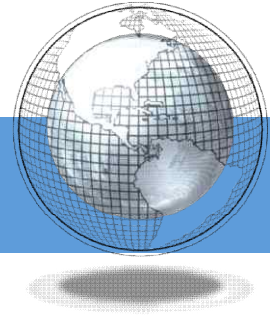


리눅스 가상화 기술의 현재와 미래



수퍼유저코리아



목 차

1

가상화란 무엇인가?

2

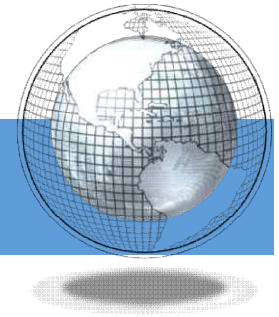
리눅스 가상화 유형과 중요성

3

리눅스 가상화 방식과 특징

4

리눅스 가상화의 미래와 숙제



1

가상화란 무엇인가?

2

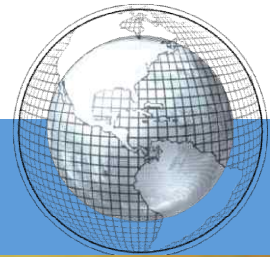
리눅스 가상화 유형과 중요성

3

리눅스 가상화 방식과 특징

4

리눅스 가상화의 미래와 숙제



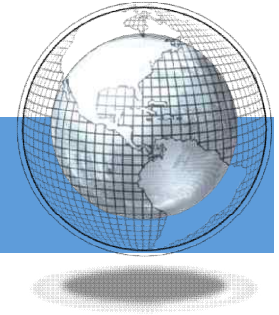
1.1 가상화?

네트워크 가상화

스토리지 가상화

파일 가상화

마이크로프로세서 가상화



1.1 가상화란 무엇인가?



실제 자원

가상화 작업

가상의 자원들을 만들어 실제 자원과 맵핑
소프트웨어 및 펌웨어 레벨에서 주로 구현

생성 자원



가상화된 자원들

가상화란?

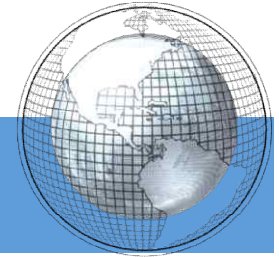
실제 존재하는 물리적인 다양한 자원들을 논리적인 자원들의 형태로 만들어 표시하는 기술

물리적 자원을 이용하는 사용자에게는 논리적인 형태로만 보임

물리적인 형태의 자원과 논리적인 형태 의 자원 중간의 중재자 역할

가상화 자원을 사용하는 사람들은 세부 자원에 대한 정보가 필요 없음

큰 자원을 작은 자원으로, 작은 자원을 큰 자원으로 가공하는 기술



1.2 가상화 역사

가상화의 오랜 전통을 가지고 있는 IBM

발
전
사

Pseries 마이크로 파티셔닝 지원

Iseries 마이크로파티셔닝 / **pSeries LPAR** 지원

Iseries LPAR 완성

Power LPAR 설계시작

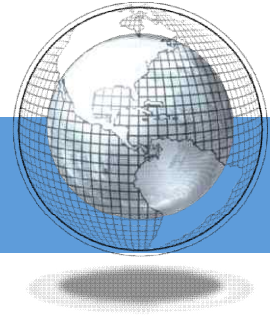
Physical Partition 지원하는 마지막 프레임(**ES/9000**)

메인프레임 **LPAR** 발표

Physical Partitioning (S/370 158 alc 168 모델)

