

Virtualisering

Virtualisering

Virtualisering

Hvorfor virtualisering?

Isolasjon

Ressurs sparing

Fleksibilitet

Skalering av ressurser

Programvareutvikling

Skytjenester

Historie

Krav til virtualisering

Hardware støttet virtualisering

Type 1 hypervisor

Type 2 hypervisor

Binær oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live migration

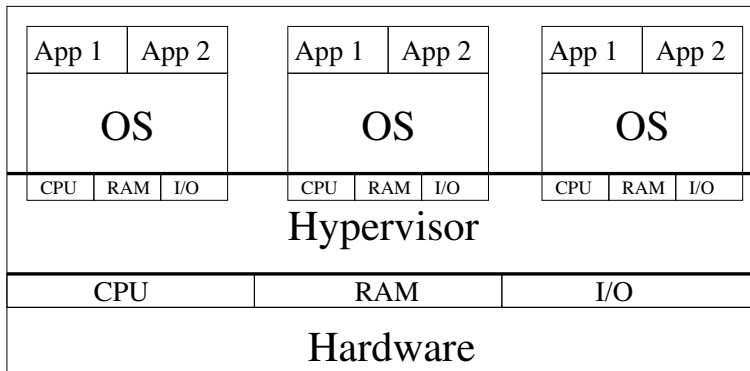
Simulering av migration

Migration av Ubuntu

Migration av IncludeOS

Nested virtualisering

- Virtualisering av server/desktop hardware
- En hypervisor simulerer hardware ved å gi samme grensesnitt som virkelig hardware gir
- Operativsystemene som kjører på en virtuell maskin, tror de kjører på ekte hardware



Hvorfor virtualisering?

Virtualisering

Virtualisering

Hvorfor virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av ressurser

Programvare-utvikling

Skytjenester

Historie

Krav til virtualisering

Hardware støttet virtualisering

Type 1 hypervisor

Type 2 hypervisor

Binær oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live migration

Simulering av migration

Migration av Ubuntu

Migration av IncludeOS

Nested virtualisering

- Isolasjon
- Ressurssparing
- Fleksibilitet
- Programvare-utvikling
- Skytjenester

Alle disse fordelene gjelder også for containere og Docker, bortsett fra den første, isolasjon og sikkerhet. Men fleksibiliteten blir enda større med containere.

Isolasjon

Virtualisering

Virtualisering

Hvorfor virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av ressurser

Programvareutvikling

Skyltjenester

Historie

Krav til virtualisering

Hardware støttet virtualisering

Type 1

hypervisor

Type 2

hypervisor

Binær

oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live

migration

Simulering av migration

Migration av Ubuntu

Migration av IncludeOS

Nested

virtualisering

- Tjenester og programmer kan kjøre på hver sin dedikerte server
- Unngår at de forskjellige tjenestene ødelegger for hverandre
- Men hva om den fysiske serveren eller hypervisor feiler?
- Det meste av nedetid og feil skyldes ikke hardware men software. Og software for en hypervisor er generelt mindre kompleks enn all programvaren på en hel maskin
- Sikkerhet: hvis en tjeneste blir hacket, vil det ikke påvirke de andre tjenestene
- Dette er fordi Operativsystemet og applikasjonene kun kommuniserer mot det virtuelle hardware-API'et som hypervisor gir dem tilgang til. De har ingen mulighet til å kommunisere med andre deler av en hypervisor eller andre VMer.

