redirection Among Multiple Displays and Machines. HCI Group, University of Stanford, Report No. CS-2000-03, 2000.
- [6] T. Richardson, et al., "Virtual Network Computing," *IEEE, Internet Computing*, vol.2, 30.1, 1998.
- [7] Fox, A., Johanson, B., Hanrahan, P., Winograd, T., "Integrating Information Appliances into an Interactive Workspace," *IEEE Computer Graphics and Applications*, vol.20. no.3, pp.54-65, May 2000.
- [8] Brumitt, B., Meyers, B., Krumm, J., Kern, A., Shafer, S., "EasyLiving: Technologies for Intelligent Environments," *Handheld and Ubiquitous Computing (HUC) 2000*, Bristol, UK, pp.97-119, Sep. 2000.
- [9] F. Bellard, QEMU, a Fast and Portable Dynamic Translator, in *proceedings of the annual conference on USENIX Annual Technical Conference*, pp.41-41, Anaheim, CA, 2005.
- [10] Tight VNC, <http://www.tightvnc.com>.
- [11] Shneiderman, B., *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 3rd ed., Addison-Wesley, Reading, MA, 1998.