

## Xen 기반 가상 데스크탑 서비스를 위한 부하 분산 정책

손성훈\*, 전상일\*\*

### Load Balancing Policy for Xen-based Virtual Desktop Service

Sunghoon Son\*, Sangil Jun\*\*

#### 요약

가상의 컴퓨팅 환경을 제공하는 가상화 기술은 최근 많은 연구가 진행되고 있는 분야이며, 이에 따라 가상화 기술을 이용한 많은 제품들이 상용화, 출시되는 추세이다. 가상화의 여러 분야 중 특히 서버 가상화는 실제 필요한 서버의 수보다 적은 수의 물리적 서버들로 다수의 가상화된 서버를 수행함으로써 네트워크 서비스 구축에 드는 비용을 절감할 수 있는 효율적인 방안이다. 본 논문에서는 서버 가상화 기술을 기반으로 하여 원격 데스크탑 서비스를 위한 서버를 가상화한 가상 데스크탑 서비스를 설계, 구현하였다. 또한 제안된 가상 데스크탑 서비스 상에서 특정 물리적 서버에 부하가 집중되는 것을 방지하기 위한 부하 분산 정책을 제안하였다. 제안된 가상 데스크탑 서비스와 부하 분산 정책은 사용자 관점 및 시스템 관점에서의 성능 향상을 모두 기대할 수 있는 효율적인 원격 데스크탑 서비스가 될 것이다.

#### Abstract

Recently virtualization becomes one of the most popular research topics, and a lot of software products which is related to virtualization have been released accordingly. Server virtualization, which virtualize physical servers to supply many virtual servers, provides very efficient way to build network-based servers. In this paper, we design and implement virtual desktop service which is based on server virtualization. We also propose load balancing scheme for the virtual desktop service. The proposed virtual desktop service and its load balancing scheme provides a cost-effective way to build high-performance remote desktop service.

▶ Keyword : 가상화(Virtualization), 서버 가상화 (server virtualization), 가상 데스크탑 서비스 (Virtual Desktop Service), 부하분산(Load Balancing)

• 제1저자 : 손성훈

• 접수일 : 2007. 12. 27, 심사일 : 2008. 1. 7, 심사완료일 : 2008. 1. 25.

\* 상명대학교 소프트웨어학부 조교수 \*\* 상명대학교 대학원 컴퓨터학과 석사 과정

※ 이 논문은 2007년도 상명대학교 일반연구기관 선발과제의 연구비를 지원 받았음

## I. 서론

가상화(virtualization)란 한 컴퓨터의 물리적 자원을 여러 것으로 나누어 사용할 수 있도록 하거나, 다수의 컴퓨터들의 물리적 자원을 묶어 하나의 컴퓨터처럼 제공하거나, 에뮬레이션 등을 통하여 존재하지 않는 자원을 생성, 가상의 컴퓨팅 환경을 제공하는 것과 관련된 일련의 컴퓨터 구조, 운영체제, 컴파일러 기술을 일컫는다[1]. 최근 이러한 가상화에 대한 연구가 활발히 진행되고 있으며, VMware나 Mojopac, Thinstall 등과 같이 가상화 기술을 상용화한 제품들도 출시되고 있다.

가상화의 여러 응용 분야들 중 서버 가상화(server virtualization)는 적절한 가상화 기법을 사용하여 한 대의 물리적 서버에 여러 대의 가상 서버들을 생성함으로써 마치 여러 대의 서버가 존재하는 것과 동일한 효과를 내는 기술이다. 서버 가상화를 활용하면 제한된 컴퓨팅 자원으로도 대규모 서버의 구축이 가능하여 서버 구축에 필요한 비용을 크게 절감할 수 있다. 본 논문에서는 이러한 서버 가상화를 기반으로 다수의 가상 머신들을 생성, 이를 대규모 원격 데스크탑 서비스의 서버로 사용하는 가상 데스크탑 서비스(virtual desktop service)를 설계, 구현하였다. 제안된 가상 데스크탑 서비스는 대표적인 가상화 기법 중의 하나인 Xen을 사용하여 한 대의 물리적 서버 상에 여러 개의 가상 머신(virtual machine)을 생성하고, 원격지의 사용자가 단말을 통해 서비스에 접속할 때마다 하나의 가상 머신을 할당, 마치 해당 사용자의 개인용 컴퓨터처럼 사용할 수 있도록 해준다.

또한 본 논문에서는 Xen의 라이브 마이그레이션(live migration) 기능을 활용하여, 제안된 가상 데스크탑 서비스 운영 중 특정 물리 서버에 부하가 집중되는 경우, 일부 가상 머신을 다른 물리 서버로 이동시킴으로써, 가상 머신들을 여러 물리 서버에 분산시킬 수 있는 부하 분산 정책을 제안한다. 마이그레이션 기능의 사용에 대한 적절한 정책은 효율적인 물리적 자원의 분배라는 차원에서 상당히 중요한 요소이다. 적절한 마이그레이션을 통해 여러 물리 서버들에 부하를 고르게 분산시킴으로써 좀 더 많은 서비스 요청을 수용하고 사용자에게 양질의 서비스를 제공할 수 있게 된다. 본 논문에서는 제안된 가상 데스크탑 서비스 하에 사용자 관점에서의 성능과 시스템 관점에서의 성능을 증대시킬 수 있는 효과적인 부하 분산 정책을 제시한다.

본 논문의 구성은 다음과 같다. 우선 2 장에서는 본 논문에서 사용하는 Xen을 포함하여 가상화와 관련된 기술들을 살펴본다. 또한 제안하는 부하 분산 정책과 관련된 기존 연구들을 소개한다. 3 장에서는 제안된 가상 데스크탑 서비스의 전체적인 설계 및 구현에 대해 알아보고, 4 장에서는 제안된 가상 데스크탑 서비스 하에서의 부하 분산 정책에 대해 설명한다. 이를 바탕으로 5 장에서 성능평가를 통하여 가장 효율적인 부하 분산 정책을 제시할 것이며, 6 장에서 결론 및 향후 연구 방향을 제시한다.

## II. 관련 연구

본 장에서는 우선 Xen을 포함한 기존 가상화에 대한 기존 연구들을 소개한다. 또한 기존의 가상 데스크탑 서비스 소프트웨어들을 설명하고, 마지막으로 기존의 부하 분산 기법에 대한 기존 연구들을 살펴본다.

### 2.1 가상화

흔히 가상화 기술은 크게 5 가지 종류로 나누어 볼 수 있다.

Emulation은 binary translation을 이용하여 호스트 ISA (Instruction Set Architecture) 상에서 특정 guest 의 ISA를 시뮬레이션 하는 기법이다 [2]. 이를 통해 호스트 A에서 수행되던 운영체제 1에서 다른 호스트 B에서 수행되던 운영체제 2가 수행될 수 있기는 하나, 수행 속도가 매우 느려서 실용적이지 못한 단점을 가지고 있다.

Native virtualization (또는 Full virtualization)은 host ISA와 guest ISA가 같은 Emulation 방식이다 [2]. 호스트 A에서 수행되던 운영체제 1 상에서 같은 호스트 A에서 수행 가능한 운영체제 2가 수행되는 것이다. Emulation 보다는 속도가 빠르고 한 기계(host) 상에서 여러 다른 운영체제를 수행시키는데 많이 사용한다.