Ressurssparing

Virtualisering

Virtualisering

Hvorfor virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av ressurser

Programvareutvikling

Skytjenester

Historie

Krav til virtualisering

Hardware støttet virtualisering

Type 1

hypervisor

Type 2

hypervisor

Binær oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live

migration

Simulering av migration

Migration av

Ubuntu

Migration av

IncludeOS

Nested

virtualisering

- Man kan oppnå isolasjon ved å ha en fysisk server for hver tjeneste, men det gir store driftskostnader
- Med virtualisering kan det samme oppnås på en enkelt server
- Virtuelle maskiner (VMer) som for eksempel bruker lite CPU kan settes på samme fysiske server
- VMer kan enkelt flyttes til og fra fysiske servere og man kan dermed spare hardware og strøm
- Hage, Thomas: *The CERES project - A Cloud Energy Reduction System*, Veileder: Kyrre Begnum (Masteroppgave i Nettverk og Systemadministrasjon, HiOA/UiO)
- <http://www.cs.hioa.no/teaching/materials/MS100A/html/NSA.html>

Fleksibilitet

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

- Kapasiteten kan enkelt økes ved å legge til flere VMer, lastbalansering blir enklere
- Elastisitet: Man kan dynamisk tildele CPUer og internminne til VMer
- Har en VM blitt ødelagt eller kompromittert kan man enkelt starte opp en ny kopi
- Tradisjonelt er det arbeidskrevende å flytte en tjeneste eller et softwareprosjekt til en ny server på grunn av avhengighet av operativsystemet og annen programvare: når noe er utviklet på en VM så kan hele VMen flyttes eller kopieres
- Live migration: Hele VM flyttes til annen fysisk server uten nedetid på tjenestene
- Ung, Fredrik: *Towards efficient and cost-effective live migrations of virtual machines*, NSA masteroppgave
- Ahmad, Bilal: *Coordinating vertical and horizontal scaling for achieving differentiated QoS*, NSA masteroppgave, Veiledere: Anis Yazidi og Hårek Haugerud

Skalering av ressurser

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Flexibilitet

**Skalering av
ressurser**

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live

migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

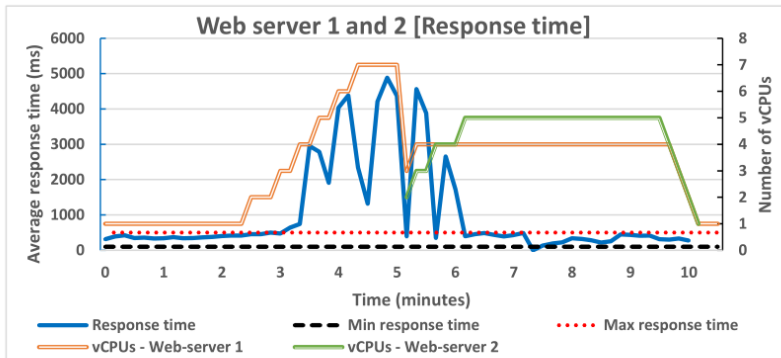


Figure 6.5: Web server 1 and 2 - response time in relation to vCPUs

Programvare-utvikling

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurs sparing

Fleksibilitet

Skalering av
ressurser

**Programvare-
utvikling**

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

- Man kan raskt teste ut programvare på forskjellige operativsystemer, Window, Linux, Mac, etc. ved å kjøre VMer med en rekke forskjellige OS
- Det er enklere å automatisere tester på flere plattformer (Test Driven Development)
- Ønsker man å teste ut nye ideer, kan man raskt sette opp miljøer for å teste dem ut

Skytjenester

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurs sparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

- Virtualisering er grunnlaget for fleksible skytjenester
- Kunder kan gis egne VMer med et antall CPUer, disk og minne
- Disse kundene kan dele fysiske servere, noe som gir store besparelser av hardware
- Nettbokhandelen Amazon startet med skytjenester fordi de bare hadde bruk for store mengder hardware til webserverene sine før jul og tenkte at de kunne leie ut hardware-ressursene resten av året

Historie

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressursparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

- IBM startet med virtualisering av stormaskiner på 1960-tallet
- En VMM (Virtual Machine Monitor) styrte flere virtuelle maskiner på samme fysiske maskin
- Første virtualiseringsløsning for x86: VMware i 1999
- Deretter fulgte Xen, VirtualBox, KVM og mange andre
- Hardware-støtte for x86 virtualisering kom først i 2005

Krav til virtualisering

Virtualisering

Virtualisering

Hvorfor virtualisering?

