

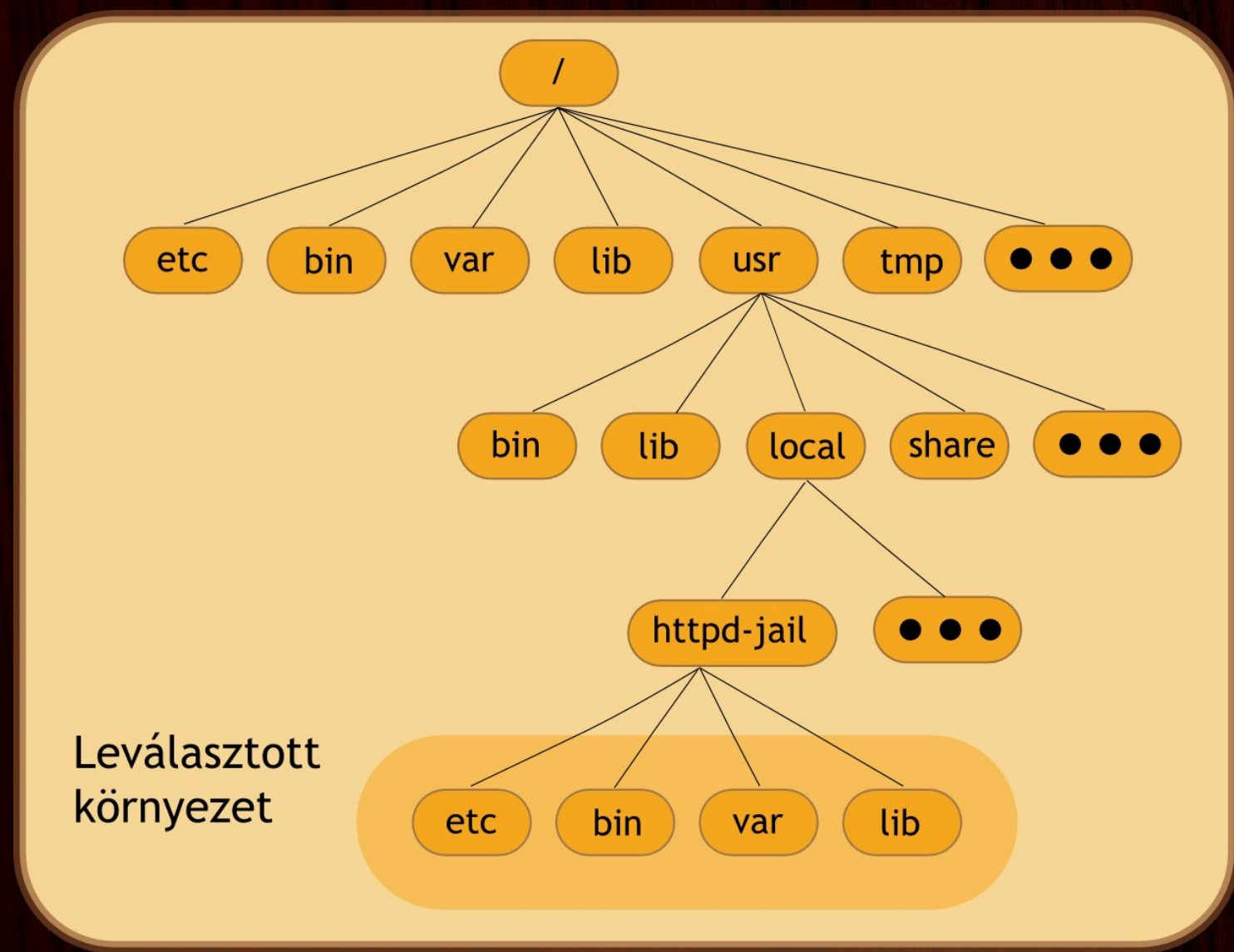
Szerverterem egy számítógépben avagy hogyan élnek a barack lakói

Mátó Péter <mato.peter@andrews.hu>

A barack rövid története I.