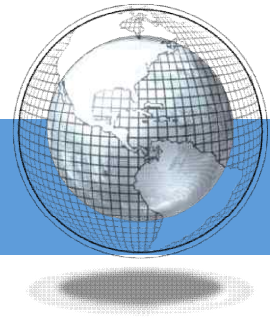
CP-67 및 하이퍼바이저 개발

1967년

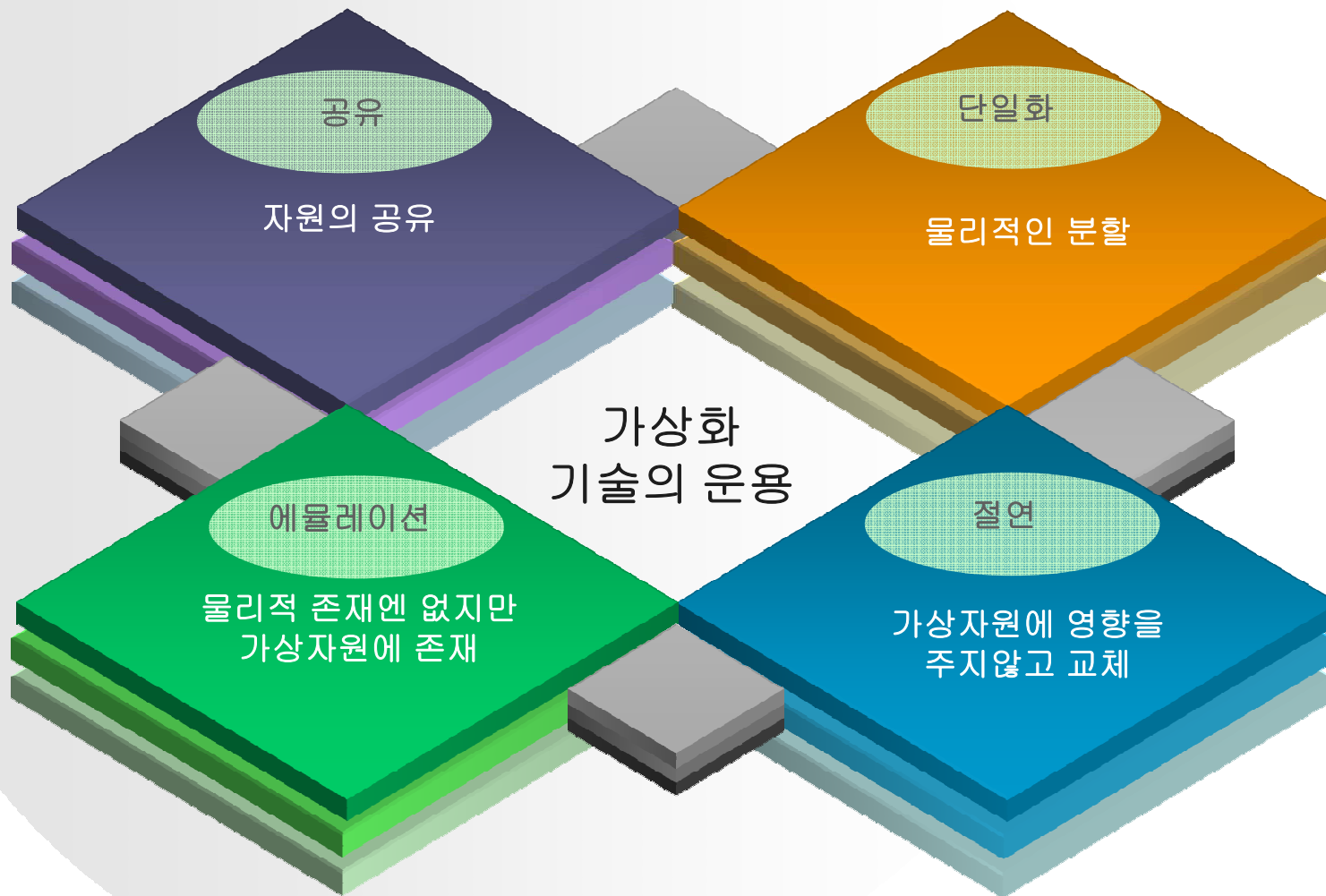


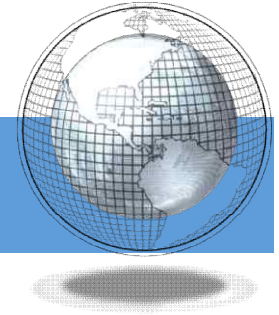
1.3 가상화 분류





1.4 가상화 기술의 기능





1.5 가상화 효과

높아진 자원의
활용률

물리적 자원들과 자원풀에 대한 동적인 공유 가능
더 높은 자원의 활용률을 얻을 수 있다. 가변적인 워크로드 상황에서 높은 효과

낮아진
관리비용

관리해야 하는 물리적 자원의 대수를 줄임으로써 관리 인력 생산성 향상
물리자원의 복잡성을 숨겨주고, 자동화, 정보화 및 중앙화를 통해 공통관리작업 단순화
워크로드 관리의 자동화를 가능 및 이기종 플랫폼 환경에서도 관리도구 공통 사용