Isolasjon

Ressurssparing

Flexibilitet

Skalering av ressurser

Programvareutvikling

Skytjenester

Historie

Krav til virtualisering

Hardware støttet virtualisering

Type 1

hypervisor

Type 2

hypervisor

Binær oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live migration

Simulering av migration

Migration av Ubuntu

Migration av IncludeOS

Nested virtualisering

- Popek og Goldberg, 1974: En maskin kan bare virtualiseres hvis alle **sensitive instruksjonene** også er **priviligerte instruksjoner**
- En **Sensitiv instruksjon** kan bare utføres i kernel mode
- En **Priviligert instruksjon** forårsaker en trap til kernel mode hvis den gjøres i user mode

Eksempel:

- X86 instruksjonen POPF (skruer av og på interrupts) er en sensitiv instruksjon
- Hvis den utføres i user mode vil ingenting skje, som for NOP (No Operation)
- Instruksjonen CLI (CLear Interrupt flag) er en sensitiv instruksjon, men den er også privilegert. Hvis den utføres i user mode, gjøres en trap til kernel mode
- Vanlige instruksjoner som ADD, CMP og MOV er hverken sensitive eller privilegerte.

Hardware støttet virtualisering

Virtualisering

Virtualisering

Hvorfor virtualisering?

Isolasjon

Ressurs sparing

Fleksibilitet

Skalering av ressurser

Programvareutvikling

Skytjenester

Historie

Krav til virtualisering

Hardware støttet virtualisering

Type 1

hypervisor

Type 2

hypervisor

Binær oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live migration

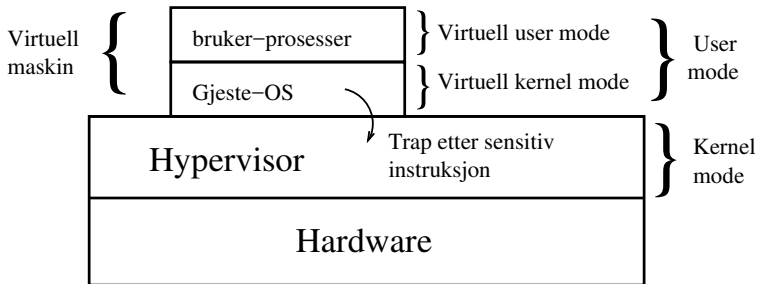
Simulering av migration

Migration av Ubuntu

Migration av IncludeOS

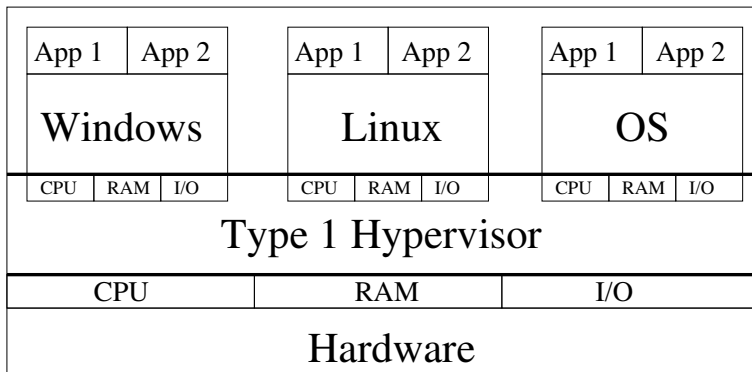
Nested virtualisering

- Hardware-støtte for x86 virtualisering kom først i 2005
- Inntil da fantes det sensitive instruksjoner (som POPF) som bare ble droppet
- Gjeste-OS kjører i user mode og vil ikke fungere som på vanlig hardware om en slik instruksjon droppes. Det vil føre til uforutsigbar oppførsel og kan føre til at OS crasher i verste fall.
- Med Intel VT-x og AMD-V vil alle sensitive instruksjoner trap'et til kernel mode når de utføres i user mode