- A rendszer az OpenOffice.org menüjének honosítása után, a projekt weboldalának indult
- Először még elég szerény vas volt, de hamarosan a 23VNet jóvoltából kiváló szerverhotelt kaptunk
- A jó helyre össze akartunk hozni egy tisztességes gépet, ez egy öreg óriástoronyba került, nagyjából ilyen alkatrészekből:
 - P4 2,4GHz, valami alaplap, 512MB mem, 80GB diszk
- Ekkor még chroot környezetekből állt a rendszer

Chroot környezet



A barack rövid története II.

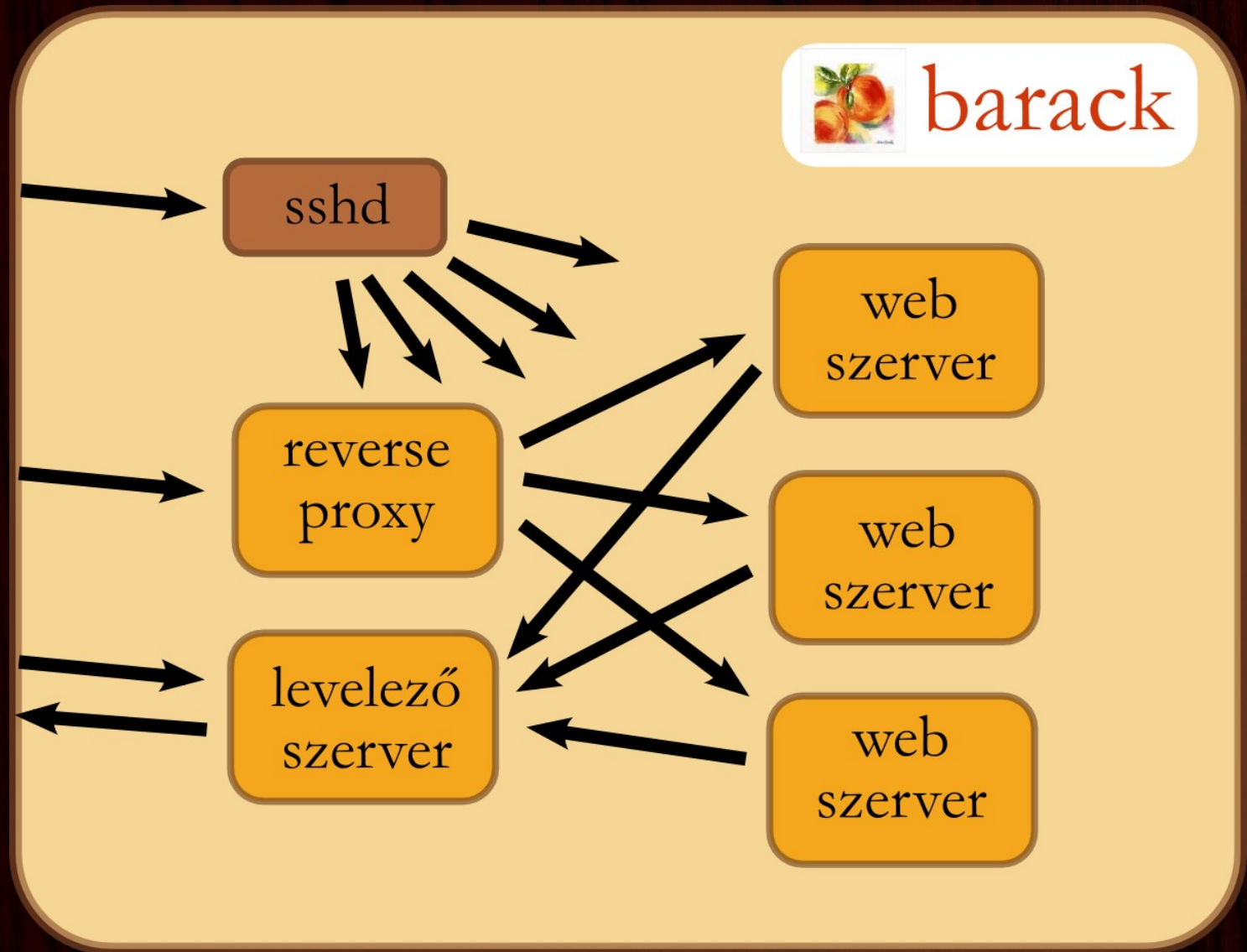
- Kicsit több lehetőségünk lett (gyk. pénzünk)
- Egy elszállás (az durva volt), több leállítás és erőforrás probléma után úgy döntöttünk, hogy kell egy megbízhatóbb gép
- Igyekeztünk előre mutató konfigot összeállítani, tehát valami ilyesmi lett:
 - Core2 Duo 2,1GHz, valami alaplapp, ami tud 8G-ig memóriát, 2x1G Geil mem, 150GB diszk
- Meghívogattunk, bővítettünk

