



건국대학교 컴퓨터공학과

졸업작품보고서

지도교수 : 장 천 현 교수님

학번 : 200812423

이름 : 김준식

1. 개발 계획

i. 개발목표 및 개요

“ZFS기반의 KVM을 이용하여 사용자가 원하는 가상머신을 효율적으로 생성 및 사용하는 서비스 개발.”

1. 배경

리눅스를 사용하기위해 시스템 가상화 툴을 이용한 수업을 들은 이후로, 가상화에 관심을 갖게 되어, 오픈소스 가상화 기술을 알게 되었다. 그중 KVM은 리눅스를 기반으로 동작하는 type 2 의 하이퍼바이저로서 널리 사용되고 있다. 단순히 가상머신을 만드는 데 그치지 않고, 어떻게 하면 더 좋은 성능을 낼 수 있는지, 어떻게하면 사용자가 편리하게 가상머신을 사용할 수 있을지를 고민했다. 그러던 중 *joyent*사에서 같은 UNIX기반의 open Solaris에 KVM을 포팅하여 Opensource로 배포하기 시작했다. 구조적으로 ext4파일 시스템 기반이었던 리눅스와는 다르게 ZFS파일 시스템을 이용하여 만들어지는 독립적인 공간인 Zone과 빠른 snapshot, clone기능을 통해 가상머신 생성 및 제어에 적합하다고 판단 되었다.

2. 목표

SMARTOS에서 원격 가상 데스크톱 환경을 구축하여 사용자가 원하는 가상머신을 생성하여 사용 가능한 플랫폼을 구축한다.

3. 개요

- a. SMARTOS에 가상머신을 생성 및 제어하는 모듈을 개발한다.
- b. 가상머신 Pool을 만들어 필요할 때 마다 복제하여 빠른 생성 및 제공이 가능하도록 한다.
- c. SPICE 프로토콜을 이용하여 가상머신 사용환경을 구축한다.
- d. 원격에서 서버에 접속하여 원하는 가상머신을 생성및 실행이 가능한 관리 서버를 구축한다.

ii. 개발내용

1. 기반기술

a. Virtualization (시스템 가상화)

- 가상화 : 컴퓨터의 한정된 자원을 다수의 운영체제나 응용프로그램들이 나눠 사용하는 개념.
- 가상화의 분류

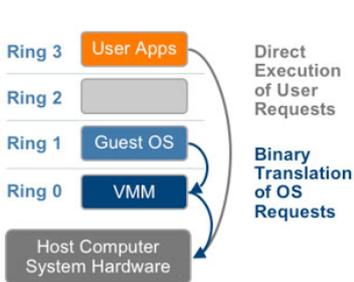


그림1. 전가상화 계층도

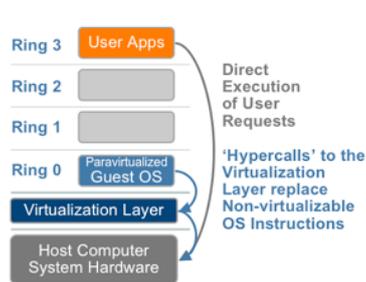


그림2. 반가상화 계층도

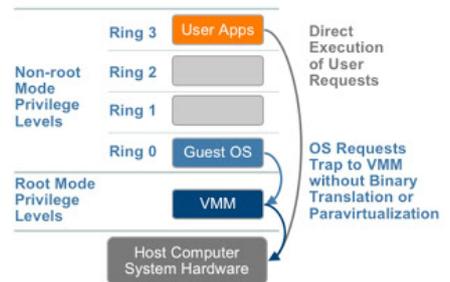


그림3. 하드웨어 지원 가상화 계층도

• 전가상화[Full]

- 전가상화된 OS들은 자신들이 가상머신인줄 모르고 하드웨어 자원을 독점한다고 생각하며 사용된다.
- 가상화된 OS는 Ring0의 특권 명령 권한이 필요한 경우에 하이퍼 바이저가 해석을 통해 전달해야하는 오버헤드가 존재한다.
- 유저레벨수준의 명령어는 해석작업이 필요없어 빠른속도로 사용된다.
- 기반의 가상머신의 영향을 많이 받는다
- VMware ESX등이 있다.

• 반가상화[Para]

- GuestOS의 수정을 통해 Hypervisor의 API를 통해 특권 명령어를 사용하게되어(Hyper call) 해석의 오버헤드가 적다.
- 반가상화의 GuestOS의 수정이 전가상화의 구현보다는 편하기 때문에 많이 사용된다.
- Xen등이 있다.

- 하드웨어지원가상화[Hardware Assisted]

-가상화의 지원을 CPU자체에서 받는 방식으로, Intel-VT/AMD-V기술의 등장 이후로 발전하고 있다.

-Ring -1레벨이 가상화된 OS에게 주어지게 되면서 특권 명령어를 사용할때의 중간계층의 오버헤드가 존재하지 않게되었다.

