

Virtualizáció szabad szoftverekkel

Mátó Péter <mato.peter@andrews.hu>

Néhány érdekes adat

A ma kapható legkisebb számítógép:

processzor: 1,6GHz, memória: 1GB, diszk: 160GB

Egy átlagos céges weblap igényei (Drupal):

processzor 1% (max. 20%), memória: 20%, diszk: 5%

Gazdaságosan összerakható SAN: 3TB

Átlagos cég, átlagos alkalmazás DB: 5GB

Milyen az élet?

- Ki szeretnék próbálni egy új operációs rendszert, mielőtt éles szerveren használnám
- Soknyelvű szoftver hibajelentéseinek ellenőrzése, a javítások tesztelése (hihetetlen idő és hely megspórolható a virtualizált tesztfarmokkal)
- Úgy érzem, hogy már túl sok a Linux felhasználó, egyre nagyobb a csábítás a sötét oldalon, hogy ezt a rendszert is támadják

Az előzmények

- 1960: IBM CP-40
- PC
 - Unix: chroot (4.2BSD: 1982), BSD: jail (FreeBSD: 2006)
 - VMWare (VMW WS: 1999, VMW GSX: 2001)
 - Windows Virtual PC, Solaris Zones
 - UML (User Mode Linux), QEmu, XEN, VirtualBox, KVM, OpenVZ

Virtualizáció, csoportosítások

- Teljes számítógép virtualizáció (rendszer VM)
 - pl.: KVM, QEMU, XEN, VirtualBox, VMWare, DOSBox
- Részleges virtualizáció (processz VM)
 - speciális futtató környezet létrehozása
 - pl. Java, .NET, Dalvik /Android/, Parrot /Perl 6/, Flash Player
 - szeparált futtató környezet (pl. jail, chroot)
- Storage, adatbázis, hálózat

Néhány alapfogalom

- Host, Guest vagy VM
- Hardver támogatott vagy nem
- Szerver, desktop virt.
- Hypervisor alapú virtualizáció
- Teljes virtualizáció – paravirtualizáció (részleges, a guest módosítása szükséges)
- Idegen architektúra támogatás (pl. QEMU)