사용의
유연성

사용자들의 환경에 맞는 자원들이 동적으로 재구성되어 활용

보안
향상성

분리와 격리를 가능하게 하여 데이터 서비스에 대한 통제 관리

높은
가용성

사용자에 영향없이 물리적 자원의 변경 및 제거, 분리, 업데이트

상호
운영

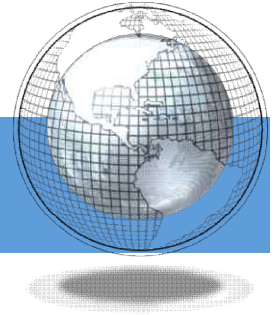
기존 물리적 자원들간 불가능한 인터페이스와 호환성 제공

프로비저
닝

자원 할당을 개별 물리적 단위보다 세밀한 조각단위로 가능

워크로드란?

트랜잭션 응답시간, 입출력 서비스 레이트, CPU 사용시간 등의 최대 또는 최솟값을 측정함으로써 프로세서가 처리하는 부하의 크기를 정량화한 것.



1

가상화란 무엇인가?

2

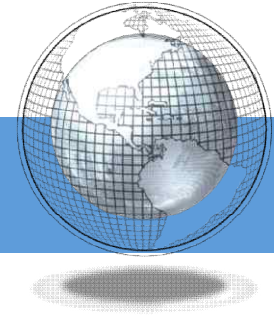
리눅스 가상화 유형과 중요성

3

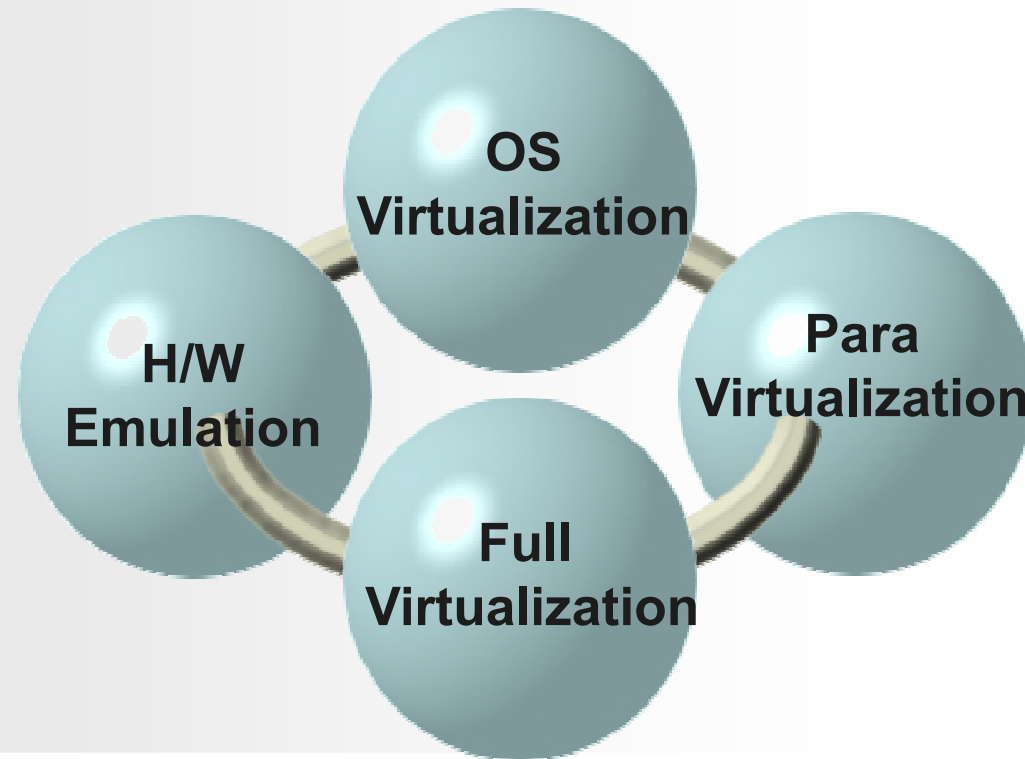
리눅스 가상화 방식과 특징

4

리눅스 가상화의 미래와 숙제

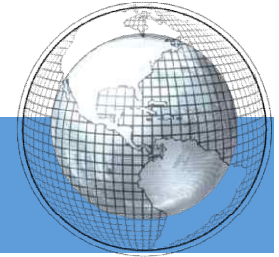


2.1 리눅스 가상화와 유형



가상화 유형은 한가지 방법만 있는 것이 아니라 다양한 레벨의 추상화를 통하여 같은 결과를 도출할 수 있다.