Para virtualization은 가상 머신의 하드웨어를 시뮬레이션 하는 대신 하이퍼바이저(hypervisor)라는 제어 프로그램을 통해 한 호스트의 하드웨어 자원을 여러 운영체제가 나누어 사용하도록 하는 기법이다 [4]. 이 때 가상머신 상에서 수행되는 운영체제들은 새로운 하드웨어 위에서 수행되는 셈이므로 커널의 수정 작업이 없으면 수행 할 수 없게 된다. 본 논문에서 주로 사용하고 있는 Xen 시스템은 이 Para Virtualization 기법을 사용하여 리눅스 운영체제를 수행시킨다.

OS-level virtualization은 기본적으로 하나의 CPU에 하나의 운영체제만이 수행되고 그 위에서 마치 여러 OS들이 수행되는 것과 같은 효과를 내지만, 실제 운영체제 설치에 필요하지 않은 수행 환경을 제공한다[5]. 실제 동작하는 운영체제는 하나 밖에 없으므로 호스트나 게스트의 구분이 없으며, 단지 여러 운영체제들이 동시에 수행 되는 것 같은 가상 수행 환경을 제공한다.

Application-level virtualization은 한 운영체제 상에서 특정 응용 프로그램 집합에게만 가상의 수행 환경을 제공하는 기법이다 [6]. 해당 응용 프로그램은 호스트 환경과는 분리된 자원을 사용하게 되어, 한 운영체제 상에서 가상화된 응용 프로그램과 그렇지 않은 응용 프로그램이 동시에 수행되더라도 파일 시스템이나 레지스트리 등의 자원은 모두 철저히 분리된 독립된 환경에서 수행된다.

이상의 가상화 기법들은 서버 가상화 등 다양한 분야에 응용될 수 있다. 가상화된 컴퓨팅 환경이 제공하는 장점은 첫째, 서버 가상화로 인해 물리적 서버의 수를 감소시킬 수 있다. 둘째, 동일한 운영체제도 적용 분야에 따라 솔루션 별로 다양한 버전을 제공할 수 있다. 셋째, 서로 분리된 환경에서 가상

머신들이 동작하므로 보안을 강화 할 수 있다. 넷째, 소프트웨어 개발자들을 위한 독립된 환경을 다양하게 제공할 수 있다. 다섯째, 가상 머신을 실시간으로 마이그레이션 할 수 있으므로 물리적 기계의 오류나 부하 급증에 대해 빠르고 안전하게 대처할 수 있다. 여섯째, 서버 프로그램을 여러 가상의 서버에 분리하여 운용함으로써 한 서버 프로그램의 오류가 다른 프로그램에 영향을 미치는 것을 방지할 수 있다.

## 2.2 Xen

Xen은 IA-32, IA-64, PowerPC 등의 CPU를 기반으로 하는 컴퓨터에서 사용할 수 있는 가상 기계 모니터(virtual machine monitor)이다 [8]. Xen은 하나의 물리적 하드웨어를 기반으로 동작하는 호스트 운영체제 상에서 동시에 여러 개의 게스트 운영체제를 수행시킬 수 있는 환경을 제공한다. 호스트 운영체제로는 수정된 리눅스나 NetBSD가 가능하며, 다양한 유닉스 계열의 운영체제들이 게스트 운영체제로 사용될 수 있다. 특히 Xen 버전 3.0 이후부터는 수정되지 않은 윈도우 운영체제 또한 게스트 운영체제로 사용될 수 있다.

Xen은 다음과 같은 특징을 가지고 있다. 첫째, 소프트웨어 계층이 하드웨어와 운영체제 사이에 설치된 하이퍼바이저라는 소프트웨어 계층이 가상화된 층을 제공하고, 이 가상화된 층은 각각의 물리적 서버가 하나 혹은 더 많은 가상 서버를 실행할 수 있게 한다. 둘째, 하나의 가상 서버 이미지가 만들어 지게 되면, 이 이미지는 서버의 종류나 시간에 관계없이 수행이 가능하다. 그리고 여러 개의 서버들은 하나의 서버를 동시에 공유할 수 있어 유연성과 낮은 비용을 제공한다. 셋째, Xen은 50,000 라인 정도의 작은 소스 코드를 가지고 있으며, 기존의 장치 드라이버들을 그대로 사용할 수 있는 장점을 가지고 있다. 넷째, 동작 중인 가상 머신을 다른 곳으로 옮길 수 있는 라이브 마이그레이션 (live migration) 기능으로 자원의 재할당이 쉬워 자원의 사용률을 증가시키고, 적절한 자원 분배를 가능하게 한다. Xen의 이러한 자원 관리 모델은 보안 문제 또한 개선해 주었다.

## 2.3 가상 데스크탑 서비스

최근 다음과 같이 데스크탑 서비스가 서버 형태로 제공되는 몇몇 서버 가상화 소프트웨어들이 소개된 바 있다.

VMWare 사는 2006년 7월에 기존 ESX Server의 새로운 버전인 Virtual Desktop Infrastructure 3를 출시하였다. 여기에는 가상화 환경 관리 도구인 VMware VirtualCenter와 가상머신의 동적 이동(Migration)을 가능하게 하는 VMware VMotion, 하드웨어 자원의 동적 이동을 지원하는 VMware DRS 등이 포함된다. 특히 VirtualCenter는 자동적으로 가상머신을 다른 하드웨어 상의 ESX 서버로 전환시키는 기능을 가지고 있다.

마이크로소프트의 Virtual Server 2005는 현재 Virtual Server 2005 R2로 업데이트되어 게스트 운영체제로 윈도우즈뿐만 아니라 리눅스 운영체제도 지원하고 있다.

선마이크로시스템의 Solaris 10에 포함된 가상화 기능은 한 대의 물리 서버에 소프트웨어와 파티션을 운용함으로써 다

수의 Solaris 10 환경을 제공하는 것으로 각 가상서버는 완전한 독립적 운영체제로 동작한다. 특히 Xen을 하이퍼바이저 형태로 Solaris에 포함시켜 성능을 향상시키고 있다.

## 2.4 기존 부하 분산 정책

서버 가상화를 통한 서버 구축 시의 부하 분산 정책에 대한 연구는 아직 초기 단계에 머무르고 있다. 다만 각종 인터넷 서비스에 사용되는 다중 접속 서버와 관련된 부하 분산 정책에 대한 연구는 다음과 같다. 우선 Static Load Balancing 기법은 클라이언트의 요청에 대해 정적으로 서버에 연결하여 서비스를 할 수 있도록 한다. 한 클라이언트는 언제나 같은 서버에서 서비스를 받게 되고 다른 서버에는 연결되지 않는다. Round Robin Load Balancing 기법은 클라이언트의 서버 연결 메시지를 중간 라우터가 라운드 로빈 방식으로 서버들에 메시지를 전달하여 연결을 한다. Diffusive Load Balancing 기법은 클라이언트의 서비스 요청을 전달하는 라우터가 좀 더 부하가 적은 서버로 요청을 전달한다. Shortest Queue Load Balancing 기법은 클라이언트의 요구를 부하가 가장 적은 host로 이동시키되, 클라이언트가 서비스를 받는 동안에도 이동이 이루어 질 수 있다는 것이다 [11]. Local Approach 방식은 부하가 많아진 서버의 부하를 이웃의 서버로 옮겨서 지역적으로 부하를 분산시킨다[13][14]. 이 방식에서 서비스를 이동시킬 서버는 이웃 서버의 정보만을 필요로 하기 때문에 부하 이동에 대한 오버헤드가 적다는 장점이 있다. Global Approach 방식 모든 서버의 부하 정보는 한 시스템에 유지하고, 이를 기반으로 전체 서버에 골고루 분산하는 방법이다[15].