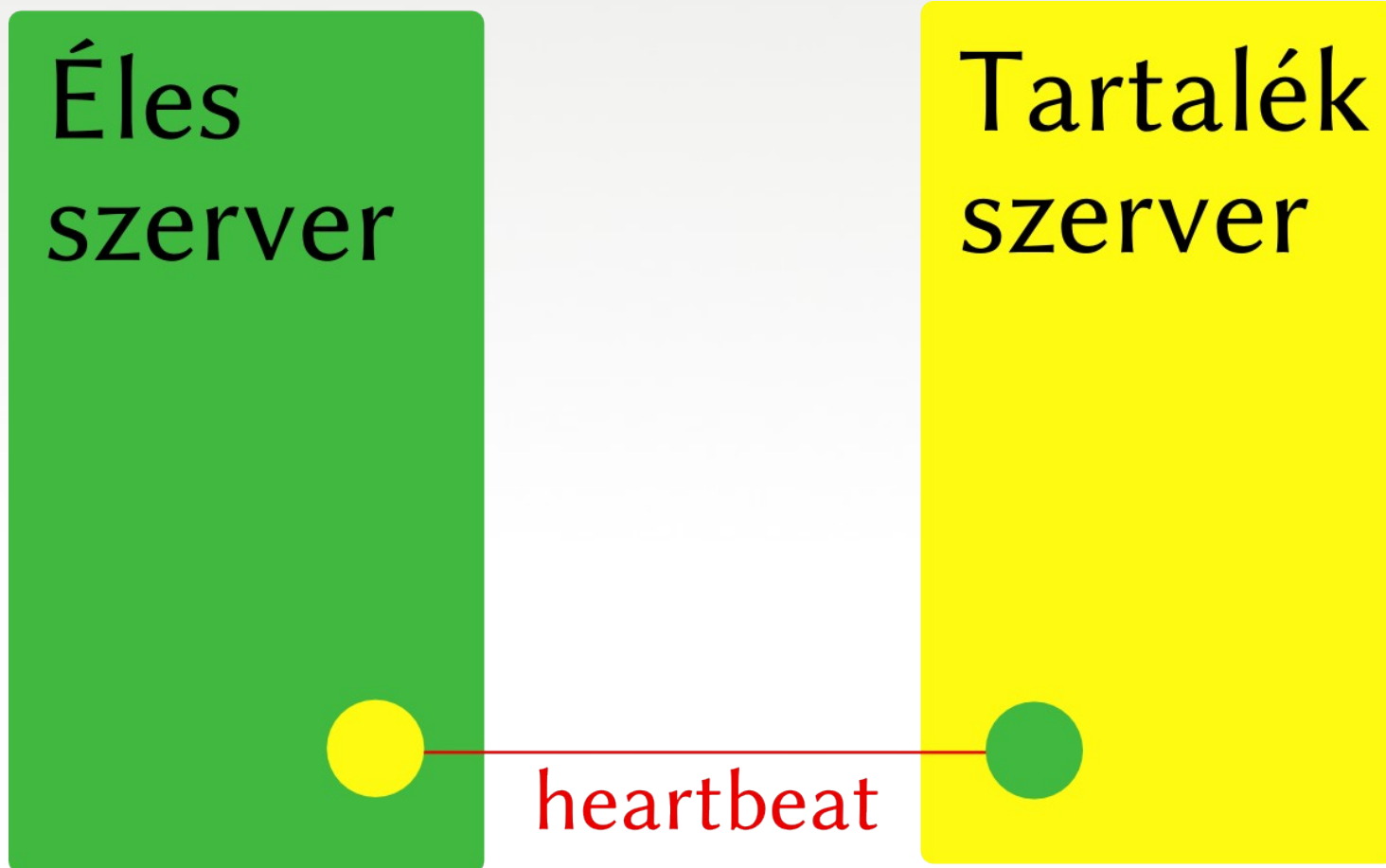
Mi indította el a térhódítását?

- Egyre erősebb x86 gépek, több szolgáltatást elbírnak
- A virtualizáció nagyon rugalmassá teszi az IT-t
- A gépek és a hűtés sokat fogyasztanak, de egyre fontosabb, hogy kicsi legyen az áramszámla
- Kevesebb helyet kell bérelni az adatközpontban
- Gépeket vásárolni és összerakni macerás és sok a drága kidobott idő (nem ritkán akár két hónap)

