
가상화 분야 Stack 통합 테스트 결과보고서 [KVM]

2012. 05.

목 차

I. Stack 통합 테스트 개요	3
1. 목적	3
II. 가상화 소개	4
1. 가상화	4
2. 가상화 주요 혜택	4
3. 가상화 분야 주요 공개SW	6
III. 테스트 대상 소개	7
1. KVM 소개	7
IV. Stack 통합 테스트	9
1. 테스트 환경	9
2. 주요 테스트 방법	9
3. 테스트 시나리오 현황	10
4. 기능 테스트 수행 결과	11
V. 종합	12
※ 참고자료	13
[별첨 1] KVM 테스트 케이스	
[별첨 2] KVM 설치 가이드	

I. Stack 통합 테스트 개요

공개SW Stack 통합테스트는 여러 공개SW들의 조합으로 시스템 Stack을 구성한 후 Stack을 구성하는 공개SW의 상호운용성에 중점을 두고 기능 및 성능테스트 시나리오를 개발하여 테스트를 진행한다.

본 통합테스트를 통해 안정된 Stack 정보를 제공하여 민간 및 공공 정보시스템 도입 시 활용될 수 있도록 한다.

1. 목적

- 공개SW Stack 통합 테스트 수행 목적
 - 공개SW로 구성된 Stack이 유기적으로 잘 동작함을 확인
 - 다양한 Stack 구성에 기반을 둔 테스트를 통해 안정된 Stack 조합 규명
 - 공개SW 시스템 도입을 위한 Stack 참조모델의 신뢰성 정보로 활용
 - 공개SW의 신뢰성과 범용성에 대한 사용자 인식 제고

II. 가상화 소개

1. 가상화(Virtualization)

컴퓨터 리소스의 추상화를 일컫는 광범위한 용어이다. “물리적인 컴퓨터 리소스의 특징을 다른 시스템, 응용 프로그램, 최종 사용자들이 리소스와 상호작용하는 방식으로부터 감추는 기술”로 정의할 수 있다.

이것은 다중 논리 리소스로서의 기능을 하는 것처럼 보이는 서버, 운영체제, 응용 프로그램 또는 저장 장치와 같은 하나의 단일 물리 리소스를 만들어 낸다. 또한 단일 논리 리소스처럼 보이는 저장 장치나 서버와 같은 여러 개의 물리적 리소스를 만들어 낼 수 있다.

기업은 서버 가상화를 통해 하나의 컴퓨터에서 동시에 1개 이상의 운영체제를 가동시킬 수 있다. 대부분의 서버는 단지 용량의 10~15%만 사용하는데, 가상화는 이런 서버의 효율률(utilization rate)을 70% 그 이상으로 올릴 수 있다. 높은 수준의 효율률은 같은 분량의 업무처리에서 요구하는 컴퓨터 수를 줄여준다.

(출처 : 위키백과)

2. 가상화 주요 혜택

높아진 자원 활용률

물리적 자원들과 자원 풀에 대한 동적인 공유를 가능하게 해 주며, 이를 통해 더 높은 자원의 활용률을 얻을 수 있다. 특히 평균 워크로드가 전체 자원의 워크로드보다 훨씬 적은 가변적인 워크로드 상황에서는 더 높은 효과를 얻을 수 있다.

낮아진 관리 비용

관리되어야 하는 물리적 자원들을 줄여 줌으로써 관리 인력의 생산성을 향상시킬 수 있다. 또한 물리적 자원들의 복잡성을 숨겨 주고, 자동화·정보화·중앙화를 통해 공통된 관리 작업을 단순화시키며, 워크로드 관리의 자동화를 가능하게 해 준다. 그뿐만 아니라 가상화는 다른 기종의 플랫폼 환경에서도 관리 도구를 공통으로 사용할 수 있게 해 준다.

□ 향상된 보안

단순한 공유 메커니즘에서는 불가능한 분리와 격리를 가능하게 해 주어 데이터와 서비스에 대해서 통제되고 안전한 접근을 제공한다.

□ 높아진 가용성

사용자 레벨에 아무런 영향을 주지 않고도 물리적 자원이 제거되거나 업그레이드 또는 변경될 수 있도록 지원해 준다.

□ 증가된 확장성

물리적 자원의 구성 변경이 없어도 필요한 만큼의 적절한 확장성을 얻을 수 있다.

□ 상호 운영성 및 투자의 보호

가상화 자원들은 기존 물리적 자원들 사이에서는 불가능한 인터페이스와 프로토콜 레벨에서의 호환성을 제공한다.

□ 향상된 프로비저닝

자원의 할당을 개별 물리적 단위보다 더 세밀한 조각 단위에서 가능하게 해 준다.