이상의 연구들은 모두 각종 인터넷 서버들을 위한 부하 분산 정책들이다. 인터넷 서비스 요청은 (웹서버 같은 경우) 대부분의 경우 클라이언트의 요청에 따라 특정 페이지를 보여주거나, (검색과 같은 경우) 간단한 응용 프로그램을 실행시키고 결과를 보여 주는 정도이다. 반면에 가상 데스크탑 서비스의 경우 하나의 서비스 요청을 처리하는데 드는 서버의 자원은 인터넷 서비스에서의 경우와는 비교할 수 없을 정도로 크다. 예를 들어 한 서비스 요청에 대해서 운영체제가 수행될 수 있을 만큼의 메모리와 운영체제, 그 위에서 응용 프로그램이 수행되는 동안 사용할 CPU 및 기타 자원들이 소모된다. 따라서 같은 사양의 서버 상에서 가상 데스크탑 서비스를 사용하는 경우는 일반적인 인터넷 환경의 서비스를 이용하는 경우에 비해 조금만 서비스 요청이 많이 들어와도 금방 시스템의 자원이 고갈되는 상태에 이르게 된다. 서비스 요청이 들어와서 가상머신을 서비스하는 가상머신의 lifetime도 짧게는 수 분에서 많게는 수 일 동안 한 서비스 요청에 대해 서비스를 해야 하기 때문에 인터넷 서비스에 비해 상당히 길다. 따라서 사용자에게 서비스를 제공하고 있는 도중에도 경우에 따라서 서비스 중단 없이 다른 서버로 즉시 이동이 가능해야 한다. 클라이언트와 연결 성능을 중요시하고 라우터에서의 큐잉 지연 등이 존재하는 인터넷 서비스와 다른 가상 데스크탑 서비스의 이러한 기본적인 특징들로 인해 위에서 언급한 부하 분산 정책들은 가상 데스크탑 서비스에 그대로 적용하기가 힘들다. 본 논문에서는 이런 이유로 인해서 가상 데스크탑 서비스를 위한 새로운 부하 분산 정책을 제시하는 것이다.

### III. 가상 데스크탑 서비스와 Migration정책

본 장에서는 제안된 가상 데스크탑 서비스의 설계 및 구현에 대해 다룬다. 특히 가상 데스크탑 서비스의 전체 구조, 데스크탑 서비스의 종류, 서비스 구성 요소들 간의 구체적인 동작 등을 설명한다.

#### 3.1 가상 데스크탑 서비스

본 논문에서 제안하는 가상 데스크탑 서비스의 전체 구성은 그림 1과 같다. 시스템은 데스크탑 서비스 사용자가 사용하는 개인용 컴퓨터나 간단한 터미널, 전체적인 서비스를 관리하는 관리 서버, 실제 가상 머신들이 수행되는 호스트들 및 가상 머신들의 디스크 역할을 하는 파일 서버로 이루어진다.

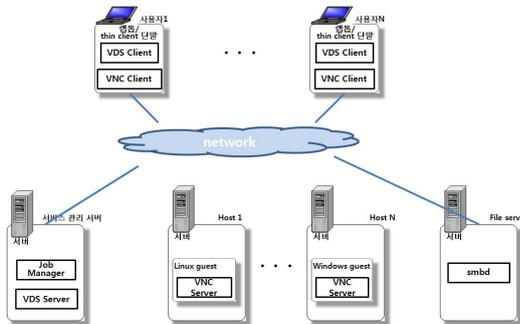


그림 1 가상 데스크탑 서비스  
Figure 1 Virtual Desktop Service

VDS Server는 관리 서버에 동작하는 서버 소프트웨어이다. 각 host서버들과 guest 운영체제의 자원을 감시하여 guest 운영체제들의 동작을 제어하고 Migration 등을 수행한다. VDS Server는 각 host서버의 Xen 시스템에 명령을 내기 위한 JobManager를 이용할 수 있는 JobManagerAPI를 통해 각 host서버에 명령을 전달하게 되고 각 host서버와 guest 운영체제들은 그 명령에 따라 가상 머신과 관련된 각종 동작을 수행하게 된다. VDS Client는 개인 사용자의 단말기에 설치되어 있으며 로그인 기능과 guest 운영체제에 대한 원격접속 기능을 담당한다. 각 guest 운영체제들에는 VNC Server가 동작하고 있고 VDS Client들에는 VNC Client가 동작하고 있어서 VNC를 이용하여 원격접속을 하여 사용자에게 서비스를 제공한다. Migration을 위해 각 host서버들은 NFS를 이용하여 디스크를 공유하고 있으며 모든 host서버들은 "/vm/host번호" 형식으로 같은 모습의 디렉토리 구조를 사용하게 된다.

#### 3.2 데스크탑 서비스의 종류

제안하는 가상 데스크탑 서비스에서는 공용 데스크탑 서비스와 개인용 데스크탑 서비스의 두 가지 유형의 데스크탑 서비스를 제공한다.

우선 공용 데스크탑 서비스는 공공장소 등에서 누구나 이

용할 수 있는 공용 컴퓨터와 같은 개념이며 이 서비스에 사용되는 가상머신을 CVM(Common Virtual Machine)이라고 한다. CVM은 모든 서비스 사용자들에게 임의로 주어지는 데스크탑(가상머신)으로 비교적 짧은 시간동안 웹서핑, 이메일 체크 등과 같이 간단한 사용 후 종료하는 경우에 적합하다.

공용 데스크탑은 사용자의 로그인을 하면 부팅을 하여 데스크탑을 서비스하는 것이 아니라 suspend 혹은 save 되었던 가상머신을 run 시켜서 데스크탑으로 서비스 한다. 또한, 서비스 종료 시에는 가상머신을 종료시키는 것이 아니라 단지 연결만 종료하게 되는 것이다. 연결이 종료된 가상머신은 시간의 흐름에 따라 suspend, save 상태로 상태가 전이 된다.

모든 서비스 사용자들은 마치 자신의 개인용 컴퓨터처럼 개인 데스크탑을 할당받는다. 그렇게 할당받는 가상머신을 PVM(Private Virtual Machine)이라고 한다. 서비스 연결 시에 할당받는 데스크탑은 그동안 사용하여 오던 자신만의 데스크탑을 할당 받는다. 그렇기 때문에 매 서비스 사용 때마다 같은 데이터를 사용하고 이전에 설치되었던 프로그램들을 그대로 사용하게 된다. 각 개인 데스크탑들은 서로 독립적으로 사용되고 있어서 서로의 데이터에 접근할 수 없을 뿐만 아니라 각 개인 데스크탑별로 samba서버로 제공되는 데이터 저장을 위한 공간을 제공한다. 개인용 데스크탑 서비스에 사용자가 로그인을 하면 가상머신은 부팅과정을 거치게 된다. 따라서 간단한 웹서핑, 이메일 체크 등의 짧은 작업에는 부팅과정을 거쳐야 하는 개인용 데스크탑 서비스가 사용자에게는 불편할 수 있다. 하지만 완전한 개인적인 공간이나 프로그램의 사용을 위해서는 개인용 데스크탑 서비스에 로그인하는 것이 좋다. 서비스의 종료 시에는 데스크탑의 종료 과정을 거치게 되어 가상머신은 완전히 종료하게 된다.