Type 1 hypervisor

- En type 1 hypervisor kjører direkte på hardware som et OS
- Alle sensitive instruksjoner som utføres i user mode av gjeste-OS må trap'e til kernel mode og fanges opp av hypervisor
- Eksempler: VMware ESX og vSphere, Xen, Hyper-V(Microsoft)



Virtualisering

Virtualisering

Hvorfor virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av ressurser

Programvareutvikling

Skytjenester

Historie

Krav til virtualisering

Hardware støttet virtualisering

Type 1 hypervisor

Type 2 hypervisor

Binær oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live migration

Simulering av migration

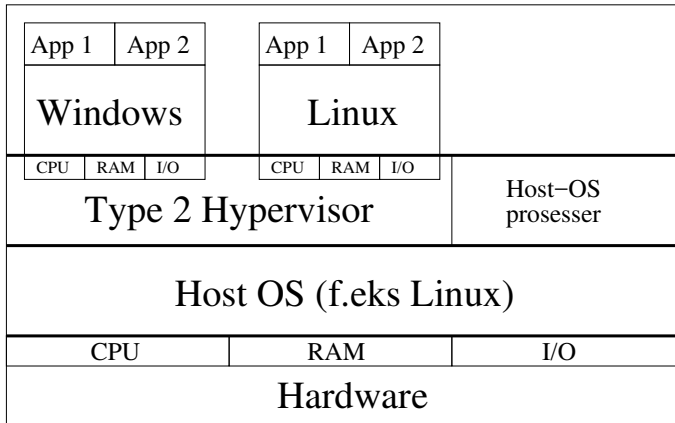
Migration av Ubuntu

Migration av IncludeOS

Nested virtualisering

Type 2 hypervisor

- En type 2 hypervisor kjører oppå et eksisterende OS
- Deler av hypervisor kan inngå i det underliggende OS'et i form av kjernemoduler
- Det kan være litt flytende grenser mellom type 1 og type 2
- KVM/Qemu(Linux), VirtualBox, VMware Fusion (Mac)



Virtualisering

Virtualisering

Hvorfor virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av ressurser

Programvareutvikling

Skytjenester

Historie

Krav til virtualisering

Hardware støttet virtualisering

Type 1 hypervisor

Type 2 hypervisor

Binær oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live migration

Simulering av migration

Migration av Ubuntu

Migration av IncludeOS

Nested virtualisering

Binær oversettelse

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

**Binær
oversettelse**

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

- Før 2005 måtte alternative metoder brukes, uten hardware-støtte
- VMware lagde en hypervisor som mens et program kjører scanner koden etter sensitive instruksjoner
- Dette gjøres for hver kodeblokk som ender i jump, call, trap eller lignende
- Sensitive instruksjoner oversettes til kall til VMware-prosedyrer i hypervisor
- De oversatte kodeblokkene caches og dette gjør kjøringen effektiv
- Hardware med VT-støtte genererer mange traps og dette tar lang tid
- Noe binær oversettelse finnes også i VirtualBox

Paravirtualisering

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

- Paravirtualisering krever at Gjest-OS endres
- Alle sensitive instruksjoner erstattes med kall til hypervisor
- Gjest-OS kan optimaliseres for virtualisering
- Ved å installere drivere laget for paravirtualisering, kan denne metoden bli meget effektiv

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

- Ytelesen reduseres generelt noe ved virtualisering
- Binær-oversettelse kan i noen tilfeller gi bedre ytelse enn bare metal (kjøre rett på hardware)
- Yaqub, Naveed: Comparison of Virtualization Performance: VMWare and KVM, NSA masteroppgave (2012)
- Mest nedlastede masteroppgave fra UiO (mer enn 15.000 nedlastninger)
- Sammenlignet: KVM vs VMWare ESXi 5.0 vs Bare Metal
- Red Hat Enterprise Linux 6.1, kernel 2.6.32, 8 CPUer, 72 GB RAM