같은 가상화 방식을 설명하는 경우에도 다른 용어를 사용하는 경우도 있다



2.1 리눅스 가상화



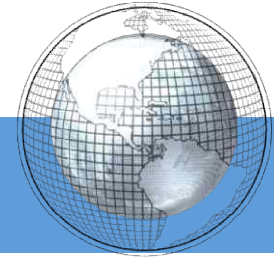
에뮬레이터

MAME

Multipale-Arcade Machine Emulaor

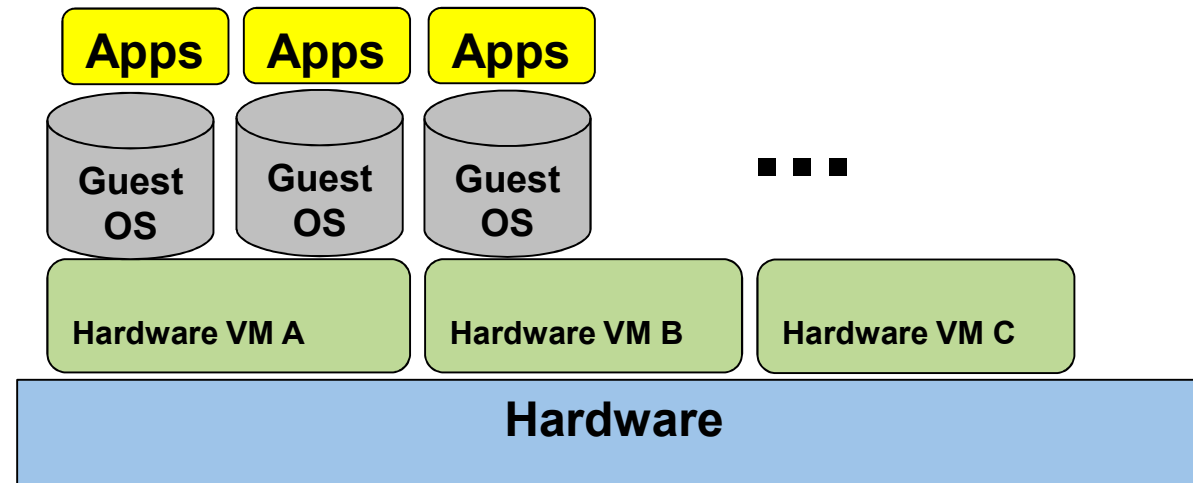
- 게임에 사용되는 프로세스 가상화
- 사운드 그래픽 등의 하드웨어 가상화
- 게임 컨트롤러 등의 전체 머신 가상화
- 펌웨어와 하드웨어의 공동개발