현재는 CVM의 상태는 Inactive, Ready, Running, Saved, Paused 중 하나이며 모든 CVM은 서비스 개시 직후 Ready 상태에서 시작한다. Ready상태의 CVM은 VNC 세션 연결 없는 상태로 일정 조건을 만족하면 Paused 상태로 바뀔 수 있으며, Paused 상태의 CVM은 다시 일정 조건을 만족하면 Saved 상태로 바뀔 수 있다. 새로운 VNC 세션 접속 요청 시 우선 Ready 상태의 CVM 중 하나를 할당하고 없으면 Paused 상태의 CVM 중에서 그마저도 없으면 Saved CVM 중에서 할당한다. Paused CVM이나 Saved CVM 중에서 할당하는 경우, 일단 해당 CVM의 상태가 Ready로 바뀐 후 VNC 세션이 형성되면 해당 CVM의 상태는 Running 으로 바뀐다. VNC 세션이 종료되면 해당 CVM의 상태는 Running 에서 Ready 로 바뀐다.

다음의 그림은 이런 CVM의 각 상태와 상태의 전이를 도식화한 것이며 상태 변화에 대한 명령들은 해당 명령에 해당하는 Xen명령 <sup>1)</sup>으로 표현하였다.

1) Xen의 제어를 위한 명령은 호스트머신의 shell 터미널에서 내리기 위해 제공되는 command(xm) 이며 본 논문에서는 create(가상머신의 시작), shutdown(가상머신의 종료), pause(가상머신 일시정지), unpause(가상머신 일시정지해제), save(가상머신을 파일형태로 저장), restore(저장된 가상머신을 불러오기)와 같은 sub command를 사용한다.

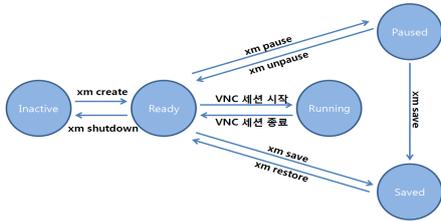


그림 2 CVM의 상태 전이  
Figure 2 State transition of CVM

모든 PVM들은 Inactive, Ready, Running 상태 중 한 가지 상태로 존재한다. PVM은 xm create로 생성되고 사용 후 pause나 save 없이 항상 shutdown(or destroy)된다.

다음의 그림은 이런 PVM의 각 상태와 상태의 전이를 도식화한 것이며 상태 변화에 대한 명령들은 해당 명령에 해당하는 Xen명령으로 표현하였다.



그림 3 PVM의 상태 전이  
Figure 3 State transition of PVM

### 3.3 가상 데스크탑 서비스의 동작

#### 3.3.1 데스크탑 서비스 연결

서비스 이용을 위해 사용자가 VDS Client를 통해서 VDS Server에 서비스 요청을 한다. 이 때 사용자는 로그인 계정에 대한 정보와 공용 데스크탑 서비스를 이용할 것이라는 정보를 입력한다. 서비스 요청을 받은 VDS Server는 Ready 큐 중에서 한 가상머신을 선택하여 VDS Client에게 할당하여 주고 VDS Client에게로 할당된 guest의 ID와 IP를 통보하여 준다. 가상머신을 할당받은 VDS Client는 VNC Client를 통해서 할당 받은 가상머신의 guest운영체제에 연결을 하게 된다. Ready큐에 가상머신이 없을 경우 Paused 큐에서 가상머신을 찾아서 Unpaused로 변경한 다음 가상머신을 할당한다. Paused큐에서조차 할당할 가상머신이 없을 경우 Saved큐에서 가상머신을 찾아서 할당하고 그마저도 없으면 할당할 수 없음을 반환한다.

서비스 이용을 위해 사용자가 VDS Client를 통해서 VDS Server에 서비스 요청을 한다. 이 때 사용자는 로그인 계정에 대한 정보와 개인 데스크탑 서비스를 이용할 것이라는 정보를 입력한다. 서비스를 요청받은 VDS Server는 DB에서 사용자의 가상머신을 찾아서 적절한 host서버에 생성을 하게 된다. 가상머신의 생성 위치(host서버)는 지속적으로 모든 host의 자원사용량을 감시하고 있던 VDS Server가 자원사용량이 적은 적절한 host를 찾아 그 host에서 생성 한다. 생성 명령을 받은 host는 가상머신을 생성하고 가상머신은 부팅과정을 거친다. 부팅과정 중에 가상머신의 guest운영체제에 파일서버로부터 해당 guest운영체제가 사용하던 디스크 공간을 할당받아 마운트 시켜 주고 부팅이 완료 되면 guest운영체제 내의 작은 응용프로그램에 의해서 VDS Server로 부팅완료 메시지를 전

송하게 된다. 부팅이 완료 되었다는 메시지를 받은 VDS Server는 생성된 guest운영체제의 ID와 IP를 VDS Client에게 통보하고 VDS Client는 VNC Client를 통해서 guest 운영체제에 연결하여 서비스를 제공한다.

#### 3.3.2 가상머신의 Migration

VDS Server는 항상 각 host들의 자원상황을 체크하고 있어서 특정 host의 작업사용량이 급증하였을 경우 Migration을 결정하게 된다. Migration에 대한 자세한 정책은 'IV Migration 정책'에서 다루기로 한다. 특정 서버에 자원 사용량이 급증하였다는 것을 감지하게 되면 Migration 정책에 의해 Migration의 대상이 되는 가상머신을 선택하게 하고 Migration 정책에 의해 가상머신이 옮겨져서 계속 수행 될 host가 선택된다. 선택되어진 가상머신을 Migration하기 위해 VDS Server는 해당 가상머신이 있는 host에 Migration 명령을 내리고 가상머신은 Migration 된다.

그림 4는 host 2에서 자원사용량이 급증하여 guest 10을 host 3으로 Migration하는 과정을 보여주고 있다. VDS Server가 각 host들의 자원사용량을 확인하던 중 host 2에서 일정시간 동안 자원사용량이 급증하였음을 감지한다. VDS Server는 Migration이 필요하다고 판단하고 Migration 정책을 의해서 host 2의 guest 10을 host 3으로 Migration할 것을 결정한다. VDS Server는 host 2에 Migration을 수행 하라고 명령하고 guest 10은 host3으로 Migration 된다. host 2는 Migration 완료 후 VDS Server로 Migration이 완료되었음을 통보한다. 사용자는 이러한 Migration 동작이 일어났는지 인식하지 못하고 계속해서 서비스를 이용한다.

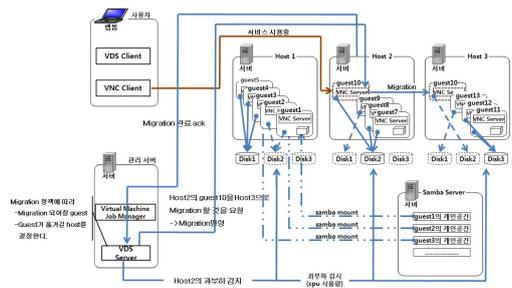


그림 4 Migration 시나리오  
Figure 4 Migration scenario

#### 3.3.3 데스크탑 서비스 종료

공용 데스크탑 서비스의 종료 과정은 host서버들이나 guest운영체제들과는 무관하게 이루어진다. 서비스를 종료하겠다는 사용자의 동작에 따라 VNC Client는 종료 시그널을 VDS Client에게 전달하고 VDS Client는 VDS Server에게 서비스 종료를 요청한다. 서비스 종료 요청을 받은 VDS Server는 사용자가 사용하던 CVM을 Ready큐에 넣고 서비스 종료를 통보한다.