Disk ytelse

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

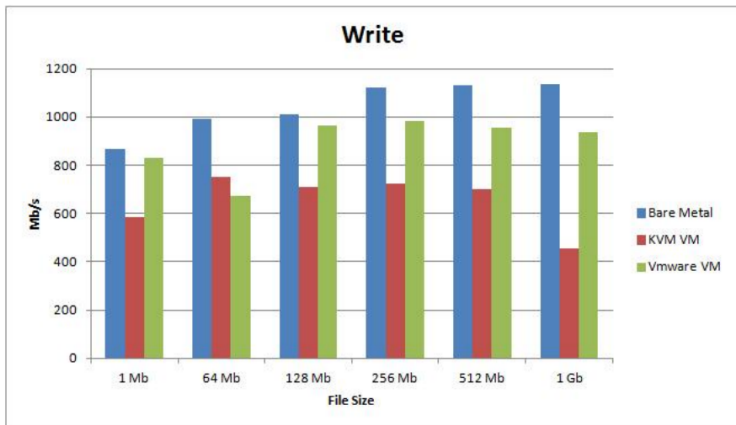


Figure 5.1: Iozone average write

RAM ytelse

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressursparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live

migrasjon

Simulering av
migrasjon

Migrasjon av
Ubuntu

Migrasjon av
IncludeOS

Nested
virtualisering

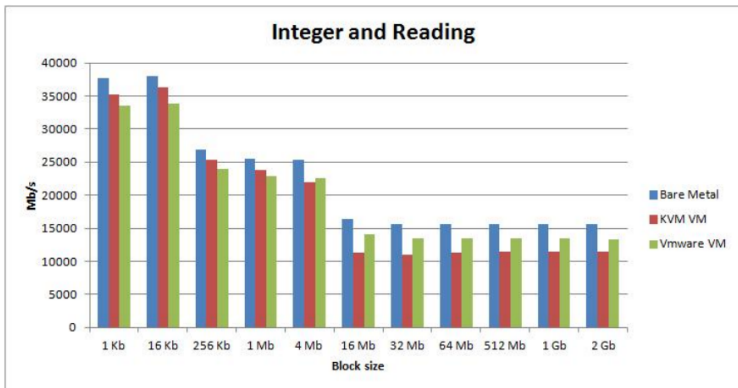


Figure 5.17: Ram speed average integer and reading

Cache: L1: 256 KB, L2: 1 MB, L3: 12 MB

CPU ytelse

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1

hypervisor

Type 2

hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live

migration

Simulering av

migration

Migration av

Ubuntu

Migration av

IncludeOS

Nested

virtualisering

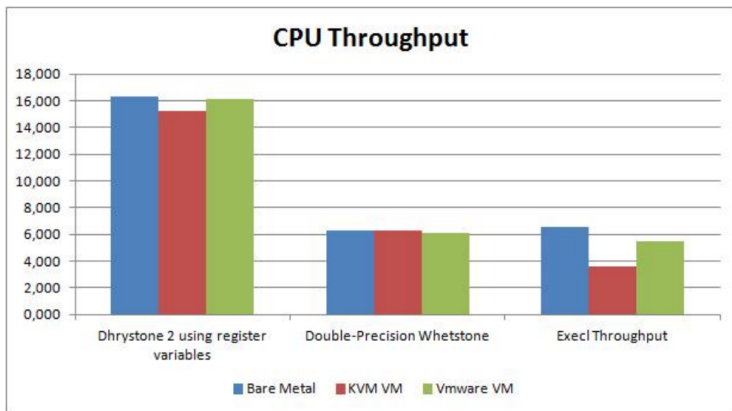


Figure 5.30: UnixBench CPU Throughput

Live migration

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressursparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

**Live
migration**

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

- Ved live migration flyttes en VM fra en fysisk server til en annen samtidig som den kjører kontinuerlig
- De statiske delene av minnet kopieres over mens VM'en kjører og tas imot på den andre serveren
- Når nesten alt er kopiert, stoppes VM'en, siste rest kopieres over og den startes opp igjen på den andre serveren
- Disken med VM-imaget og VM sitt filsystem er vanligvis et nettverksfilssystem, for eksempel NFS
- Wu, Hai: *Simulating Live Migration using Micro Virtual Machines* NSA masteroppgave (2015)

