Szerver virtualizációs technológiák

Beregszászi Alex, Szalai Ferenc

2006. november 21.

Tematika

1. rész Alapfogalmak és az elméleti háttér tisztázása
2. rész Emulációs technikák bemutatása
3. rész Virtualizációs technikák bemutatása
4. rész A közös területek mélyebb bemutatása (diszk formátumok)
5. rész XEN es QEmu használat közben

Miért?

- Miközben a hardver ára állandó vagy csökken a növekvő komplexitású informatikai infrastruktúrát egyre nehezebb és költségesebb üzemeltetni.
- Legfontosabb motivációs tényezők:
  - költségtakarékosság
  - üzemeltetés egyszerűsítése
  - flexibilis infrastruktúra
  - leállási idő csökkentése
  - hely és energia takarékosság
  - skálázhatóság növelése
  - megbízhatóság növelése

Mit?

- storage (adattároló alrendszerek): elrejteni a különféle gyártók SAN rendszereit, összevonni az elemi tároló kapacitásokat (pl.: IBM SVC)
- I/O virtualizáció: dinamikus sávszélesség és QoS allokáció fizikai csatornákban (pl.: Infiniband)
- szerver virtualizáció: erről fogunk részletesen beszélni

Hogyan?

Szerver virtualizációs technikák fő kategóriái:

- Emuláció: teljes utasítás készlet transzformáció
- Para-virtualizáció: fizikai hardver elérés a hipervizoron keresztül
- Operációs rendszer szintű virtualizáció:
- API virtualizáció
- Alkalmazás szintű virtualizáció

Egzotikumok:

- Vitual SMP rendszerek: más néven elosztott, közös memóriájú klaszterek (NUMA). Kis késleltetésű, gyors hálózat (pl. Infiniband, PCI-X) kell hozzá.
- PC architektúrán túl: pl.: Power 5 (mikrovirtualizáció)
- IBM S/360 hypervisor OS

Alapfogalmak és az elméleti háttér tisztázása

- **virtuális gép**: absztrakció, olyan szoftver ami fizikai eszközök virtualizálásával teremet alkalmazások számára környezetet.
- **host gép/operációs rendszer**: virtuális gépeket befogadó fizikai eszköz.
- **guest/vendég operációs rendszer**: virtuális gépben futó operációs rendszer
- **hipervisor**: szuper-privilegizált módban futó kernel amin a virtuális gépek futnak (paravirtualizáció)
- **JIT - just in time**: futás közben az adott processzorra "utasításcsomagok" készítése - jelentős sebességnövekedés céljából

Emuláció

Előnyök:

- változatos architektúrák (PC, Amiga, stb.)
- teljes hardver kontroll (BIOS, VGA stb.)

Hátrányok:

- sebesség
- kell hozzá hoszt rendszer
- overhead a hoszt rendszeren

Fő felhasználási terület:

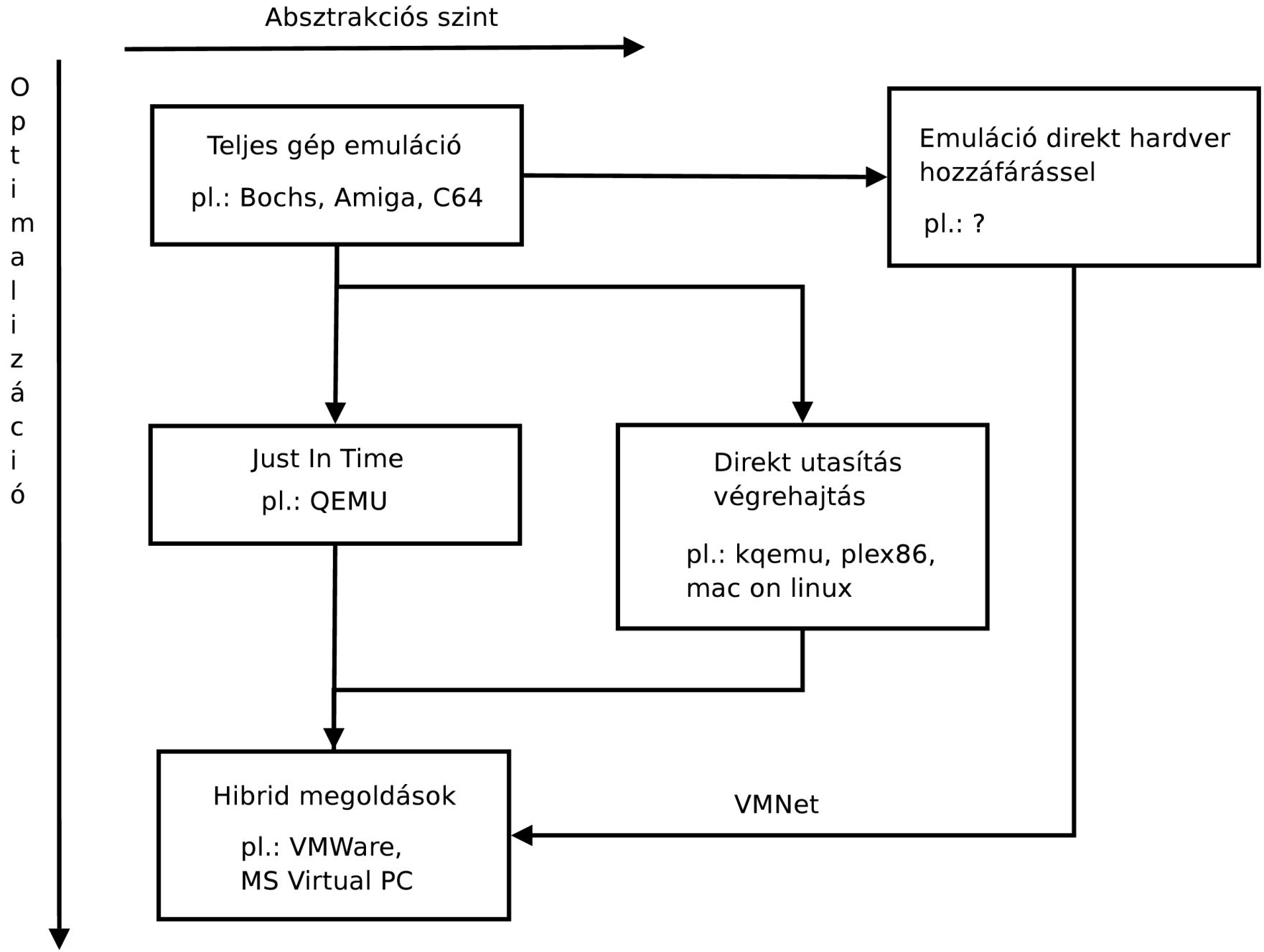
- fejlesztő környezet, pl.: új hardver fejlesztése
- zárt kódú operációs rendszerek virtualizációja