-KVM등이 있다.

b. KVM (Kernel based Virtual Machine)

- KVM / QEMU

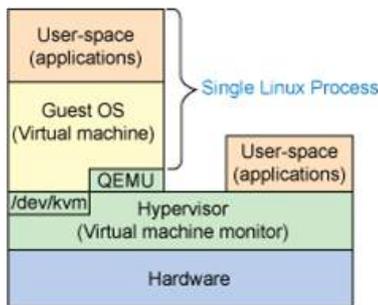


그림4.KVM/QEMU가 사용된 가상화 계층도

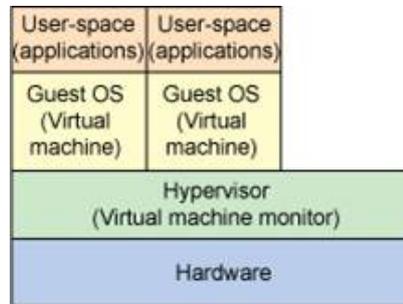


그림5.VMware가 사용된 가상화 계층도

KVM

-리눅스기반의 하이퍼바이저로서 독립적인 메모리할당과 다양한 OS를 에뮬레이팅시켜준다.

-KVM은 하나의 VM(GuestOS)을 하나의 리눅스 Process로 사용한다.

-유저영역의 프로그램들은 /dev/kvm 를 인터페이스로 사용하여 가상머신의 메모리로 접근한다.

-Hypervisor를 모듈화시켜 커널의 구동화 함께 지속적으로 가상머신을 관리한다.

-Intel-VT/AMD-V기술을 지원하는 CPU를 필요로 한다.

-시스템상에 두가지 모듈로 존재하는데 KVM.ko는 시스템가상화인프라를 담당하고 KVM-intel /amd모듈은 CPU의존적 모듈이다.

QEMU

-에뮬레이터로서 가상화된 OS와의 I/O인터페이스를 담당한다.

-가상화된 OS의 명령어 셋을 번역한다.

-하드웨어지원(KVM)을 이용한 가상화 방식과 순수 에뮬레이팅 방식이 존재한다.

-KVM은 메모리와 CPU명령권한을 관리하고 실질적인 에뮬레이팅은 QEMU가 수행한다.

c. SMARTOS

- ZFS (The Z File System)

Ext4 대신 ZFS 파일시스템을 이용하는 SMARTOS를 이용해 가상머신은 RAW방식으로 저장 공간을 사용하고 파일시스템, 저레벨 수준의 snapshot/clone 기능을 구현한다. 이를 통해 snapshot에 더 나은 성능을 가지게 된다.

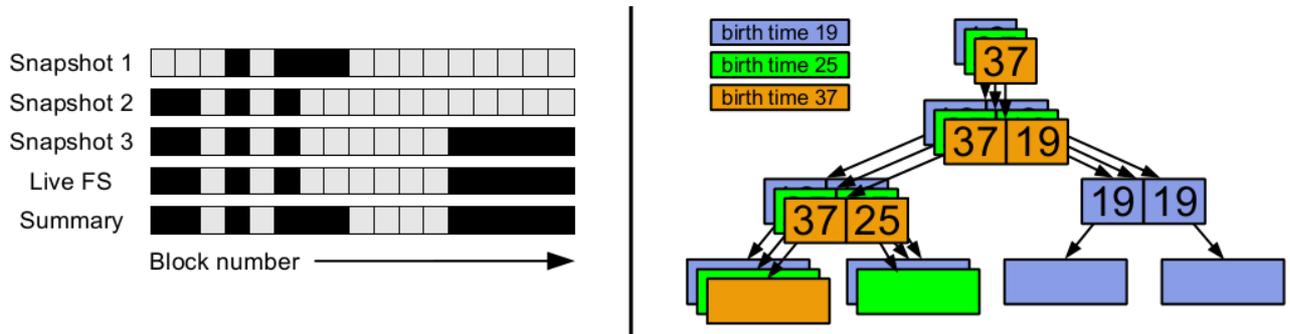


그림6. ext4 와 ZFS의 Snapshot 방식 비교

	VM		
VM's disk allocation strategy	RAW	qcow2	RAW
Virtualization	kvm/qemu		
File system	ext4		ZFS
OS	Linux		SMART
	-better performance	-snapshot -less occupy	-low level snapshot -isolation