대상 하드웨어 **VM**을 사용하여 시뮬레이트
할때 나타난 실제 코드에 대하여
유효성 검사



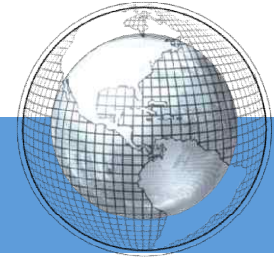
2.1 리눅스 가상화와 유형

**H/W
Emulation**



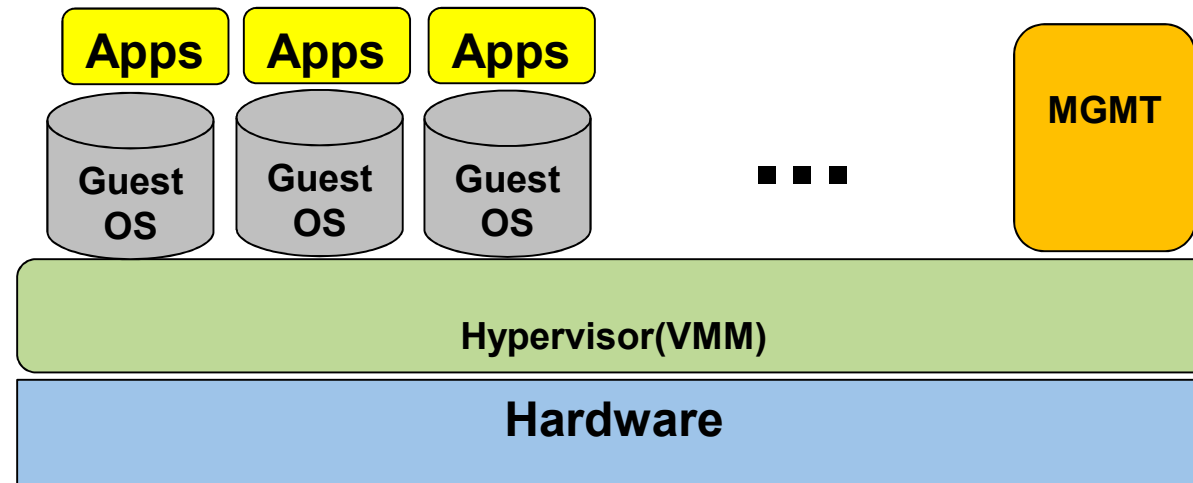
하드웨어 에뮬레이션의 특징

- 가장 복잡한 가상화 방식
- 모든 명령어들이 기반 하드웨어에 시뮬레이트 되어야 하므로 속도가 느림
- powerPC의 수정되지 않은 OS를 ARM 호스트에서 실행가능
- 다른 프로세서를 시뮬레이트 하는 다수의 가상 머신들 실행가능



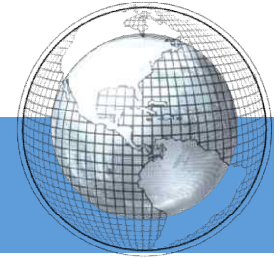
2.1 리눅스 가상화와 유형

Full
Virtualization



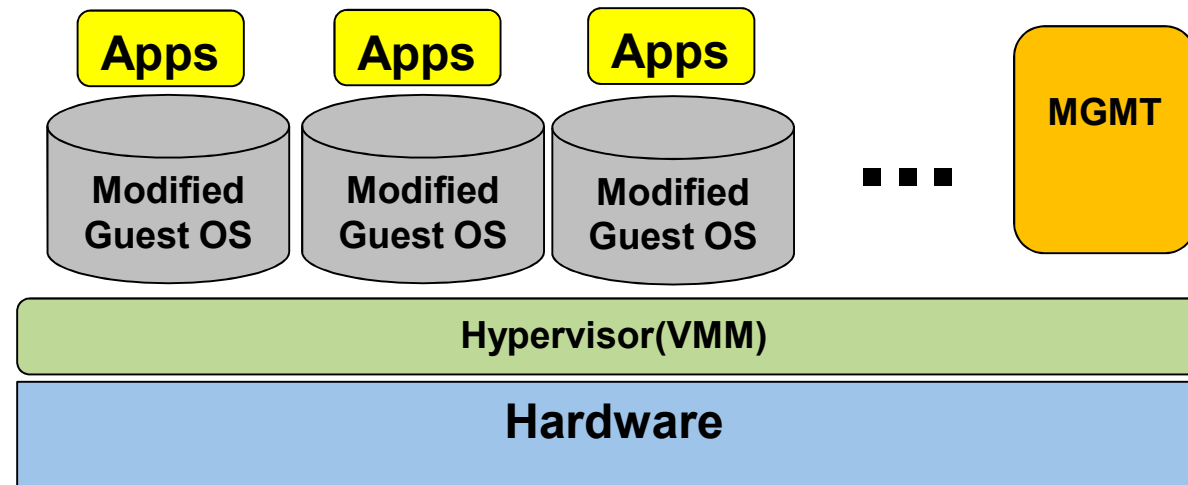
전체 가상화의 특징

- 보호를 받는 특정 명령어들은 VMM내에서 컨트롤 되어야 한다
- 하이퍼바이저를 통하여 기반 하드웨어를 공유
- 하드웨어 에뮬레이션보다는 빠르지만 실제 하드웨어보다는 성능이 낮음
- OS를 수정하지 않고 실행가능(단 OS가 기반 하드웨어를 지원해야 함)



