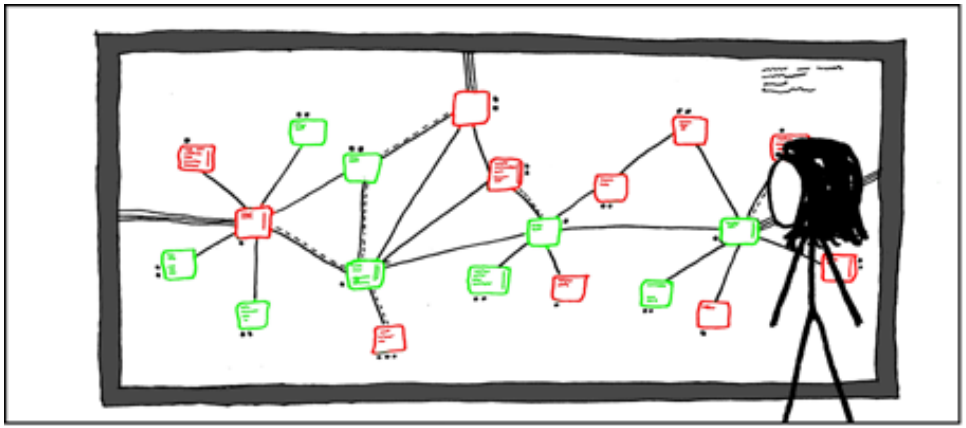
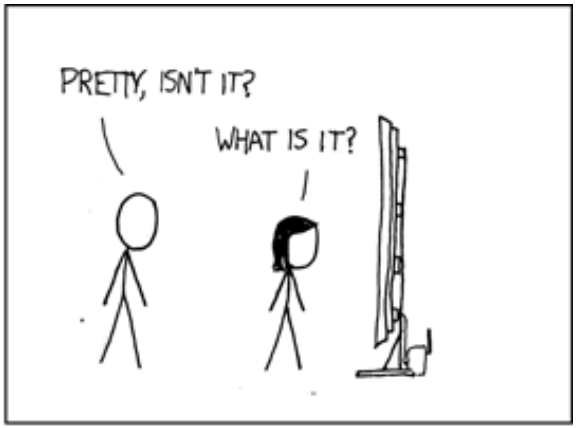
# Virtualizace na Linuxu

Radim Roška

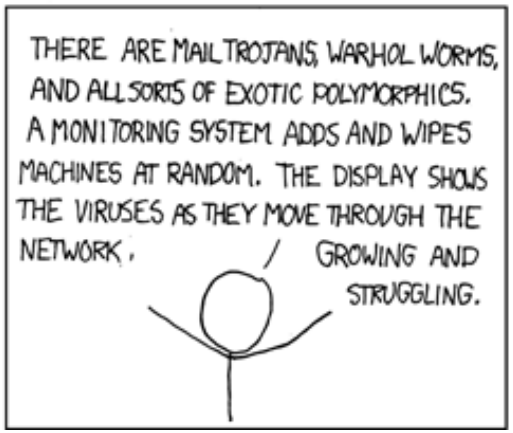
Středisko Un*xových technologií
Silicon Hill

13.4.2010











zdroj:xkcd.com

# Outline

1. Úvod
   - Vysvětlení pojmů
   - Základní typy virtualizace
2. Konkrétní řešení
   - Ostatní
   - KVM
   - OpenVZ
3. Hrajeme si s KVM a OpenVZ

# Co to je virtualizace

- obecně = abstrakce počítačových zdrojů
- konkrétně pro nás = technika, který na jednom fyzickém počítači umožní spustit řadu virtuálních počítačů