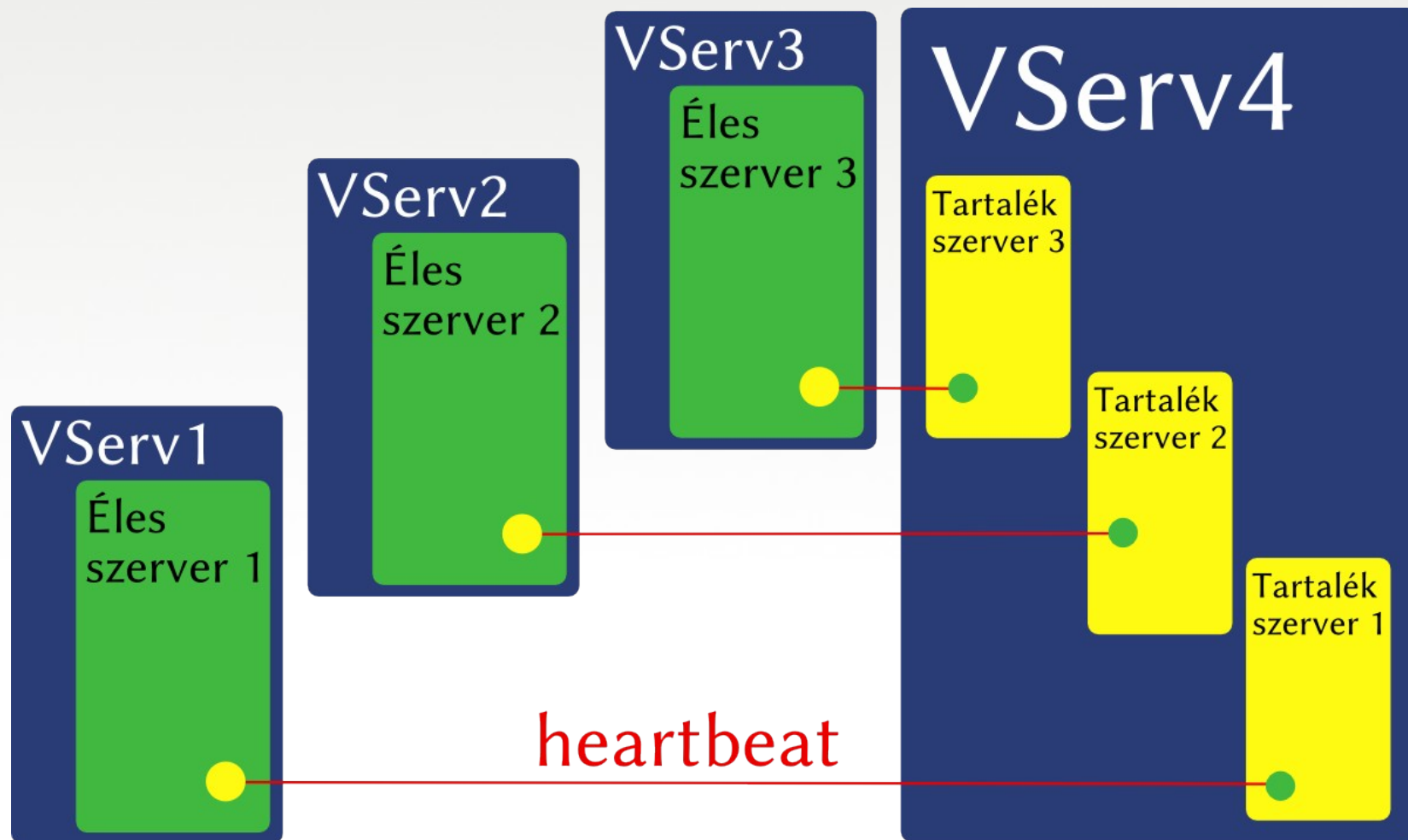
Mire használható?

- Szerverek virtualizációja
 - kevesebb gép, kényelmesebb adminisztráció
- Tesztkörnyezetek kialakítása
 - adminisztrátorok, fejlesztők, diákok
- Speciális helyzetek kezelése
 - aszimmetrikus HA-k kialakítása
 - ha a tech támogatja, akkor spéci HA megoldások

Klasszikus HA



Virtualizált HA



Mire használható még?

- Nincs többé újratelepítés vascsere miatt
- Egyszerűbb a mentés, archiválás, katasztrófa helyreállítás
- Vas migráció elvégezhető online (live migration módszer használatával)
- A gépek erőforrásai nagyon rugalmasan változtathatók

Pénzügyi megtakarítások

- Csökken az áramszámla
- Csökken a hűtés miatti áramszámla
- Csökken a méret, kevesebb rack-et kell bérelni
- Növekszik az adminisztrátor hatékonysága, kevesebb emberrel is el lehet látni ugyanazt a feladatot
- Rugalmasabb, hibatűrőbb rendszer, kevesebb kiesés, kevesebb veszteség

A lehetséges problémák

- Ha a hosthoz hozzáférnek, akkor minden rajta futó virtuális szolgáltatáshoz is hozzáférnek
- Ha a host valamilyen hiba folytán leáll, akkor az egész rendszer halott, minden komponensével

Paravirtualizáció

- Kissé kényelmetlenné teszi a rendszer használatát, mivel módosítani kell a guest oprendszerét
- Ennek azonban megvan az előnye is
 - virtualizációra kiélezett guest rendszerek (pl. JEOS)
 - speciális hardver meghajtók (pl. Linux VIRTIO, VirtualBox guest additions...), speciális lehetőségek
 - a sebesség jobb lehet, mint a teljes virtualizációnál

Zárt vagy szabad?