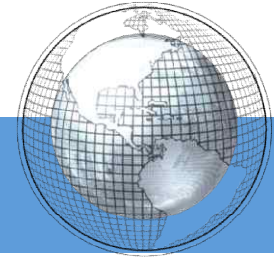
2.1 리눅스 가상화와 유형

Para
Virtualization



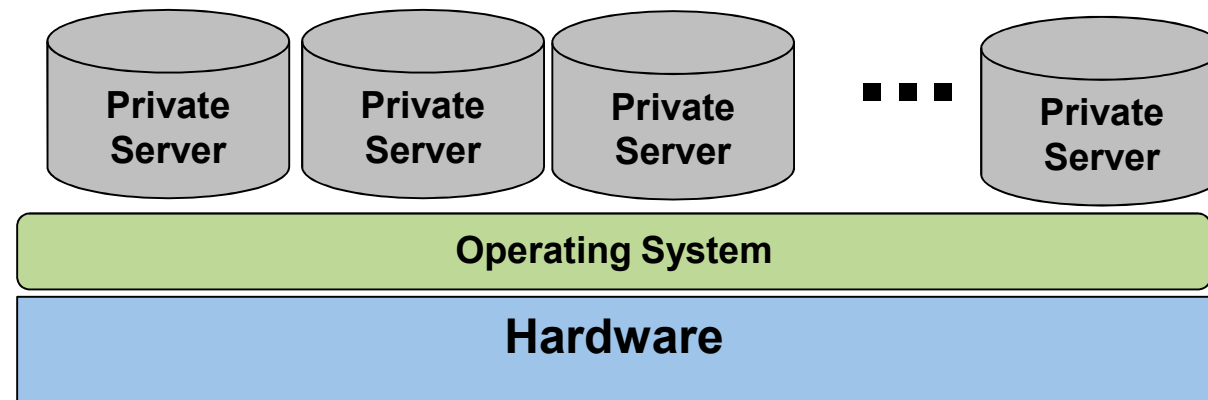
반가상화의 특징

- 가상화 인식을 OS에서 통합하여 사용
- 재컴파일 및 트래핑등의 작업이 필요가 없음
- 게스트 OS들이 하이퍼바이저에 맞게 수정하는 작업이 필요
- 가상화 되지 않은 상태의 시스템 성능에 가장 근접하며 전가상화와 같이 여러다른OS 동시지원



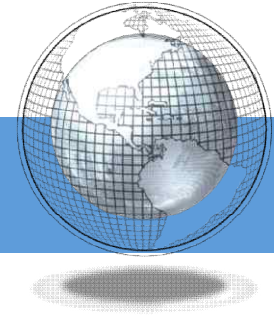
2.1 리눅스 가상화와 유형

OS Level
Virtualization



OS 레벨 가상화의 특징

- OS에서 서버들을 가상화
- 하나의 OS지원에 서버들을 분리
- OS 커널을 환경에 맞게 수정
- 성능이 우수



2.2 리눅스 가상화 중요성

비즈니스 관점

서버통합(사용도가 낮은 시스템들을 하나의 서버에 가상화)

장 점

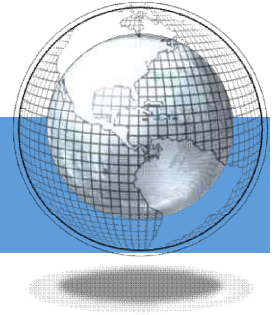
서버관리(전력, 공간, 냉각) 등에서의 유지비 절감효과
가용 하드웨어에 대해 로드를 조정할 수 있다

개발자 관점

다양한 오류, 버그로부터의 디버깅 환경 제공

장 점

커널 또는 드라이버의 오류로 인한 **OS** 전체에 대한 영향분산
커널 디버깅이 가능
오류나 버그로 인한 **OS** 문제 발생시 다른 **OS**들은 계속 실행가능



1

가상화란 무엇인가?

2

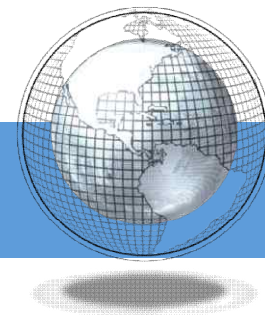
리눅스 가상화 유형과 중요성

3

리눅스 가상화 방식과 특징

4

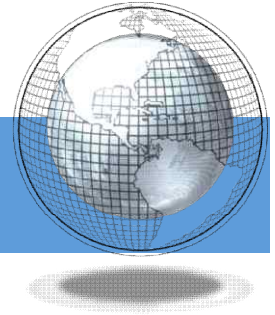
리눅스 가상화의 미래와 숙제



3.1 리눅스 가상화 방식과 특징

리눅스 가상화	
Bochs	Emulation
QEMU	Emulation
VMware	Full Virtualization
Z/VM	Full Virtualization
Xen	Para Virtualization
UML	Para Virtualization
Linux-VServer	Operating system-level Virtualization
OpenVZ	Operating system-level Virtualization