# K čemu to je?

- zefektivňuje využití hardware v datacentrech...
- učební prostředí
- sandbox

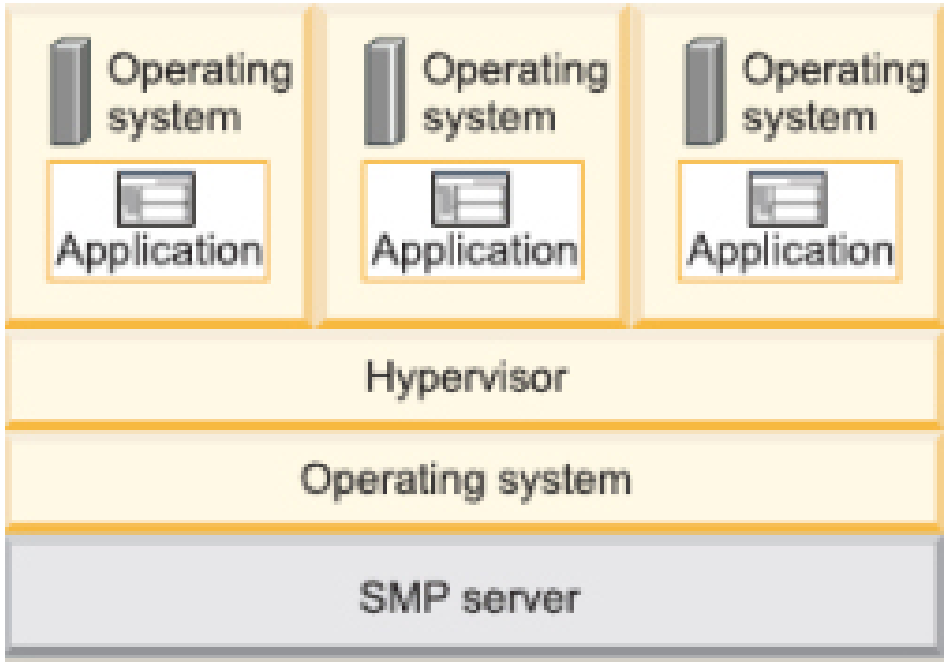
# Základní pojmy

- host OS
- guest OS
- hypervisor
- VM - virtual machine

# Hypervisor - virtual machine monitor

- umožňuje běh více OS na hostovi - poskytuje virtualní platformu
- monitoruje a řídí běh guest OS z hlediska hw prostředků
- izoluje jednotlivé guesty
- 2 typy:
  - typ I - native: hypervisor běží přímo na hw
  - typ II - hosted: hypervisor je sw spuštěný v běžném OS

# Hypervisor - typ II



# Požadavky virtualizovaného prostředí

Popek a Goldberg:

- Software nepozná, že běží na VM
- Operace(chyby,...) v jednom VM nemohou ovlivnit jiný VM ani hosta.
- Významná část instrukcí by měla proběhnout bez zásahu VMM (efektivita)

Theorem : For any conventional third-generation computer, a VMM may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions.

# Outline

1. Úvod
   - Vysvětlení pojmů
   - Základní typy virtualizace
2. Konkrétní řešení
   - Ostatní
   - KVM
   - OpenVZ
3. Hrajeme si s KVM a OpenVZ

# Full virtualization(plná virtualizace)

- virtualizují se všechny hw součásti PC
- guest není modifikovaný
- přístup k fyzickým zdrojům zprostředkován hypervisore

To přináší vykonostní postih.

# Paravirtualization

- částečná abstrakce, co jde to se nechá zpracovat přímo na hw - bez emulace za využití specialního API
- guest musí být upraven
- jednodušší hypervisor

Výkonější než plná virtualizace.

# Hardware-assisted virtualization

- plná virtualizace za podpory procesoru (Intel VT,AMD-V) => bez emulace
- guest nemusí být modifikován
- lepší výkon
- současný trend: KVM, Virtualbox, VMWare Workstation, Xen,...

```
egrep '(vmx|svm)' /proc/cpuinfo
```

# Operating system-level virtualization

- kernel(opatchovaný) umí pracovat s více izolovanýma user-space instancema operačního systému => linux only
- guesti a host sdilí tentýž kernel
- kernel poskytuje také nástroje na řízení prostředků(paměť, hdd,...)
- hodně pokročilý chroot
- container, jail, zone...
- bezpečnost, dynamický load balancing mezi nody v clusteru

# Outline

1. Úvod
   - Vysvětlení pojmů
   - Základní typy virtualizace
2. Konkrétní řešení
   - Ostatní
   - KVM
   - OpenVZ
3. Hrajeme si s KVM a OpenVZ

# Ostatní

- VirtualBox
- Xen
- VMWare

# Universální ovládání

- libvirt (by Red Hat) - knihovna, kterou lze ovládat téměř všechny hypervisory (Xen, KVM, OpenVZ,VirtualBox, VMWare...)
- = naprostá volnost v automatizaci některých úkonů, úspora spotřeby etc...