Emuláció - folytatás

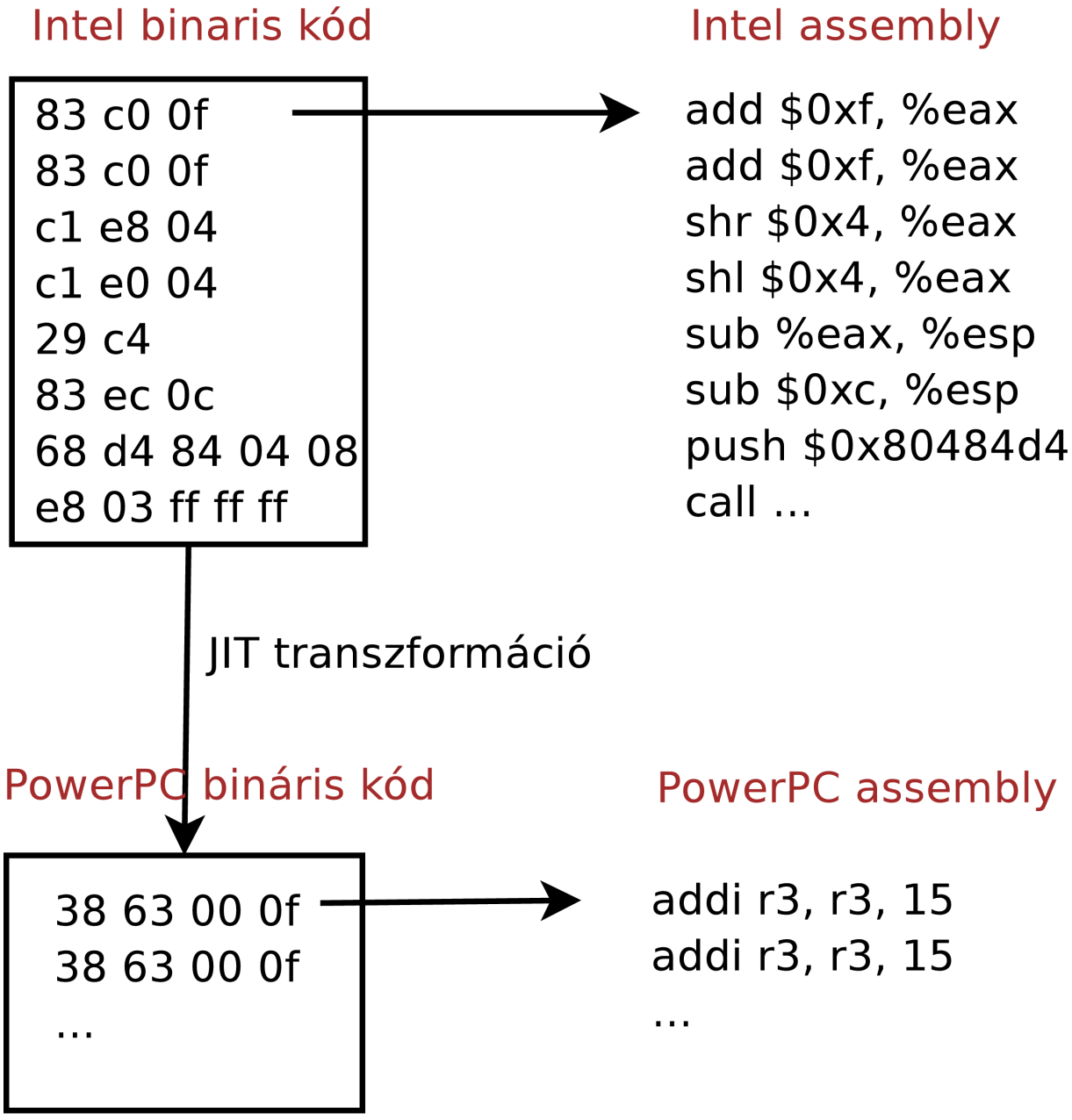
Legismertebb szoftverek:

- VMware Workstation, Player, Server
- MS Virtual PC és szerver
- QEmu (GPL, részben LGPL)
- Bochs (GPL), Plex86 (GPL)
- MacOnLinux
- Amiga, C64, PPC stb. emulátor
- Mobiltelefon fejlesztő környezet
- Mikrokontoller emulátorok

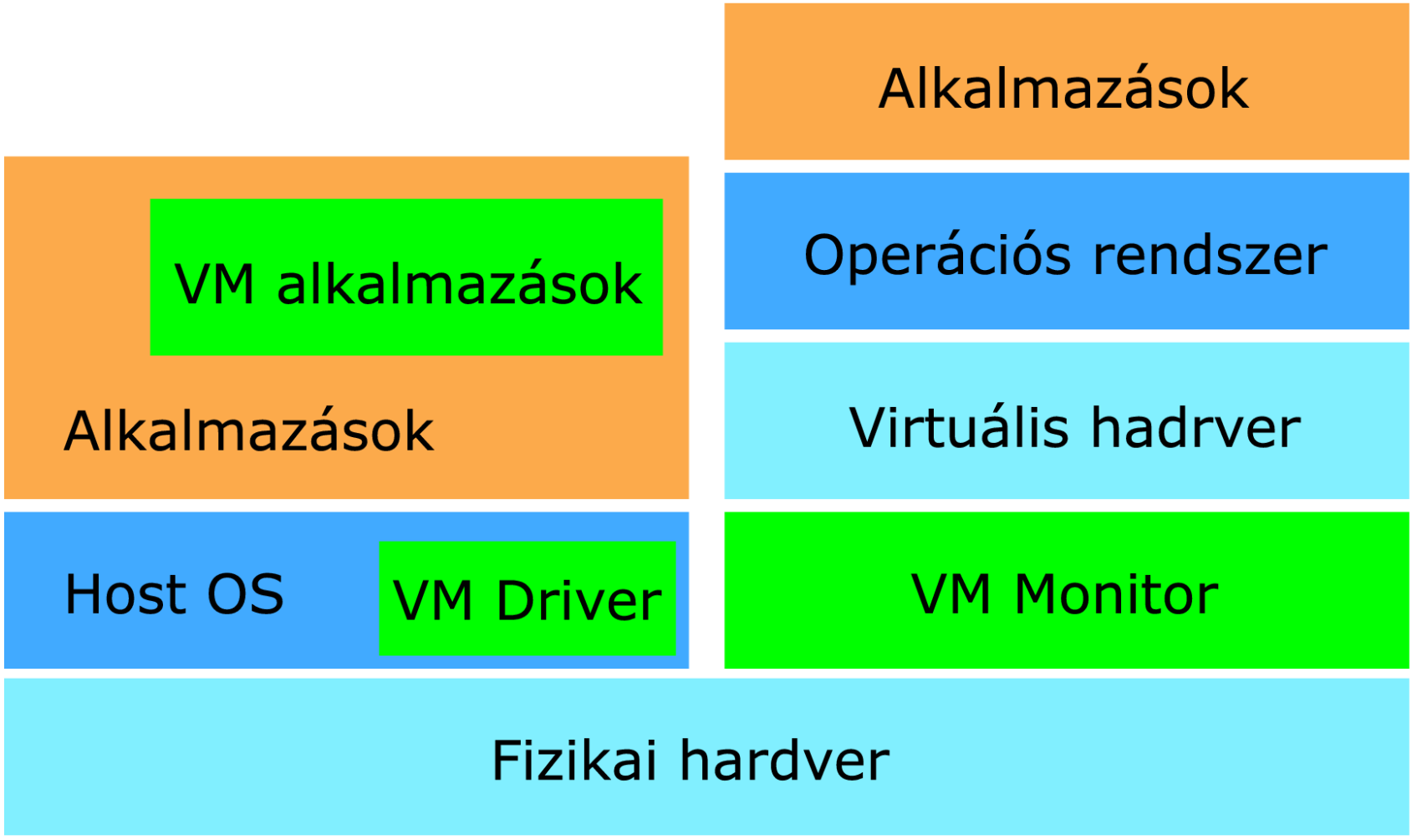
Emuláció


Absztrakciós szint
Optimalizáció
Teljes gép emuláció
pl.: Bochs, Amiga, C64
Emuláció direkt hardver hozzáfárással
pl.: ?
Just In Time
pl.: QEMU
Direkt utasítás végrehajtás
pl.: kqemu, plex86, mac on linux
Hibrid megoldások
pl.: VMWare, MS Virtual PC
VMNet