그림7. Linux와 SMARTOS의 가상머신 환경 비교

- Zone

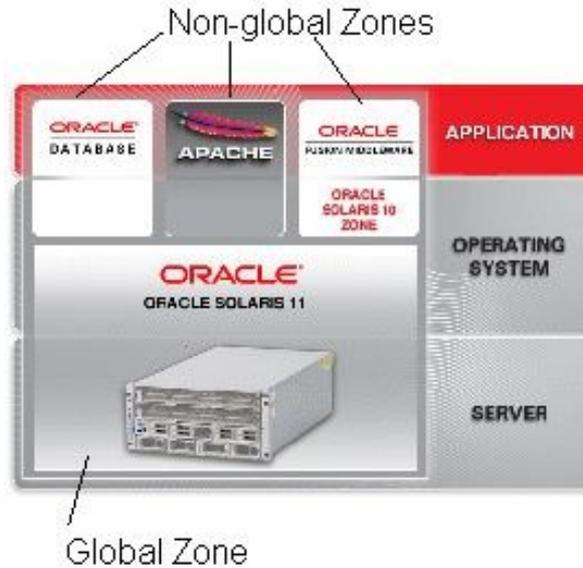


그림8. Solaris Zone 개념도

Smart OS 통해 만들어지는 가상머신은 Zone영역에 생성된다. Zone은 가상화된 운영체제 별로 독립적인 공간을 가지게 한다. 각각의 OS 별로 CPU, Memory, network를 할당을 관할하는 권한이 주어지기 때문에 Hypervisor기술보다 가상화 기술에 더 적합하다고 할 수 있으며 가상머신과 OS(Server)간의 적은 오버헤드를 발생시킨다. 또한, 가상머신 생성, 소멸시 상대적으로 적은 자원으로 생성이 가능하다.

- Network virtualization

Network 가상화를 통가해 자원 할당 및 제어, Live 모니터링이 가능해진다. 이는 이전의 단순한 QoS(Quality of Service)방식보다 한 단계 더 발전한 방법으로 모든 패킷을 받고 그 이후에 처리량에 따라 분배하여 발생하는 불필요한 패킷손실, 연산처리 등이 발생하지 않는다. 이는 더 나은 성능의 Network I/O를 제공하며 가상머신간의 실시간 상황에 따라 동적으로 처리하는 것이 가능하게 해준다. Solaris에 존재하는 Crossbow기술을 활용하여 Network 가상화를 구현하고 Quality of Service를 제공하는것이 가능하다.

d. SPICE

원격에서 컴퓨터를 사용하게되면 생기는 많은 시공간적인 비용의 의해서 개발된 Remote Desktop Protocol은 발전을 거듭해 현재 VNC, NX, SPICE, MS-RDP등 다양한 프로토콜이 생겨나게 되었다.

가상머신의 화면을 직접 송출하지 않고, 비디오 드라이버가 클라이언트와 직접 연결되어 일반 화면, 비디오, 오디오 등을 따로 코덱을 이용하여 압축, 멀티채널을 통해 전달하여 실제 컴퓨터에 근접한 환경을 제공한다.

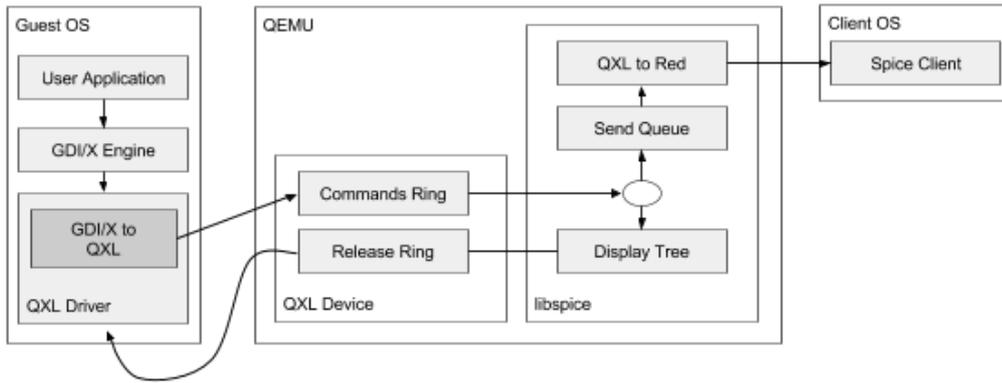


그림9. SPICE를 사용하는 QEMU

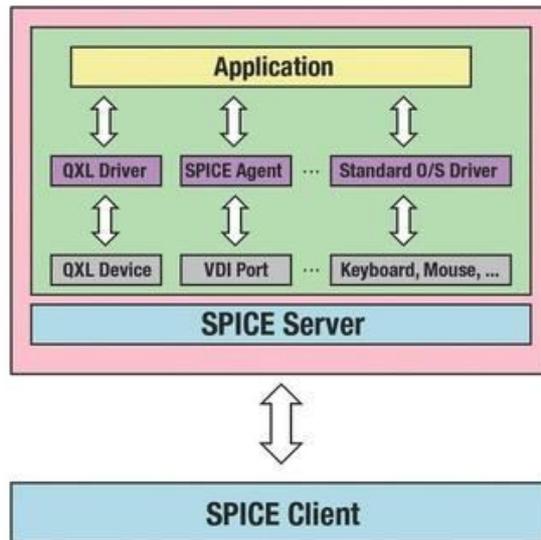


그림10. SPICE 멀티채널 통신 구조도

GuestOS에 설치되는 SPICE Driver, Hypervisor에 설치되는 SPICE Device, 클라이언트에 설치되는 SPICE Client로 구성된다.