(출처 : 가상화 기술의 새로운 패러다임)

3. 가상화 분야 주요 공개SW

[표 II-1. 주요 공개SW]

제품명	Stack 환경		홈페이지	비고
KVM	Host OS	Linux Platform	http://www.linux-kvm.org	Hypervisor ¹⁾
	Guest OS	Cross-platform		
Qemu	Host OS	Cross-platform	http://www.qemu.org	Emulator ²⁾
	Guest OS	Cross-platform		
VirtualBox	Host OS	Cross-platform	http://www.virtualbox.org	Hypervisor
	Guest OS	Cross-platform		
Xen	Host OS	Cross-platform	http://www.xen.org	Hypervisor
	Guest OS	Cross-platform		

1) 호스트 컴퓨터에서 다수의 운영 체제(operating system)를 동시에 실행하기 위한 논리적 플랫폼(platform)을 말한다. 가상화 머신 모니터(virtual machine monitor, 줄여서 VMM)라고도 부른다.

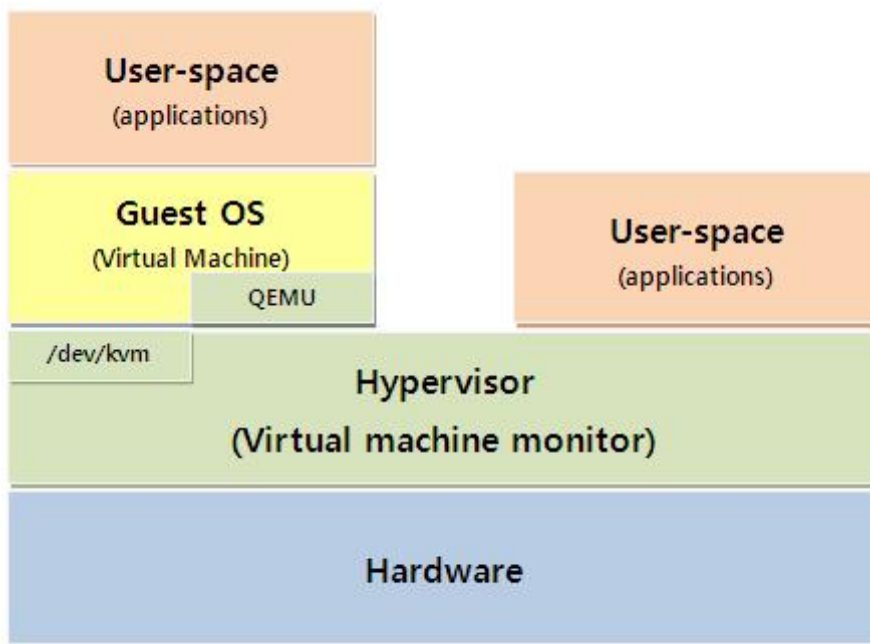
2) 한 시스템에서 다른 시스템을 복제한다. 그리하여 두 번째 시스템이 첫 번째 시스템을 따라 행동하는 것이다. 외부의 행동에 대해 똑같이 따라하려고 하는 이 관점은 시뮬레이션과는 정 반대이다. 왜냐하면, 시뮬레이션은 자주 내부 상태와 관련하여, 흉내내는 시스템의 추상적인 모델과 관계가 있기 때문이다.

III. 테스트 대상 소개

1. KVM 소개

KVM(Kernel-based Virtual Machine)은 x86기반의 하드웨어(Intel VT 또는 AMD-V)를 지원하는 전가상화(full virtualization)³⁾ 기반 가상화 솔루션으로 리눅스, 윈도우즈 등 OS(Operating System)의 수정 없이 다양한 가상머신(Virtual Machine)⁴⁾을 사용할 수 있는 이점이 있다.

KVM은 리눅스 커널 2.6.20 버전부터 리눅스 커널에 포함되어 배포되고 있다.



[그림 III-1. KVM 가상화 구성 요소]

3) 하드웨어 자원(CPU, Memory, Disk, Network 등)에 대한 관리를 하이퍼바이저(Hypervisor)가 담당하며, Guest OS는 수정 없이 사용할 수 있고 이러한 특징으로 인하여 다양한 OS를 지원 가능하다. 단지, 전가상화를 지원하기 위해서는 CPU가 가상화 기술을 지원해야 한다.

4) 컴퓨터 과학 분야에서 특정한 프로그램(OS, Apps 등)들을 실행하는 컴퓨터를 실제 컴퓨터 같이 만들어 주는 소프트웨어를 말하며, 본 보고서에서는 Guest OS로 표현한다.