사용자가 VNC Client를 종료하여 개인용 데스크탑 서비스의 종료를 요청하면 VNC Client는 VDS Server에 서비

스 종료료를 요청한다. 서비스 종료료를 요청받은 VDS Server는 해당 PVM이 수행되고 있는 host에 shutdown명령을 내리고 PVM은 종료하면서 파일서버와의 자료 저장공간 공유를 해지한다. PVM이 종료되면 VDS Client에 서비스 종료료를 통보하여 서비스 종료 과정을 끝낸다.

### IV. Migration 정책

본 절에서는 Xen 상에서의 가상 데스크탑 서비스를 위한 몇 가지 Migration 정책을 제시하고, 그 정책들 중 최상의 서비스를 제공할 수 있는 정책을 제안한다.

#### 4.1 VDS에서의 migration

Migration은 하나의 물리 머신에서 다른 머신으로 가상머신 state를 인터럽트 없이 움직이는 것을 말하는데 IP 주소가 가상머신을 따라 이동하기 때문에 네트워크도 계속 가능하다. 특히 down-time을 최소화시키기 위해서 진행할 수 있는 live migration은 가상머신의 페이지를 스트리밍으로 네트워크를 통해서 가상머신이 동작할 때 보내기 때문에 서비스의 적용에 크게 의미 있다. live migration을 위해 우선 page들을 모두 대상 노드로 전송한 후, 대상 노드에 전송하는 동안 dirty bit에 mark된 page들을 다시 전송하게 된다. 이를 몇 차례 진행함을 통해서 원 노드의 guest os를 suspend시키고 대상 노드로 옮겨간 os를 resume 시키게 된다. 이때 dirtying rate가 가상머신의 down-time을 결정하게 된다.

물리적 서버 상에서 여러 가상머신이 서버로 수행되고 있는 환경에서 특정 물리적 서버의 자원 사용량이 급증할 수 있다. 그것은 물리적 서버에 대한 과도한 서비스 요청이나 가상머신의 많은 자원 사용이 문제가 될 수 있다. 하지만 본 논문에서 제시하는 서비스에서 물리적인 서버는 실제로 일반 사용자에게 어떠한 서비스 2)를 제공하지 않는다. 그렇기 때문에 물리적 서버의 과도한 부하는 가상머신으로부터 유래한다고 제한하기로 한다.

특정 물리적 서버에서 지속적으로 과도한 부하가 감지되었을 때 속히 그 서버에 대한 부하를 낮추어 주지 않는다면 경우에 따라 물리적 서버가 서비스 불능 상태에 빠질 수 있다. 그것은 곧 그 물리적 서버에서 수행되고 있던 다수의 가상머신들이 서비스를 할 수 없게 되는 것과 같다. 그렇기 때문에 이러한 물리적 서버의 과도한 부하를 다른 유휴 자원이 있는 서버로 나누어 서비스 할 수 있어야 하며 이러한 목적을 위해서 Migration기능을 이용하게 된다.

Migration을 통해 서비스 과다로 인한 물리적 서버의 서비스 불능 상태를 예방하고 전체 물리적 서버들 간에 자원 사용량을 공평하게 유지할 수 있게 된다. 그렇게 하면 특정 서버로 서비스 요청이 몰려 서비스를 할 수 없게 되는 일을 예방할 수 있게 되어 서비스 요청을 서버가 거절하거나 서비스를 받기 위해 사용자가 장시간 기다리는 시간을 줄일 수 있게 되어 시스템의 서비스 질이 향상될 것이다.

2) 여기서 말하는 어떠한 서비스란 웹서버, 메일서버, 원격 접속 서비스 등과 같이 일반적인 서버들이 제공하는 서비스를 말한다.

#### 4.2 Migration 동작

Migration을 위해서는 우선 각 호스트머신들의 정보와 가상머신들의 정보가 필요하다. 이들이 각각 Migration을 위해서 필요로 하는 정보들을 정리하면 다음과 같다.

■ 호스트머신: CPU 사용량, IP 주소, 고유 ID, 메모리 크기, 메모리 사용량

■ 가상머신: IP 주소, 고유 ID, 메모리 크기

이러한 정보를 바탕으로 Migration의 대상이 되는 호스트머신과 가상머신을 결정하고 옮겨지게 될 목적지 호스트머신을 결정한다.

#### 4.3 Migration 정책

Migration정책에 있어서 고려되는 자원은 CPU 사용량이며 가상머신의 자원사용량은 고려하지 않는다. 가상머신의 CPU 사용량의 증가는 곧 호스트머신의 CPU 사용량 증가와 같고 사용 메모리는 이미 정적으로 정해지기 때문에 가상머신의 생성 및 Migration 가능성 테스트에만 이용된다.

Migration정책은 다음과 같다.

1. 모든 호스트머신의 CPU 사용량을 항상 감시하고 있다.
2. 특정 호스트머신의 CPU 사용량이 증가함을 감지한다.
  - CPU 사용량은 일정시간 동안 평균을 구한 값이 일정 크기 이상일 경우 Migration이 필요하다고 판단
3. Migration 대상 guest 선정 정책으로 가상머신을 선택한다.
4. Migration 목적지 선정 정책으로 호스트머신을 선택한다.
5. 선택된 호스트머신으로 Migration이 가능한지 판단한다.
  - 선택된 호스트머신의 메모리 사용량을 고려한다.
6. Migration이 가능하다면 Migration 수행 후의 CPU 사용량이 지금의 호스트머신의 CPU 사용량보다 많아지는 것인지 확인한다.
7. Migration대상 호스트머신의 CPU사용량보다 많아진다면 Migration을 포기 한다.
8. Migration대상 호스트머신의 CPU 사용량보다 많아지지 않는다면 Migration을 수행한다.
9. Migration 불가능하다면 Migration 목적지 선정 정책으로 다음 목적지 호스트머신을 선택한다.
10. Migration을 수행한 후에는 관련 정보를 갱신한다.

아래의 그림은 이러한 Migration정책을 도식화한 것이다.

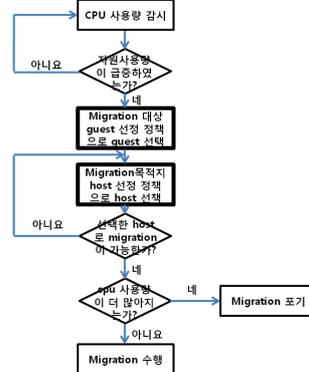


그림 5 Migration 정책  
Figure 5 Migration policy

#### 4.4 Migration 정책

그림의 굵은 사각형으로 표현된 Migration 대상 guest 선정 정책과 Migration 목적지 선정 정책은 서로 관계를 가지고 하나의 정책을 구성한다. 다음의 Migration 대상 guest 선정 정책과 Migration 목적지 선정 정책의 조합에 의해 Migration 대상 guest 선정 정책과 Migration 목적지 선정 정책의 번호를 따서 1A, 2A, 2B, 3A, 3B, 4A, 4C로 명명한다.

##### ■ Migration 대상 guest 선정 정책

Migration의 대상이 될 가상머신들은 모두 지속적인 CPU 사용량 감시를 통해서 일정 시간 동안 평균적으로 가장 많은 CPU 사용량을 보이는 호스트머신의 가상머신들이다. 그 중 Migration을 하기 위해 선택되어지는 가상머신은 각 정책 별로 다음과 같이 선택한다.