GuestOS내부에 SPICE Driver를 설치하게 되면 GuestOS내부에서 화면을 객체화시켜 전달되고 렌더링된 후 유저에게 보여지고 조작되어 다시 객체화 되기때문에 이론상 화면의 딜레이가 원거리에 있더라도 존재하지 않는다. 하지만 대역폭과 QoS등의 이유로 다소 지연이 있다. 하지만 원격에서 사용할경우 기존의 다른 통신프로토콜보다 부드러운 움직임을 보여준다.

e. Ruby on Rails

루비 온 레일즈(Ruby on Rails)는 루비로 작성된 MVC 패턴을 이용하는 오픈 소스 웹 프레임 워크이다. 줄여서 레일즈라 불리기도 한다. 덴마크의 David Heinemeier Hansson에 의해 시작되어 오픈 소스로 개발되고 있다. 특히 데이터베이스를 이용한 웹 애플리케이션을 개발할 때 반복되는 코드를 대폭 줄여 개발 기간을 단축하는 것으로 인기를 끌고 있다.

루비 온 레일즈는 모델, 뷰, 컨트롤러의 템플릿 생성 및 테스트 등 자동화하는 툴, 테스트를 위한 웹서버 등을 포함하고 있다. 또한 루비 온 레일즈가 기반을 두고 있는 ActiveRecord는 데이터 베이스 작업을 추상화하여 생성/읽기/갱신/삭제(Create/Read/Update/Delete, CRUD) 작업

을 자동화 및 단순화 시켜준다. 웹 애플리케이션의 동작은 대부분 CRUD 작업을 통해 이루어지므로, 이와 같은 루비 온 레일즈의 특성은 웹 애플리케이션을 빠르게 개발할 수 있도록 도와준다.

2. 구현 내용

a. Virtual Machine Management Module

가상 머신의 생성 삭제 등 전체적인 관리 기능의 통합 모듈을 개발 한다. 가상 머신을 설치 또는 삭제 하기 위해서는 기본적인 운영체제 이미지 설치 및 환경설정 등 복잡한 작업이 요구되어진다 가상 머신유지 관리에 필요한 복잡한 작업들을 자동화 하고 가상머신의 환경설정에 필 요한 각각의 기능들을 모듈화 하여 통합적인 관리가 가능하도록 한다.

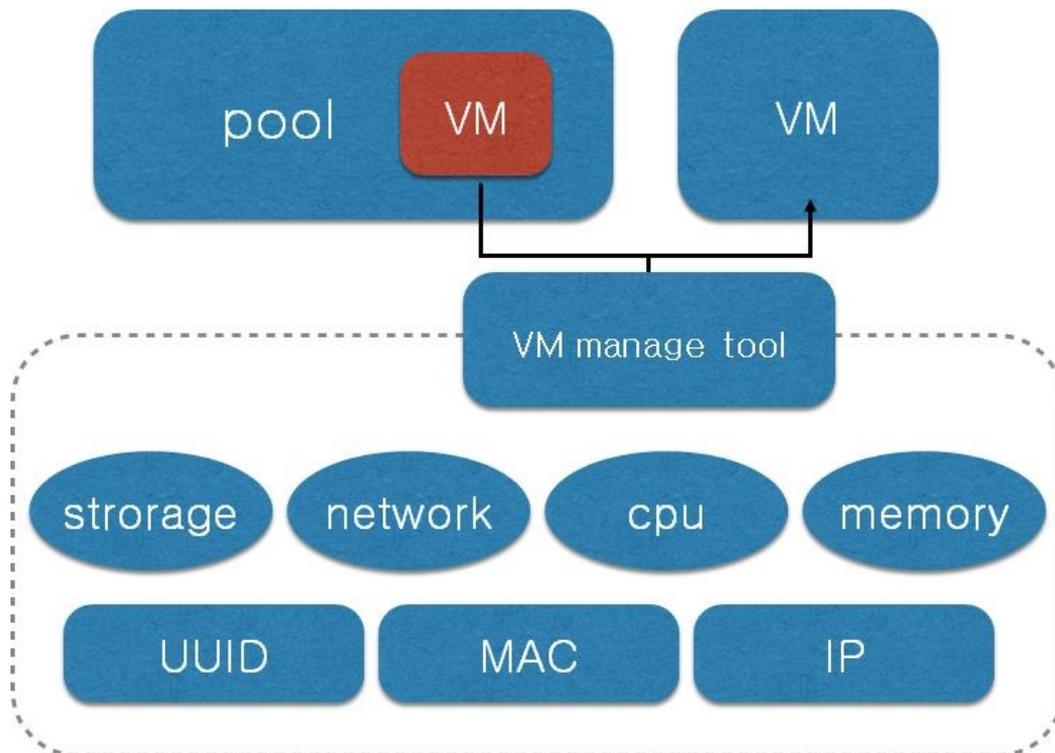


그림11. Pool방식 가상머신(VM) 관리 모듈

b. 가상머신 생성 pool 구축

사용자가 원하는 OS를 선택한 이후 바로 사용이 가능하도록 설치 완료된 상태의 템플

릿을 설치해두고, 이를 Pool로 활용하기위해 ZFS파일 시스템의 Snapshot을 사용하여 사용자

가 원하는 Storage, CPU, Memory 등을 선택하여 빠르게 제공할 수 있도록 구축한다.