Utasítás olvasás és JIT



Emuláció - VMware desktop architektúra



Paravirtualizáció

Előnyök:

- jó teljesítmény (2-5 % veszteség)
- hardver támogatás

Hátrányok:

- portolni kell a guest OS-t csak hardver támogatással képes módosítatlan operációs rendszert futtatni
- fiatal technológia ezért management eszközök hiányosak

Tipikus felhasználás:

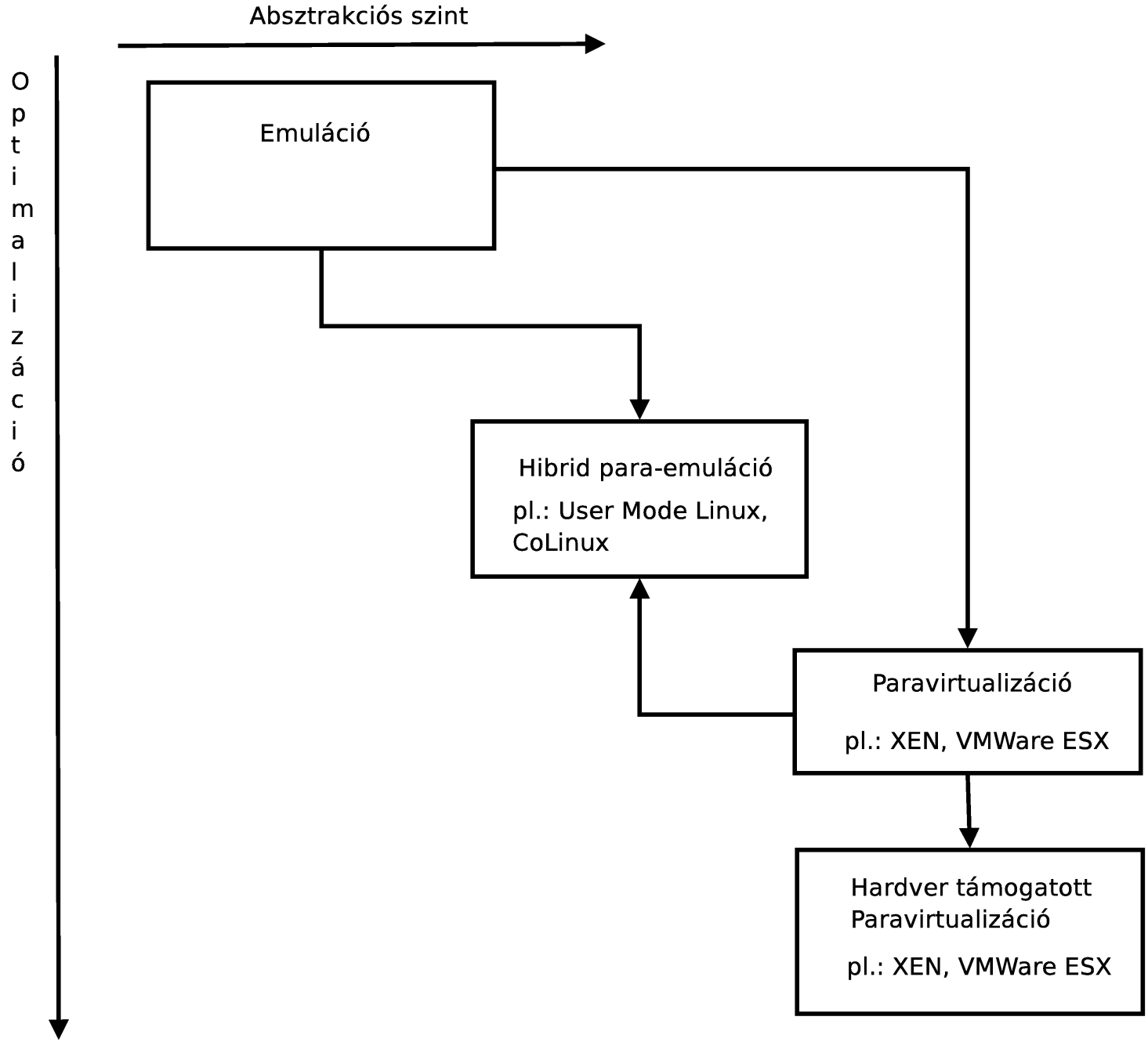
- virtualizált infrastruktúra
- hosting

Paravirtualizáció - folytatás

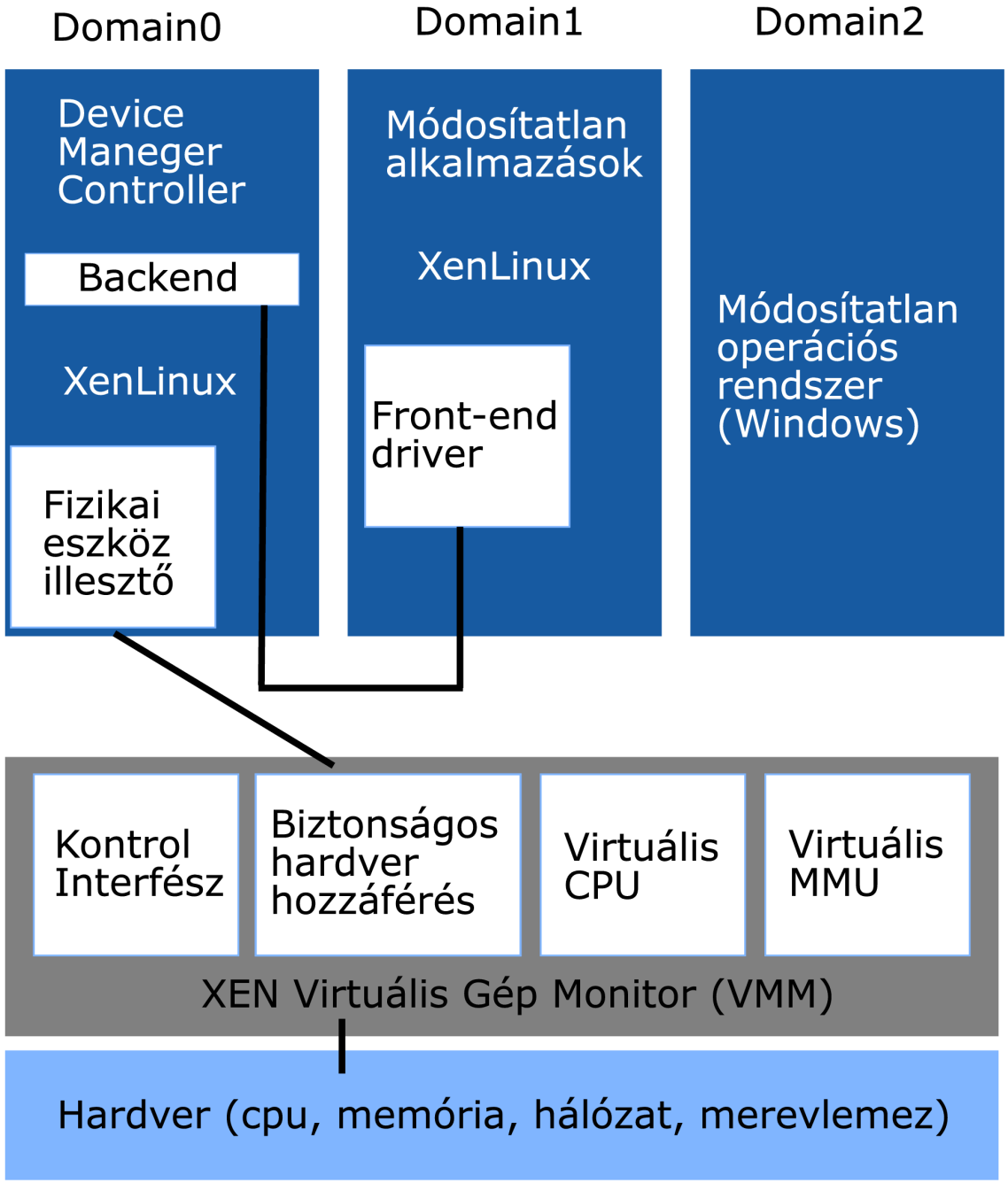
Legismertebb szoftverek:

- XEN
- User Mode Linux (UML) - emulációs és paravirtualizációs elemeket is tartalmaz
- CoLinux - Linux kernel futtatása Windos-on
- VMware ESX - nem tiszta de paravirtualizációs elemelket is tartalmaz
- Denali, Trango: egyetemi projektek később kereskedelmi termékek

Paravirtualizáció



Paravirtualizáció - XEN architektúra



Operációs rendszer szintű virtualizáció

Előnyök:

- teljesítmény (2-5% veszteség)
- pehelysúlyú
- jó virtuális gépenkénti erőforrás allokáció

Hátrányok:

- nem lehet több különböző operációs rendszer típus
- kernel módosítást igényel

Tipikus felhasználás:

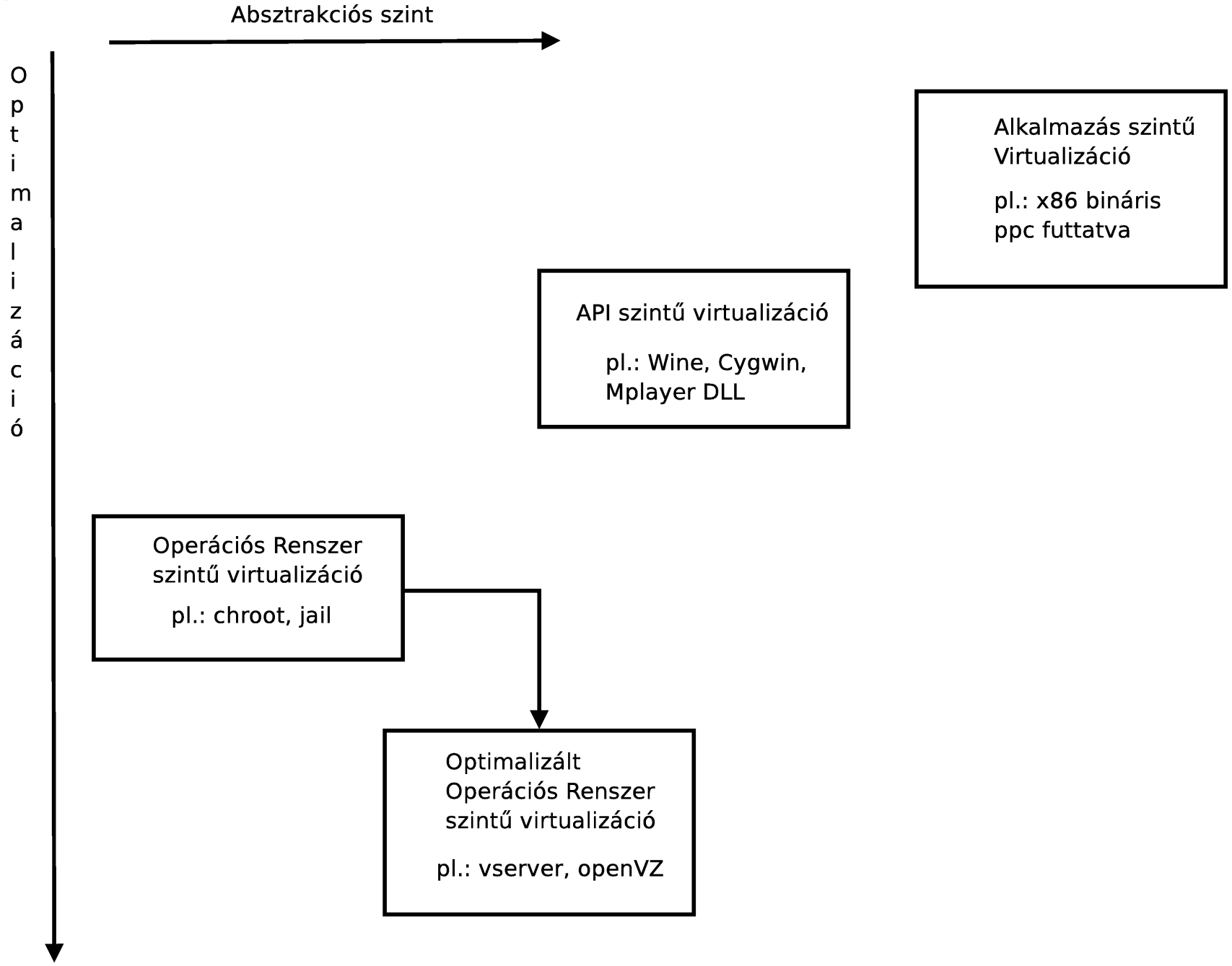
- hosting
- virtuális hálózat szimuláció

Operációs rendszer szintű virtualizáció - folytatás

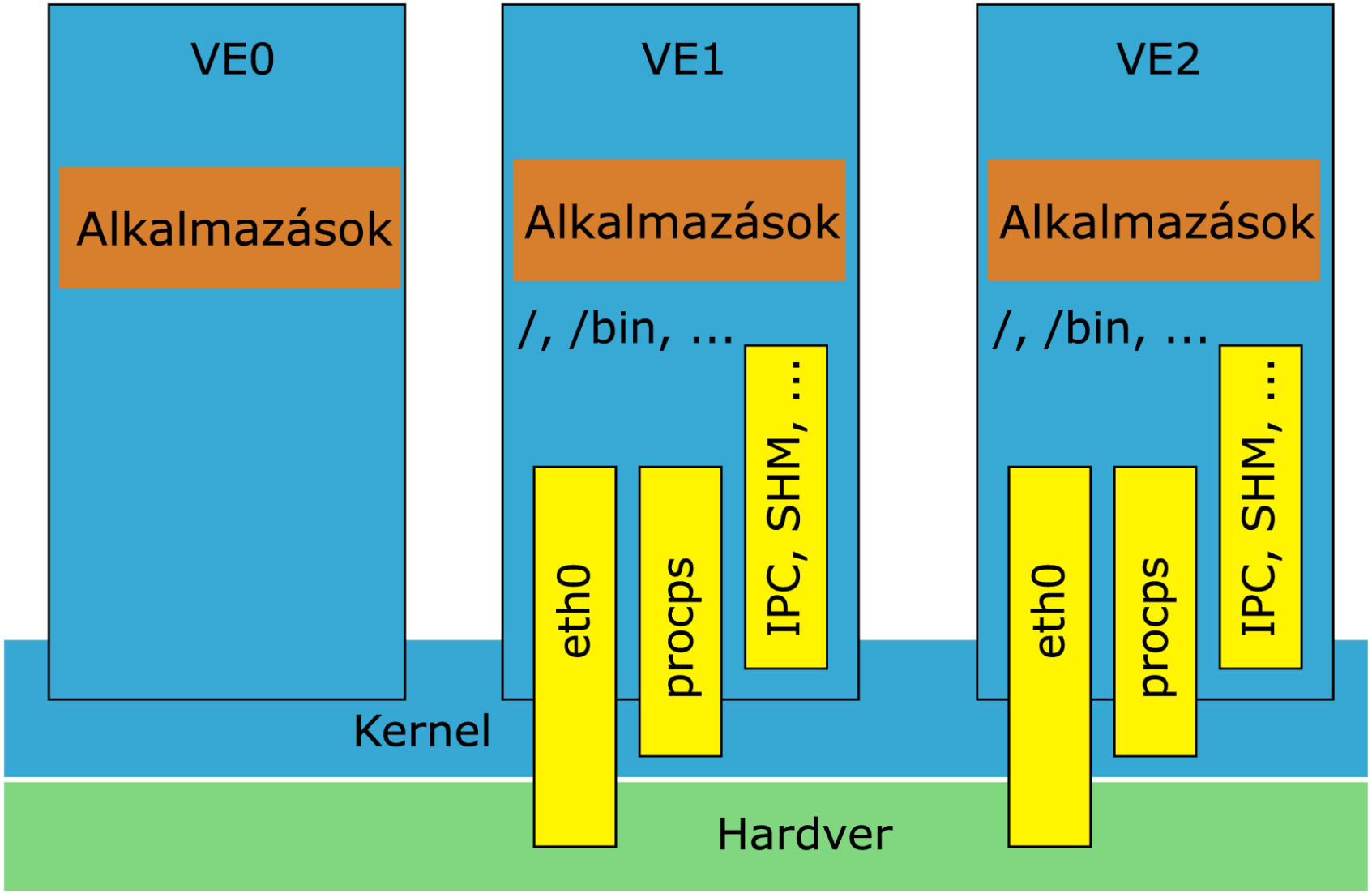
Legismertebb szoftverek:

- chroot, BSD jail
- Linux-VServer
- OpenVZ, Virtuozzo
- Solaris Container

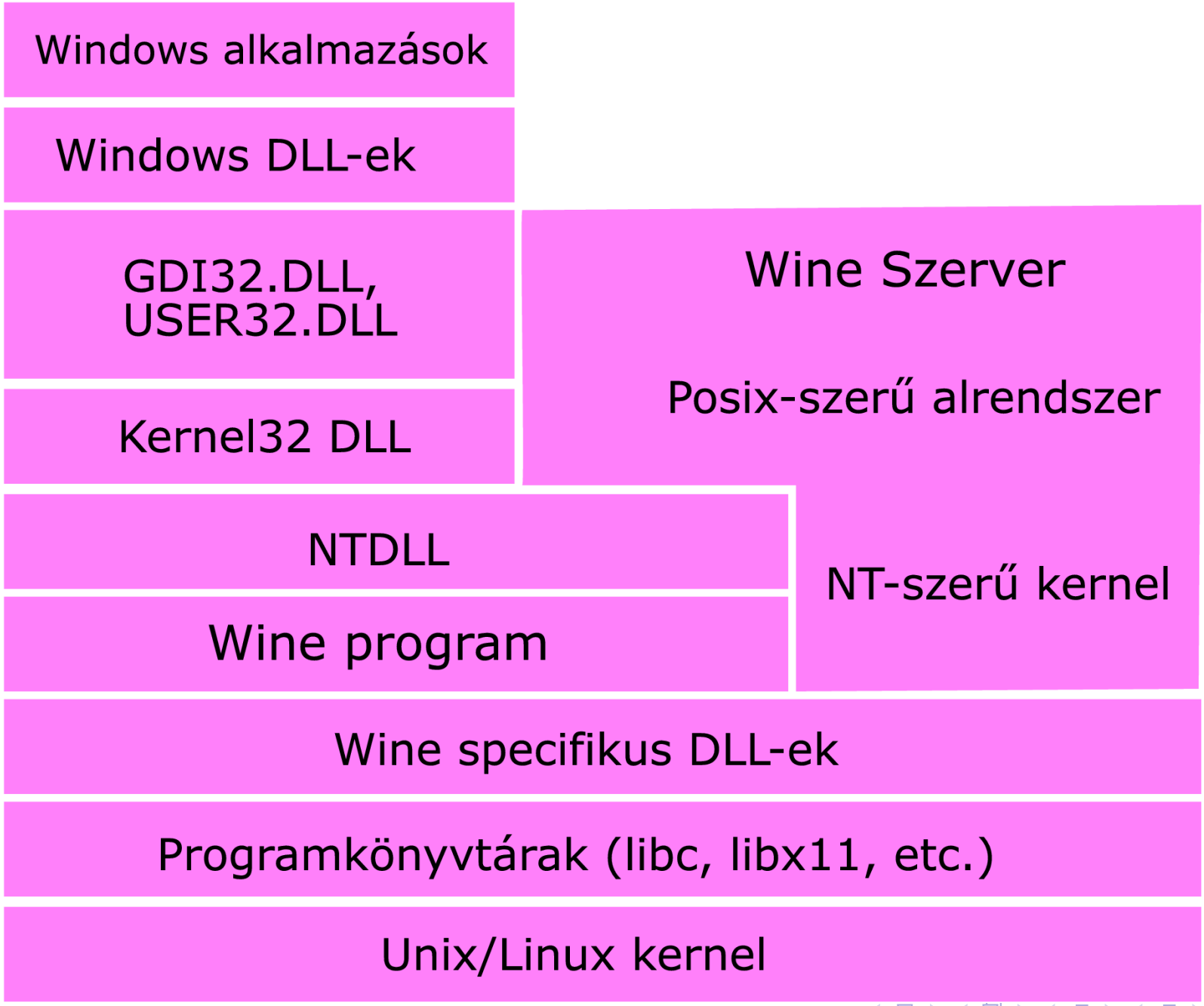
Egyéb


Absztrakciós szint
Optimalizáció
Alkalmazás szintű Virtualizáció
pl.: x86 bináris ppc futtatva
API szintű virtualizáció
pl.: Wine, Cygwin, Mplayer DLL
Operációs Renszer szintű virtualizáció
pl.: chroot, jail
Optimalizált Operációs Renszer szintű virtualizáció
pl.: vserver, openVZ