오픈 소스
솔루션
프로젝트



3.2 리눅스 가상화 방식과 특징

Bochs (에뮬레이션)

- X86 컴퓨터 시뮬레이터로 이식성 있음
- X86, power PC 등의 다양한 플랫폼에서 실행가능
- 키보드 마우스 등 주변기기들을 모두 포함하여 컴퓨터시뮬레이트

QEMU (에뮬레이션)

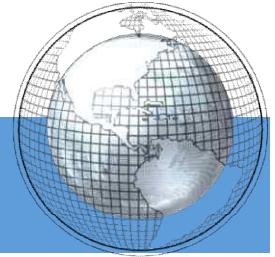
- 두 개의 연산 모드를 지원
(full System Emulation, User Mode Emulation)
- 아키텍처용 컴파일 된 바이너리가 x86 실행되는 리눅스에서 구동

VMware (전체 가상화)

- 가용 I/O 하드웨어를 가상화
- 가상화 된 환경은 파일로 저장되므로 새로운 호스트로 쉽고 빠르게 마이그레이션 가능

z/VM (전체 가상화)

- System z는 오래된 system /360 계열과 백워드 호환성
- System z용 os 하이퍼바이저로 코어에는 리눅스 포함하여 게스트 OS에 물리적 리소스들을 가상화



3.2 리눅스 가상화 방식과 특징

Xen (paravirtualization)

OS 레벨 paravirtualization용 무료 오픈 소스 솔루션

- 하이퍼바이저와 os가 가상화에서 협업하고 os는 변하지만 성능우수
 - 협업이 필요하고 패치된 os들만이 xen을 통해 가상화 가능
- 전체적인 지원 관점에서는 단점일 수 있음

유지모드리눅스 (paravirtualization)

OS가 사용자 공간에서 다른 리눅스 **OS**를 실행할 수 있도록 지원

- 리눅스 커널들이 하나의 리눅스 커널 정황 내에서 실행
- 게스트 커널들은 애플리케이션 공간에서 실행되기 때문에 컴파일필요 커널들이 중첩될수있고 또 다른 게스트 커널에서도 실행가능

Linux-Vserver (OS 레벨 가상화)

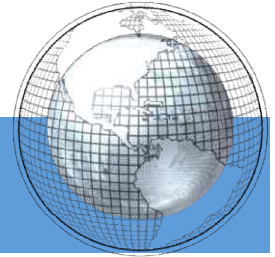
OS레벨 가상화 솔루션

- 여러 사용자 공간 환경들이 서로를 인식하지 못한 채 독립적으로 실행
 - 리눅스 커널로의 수정을 통해 사용자 공간 고립화
- 커널과 내부 데이터 수정이 필요

Open VZ (OS 레벨 가상화)

고립된 사용자공간과 관리용 사용자를 지원하는 가상화 인식 커널

- 명령어로 새로운 **VPS**를 쉽게 생성
- 하나의 물리적 서버에서 또 다른 서버로 체크포인트 및 마이그레이션 할 수 있는 기능



3.2 리눅스 가상화 방식과 특징

KVM
(Kernel Virtual
Machine)

- 리눅스 커널을 커널 모듈을 사용하는 하이퍼바이저로 전환
- 게스트 OS들이 호스트 리눅스 커널의 사용자 공간에서 실행가능
- 가상화 기술이 리눅스 커널의 일부가 된 계기
- 가상화 지원 H/W 실행시 리눅스와 윈도우의 게스트 지원가능

Kernel
Virtual
Machine

Apps Apps Apps

...

Apps

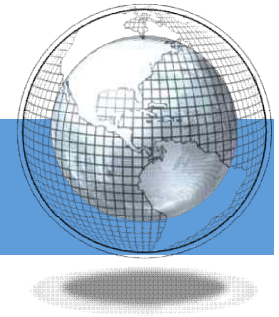
MGMT

QEMU

Linux

KVM Module

Hardware Virtualization Acceleration



1

가상화란 무엇인가?