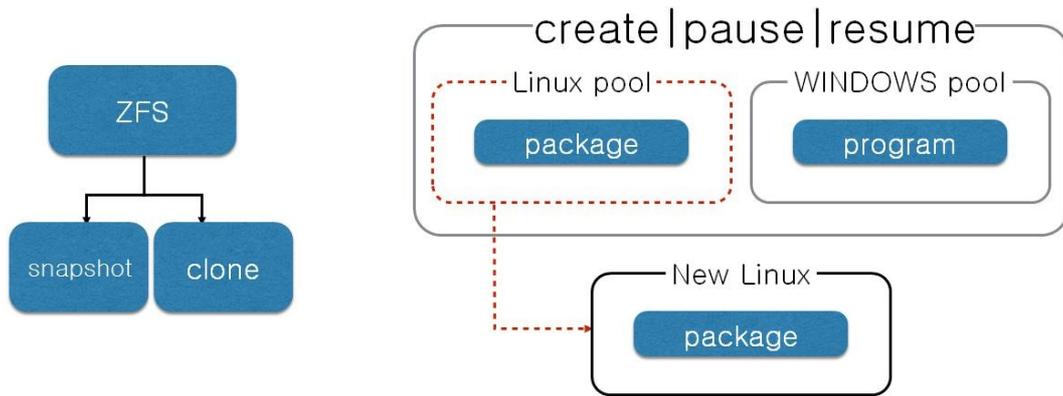


그림12. Pool 방식 가상머신(VM)

c. SMARTOS/KVM 기반의 특정 가상머신 Provision 및 Template화

사용자가 Guest VM SPICE 통신을 이용해 원격으로 사용하기 위해서는 Agent설치를 통해 웹 상으로 전송이 가능하다, Template화 된 이미지를 통해 즉시 원하는 크기로 확장되고, CPU 갯수와 Memory확장이 가능하도록 정형화된 이미지들을 구축한다.

사용 툴/기능

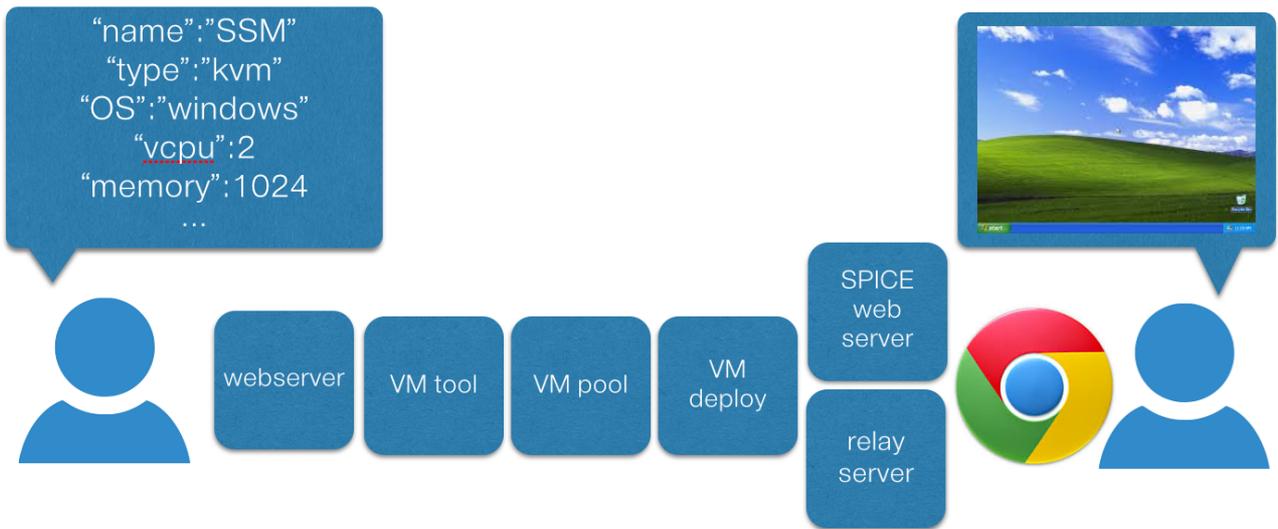
- ZFS Clone/Snapshot
- KVM -SPICE-HTML5
- vmadm
- imgadm