A barack vas jelenleg

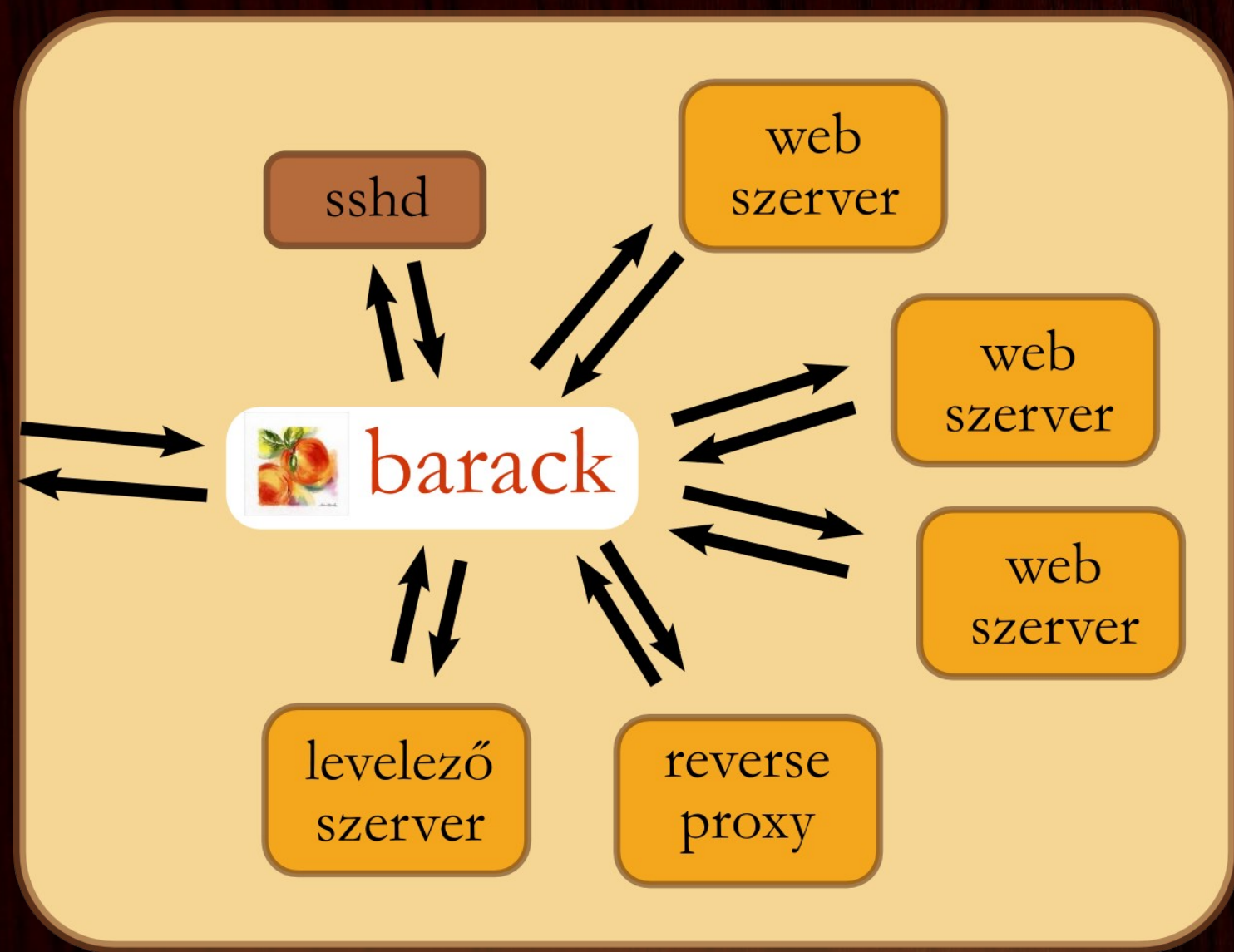
- Chieftec 400W ház
- Core2 Quad 2,4GHz
- ASUS P5B, NIC GB + valami videokártya
- 4x2GB Geil memória
- 2x320GB Seagate Barracuda SATAII
- valami régi 150G diszk mentésre