2

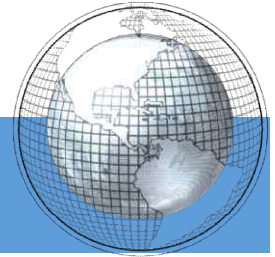
리눅스 가상화 유형과 중요성

3

리눅스 가상화 방식과 특징

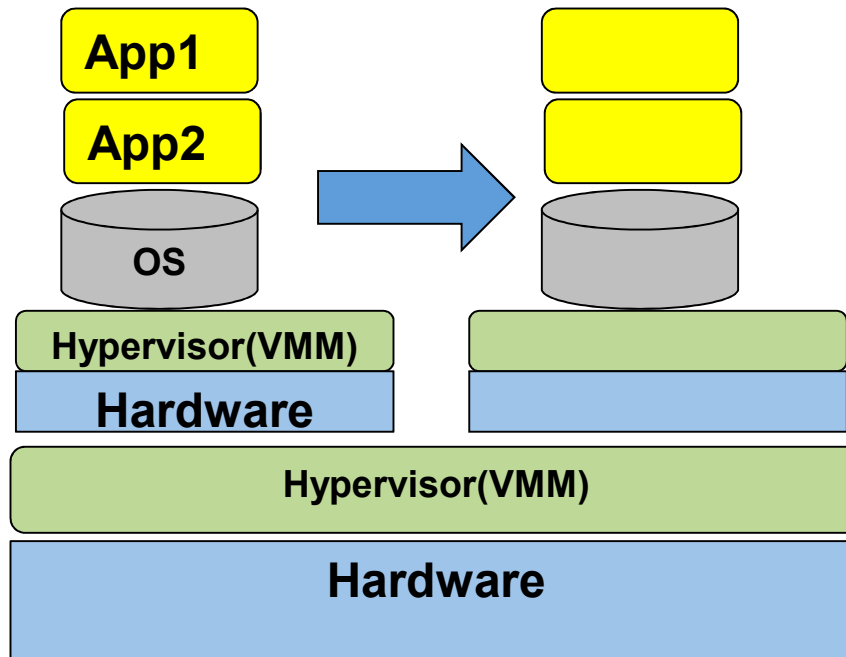
4

리눅스 가상화의 미래와 숙제



4.1 리눅스 가상화 미래

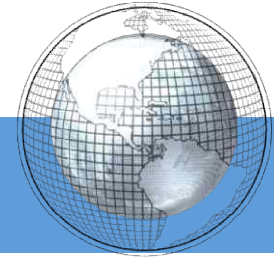
멀티시스템간의 가상화



중단없이 수행중인
파티션이나
어플리케이션을
다른 물리적인
서버로 옮기는 작업

시나리오

워크로드의 증가 - 증가하는 SMP 서버들의 워크로드가 더 강력한 다른 SMP 서버로 이전
서버수리/업데이트 - 정기점검 또는 업데이트를 위한 SMP 서버들이 셧다운
서버 콘솔리데이션 - 2개 또는 그 이상의 OS 이미지들이 하나의 SMP 서버안에 통합
워크로드 밸런싱 - 자원 활용률을 최적화하기 위해 워크로드 재분배 필요성 대두



4.2 리눅스 가상화 숙제

- | | |
|------------------------------|---|
| ● 정보보안 배제한 가상화 사업 | 가상화 사업 초기에 정보보안을 고려하지 않고 진행 |
| ● 가상화 계층 보안관리 취약혜손 | 가상화 계층의 혜손은 호스팅 된 모든 워크로드들의 혜손으로 이어질 가능성 대두 |
| ● 내부 가상네트워크 가시성, 제어부재 | 가상화 머신들간 통신 시 침입방지시스템과 같은 네트워크 기반 보안장치들이 VM 들간의 통신을 감지하지 못함 |
| ● 무분별한 물리적 서버통합 | 서로다른 신뢰 수준의 워크로드들이 충분한 구별없이 하나의 물리적 서버로 통합 |
| ● 하이퍼바이저/ VM 관리자 통제조치 | 하이퍼바이저/가상화 머신 모니터 계층 및 관리 도구에 관리지 액세스에 적용 되어야 할 통제조치가 부족 |
| ● 네트워크 보안제어 불분명한 역할 | 복수의 물리적 요소들이 하나로 통합되면 시스템 관리자나 사용자가 의도치 않게 보안범위 침해 및 허용되지 않는 데이터에 접근하게 될 위험증가 |
| ● 책임소재에 대한 명확한 구분 | 검증되지 않은 자원 및 기타 자료들에 대한 공유 및 업로드에 대한 대안마련 미흡 |



감사합니다!



수퍼유저코리아