d. webserver

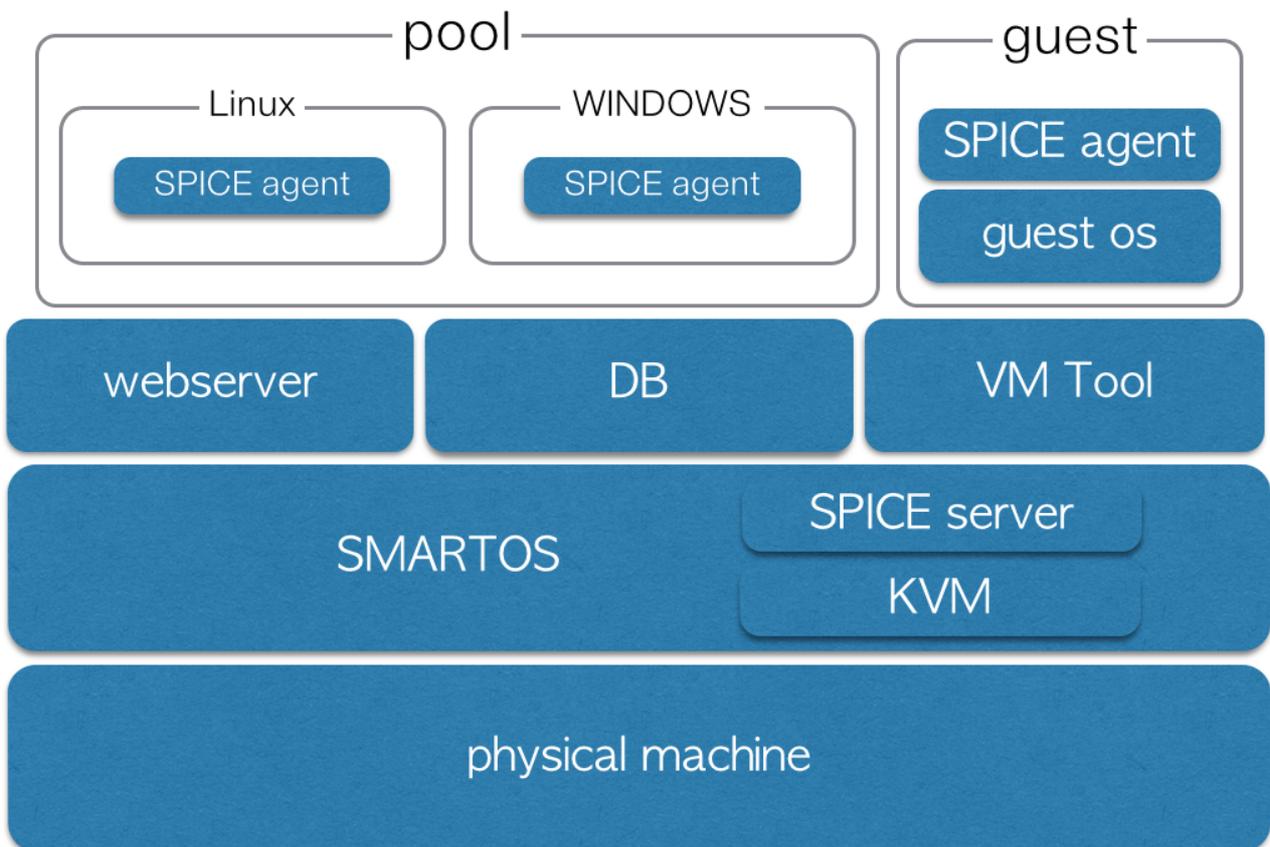
웹접근을 통해 사용자가 원하는 가상머신 생성과 사용이 가능하도록 하기위해서 가상머신을 생성하는 서버와 사용자 사이의 세션을 관리해주고 명령어를 실행시켜주는 서버가 필요하다.

사용자는 모든 작업을 웹을 통해 이용하게 된다. 가상머신 생성, 삭제, 관리 등의 기능을 제공하고 세션 유지 및 관리가 가능하도록 한다. 또한, 각 계정마다 서버에 접근 시 리다이렉션을 통해 중복 실행을 방지하고 연속적인 작업이 가능하도록 한다. Ruby on rails frame work와 mongoddb로 web server를 구축한다.