# Outline

1. Úvod
   - Vysvětlení pojmů
   - Základní typy virtualizace
2. Konkrétní řešení
   - Ostatní
   - KVM
   - OpenVZ
3. Hrajeme si s KVM a OpenVZ

# Kernel-based virtualization

- ..aneb tučňák hypervisorem
- VM je běžný proces
- process management, memory management řeší Linux => hypervisor může být jednodušší
- I/O operace (emulace HW) řeší qemu
- přidán guest mód

# KVM way

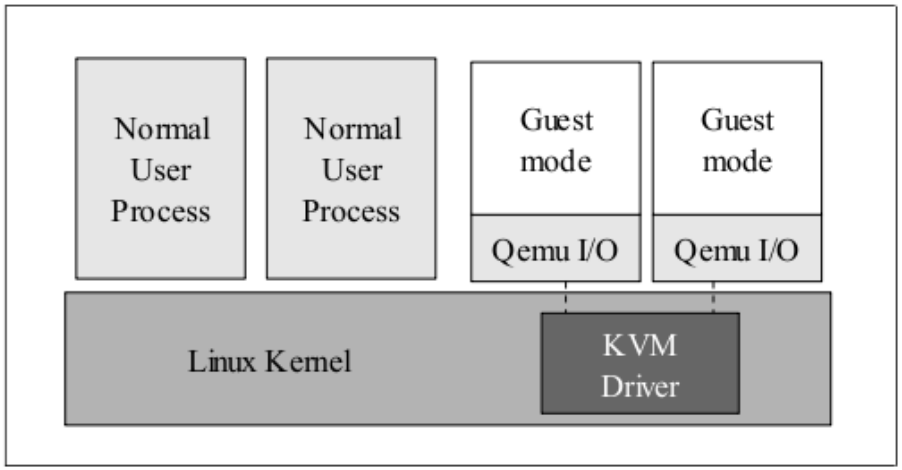
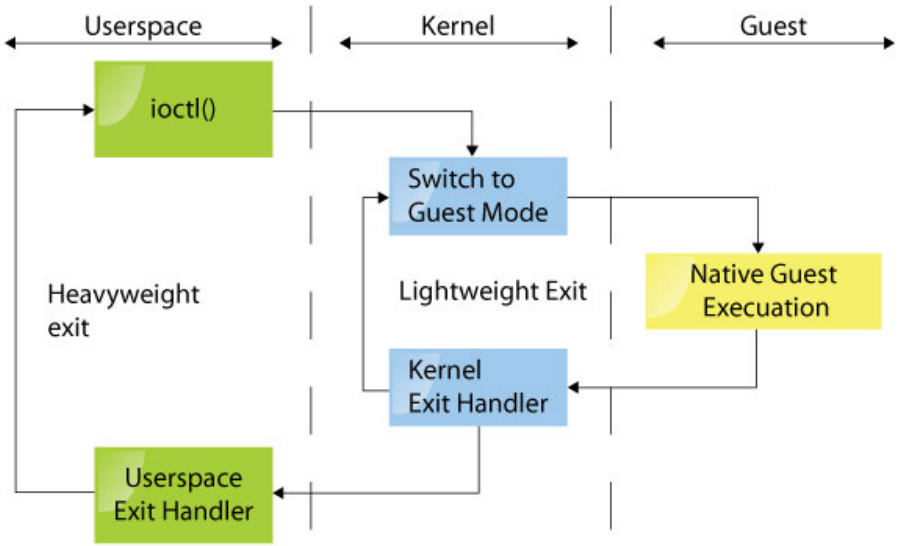



*Figure 2 – kvm Based Architecture*

# Guest mode

- Guest mode: spouštění non-I/O guest kódu
- Kernel mode: přepínání do guest modu a zpracování výstupů z něj kvůl i I/O operacím
- User mode: provedení I/O operací pro guesta