Simulering av migration

Virtualisering

Virtualisering

Hvorfor virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av ressurser

Programvareutvikling

Skytjenester

Historie

Krav til virtualisering

Hardware

støttet virtualisering

Type 1

hypervisor

Type 2

hypervisor

Binær

oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live

migration

Simulering av migration

Migration av

Ubuntu

Migration av

IncludeOS

Nested

virtualisering

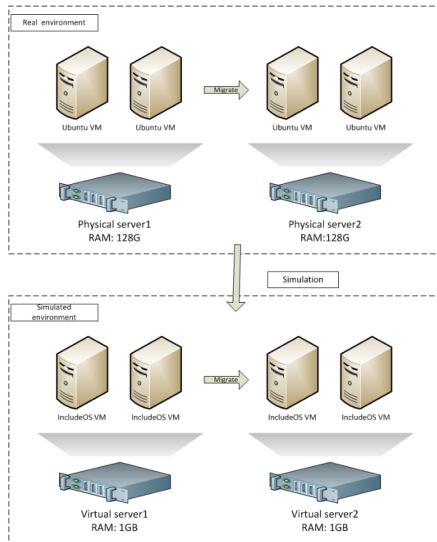


Figure 3.2: Modeling the data center

Migration av Ubuntu

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1

hypervisor

Type 2

hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live

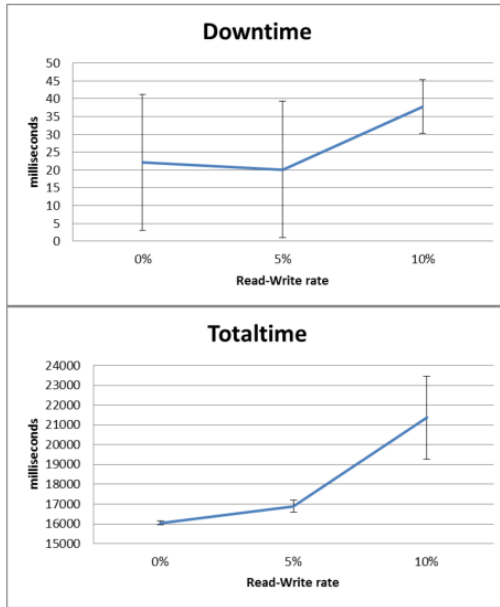
migration

Simulering av
migration

**Migration av
Ubuntu**

Migration av
IncludeOS

Nested
virtualisering



Migration av IncludeOS

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressursparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1

hypervisor

Type 2

hypervisor

Binær

oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

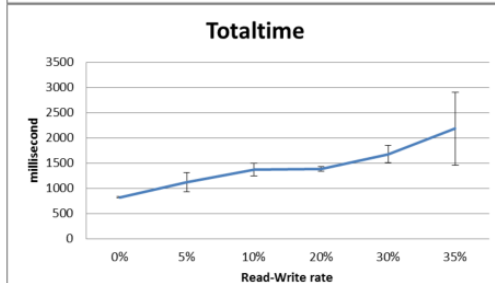
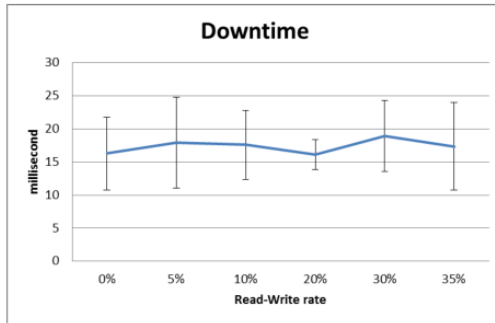
Live
migration