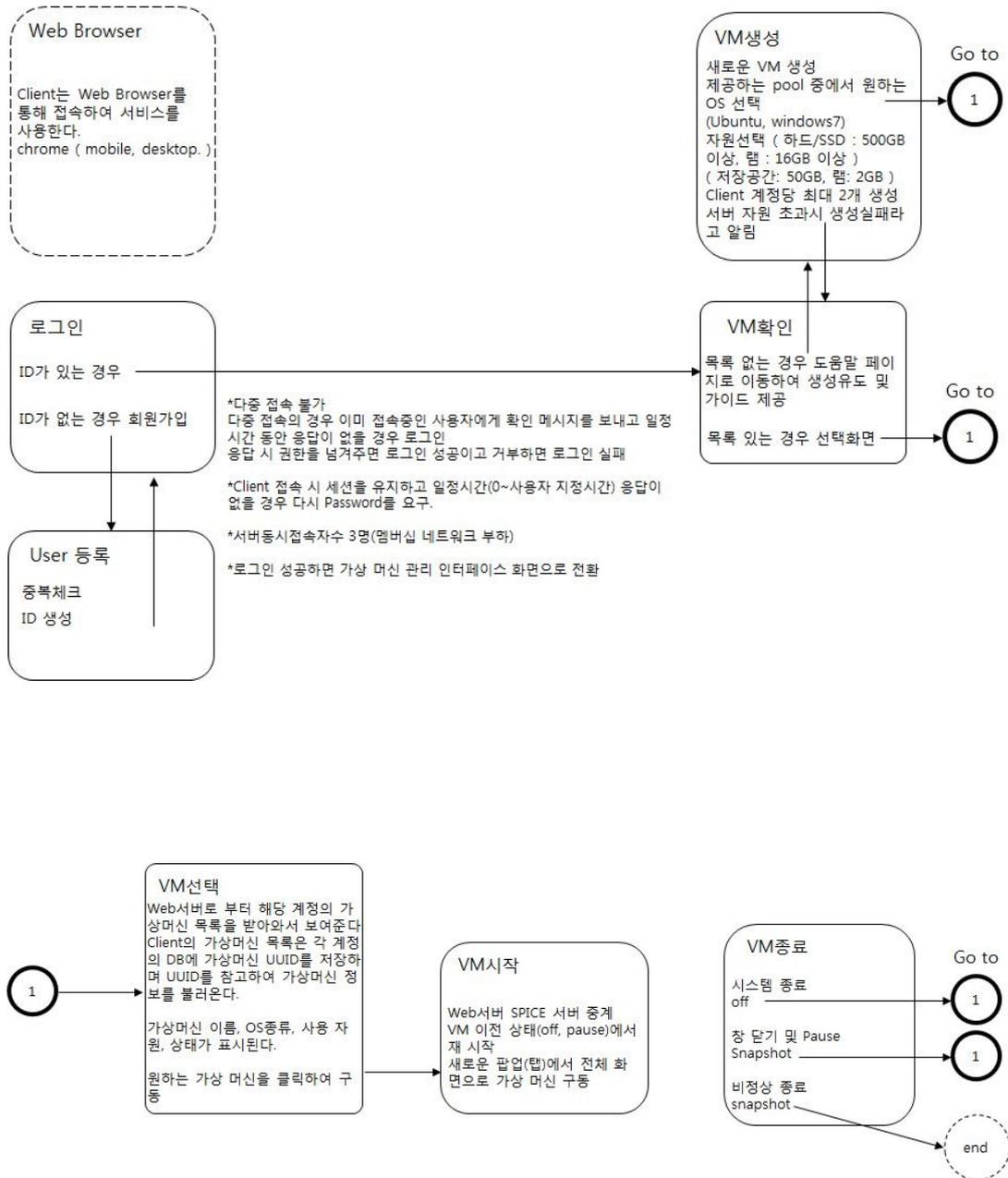
iii. system flow diagram



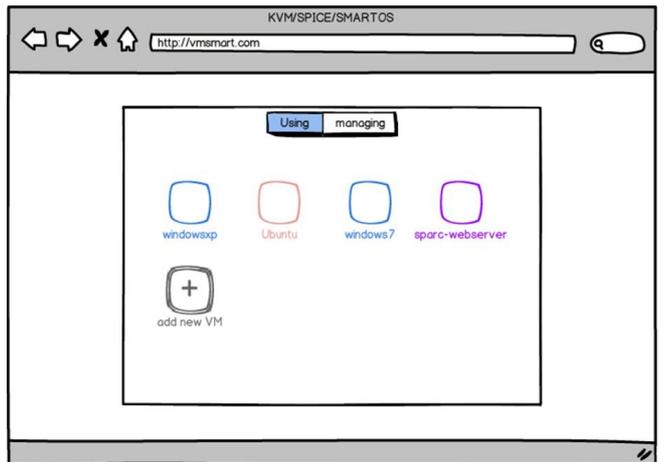
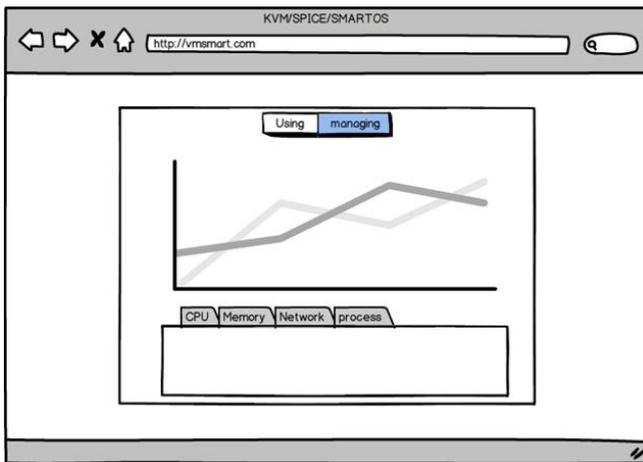
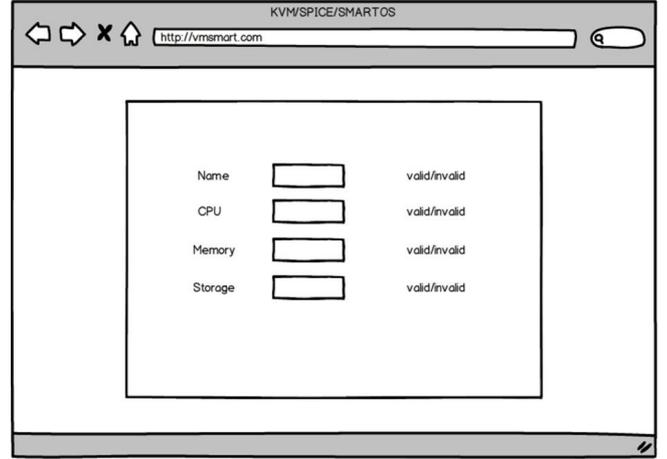
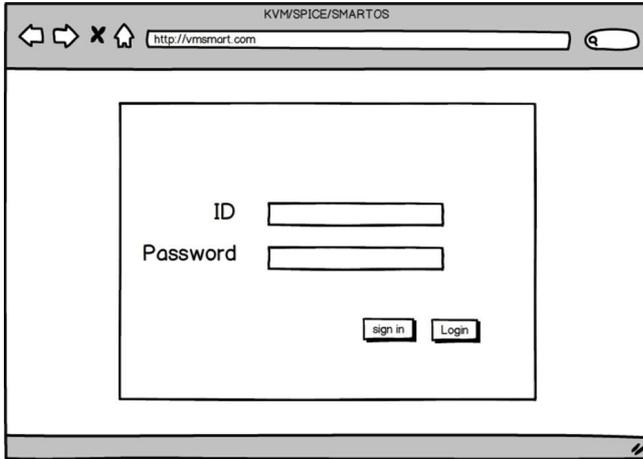
iv. 시스템계층도