1. 가장 많은 CPU 사용량을 보이고 있는 가상머신을 선택하여 Migration을 한다.
2. CPU 사용량이 호스트머신내의 가상머신들 중에서 하위 50%에 속하는 가상머신들을 선택하여 Migration한다.
3. 호스트머신내의 가상머신들 중에서 CPU 사용량이 가장 많은 가상머신을 제외한 CPU 사용량 상위 40%의 가상머신들을 선택하여 Migration한다.
4. CPU 사용량이 호스트머신내의 가상머신들 중에서 상위 10%에 속하는 가상머신들을 선택하여 Migration한다. 이때 CPU 사용량이 가장 많은 가상머신도 Migration 대상에 포함한다.

이렇게 선택되어진 가상머신들은 Migration 목적지 호스트머신 선정 정책에 의해 선택된 1개 혹은 다수의 호스트머신으로 Migration 된다.

##### ■ Migration 목적지 선정 정책

Migration되기 위해 선정된 가상머신들이 옮겨지게 되는 호스트머신을 선택하기 위한 정책으로 지속적인 CPU 사용량 감시를 통해서 얻어진 평균적인 CPU사용량을 기준으로 호스트머신을 선택한다. 그 선택 정책은 각 정책별로 다음과 같다.

- A. 가장 CPU를 적게 사용하고 있는 호스트머신으로 이동시킨다.
- B. 가장 CPU를 적게 사용하고 있는 호스트머신 가운데 하위 50%에 속하는 호스트머신들로 분산하여 이동시킨다. 이 때 CPU 사용량이 많은 가상머신일수록 CPU 사용량이 적은 호스트머신으로 이동시킨다.
- C. 가장 CPU 사용량이 적은 호스트머신 가운데 하위 10%에 속하는 호스트머신들로 분산하여 이동시킨다. 이 때 CPU 사용량이 많은 가상머신일수록 CPU 사용량이 적은 호스트머신으로 이동시킨다.

조합에 의해 1A 정책은 가장 CPU 사용률이 높은 가상머신을 CPU 사용률이 가장 낮은 호스트머신으로 옮긴다. 2A 정책은 CPU 사용률이 낮은 하위 50%내의 가상머신을 CPU 사용률이 가장 낮은 호스트머신으로 옮기고 2B 정책은 CPU 사용률이 낮은 하위 50%내의 가상머신을 CPU 사용률이 낮은 하위 50%내의 호스트머신으로 옮긴다. 3A 정책은 CPU 사용률이 높은 상위 40%내의 가상머신 중 가장 높은 CPU 사용률을 가지는 가상머신을 제외한 가상머신들을 CPU 사용률이 가장 낮은 호스트머신으로 옮기고 3B 정책은 CPU 사용률이 높은 상위 40%내의 가상머신 중 가장 높은 CPU 사용률을 가지는 가상머신을 제외한 가상머신들을 CPU 사용률이 낮은

하위 50%내의 호스트머신으로 옮긴다. 4A 정책은 CPU 사용률이 높은 상위 10%내의 가상머신을 CPU 사용률이 가장 낮은 호스트머신으로 옮기고 4C 정책은 CPU 사용률이 낮은 하위 10%의 호스트머신으로 분산하여 이동시킨다.

## V. 성능평가

### 5.1 실험

본 논문에서는 앞서 언급한 Migration 정책의 성능을 테스트하기 위하여 시뮬레이션을 수행하였다. 시뮬레이터를 수행시키면 쓰레드(thread)로 생성된 수십 대의 서버에 수백 대의 가상머신이 임의의 시간 간격으로 실행되고 임의의 시간 후에 종료된다. 이때의 서버와 가상머신은 시뮬레이션상의 상상의 서버와 가상머신들이다. 각 가상머신들은 임의의 시간 간격으로 임의 크기의 부하를 발생시키며 시뮬레이터는 지속적으로 부하 상황을 감시하여 VDS Server의 역할을 대신한다. 일정 시간 간격으로 부하 감시를 수행하고 부하 감시를 수행할 때마다 필요한 데이터들을 출력하게 된다. 모든 실험은 동일한 횟수의 부하 감시 수행 횟수 동안 수행하여 얻어진 데이터를 이용한다.

시뮬레이션을 위한 파라미터는 다음과 같다. 우선 고정 파라미터는 모든 테스트에 동일한 환경 구성을 위해 변화시키지 않은 값들이며, 가변 파라미터는 테스트 과정에서 랜덤하게 혹은 정해진 값들로 변화시키면서 테스트하게 되는 값들이다. 5.2 실험 결과 분석에서 사용된 데이터들은 가변 파라미터들을 변화시켜 가면서 얻은 데이터이다.

#### ■ 고정 파라미터들

- HOSTMEM 은 호스트 머신의 메모리 크기를 표현하며 4096을 설정하였다.
- GUESTMEM 은 가상머신의 메모리 크기를 표현하며 368을 설정하였다.
- HOST\_MIN\_MEM 호스트 머신의 최소 메모리 크기로 256을 설정하였다.
- HOSTS 은 호스트 머신의 수를 나타내고 30개의 호스트를 설정하였다.
- GUESTS 은 시스템 전체의 가상머신 수를 나타내고 400을 설정하였다.
- MIGRATE\_TERM 은 5초로 Migration을 위해 모든 호스트 머신의 자원사용량을 감시하기 위한 시간 간격이다.
- WINDOW\_SIZE 은 7로 Migration을 결정하기 위해 호스트들의 자원사용량 평균값 계산을 위한 윈도우 크기이다.

#### ■ 가변 파라미터들

- MAXLOAD 은 가상머신의 부하를 발생시킬 때의 랜덤 최대값이다. 50으로 그 값을 설정하기로 한다.
- LOADTERM 은 3으로 가상머신이 load를 발생시키는 간격의 랜덤 최대값이다.
- MINLOADTERM 은 3으로 최소한 이 값만큼의 간격을 가지고 부하를 발생시킨다.
- GUEST\_CREATE\_TERM 은 1로 가상머신의 생성 간격의 랜덤 최대값이며 생성 간격은 매 생성시마다 변한다.
- MIN\_GUEST\_CREATE\_TERM 은 1로 최소한 이 값만큼

의 간격은 가지고 가상머신을 생성하기 위해 설정한다.

- CRITICAL\_LOAD 은 이 값 이상의 cpu 사용량을 보이는 가상머신이 Migration 대상 호스트로 선정되며 여러 값을 가지고 테스트 한다.
- MAXLOADCOUNT 은 가상머신이 생성되어 종료 될 때까지 몇 번이나 load를 생성하는지를 결정하는 랜덤 최대값이며 가상머신마다 다르게 설정된다.
- UPPER\_RATE\_GUEST 은 정책 3A, 3B, 4A, 4C에서 Migration 대상 가상머신을 선택하기 위한 비율을 설정한다.
- UNDER\_RATE\_HOST 은 정책 2B, 3B, 4C에서 가상머신들의 Migration 될 목적지 호스트를 선택하기 위한 비율을 설정한다.

### 5.2 실험 결과 분석

Migration 정책의 효율성을 언급하기 위해 가장 중요한 요소는 상기한 바와 같이 사용자 요청에 대한 가상머신의 생성 거부 횟수와 Migration 수행 회수이다. 가상머신의 생성 거부 횟수가 증가한다는 것은 일반 사용자가 서비스를 사용하기 위해 서비스 요청을 했을 때 서비스를 이용할 수 없거나 장시간 대기한 후에야 서비스를 이용할 수 있음을 의미하므로 가상 데스크탑 서비스의 서비스 질이 떨어진다고 볼 수 있다. 그렇기 때문에 가상머신의 생성 거부 횟수가 작을수록 효과적인 Migration 정책이라고 볼 수 있다. 또한, Migration의 수가 많은 경우에는 서비스를 제공하는 호스트 머신들의 자원을 많이 소모하는 것과 같은 것이고, Migration하는 동안 크게 느끼지는 못하지만 가상 데스크탑이 순간적으로 느려지는 현상이 나타난다. 그렇기 때문에 사용자 관점의 서비스의 성능을 위해서는 같은 가상머신의 생성 거부 횟수를 보일지라도 Migration 수가 적을수록 효과적인 정책이 될 수 있는 것이다.