Simulering av
migration

Migration av
Ubuntu

**Migration av
IncludeOS**

Nested
virtualisering



Nested virtualisering

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

**Nested
virtualisering**

- Ved nested virtualisering kjører man en Hypervisor inne i en annen VM, slik at det blir tre lag
- Dette gir ytterligere ytelsesreduksjon
- Intel har hatt hardwarestøtte for nested virtualisering siden 2013
- Kan brukes til å flytte VMer mellom ulike Cloud-providere

Qemu og KVM

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

- QEMU (Quick EMUlator) er en open source programvare som kan emulere en komplett datamaskin
- QEMU kan emulere mange instruksjonssett, inkludert x86, MIPS, 32-bit ARMv7, ARMv8, PowerPC
- QEMU kan boote en rekke OS: Linux, Windows, Solaris, DOS, BSD
- KVM (Kernel Virtual Machine) er en Linux kjernemodul som gjør at user mode programmer kan unytte hardware virtualisering for en rekke forskjellige prosessorer, inkludert Intel og AMDs x86-prosessorer
- For å utnytte KVM må man kjøre QEMU-prosessen med opsjonen `-enable-kvm`

qemu på Linux server

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressursparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1

hypervisor

Type 2

hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live

migration

Simulering av
migration

Migration av

Ubuntu

Migration av
IncludeOS

Nested

virtualisering

- `# qemu-system-x86_64 -enable-kvm -name server2 -m 1024 -realtime mlock=off -smp 1,sockets=1,cores=1,threads=1 -hda /root/cslab/vmimages/ubuntu14.04.amd64.4G2.img -netdev tap,script=./qemu-ifup,id=hostnet0 -device rtl8139,netdev=hostnet0,id=net0 -nographic`
- `-enable-kvm` betyr at man skal bruke kjernemoduler som gjør virtualiseringen mye mer effektiv.
- En hel virtuell maskin med alt innhold kjøres som en enkelt prosess i user-mode.
- Image'et `ubuntu14.04.amd64.4G2.img` er på 2.2 Gbyte.

```
root@intel:~/vm# lsmod | grep kvm
kvm_intel          172032    6
kvm                540672    1 kvm_intel
```

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av
ressurserProgramvare-
utvikling

Skytjenester

Historie

Krav til
virtualiseringHardware
støttet
virtualiseringType 1
hypervisorType 2
hypervisorBinær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migrationSimulering av
migrationMigration av
UbuntuMigration av
IncludeOSNested
virtualisering

Vi skal nå se på fire C-program som skal kjøres virtuelt og bare metal, på en fysisk server.

```
#include <stdio.h>

#define uint64_t unsigned long int

int main(void) {
    uint64_t i, s = 0;
    for (i=0; i<50000000000; i++) {
        s = s + i;
    }
    return(0);
}
```

Kjører stort sett bare i user mode, bør ha omtrent samme ytelse virtuelt.

getppid.c

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurs sparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

```
#include<unistd.h>
```

```
int main(void) {  
    int i;  
    for (i=0; i<200000000; i++) {  
        getppid();  
    }  
    return(0);  
}
```

Dette programmet består av å gjøre systemkallet getppid om og om igjen.

gettimeofday.c

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/time.h>

int main(void)
{
    struct timeval c;
    int i;
    for(i=0; i<400000000; i++) {
        gettimeofday(&c,NULL);
    }
    return(0);
}
```

Mange systemkall som igjen bruker andre systemkall. Kan gi forskjell mellom bare metal og virtuelt.

forkwait.c

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressursparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

```
#include <unistd.h>
#include <sys/wait.h>

#include<unistd.h>

int main(void) {
    int i,pid;
    for (i=0; i<10000; i++) {
        pid = fork();
        if (pid < 0) return -1;
        if (pid == 0) return 0;
        waitpid(pid,NULL,0);
    }
    return(0);
}
```

Det siste eksempelet gir også mange systemkall og en god del minne-bruk.

Bare metal

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1

hypervisor

Type 2

hypervisor

Binær

oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live

migration

Simulering av

migration

Migration av

Ubuntu

Migration av

IncludeOS

Nested

virtualisering

Total tid programmene bruker i sekunder.