Operációs rendszer szintű virtualizáció - Linux VServer



API szintű virtualizáció - Wine



Virtuális hálózati megoldások

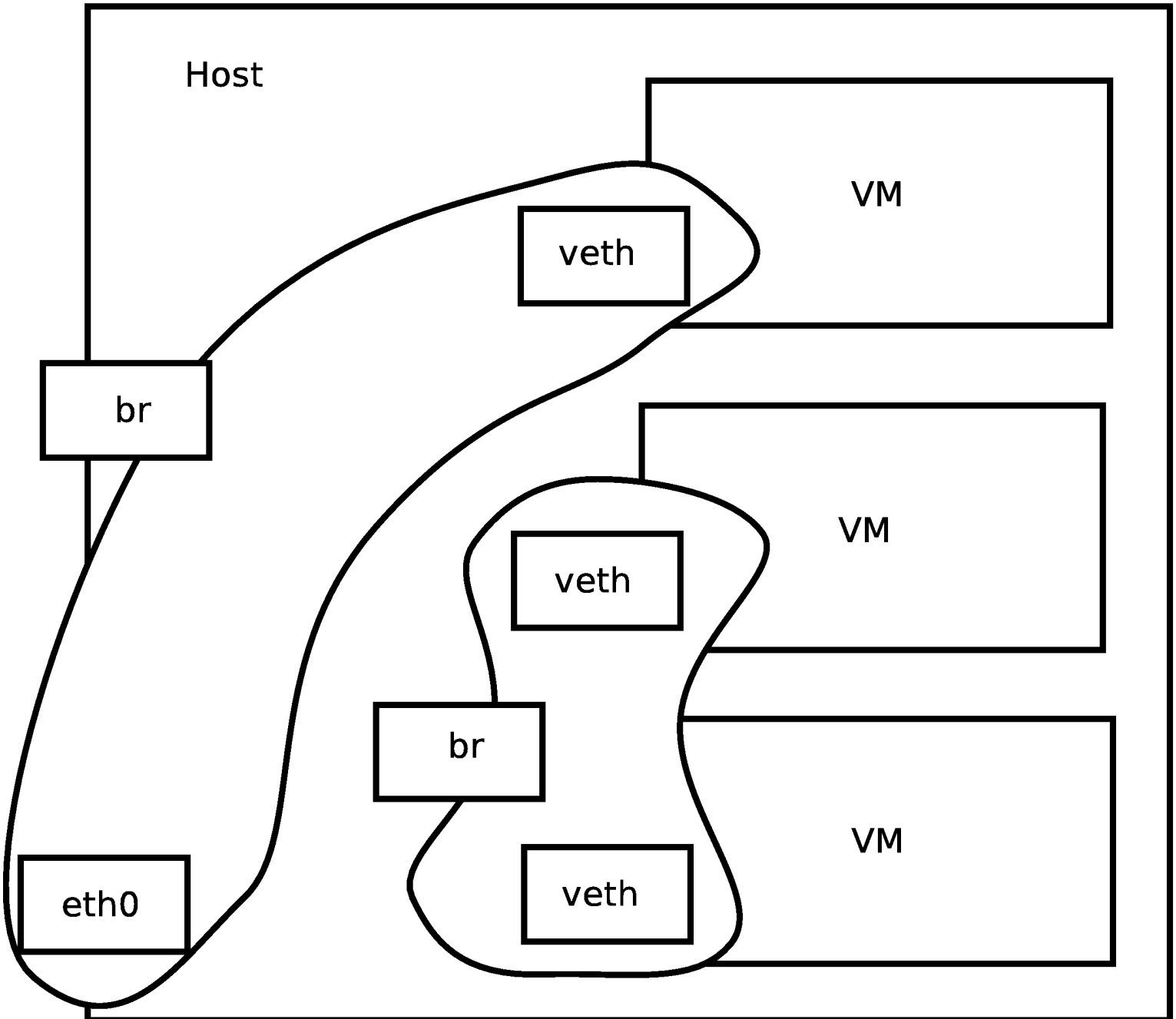
Virtualizált hálózati interfész:

- virtuális interfész a host rendszeren (XEN)
- TAP driver - TUN/Etertap - UML, CoLinux
- pcap könyvtár
- usespace megoldás pl.: switch deamon
- slirp (Serial Line Internet Protocol) - csak IP alapú kapcsolatra (userspace NAT), nem kell hozzá root jogosultság

Hálózati kapcsolat biztosítása a virtuális gépek számára:

- szoftver bridge
- NAT
- route
- fizikai hálózati eszköz delegálása virtuális gépnek közvetlen használatra

Virtuális hálózat - bridge



Diszk formátumok

Célok:

- Minél kisebb méret
- Minél gyorsabb elérés

Méret csökkentő megoldások:

- "Lyukak" a állományrendszerben
- COW (Copy On Write) és a "ritkás állomány" (sparse)
- Tömörítés
- "Snapshot mode"

Sebességnövelő megoldások:

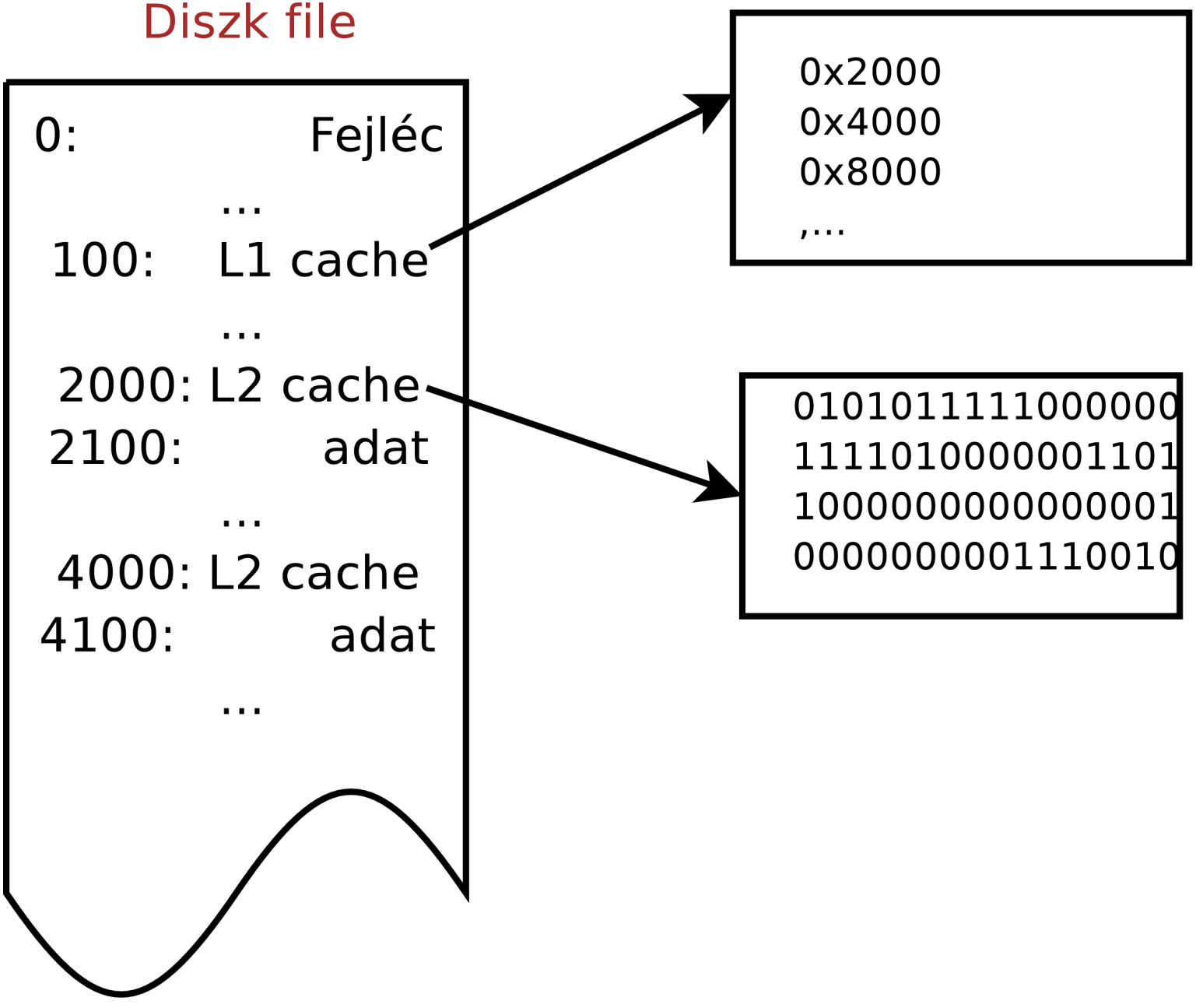
- Többszintű "cache" és "clustering"

Extrák:

- Titkosítás

Cache és clustering

Diszk file



Diszk formátumok folytatás

Típusok:

- Raw
- COW:
  - UML
  - VMware Disk
  - Conectix/Microsoft Virtual PC
- Tömörített:
  - DMG
  - QCOW