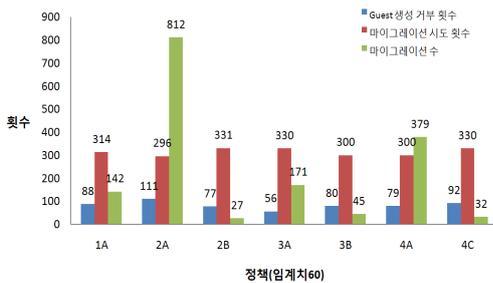


그림 6 정책별 가상머신 생성 거부 및 Migration 횟수 (부하 임계치 60)

Figure 6 Number of denials to VM creation/migration by policies (threshold value 60)

다음 그림 7과 그림 8은 부하 임계치에 따른 가상머신의 생성거부 횟수와 그때의 Migration 수를 비교하였다. 부하 임계치가 60이하이거나 90이상 이였을 경우에는 좋은 성능을 보이지 못하였기 때문에 70과 80의 임계치를 가지고 설명할 것이다. 예를 들어 60을 임계치로 잡았을 경우 위의 그림 6 '정책별 가상머신 생성 거부 및 Migration 횟수(부하 임계치 60)'과 같이 과도한 Migration을 수행 및 시도하고 많은 가상머신 생성 거부 횟수를 보이고 있다. 특히 Migration 동

작을 많이 시도 하고 Migration이 많이 실패하는 이유는 너무 임계치가 작기 때문에 Migration 대상이 쉽게 되고 목적지 호스트의 CPU 사용량이 Migration 했을 경우 가상머신이 있던 호스트에 비해 오히려 많다고 계산되는 부작용이 계속해서 나타나기 때문이다. 임계치가 90이상일 경우에는 너무 높은 임계치로 인해 Migration이 거의 일어나지 않아 Migration의 효과를 볼 수가 없어진다. Migration이 없는 경우의 성능은 다시 언급할 것이다.

그림 7과 그림 8에서 Migration 정책 중 1A, 2A, 3A, 4A의 경우에 Migration이 많이 일어나는 것을 볼 수 있다. 특히 cpu 사용량과 메모리의 부족의 이유로 가상머신 생성을 거부당한 횟수의 합(그림에서의 계)은 대체적으로 부하 임계치가 70일 경우가 80일 경우보다 2배가량 증가하는 것을 볼 수 있으며, 이는 그림 9에서 보면 좀 더 명확히 확인할 수 있다. 그림 9의 거부 횟수는 cpu 사용량과 메모리의 부족으로 인한 생성 거부 횟수를 더한 것이다. 이 그림들에서 Migration 정책에 효과적인 부하 임계치는 80임을 확인할 수 있다.

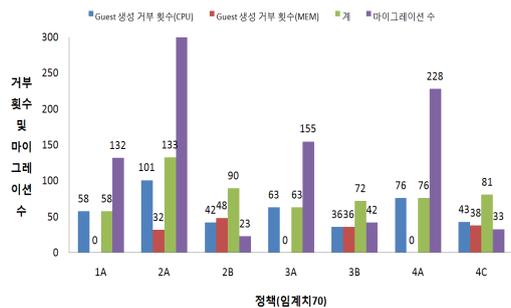


그림 7 정책별 가상머신 생성 거부 및 Migration 횟수 (부하 임계치 70)

Figure 7 Number of denials to VM creation/migration by policies (threshold value 70)

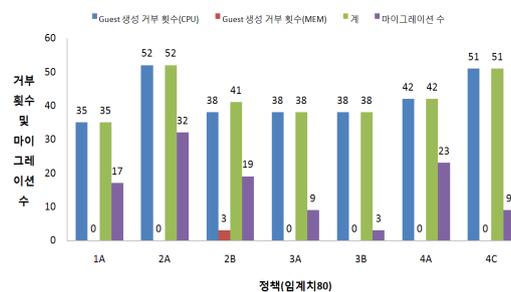


그림 8 정책별 가상머신 생성 거부 및 Migration 횟수 (부하 임계치 80)

Figure 8 Number of denials to VM creation/migration by policies (threshold value 80)

그림 8에서 가상머신의 생성 거부 횟수가 적은 정책은 1A, 3A, 3B로 나타나고 있다. 그러면서 동시에 Migration 수가 적은 정책은 3회만 Migration이 일어난 3B 정책이다.

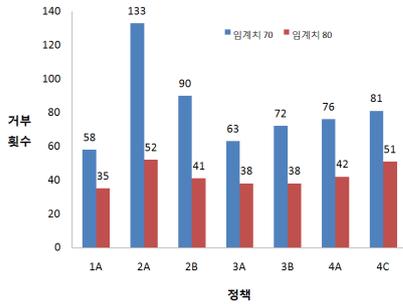


그림 9 정책에 따른 임계치별 가상머신 생성 거부 횟수  
Figure 9 Number of denials to VM creations by threshold values

4A정책과 4C정책의 경우 Migration을 하기 위한 가상머신을 선택하고 옮겨질 목적지 호스트를 선택하는 비율을 다르게 해 볼 수 있다. 앞의 경우에는 이 비율을 10%로 하였으며 아래의 그림10에서 비율을 30%로 높인 경우와 비교하여 볼 수 있다. 4A, 4C정책의 30%로 조정된 비율에서는 그림에서 어떠한 경우라도 3A, 3B보다는 가상머신 생성 거부 횟수가 많고 Migration수도 비율이 10%일 경우보다 30%일 때가 증가하는 등 성능 향상을 보기 어렵다.

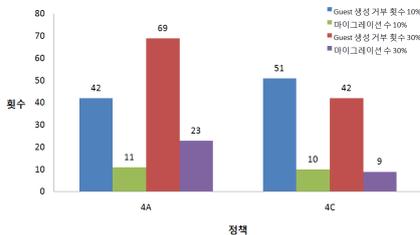


그림 10 Migration 대상 게스트와 목적지 호스트의 비율에 따른 성능 비교  
Figure 10 Comparison between 4A and 4C by target guest and destination host

테스트 결과에서 사용자가 서비스를 이용하기 위해서 가상머신을 생성할 것을 요청하였을 경우 이를 거부하는 횟수가 적으면서 Migration에 의한 시스템의 부하를 줄여 서비스의 신뢰를 높이고 시스템에 안정성을 주는 Migration 정책은 부하 임계치가 80인 상황에서의 3B 정책이라는 것을 확인 할 수 있었다. 1A, 3A, 3B 모두 정책 가상머신 생성 요청 거부 횟수가 비슷하기는 하지만 1A과 3A 정책은 Migration 수가 3B보다 많으므로 3B가 더 효과적인 정책이라고 볼 수 있는 것이다. 2A, 4C정책은 가상머신 생성 요청 거부 횟수와 마이그레이션 횟수가 많았기 때문에 성능이 좋지 않았다. 2B정책은 가

3) 3A정책과 3B정책의 Migration을 위한 가상머신 선택문제에서 cpu 사용량 기준으로 가장 많은 사용량을 가지는 가상머신을 제외한 상위 몇 %를 선택할 것인가에 대한 비율을 40%로 하고 테스트한 결과이다. 그 이하로 비율을 낮추면 Migration이 거의 일어나지 않게 되어 임계치 90을 가지고 테스트 한 결과와 같은 이유로 제외되었다.