□ CPU 지원 환경

[표 III-1. CPU 지원 환경]

CPU Type	비 고
AMD(AMD-V)	- grep svm /proc/cpuinfo 지원 여부 확인
Intel(VT-x)	- grep vmx /proc/cpuinfo 지원 여부 확인
PowerPC	- 지원 타입 <ul style="list-style-type: none"> • BookE 440 • BookE 500 • Book3S PPC64 • Book3S PPC32
S390	- 64bit 지원

□ Guest OS 지원 환경

[표 III-2. Guest OS 지원 환경]

구분	OS
Linux	- CentOS - Debian - Fedora - OpenSUSE - RHEL(Redhat Enterprise Linux) - Ubuntu
Unix	- FreeBSD - NetBSD - OpenBSD - OpenSolaris - Solaris
Windows	- Windows 2008 Server - Windows 2003 Server - Windows 7 - Windows Vista - Windows XP

※ 추가적인 자세한 정보는 아래의 링크 정보 참조

- http://www.linux-kvm.org/page/Guest_Support_Status

IV. Stack 통합 테스트

1. 테스트 환경

KVM 환경

[표 IV-1. KVM 환경]

모듈	Version
KVM	2.6.32 (Linux Kernel)
qemu-kvm	0.12.1
Virt-Manager	0.9.0.7

Stack 환경

[표 IV-2. Stack 환경]

구성	Host OS	Guest OS	
A Stack	Ubuntu 10.04 Desktop LTS (64bit)	CentOS 6.2	Windows 7
B Stack	CentOS 6.2 (64bit)	Ubuntu 12.04	Windows 7

HW 환경

[표 IV-3. HW 환경]

제조사	모델명	CPU	MEM	Disk	NIC
IBM	X3550M2	Intel Xeon(R)CPU 2.40GHz * 4	8GB	320GB	Gigabit 1Port

2. 주요 테스트 방법

시나리오 테스트

시나리오 테스트 기법은 단일 기능에 대한 결함 여부를 확인하는 것이 아니라 서로 다른 컴포넌트 사이의 상호작용과 간섭으로 발생할 수 있는 결함을 발견하기 위한 기법이다.

본 테스트에서는 사용자 시나리오 테스트 기법을 적용하여 KVM을 사용하는 사용자들이 사용할 수 있는 항목 중 Guest OS 생성, 삭제,

제어, 복제 등에 대한 사용자 시나리오 도출하였으며, 각각의 항목에서 도출한 시나리오 중 사용자가 일반적으로 수행할 수 있는 시나리오를 추출하여 테스트케이스를 작성하였다.

□ 상호 운용성 기반 테스트

애플리케이션이 지원하는 다양한 Stack을 구성하여, 각각의 서버 환경에서 결함이 없이 잘 구동되는지를 확인하며, 비즈니스 로직 및 데이터 상호 연동이 Stack 내의 컴포넌트 간의 인터페이스가 정상적으로 동작하는지 여부를 중점적으로 검증한다.

3. 테스트 시나리오 현황

[표 IV-4. 테스트 시나리오 현황]

기 능	테스트 시나리오	테스트 케이스
Guest OS 생성	2	20
Guest OS 삭제	2	8
Guest OS 제어	10	66
Guest OS 복제	4	28
합 계	18	122

4. 기능 테스트 수행 결과

테스트 수행 관련 세부 시나리오 및 테스트 결과는 별첨 「KVM 테스트 케이스」 문서를 참고한다.

테스트 결과

[표 IV-5. 테스트 결과]

분류		PASS		FAIL		N/A	
기능	개수	A Stack	B Stack	A Stack	B Stack	A Stack	B Stack
Guest OS 생성	20	20	20	0	0	0	0
Guest OS 삭제	8	8	8	0	0	0	0
Guest OS 제어	66	66	66	0	0	0	0
Guest OS 복제	28	28	28	0	0	0	0

- 기능 시나리오를 통한 테스트 수행 결과 Guest OS 생성, 삭제 등 모든 기능이 예상 결과와 동일하게 정상 동작함

참고사항

- KVM 테스트 환경 구성시 가상화 관리 도구(virt-manager)를 활용하여 테스트 시나리오 작성 및 테스트를 수행하였으며, 추가적인 테스트 환경 정보는 「KVM 설치 가이드」 문서를 참고한다.

V. 종합

- KVM Stack 통합테스트 수행 결과 공개SW로 구성된 A, B Stack 상에서 각 기능 시나리오 수행 시 치명적 오류 또는 심각한 장애가 발생하지 않았으며, Stack을 구성하는 각 공개SW가 유기적으로 동작함을 확인하였음

- 추가적으로, 성능 테스트에 대한 결과를 추후 제공 예정이며, 이러한 테스트 결과를 토대로 다양한 분야에서 KVM이 활용될 수 있을 것으로 판단됨

※ 참고 자료

- [1] <http://www.linux-kvm.org/>
- [2] <http://www.centos.org/>
- [3] <http://www.ubuntu.com/>
- [4] <http://windows.microsoft.com/ko-kr/windows7/products/home>
- [5] <http://www.apache.or.kr/>
- [6] <http://en.wikipedia.org/wiki/>
- [7] 가상화 기술의 새로운 패러다임 - 한국경제신문