jelenleg összérték kb.: 150e HUF ami, lássuk be, vicc

A barack logikai sémája



A barack még logikaibb sémája



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

- 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 Ubunut Hardy Heron 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 Vista Ultimate, 32/64bit
- MS Windows XP, 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

A virtio bekapcsolása

CONFIG_VIRTIO_PCI=y

(Virtualization -> PCI driver for virtio devices)

CONFIG_VIRTIO_BALLOON=y

(Virtualization -> Virtio balloon driver)

CONFIG_VIRTIO_BLK=y

(Device Drivers -> Block -> Virtio block driver)

CONFIG_VIRTIO_NET=y

(Device Drivers -> Network device support -> Virtio network driver)

CONFIG_VIRTIO=y (automatically selected)

CONFIG_VIRTIO_RING=y (automatically selected)

A guest rendszerek

- Egy új fogalom: JEOS
Just Enough Operating System
- Ubuntu JEOS jellemzői
 - minimalizált bináris készlet
 - speciális virtualizációra kihegyezett kernel
 - minden csomag elérhető
 - a szokásos könnyedséggel kezelhető
- Egy telepítés után könnyen klónozzható

A virtuális gép létrehozása

- dd vagy qemu-img

```
dd if=/dev/zero of=vda.img bs=1M count=2048
```
- boot Jeos iso-ról, telepítés a szokásos módon
- a vadmin felhasználó létrehozása
- ssh kulcs elhelyezése
- sudo beállítása a vadmin felhasználó számára
- kernel és boot loader eltávolítása

A virtuális hálózat

- Használt csomag: bridge-utils
- létrehozunk egy speciális NIC-et

```
# auto br0
```

```
iface br0 inet static
```

```
address 10.1.1.254
```

```
netmask 255.255.255.0
```

```
pre-up brctl addbr br0; echo '1' > /proc/sys/net/ipv4/ip_forward
```

```
post-down brctl delbr br0; echo '0' > /proc/sys/net/ipv4/ip_forward
```

```
bridge_stp off
```

A virtuális gépek erőforrásigénye

- Mini php host:
d: 600M s: 256M m: 64M p: 1
- Közepes Drupal:
d: 2G s: 256M m: 400M p: 1
- Nagyobb Drupal:
d: 10G s: 256M m: 1G p: 1

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 folyamatos figyelésével vissza húzni a szükséges méretre
- a diszk átméretezésére szerencsésebb egy új diszk

Virtuális gépek kényelmesen

- Új gépek létrehozása klónozással
- Csak néhány dolgot kell átállítani
 - hálózati cím (de ezt a MAC-ből generálhatjuk az első boot-kor)
 - gép neve
 - szükséges plusz csomagok
 - az alrendszerek beállításai
- cron-apt, és egyéb időzített ellenőrzések

Erőforrások szabályzása

- a memória szabályzása viszonylag egyszerű
- a processzor felhasználást 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)

Biztonsági kérdések

- Ha a KVM kernel része hibás, akkor bukott az egész rendszer
- 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 virtuális gépek hálózaton naplózzanak, minimum a host-ra

Kérdések?

Köszönöm a figyelmet.