상머신 생성 요청 거부 횟수는 3B 정책과 비슷하지만 마이그레이션 횟수가 너무 많아서 성능이 좋지 못하다.

3B 정책을 Migration에 대한 동작이 없는 상태로 테스트한 결과 평균적으로 46회를 가상머신 생성 요청 거부하였다. 대부분의 Migration정책보다 많은 거부 횟수를 보인 것이며 특히 효과적인 정책이라고 제시한 3B 정책에 비해서도 많은 거부 횟수를 보였다. 이 결과는 적절한 Migration정책을 통해서 자원사용량을 균등하게 유지 하지 않으면 서비스의 질이 떨어질 수 있음을 의미한다.

가상데스크탑 서비스를 서비스하는 시스템 관점에서의 성능을 비교하기 위해 중요한 것은 한 번에 얼마나 많은 사용자가 이용할 수 있는냐는 것이 중요하다. 다음 그림 11 '정책별 최대 동시 사용자 수'에서는 각 정책 별로 최대 동시 사용자 수를 보여주고 있다. 정책 3A, 3B 4A, 4C의 최대 동시 사용자 수가 높은 것을 볼 수 있고 그 중에 3A 정책이 가장 최대 동시 사용자 수가 높은 것을 확인 할 수 있다. 하지만 정책 3A, 3B 4A, 4C 들 간에는 큰 차이가 보이지 않으므로 제시된 Migration 정책들은 다소 성능차이를 보이고 3A 정책이 특히 좋은 성능을 보이고는 있으나 크게 좋은 성능을 보인다고 할 수 없다. 그렇지만 Migration 정책이 적용되지 않은 상태에서의 테스트 결과에서는 평균적으로 108 정도의 최대 동시 사용자 수를 보였다. 3A, 3B, 4A, 4C 정책이 조금은 더 성능이 좋다는 것을 의미한다.

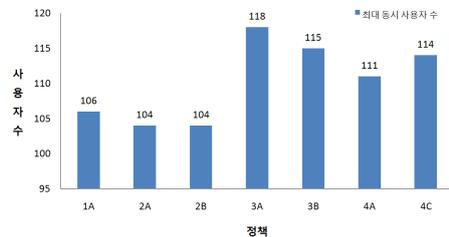


그림 11 정책별 최대 동시 사용자 수  
Figure 11 Number of concurrent users by policies

## VI. 결론

가상화 시스템 분야는 향후 몇 년간 급속히 발전할 것으로 예상된다. 가상화의 여러 응용 분야들 중 개인 사용자를 대상으로 하는 솔루션들은 VMware사의 VMware workstation이나 Microsoft사의 Virtual PC의 경우에서 보듯이 부분 혹은 전체적으로 무료화되고 있는 추세이다. 대신 가상 데스크탑 서비스와 같이 서버 가상화를 통한 서비스 플랫폼 구축 분야는 그 시장성이 인정받고 있다고 볼 수 있을 것이다.

본 논문에서 제시한 가상화된 서버에서의 부하 분산 정책들은 이러한 가상화 서비스 플랫폼의 성능 개선에 크게 기여하게 될 것이다. 또한 제안된 부하 분산 정책을 채용한 가상 데스크탑 서비스는 사용자의 서비스 질에 대한 요구를 충분히 만족 시켜 줄 수 있을 것이다.

향후 본 논문에서 수행한 테스트 결과에 대한 여러 가정과 전제들에 대한 구체적인 연구가 계속되어야 할 것이며, 이러한 지속적인 연구는 가상화 기술을 이용한 서비스의 질적 향상에 크게 기여하게 될 것이다.

## 참고문헌

- [1] 신동하, uC/OS-II 실시간 커널의 가상화를 위한 하이퍼바이저 구현, 컴퓨터정보학회 논문지, 12권 5호, pp. 103-112, 2007.
- [2] Hideo Masuda, Using coLinux to Provide a Linux Environment on Windows PC in Public Computer Labs, ACM SIGUCCS conference on User services SIGUCCS '06, Proceedings of the 34th annual, pp. 221-224, 2006.
- [3] Wikipedia, Native virtualization, [http://en.wikipedia.org/wiki/Native\\_Virtualization](http://en.wikipedia.org/wiki/Native_Virtualization), 2007.
- [4] Users' Manual v3.0, <http://www.xensource.com>, XenSource, Inc., 2005.
- [5] Yang Yu, Feather-weight Virtual Machine for Windows Applications, ACM/Userix International Conference On Virtual Execution Environments, pp. 24-34, 2006.
- [6] Wikipedia, Application Virtualization, [http://en.wikipedia.org/wiki/Application\\_Virtualization](http://en.wikipedia.org/wiki/Application_Virtualization), 2007.
- [7] 이호, The Evolution of Virtual Infrastructure Management, VMware 가상화 로드쇼 2007, Git, Inc., June 2007.
- [8] <http://www.xensource.com>, XenSource, Inc., 2005.
- [9] Paul Barham, Xen and the art of Virtualization, <http://www.cl.cam.ac.uk/research/srg/netos/papers/2006-xen-lw-masterclass.pdf>, XenSource, Inc., 2006.
- [10] 문병주, 서버가상화 기술 동향, 전자통신연구원 주간기술동향 통원 1302호, June 2007.
- [11] R. D. Nelson, An Approximation to the Response Time for Shortest Queue Routing, ACM SIGMETRICS Performance Evaluation Review, Proceedings of the 1989 ACM SIGMETRICS international conference on Measurement and modeling of computer systems SIGMETRICS '89, Volume 17 Issue1, pp. 181-189, April 1989.
- [12] Milan E, Simulation of Load Balancing Algorithms, ACM SIGCSE Bulletin, Volume 34 Issue 4, pp. 138-141, December 2002.
- [13] B. Ng., A Multi-Server Architecture for Distributed Virtual Walkthrough, In Proceedings of the ACM Symposium on Virtual Reality Software and Technology(VRST'02), pages 163-170, November 2002.
- [14] S. Pekkola, Collaborative Virtual environments in the year of the dragon, In Proceedings of the ACM International Conference on Collaborative Virtual

Environments(CVE'00), pages 11-18, 2000.

- [15] J. C. S. Lui, An efficient partitioning algorithm for distributed virtual environment systems, IEEE Transactions on Parallel and Distributed System, 13(3):193-211, March 2002.

## 저자 소개



손성훈

1991 서울대학교 계산통계학과 학사  
 1993 서울대학교 전산학과 석사  
 1999 서울대학교 전산학과 박사  
 1999 ~ 2004 한국전자통신연구원  
 임연구원  
 2004 ~ 현재 상명대학교 소프트웨어  
 학부 조교수  
 관심분야 : 저전력 임베디드 컴퓨팅, 가  
 상화, 멀티미디어 시스템 등



전상일

2006년 상명대학교 소프트웨어학부 학사  
 2006년 ~ 현재 상명대학교 일반 대학원  
 컴퓨터과학과 석사 과정  
 2007년 ~ 현재 (주) 마크애니 연구원  
 관심 분야 : 가상화, 임베디드 컴퓨팅, 실  
 시간 운영체제 등