

# Virtualizacija IT infrastrukture

## LINUX VIRTUALIZACIJA



**Algebra**

visoka škola za  
primijenjeno računarstvo

# *XEN i KVM*

## ➤ XEN hypervisor

- *Xen hipervizor kao platforma ima dva dijela - hipervizor koji se brine za sve osnovne funkcije (upravljanje CPU i memorijskim resursima, scheduling virtualnih mašina itd.) i jednu posebnu virtualnu mašinu koja se zove Domain0 (domo) koja ima direktan pristup hardveru, upravljačkim programima i kompletnom procesu upravljanja drugim virtualnim mašinama.*

## ➤ KVM hypervisor

- *Kernel-based Virtual Machine (KVM) je druga, novija generacija virtualizacijske tehnologije pod open-source operacijskim sustavima. Implementacija KVM-a je potpuno drugačija od Xen-a, pošto je KVM zapravo kernel modul koji pretvara Linux kernel u bare-metal*



**Algebra**

visoka škola za  
primijenjeno računarstvo

# KVM vs XEN

- Dva su vrlo bitna detalja zbog kojih je arhitektura KVM-a bolja od arhitekture Xen hipervizora
- 1. KVM je napravljen nakon što su Intel i AMD napravili procesore koji imaju hardverski podržanu virtualizaciju (*hardware assisted virtualization*, Intel VT-x, AMD-V). Stoga KVM nužno treba ovakve procesore da bi mogao raditi. Također, kako su u vrijeme kada je završen rad na KVM-u 64-bitni procesori već bili standard na PC platformi, KVM traži 64-bitni operacijski sustav kako bi radio što je zapravo i logično - nema nikakvog smisla koristiti virtualizaciju na 32-bitnom operacijskom sustavu zbog ograničenosti memorijskog adresiranja navedenih operacijskih sustava.
- 2. KVM ne pokušava "izmišljati toplu vodu" i koristi sve već postojeće metode koje posjeduje Linux kernel - upravljanje memorijom, procesima, ulazno/izlaznim operacijama, sigurnosnim postavkama i sl. Sve su to metode koje već postoje u Linux operacijskim sustavima i KVM ih koristi kao postojeće elemente.



**Algebra**

visoka škola za  
primijenjeno računarstvo

# Provjera

➤ `$ egrep '^flags.*(vmx|svm|lm)' /proc/cpuinfo`

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr  
pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht  
syscall nx mmxext fxsr\_opt rdtscp lm 3dnowext 3dnow  
rep\_good nopl pni cx16 lahf\_lm cmp\_legacy svm  
extapic cr8\_legacy 3dnow

- vmx znači da možemo koristiti Intel VT ekstenzije
- Svm znači da možemo koristiti AMD VT ekstenzije



**Algebra**

visoka škola za  
primijenjeno računarstvo

# *KVM arhitektura*

- U KVM-u, virtualna mašina je implementirana kao običan proces na Linux operacijskom sustavu, sa kojim upravljaju kernel i standardni Linux *scheduler*. Emulacija dodatnih uređaja se radi kroz QEMU sustav koji ima emulirani BIOS, PCI i USB sabirnice i standardne uređaje kao što su IDE/SCSI kontroleri, mrežne kartice i slično.



**Algebra**

visoka škola za  
primijenjeno računarstvo

# Mogućnosti

- **Security**

- Virtualna mašina je zapravo proces na Linuxu, pa koristi i Linux model za sigurnost. Konkretno, to znači da možemo koristiti SELinux (Security Enhanced Linux), dozvole, ACL-ove i sve ostale sustave kako bismo pristup virtualnoj mašini napravili sigurnim.

- **Memory Management**

- KVM virtualizacija koristi model upravljanja memorijom koji je integriran u Linux kernelu. Pošto se radi o običnom procesu, može se koristiti swap, 64-bitni pristup memoriji, rezervacija memoriji, zajednička memorija, i svi ostali mehanizmi koji već postoje u kernelu. Podržana je i NUMA tehnologija (da bi virtualne mašine efikasnije pristupale većoj količini memorije. KVM podržava i Intel EPT (*Extended Page Tables*) i AMD RVI (*Rapid Virtualization Indexing*) za veću efikasnost rada procesora. Podržana je i tehnologija za dijeljenje memorijskih lokacija KSM (*Kernel Same-page Merging*), koja može uštediti veliku količinu memorije na fizičkom serveru.

- **Sustav za pohranu**

- KVM može koristiti bilo koji sustav za pohranu koji je podržan u Linuxu - IDE, SCSI, SATA, NAS, NFS, SAMBA, SAN, iSCSI, FC, FCoE. Može se koristiti i GFS2 (*Global File System v2*), kako bi virtualne mašine mogle biti dijeljene između više hipervizora. Standardni disk format za KVM virtualne mašine je QCOW2 (*Qemu Copy On Write v2*), koji ima i podršku za enkripciju, snapshot i kompresiju.

- **Live migracija**

- *Live* migracija je podržana na KVM-u, što znači da virtualne mašine možemo prenositi sa jednog servera na drugi bez da ih ugasimo. Možemo napraviti i zaustavljanje virtualne mašine (*suspend*), kako bismo snimili trenutno stanje koje možemo kasnije nastaviti koristiti.



**Algebra**

visoka škola za  
primijenjeno računarstvo

# *Instalacija iz komandne linije*

```
➤ $ virt-install --connect qemu:///system \  
--name puppet --ram 512 \  
--file /nfs/vms/puppet.img \  
--network=bridge:bro \  
--accelerate -s 36 --pxe -d \  
--noautoconsole \  
--mac=54:52:00:53:20:00 \  
--nographics --nonsparse
```



**Algebra**

visoka škola za  
primijenjeno računarstvo

# *Management iz komandne linije*

- management korištenjem komande virsh

- boot VM

```
$ virsh start <guest>
```

- turn off VM

```
$ virsh destroy <guest>
```

- za pristup konzoli

```
$ virsh console <guest>
```

Za automatsko startanje virtualne mašine:

```
virsh autostart <guest>
```



**Algebra**

visoka škola za  
primijenjeno računarstvo



# *Logging KVM-a*

- provjeriti `/var/log/messages`
- `/var/log/libvirt/<guest>.log`
- povećati log level kroz `log_level` i `log_outputs` opcije u konfiguracijskoj datoteci `libvirtd.conf`



**Algebra**

visoka škola za  
primijenjeno računarstvo