- Kétségtelen, hogy a zárt virtualizációs megoldások előrébb járnak
- A legfontosabb funkciókat azonban minden rendszer megfelelően, stabilan támogatja
- A szabad szoftver megoldások nem fogják a felhasználókat csapdába ejteni
- A legrosszabb, ami történhet, hogy egy zárttá váló kiadás esetén a közösség fork-olja a fejlesztést

Mire van szükségünk hozzá?

A technológiától és az igényeinktől függően:

- Egy erős gép, sok mag, sok RAM
- 64bites rendszer a memória használat miatt
- Virtualizációs támogatás a prociba
 - AMD-v, Intel VT-x
- Közös diszk alrendszer (NAS vagy SAN)

A host: diszkek

- az adatbiztonság növelésére RAID
- a rugalmasság növelésére LVM
- A közös használathoz közös elérés kell:
 - FC hálózat, iSCSI, ATA over ethernet, DRBD
 - Spéci fájlrendszer, ami képes kezelni a párhuzamos hozzáférést (pl. GFS – Global Filesystem)
- Mindezek együtt: NAS, SAN (pl.:FreeNAS)

A host: processzor

- Mindenképp törekedni kell rá, hogy egy virtuális host-ban legyen minél több proci mag
- Nem szerencsés annyi magot a VM-nek adni, amennyi van (host OS-nek is kell, párhuzamosság, várakozás a procik felszabadulására)
- Ma már olcsón kapható sokmagos rendszer (pl. Core i7 alapon felépített rendszer)