# Guest mode, part 2



# Ovládání

- man kvm
- man kvm-img
- man virt-manager
- man virsh

# Outline

1 Úvod
- Vysvětlení pojmů
- Základní typy virtualizace

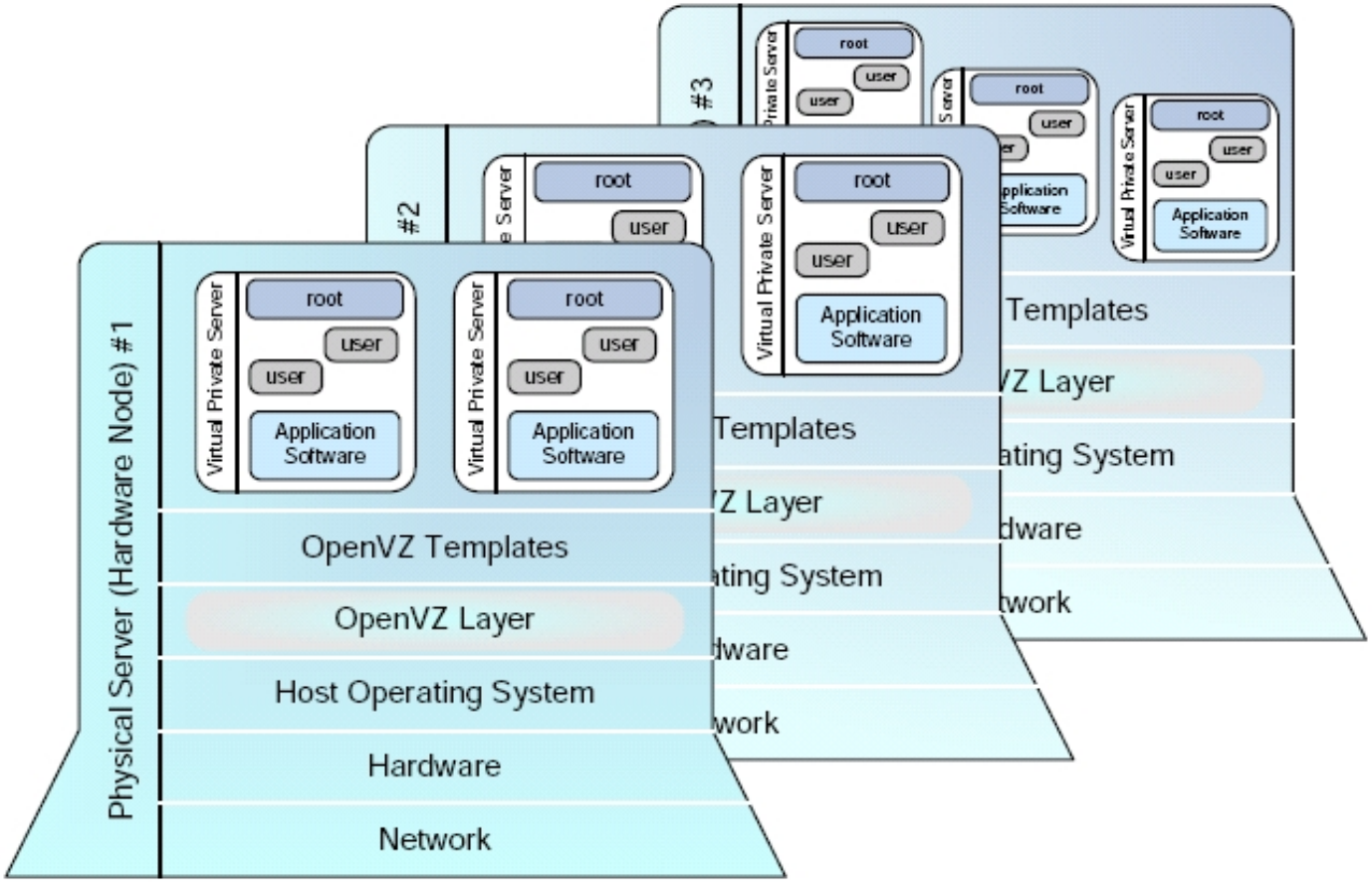
2 Konkrétní řešení
- Ostatní
- KVM
- OpenVZ

3 Hrajeme si s KVM a OpenVZ

# OpenVZ

- OpenVZ je kompletní řešení pro tvorbu tzv. VPS (Virtual Private Server)
- OS-level virtualizace
- Každý guest(VPS) se chová jako stand-alone server pro své uživatele

# OpenVZ


Physical Server (Hardware Node) #1
Virtual Private Server
root
user
user
Application Software
Virtual Private Server
root
user
user
Application Software
OpenVZ Templates
OpenVZ Layer
Host Operating System
Hardware
Network
#2
Virtual Private Server
root
user
user
Application Software
Templates
Z Layer
ating System
dware
work
#3
Virtual Private Server
root
user
user
Application Software
Templates
VZ Layer
ating System
dware
twork

# Vlastnosti

- Z pohledu apps je VPS nezávislý system <= virtualization layer v host kernelu
- virtualizace sítě - izolace síťových interfaců
- template - šablona pro tvorbu VPS
- resource management - kontroluje množství prostředků poskytnuté VPS

# 1. úkol

recept:

- 2x kvm virtual
- každý okořeníme s jedním openvz patchnutým kernelem
- a dochutíme nainstalovaným OpenVZ VPS serverem
- následně zmigrujeme OpenVZ kontejner z jednoho virtualu na druhý

# Otázky

?

# Zdroje

- wikipedia :)
- http://www.ok-labs.com/blog/entry/some-get-it-some-dont/
- http://www.linuxforu.com/how-to/kvm-virtualisation-the-linux-way/
- http://download.openvz.org/doc/OpenVZ-Users-Guide.pdf

# Appendix - poznámky z diskuze

Zrcadlení disku po síti:

- http://www.drbd.org/

Úplnou šedou kopii umí VMWare