```
root@intel:~/virt# ./runiter.bash
sum getppid gettimeofday forkwait
1.47;0.89;0.70;2.49;
1.46;0.89;0.71;2.54;
1.46;0.85;0.70;2.48;
1.47;0.89;0.71;2.49;
1.46;0.85;0.71;2.45;
```

qemu med -enable-kvm

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Flexibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1

hypervisor

Type 2

hypervisor

Binær

oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live

migration

Simulering av

migration

Migration av

Ubuntu

Migration av

IncludeOS

Nested

virtualisering

```
root@server2:~/virt# ./runiter.bash
```

```
sum getppid gettimeofday forkwait
```

```
1.46;0.75;1.69;0.44;
```

```
1.39;0.75;1.69;0.43;
```

```
1.39;0.75;1.69;0.44;
```

```
1.47;0.75;1.68;0.51;
```

```
1.39;0.75;1.69;0.46;
```

- De to første programmene går som forventet omtrent like raskt
- Noe overraskende går det litt raskere på den virtuelle maskinen
- Kall til gettimeofday går vesentlig saktere på VM'en, ikke unaturlig, trap til hypervisor tar mye tid
- Overraskende at fork-kall går raskere på den virtuelle maskinen
- QEMU gjør binær-oversettelse av koden, kan spille en rolle her

gemu uten -enable-kvm

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Flexibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

```
root@server2:~/virt# ./runiter.bash
sum getppid gettimeofday forkwait
10.91;4.38;3.67;10.47;
11.06;4.28;3.42;10.51;
10.51;4.38;3.35;10.73;
10.80;4.29;3.50;10.24;
10.04;4.34;3.54;10.58;
```

Vi ser at det går vesentlig saktere med en ren qemu-emulering uten bruk av hardware-aksellerasjon. Programmet sum som kjører i user mode bruker omtrent åtte ganger så langt tid og forkwait bruker 20 ganger så lang tid.

Unikernel

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1
hypervisor

Type 2
hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

- En Unikernel er en minimalistisk operativsystemkjerne
- Inneholder OS og selve tjenesten eller applikasjonen som skal kjøres
- Typisk inneholder en unikernel kun akkurat det som er nødvendig for at tjenesten som skal tilbys kan kjøre
- Unngår overflødig kode
- Generelle operativsystem er laget for å kunne kjøre en hvilken som helts applikasjon og er store.
- En unikernel kan klare seg med en hundredel eller mindre i minnebruk sammenlignet med Ubuntu eller Windows

IncludeOS

Virtualisering

Virtualisering

Hvorfor
virtualisering?

Isolasjon

Ressurssparing

Fleksibilitet

Skalering av
ressurser

Programvare-
utvikling

Skytjenester

Historie

Krav til
virtualisering

Hardware
støttet
virtualisering

Type 1

hypervisor

Type 2

hypervisor

Binær
oversettelse

Paravirtualisering

Ytelse

Disk ytelse

RAM ytelse

CPU ytelse

Live
migration

Simulering av
migration

Migration av
Ubuntu

Migration av
IncludeOS

Nested
virtualisering

- IncludeOS er et operativsystem som ble laget helt fra bunnen av og skrevet i assembly og C++
- Laget med tanke på at det skal kjøres som en virtuell maskin
- Bare det helt nødvendige for at en tjeneste skal kunne kjører er tatt med i operativsystemet
- Alfred Bratterud er skaperen av IncludeOS og han har hele tiden ledet utviklingen siden starten ved HiOA i 2013
- IncludeOS er single threaded, all kode kjører i kernel mode og det er ikke mulig å logge inn som en bruker
- Må kompiles på nytt om man skal gjøre endringer
- Utgangspunktet er en tjeneste skrevet i C++ og man inkluderer i programmet bokstavelig talt operativsystemet med

```
#include <os>
```