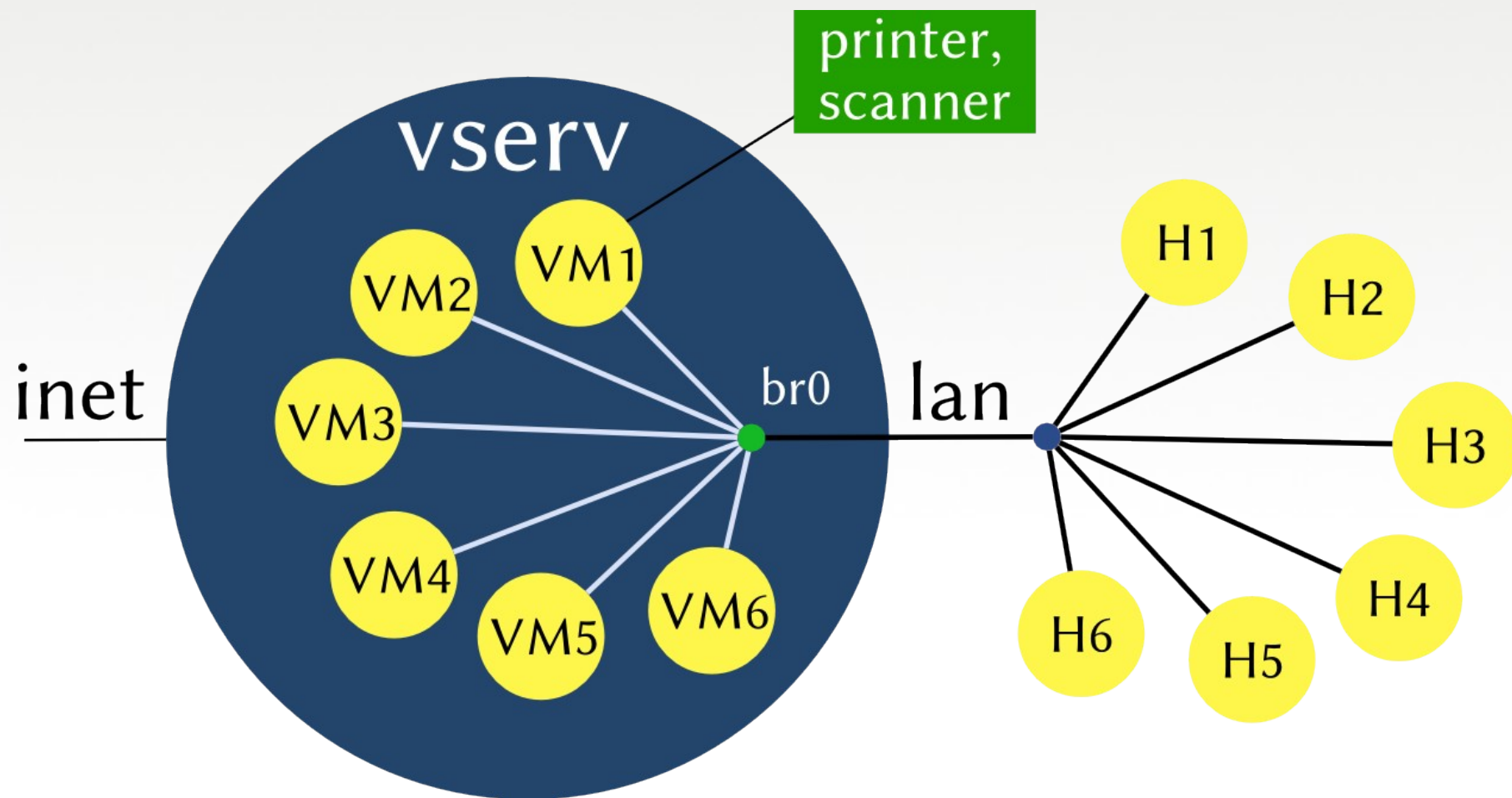
A host: memória

- Legyen benne nagyon sok
- Általában ez a szűk keresztmetszet
- Ha lehet, figyeljünk a DDR2 és DDR3 csatornák optimális kihasználására
- Nem szerver vasba jelenleg 12G, ritkán 24G tehető (de az utóbbit én csak úgy hiszem el, ha a kereskedő összerakja a vasat, mielőtt kifizetem)

A host: hálózat

- Legalább egy GB csatlakozó
- Ha iSCSI vagy AOE használt, akkor erre érdemes egy dedikált GB alhálót kiépíteni
- A hálózatnál a virtuális hálózatot úgy kell kialakítani, mint egy szerverterem hálózatát
- Vagy maga a host, vagy egy VM legyen a tűzfal a virtuális gépek felé (Fokozott biztonság szükséges!)

Egy virtualizált kisvállalat



A Qemu

- 2005-2008 Fabrice Bellard
- processzor emulátor
- x86-on virtualizál
- KQemu csak később lett GPL
- processzorok:
x86, x86_64, ARM, Sparc, PowerPC, MIPS...

A KVM rendszer

- Kernel-based Virtual Machine
- 2006. dec. 18. A KVM bekerül a 2.6.20-as kernelbe
- 2008. szept. 5. A Redhat megvásárolta a Qumranet

A KVM jellemzői I.

- stabil, kicsi és egyszerű
- szerveren remekül használható
- a Qemu-t userspace részét használta fel
- része az Ubuntu 10.04 LTS rendszernek
- módosítás nélküli guest kernellel működik
- hardver támogatás szükséges
 - Intel: Intel VT támogatás (`grep vmx /proc/cpuinfo`)
 - AMD: AMD-V támogatás (`grep svm /proc/cpuinfo`)

A KVM jellemzői II.