BIOS emuláció

Megvalósítás típusa:

- Emulátor által lekezelt néhány port / kivétel
- Akár hardveres image futtatása

Ismertebb rendszerszoftverek:

- PC BIOS
  - 16 bites
  - Kiegeszítő ROM imagek, pl. VGA vagy hálózati kártya BIOS
- OpenFirmware / SunBoot
  - 32 bites
  - Forth nyelven írt - "szkriptelhető"
- EFI
  - 32 bites
  - Bővithető, modernebb - PC-khez

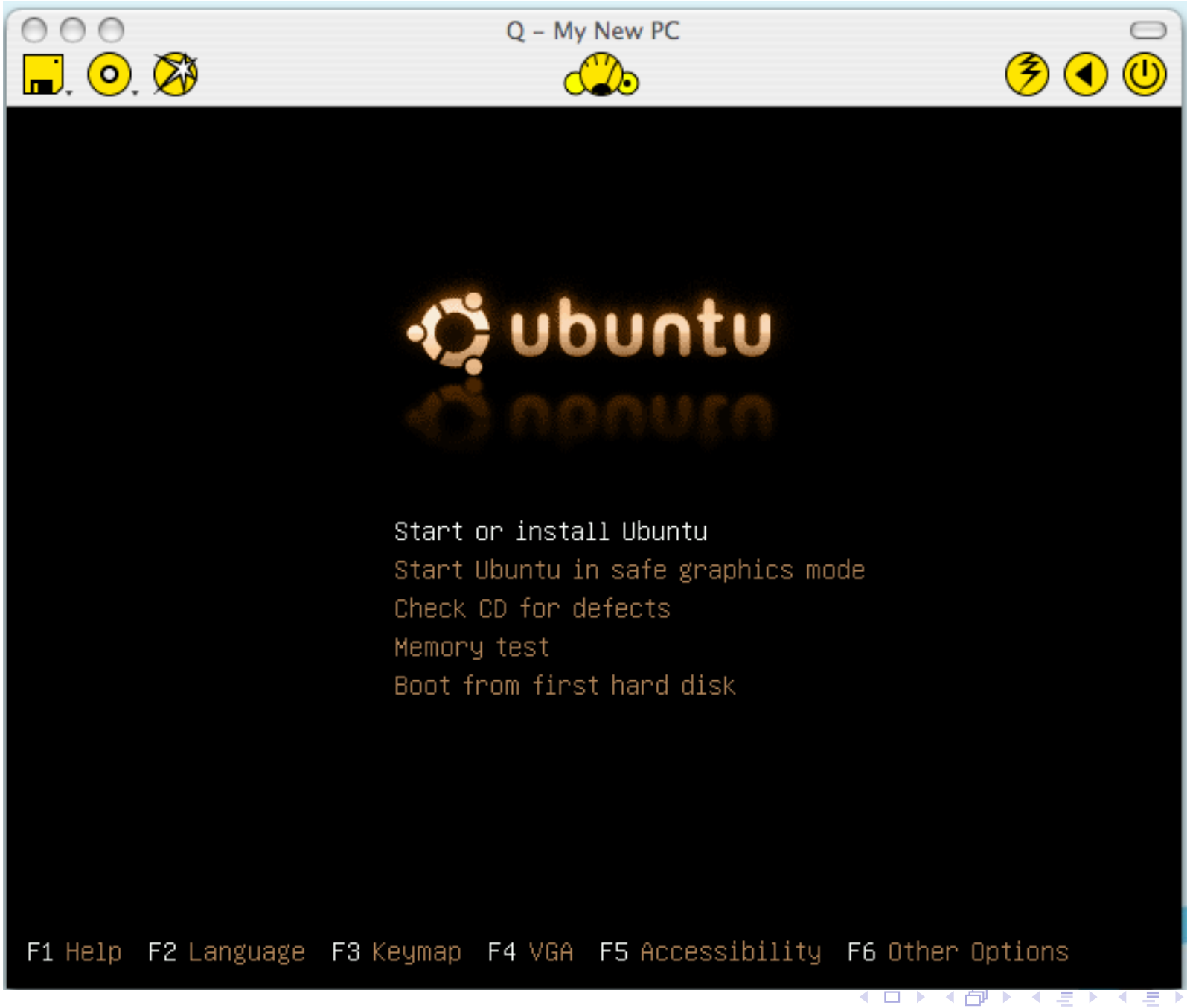
Qemu

- Nyílt forrású több architektúrát támogató emulátor
- Vegyes GPL és LGPL licensz

Kétfajta futási mód:

- User: pl. x86-ra fordított Linux bináris futtatása powerpc-n
- System: teljes gép emulációja

QEmu on Mac OS X



Qemu - támogatott rendszerek

Arch	User	System	Host
x86	I	I	I
x86-64	N	I	I
ppc	I	I	I
ppc64	N	F	N
arm	I	I	F
sparc	I	I	F
sparc64	F	F	F
mips	I	I	N
m68k	F	F	N
sh4	F	F	N
ia64	N	N	I

Qemu - optimalizáció

- JIT - 80-90%-al lassabb a natívnál
- QEMU Accelerator (kqemu) - 0-50%-al lassabb a natívnál
- qvm86

Diszk imagek:

- Bochs
- cloop
- UML COW
- DMG
- VMware v3 / v4
- VirtualPC
- Virtual FAT

XEN

- Nyílt forrású paravirtualizációs megoldás
- Ipari támogatás: XenSource, IBM, Novell, Microsoft, SuSE, RedHat stb.
- nagyobb operációs rendszer terjesztések része (Debian Etch-től, SuSE 10.x-től, RedHat 5.x-től)
- hardver támogatott virtualizációt ki tudja használni Intel és AMD processzorokban
- XEN Enterprice: javított teljesítmény, grafikus távoli management felület

XEN - telepítés/Debian Etch

- Csomagok: xen-hypervisor-3.0.3-i386, linux-image-xen-686, libc6-xen, bridge-utils (hálózathoz), xen-ioemu-3.0.3-1 (HVM támogatáshoz)

Grub konfiguráció:

```
title XEN 3.0
root (hd0,0)
kernel /boot/xen-3.0-i386.gz dom0_mem=25600
module /boot/vmlinuz-2.6.17-3-xen-686 root=/dev/sda
ro console=tty0
module /boot/initrd.img-2.6.17-3-xen-686
```