v. 시나리오



vi. 화면 구성도



vii. 추진계획

	5월	6월	7월	8월	9월	10월	11월
졸업작품 계획 및 보고서 작성 및 제출	■						
SMARTOS / KVM 환경 구축		■	■				
ZFS pool 관리 모듈 개발			■	■			
가상머신 관리 모듈 개발				■	■	■	
템플릿 Pool 설정 및 관리 모듈 개발						■	■
피드백 및 테스트							■

2. 발전 방향

기존 문제점

가상머신을 통해 원하는 OS를 사용할 수 있게 되었다. 그러나 Virtual Box, VM ware와 같은 상용 가상머신의 경우, 제약적인 GUI와 접근성은 일반 사용자들에게 주력 컴퓨터로서 가상머신이 사용되기까지 꽤 오랜 시간이 걸릴 것으로 보인다. 이런 불편한 접근성을 개선하기 위하여 웹을 통해 원하는 OS의 가상머신을 사용할 수 있는 전체적인 플랫폼을 구축하고자 한다.

발전가능성

vagrant, chef 등 과 같이 가상머신을 원하는 모양 그대로 다른 플랫폼에서 생성할 수 있고, 원하는 패키지를 사용할 수 있는 시대가 왔다. 하지만 필수적인 Infrastructure를 생성해주는 작업만은 필연적이어서 어떤 종류의 OS를 만들고 어떤 수준의 사용자 선택권을 주어 주느냐에 따라 다른 목적의 가상 플랫폼이 된다.

VDI(Virtual Desktop Interface)를 넘어 WDI(Web Desktop Interface)를 시도해 보면서 다양한 기술을 전반적으로 부딪혀 보는 좋은 계기가 될 것이라 믿는다.

3. 용어정리

1) KVM

2007년 Qumranet사가 Redhat사에 인수된 기술로 Full - Virtualization중에서도 인텔과 AMD의 하드웨어 가상화지원 기술을 이용한 Hardware assisted Virtualization기술 기반인 KVM(Kernel based Virtual Machine)은 Type-2형의 Hypervisor로서 리눅스 기반의 오픈소스다. 커널 모듈로서 구현되어 리눅스 커널자체가 Hypervisor화 됨을 이용해 Context swap을 빈도를 줄여 CPU성능에 장점이 있는 기술이다. 2012년 OpenSolaris기반인 SMARTOS에 포팅되어 13년에 민간 이용자들이 사용 가능해졌다.

2) SMARTOS

SOLARIS

SOLARIS는 Sun microsystems의 기술로 현재는 Oracle사에 인수되어 라이선스를 갖고 있다. 250MB정도의 낮은 메모리 점유를 통해 가상머신들이 메모리 사용함에 있어 여유를 준다.

ZFS

ZettaByte File System은 기존의 Linux의 ext4포맷의 페이징 파일시스템과는 다르게 Copy-On-Write기반의 논리적인 공간으로서 snapshot과 clone이 손쉽다.

File의 수정되는 부분만을 추가적으로 기록하는 구조로서 500MB의 Ubuntu의 20개 이미지 총 크기로 몇 MB로 각각 더 많은 다른 분기를 가질수록 용량을 점차적으로 커지게된다. 가상머신은 각각 고유한 Storage를 필요로 하는데 고립성, 독립성, 확장성, 안정성을 고루 갖춘 File system으로 적합하다.

DTrace

안정적이고 가벼운 자원모니터링 툴로 memory 사용량부터 각 프로세스의 움직임까지 볼 수 있다. 가상머신을 생성하고 구동하는데 있어, 어떤 지점이 병목지점인지 체크하기 위해 사용한다.

ZONE

Solaris에서는 별다른 Overhead없이 OS차원에서 독립적인 파티션을 제공한다.

OS virtualization로서 ZFS와 Crossbow와 함께 가상환경의 최적화를 제공한다.

다만 Solaris자체를 가상화 하는데 특화되어있기때문에 이기종의 OS를 가상화 하기 위해서는 파티션만을 제공하고 KVM을 통해 에뮬레이팅하게 된다. 단순 DB나 웹서버등의 기능을 사용하고자 할 때 사용하기 용이하다.