- erőforrások finomhangolása nem lehetséges
- 32 és 64 bites host támogatás
- 32 és 64 bites guest támogatás
- SMP host támogatás
- SMP guest támogatás (max. 16 CPU)
- guest swapping
- live migration

Támogatott guest rendszerek

- Csak néhány példa:
- Linux 2.6, 32/64bit
- *BSD, 32/64bit (Net 32)
- MS Windows Server 2008, 32/64bit
- MS Windows XP, 32/64bit
- MS Windows 7 és Vista, 32/64bit

A virtuális vas jellemzői

- proci - speciális PII-nek látszó
- memória - szabadon beállítható méret
- diszkek
 - IDE / SCSI / virtio
- CDROM
- hálózati eszközök
 - ne2k_pci, pcnet, rtl8139, virtio ...

A virtio jellemzői

- 2.6.25-ös kernelbe és a 60-as KVM-be került be
- A virtio diszk saját mérések szerint 10x gyorsabb, mint az SCSI emuláció
- A virtio net más mérései szerint 2-4x gyorsabb
- A guest-en a block eszköznél speciális beállításokra van szükség

VM erőforrásigénye

A gép jellege	Párhuzamos felhasználó	Proc.	Mem.	Diszk	Swap
Statikus mikrohost	2-5	1	48MB	1GB	1GB
Mini php host	2-5	1	256MB	1GB	1GB
Közepes Drupal	5-50	1	400MB	4GB	1GB
Nagyobbacska DB	12 szerver	2	2GB	20GB	2GB
Nagyobb Drupal, WP	50-500	2	2GB	4GB	2GB

Erőforrás igény felmérése

- Ha már van gép, akkor azon mérni kell az processzor, I/O, memória, hálózat és diszk terhelését
- Ha nincs, akkor becslés alapján, túlméretezve létrehozni a virtuális gépet
- utána a free, a top, az iotop, df és iptraf, vagy inkább Munin figyelésével a szükséges méret beáll.
- a diszk átméretezésére szerencsésebb egy új diszk vagy LVM használható

Virtuális gépek kényelmesen

- libvirt és Virt-Manager
- Virtuális gépek létrehozása klónozással, néhány beállítást megcsinál, csak néhány dolgot kell állítani
 - hálózati cím (a MAC-ből generálhatjuk az első boot-kor)
 - gép neve
 - szükséges plusz csomagok, az alrendszer beállításai
- cron-apt, és egyéb időzített ellenőrzések

Erőforrások szabályzása

- A memória szabályzása viszonylag egyszerű
- A processzeket nice segítségével lehet priorizálni
- Az I/O-t az ionice segítségével priorizálhatjuk
- A hálózat felhasználását a Netfilter limit moduljával szabályozhatjuk egyszerűen
- Hálózatonál lehetőség van osztály bázisú sáv szélesség szabályzásra (pl. CBQ, HTB)

Erőforrások szabályzása II.

- A Szent Grál? A „fair share scheduler”?
- Végre itt van, csak nagyon kevesen tudják. Az erőforrások finomhangolását teszi lehetővé a CGroups rendszer
- Jelenleg az elérhető dokumentációk mennyisége nagyon szerény, sokat kell kísérletezni
- Lehetővé teszi a processzor, memória, I/O és a hálózat erőforrás szabályzását

Biztonsági kérdések

- Ha a KVM kernel része hibás, akkor vége a bulinak
- A host-ra minimális hozzáférést, csak admin csatornára, és csak szabályzott forrásból
- A host csomagszűrőjét felhasználva minimalizálni kell az áthallást a virtuális gépek között
- Rendszeres frissítés mindenhol
- A VM hálózaton naplózzon, minimum a host-ra

Köszönöm a figyelmet.

Kérdések?

Virtualizáció szabad szoftverekkel

Mátó Péter <mato.peter@andrews.hu>