XEN - management

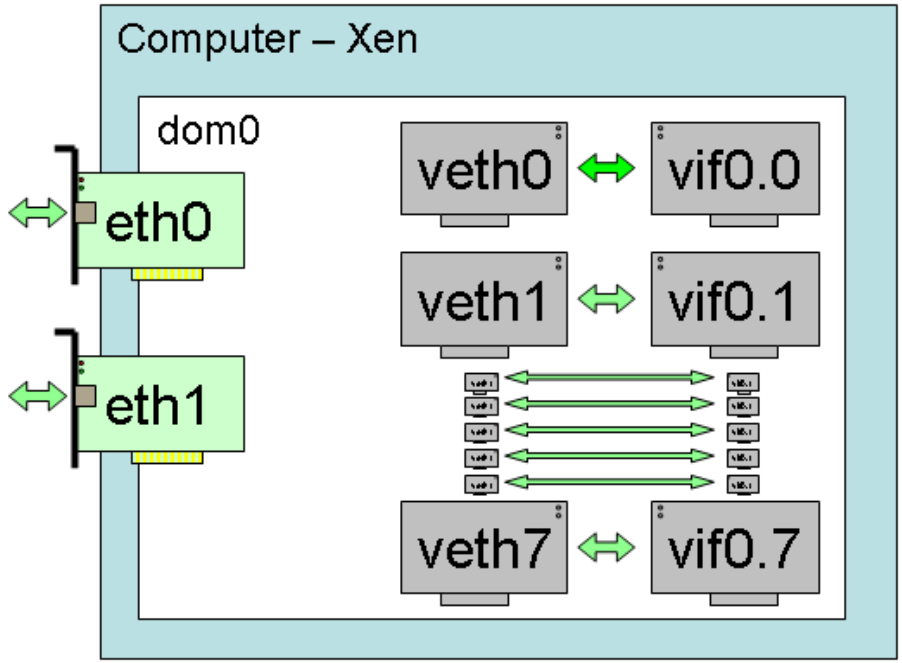
- **xend**: daemon ami a userspace eszközök és a VMM valamint a domain-ek közötti kommunikációt segíti. `/etc/xen/xend-config.sxd`
- **xm**: parancssori kliens domain-en kezelésére
  - `info`: hipervisor információ
  - `create`: domain létrehozása
  - `list`: domain-ek listázása
  - `shutdown`: domain leállítása
  - `pause, unpause`: domain működésének felfüggesztése
  - `save, restore`: domain állapotmentése és visszaállítása (a la checkpoint)
  - `migrate`: futó domain áthelyezése másik fizikai gépre (kiesés <100ms)
- minden futó domain példánynak egyedi azonosítója van (domainID), a domain nevének is egyedinek kell lennie.

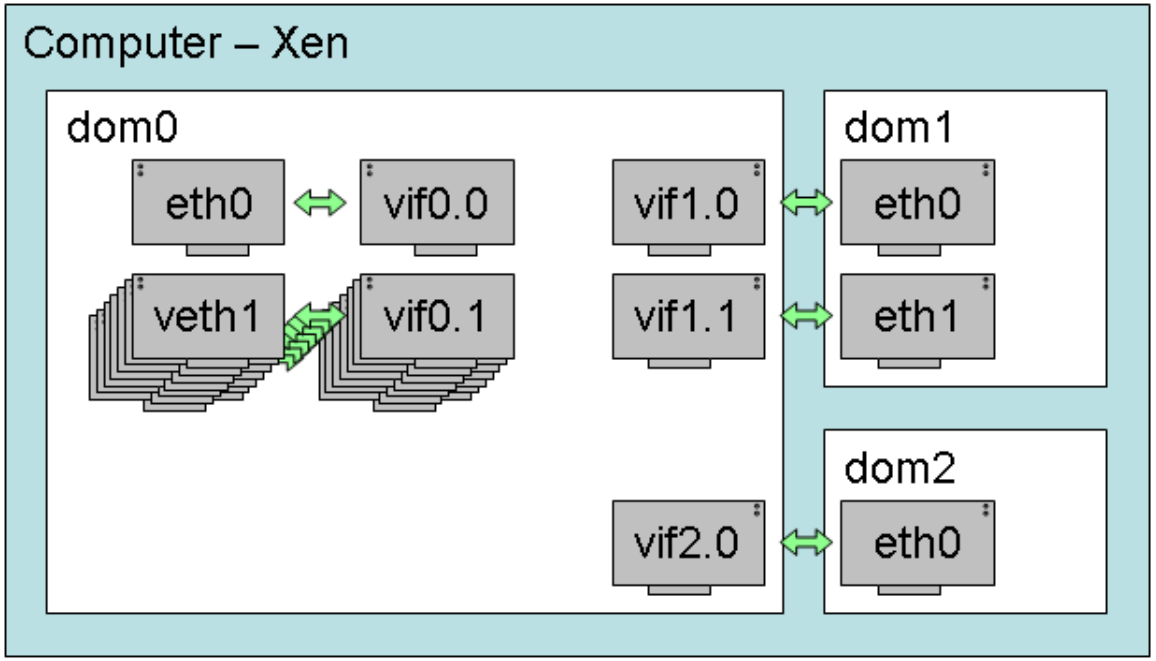
XEN - domain létrehozása

Konfigurációs állomány: python script

```
# cat /etc/xen/noc.grid.conf
name = 'noc.grid'
kernel = '/boot/vmlinuz-2.6.17-3-xen-686'
ramdisk = '/boot/initrd.img-2.6.17-3-xen-686'
memory = 256
vif = ['mac=00:16:3E:00:00:13, bridge=xenbr1']
disk = ['phy:/dev/xenimages/noc.grid,sda1,w']
root = '/dev/sda1 ro'
extra = '2'
on_poweroff = 'destroy'
on_reboot = 'restart'
on_crash = 'restart'
```

XEN - hálózat


Computer – Xen
dom0
eth0
eth1
veth0
vif0.0
veth1
vif0.1
veth7
vif0.7


Computer – Xen
dom0
eth0
vif0.0
veth1
vif0.1
vif1.0
vif1.1
vif2.0
dom1
eth0
eth1
dom2
eth0

XEN - hálózat folytatás

- `/etc/xen/scripts` alatt
- egy ’network’ script, ami a dom0-án állítja a hálózatot a xend indulásakor
- egy ’vif’ script ami a dom0-án konfigurálja a virtuális interfészeket
- minden domaint vifX.Y alakú halózati interface reprezentála a domain0-ban. (X=domainID, Y=interface szám)
- lehetőségek:
  - bridge
  - route
  - nat (problémás)
  - vegyes megoldások
  - saját scriptek

XEN - diszk hozzáférés

Lehetőségek:

- image állomány (loopback)
  `disk = ['file:/var/images/debian.img,sda1,w']`
- user space megoldások: blktap
  `disk = ['tap:aio:/dev/images/debian.img,sda1,w']`
- külön partíció
  `disk = ['phy:/dev/hda2,sda1,w']`
- logikai kötet (LVM)
  `disk = ['phy:/dev/xenimages/noc.grid,sda1,w']`
- use space megoldás: qemu-dm
  `disk = ['phy:/dev/xenimages/win2003,ioemu:hda1,w']`
- NFS root: konfigurációs állományban `nfs_server`, `nfs_root`

XEN - hol tároljuk a virtuális lemezeket

- **fizikai gépbe sok lemez**: egyszerű de nehezen skálázható, nagy rendelkezésre állás kialakítása nehézkes (DRDB)
- **NAS(NFS)**: három független állományrendszer réteg konzisztenciáját kell fenntartani
- **FC/Infiniband SAN**: nagy teljesítmény, kis késleltetés de drága
- **IP/Ethernet SAN**: közepes teljesítmény de célnak általában megfelel (iSCSI, AOE)

XEN - hova tovább?

- teljesítmény elemzés: xeno-profile
- Guest API: hipervizor inkompatibilitások csökkentése (vitual I/O interfész kód tisztázása)
- IOMMU - I/O virtualizáció
- Infiniband - I/O virtualizáció
- OS támogatás: **Linux**, *NetBSD 3.1, FreeBSD 7.0, OpenSolaris 10*, Plan9
- vanilla kernel része?
- XML konfig állományok mindenhol
- management: DTMF CIM támogatás, Xen-XML RPC
- Copy on Write (userspace dm), virtuális lemezformátumok: VMDK, MS VHD
- erőforrás kezelés: VCPU fizikai CPU-hoz rendelése dinamikusan futásidőben, QoS
- HVM támogatás javítása: Qemu ioemu-tól megszabadulni, teljesítmény fokozás, video és usb támogatás javítása
- XenFS: közös állományrendszer VM-ek között
- IA64 port, PowerPC port