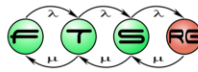


Kiszolgáló oldali virtualizáció

Tóth Dániel, Micskei Zoltán,
Szatmári Zoltán



Tartalom

- **Jellegzetességek**
- Távoli hozzáférés megvalósítása
- Háttértárak virtualizációja
- VMware ESXi architektúra
- Microsoft Hyper-V architektúra
- Xen architektúra
- Erőforrás-gazdálkodás

Szerver virtualizáció jellegzetességei

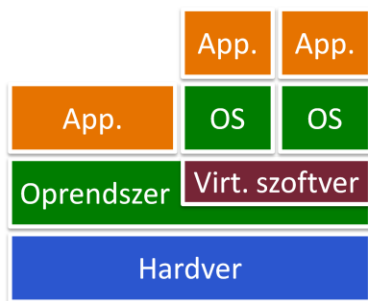
- **„Konszolidáció” – sok virtuális gép együttes futtatása, erőforrás hatékonyság, takarékoság**
- **Hangsúlyos a távoli elérés, helyi hozzáférés teljesen háttérbe szorul**
- Az előző következménye: kevesebb fajta hardvert kell támogatni, nincs „multimédiás” igény
- Karbantartási feladatok egyszerűsítése, automatizálhatóság
- Menedzsment: monitorozás, távoli beavatkozás, terhelés és teljesítményadatok figyelése
- **„Integráció” – a virtualizált szerverek egy nagyobb infrastruktúra részei**
- Hozzáférés vezérlés, biztonság, a rendszernek sok (nem feltétlenül megbízható) felhasználója lehet egyszerre
- Megbízható, hibatűrő működés



Most a vastagon kiemeltekkel foglalkozunk, a többi majd későbbi előadásokban kerül terítékre.

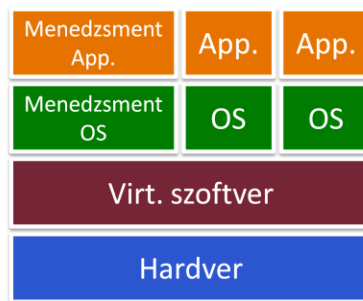
Emlékeztető – a két fő architektúra

▪ Kétféle megközelítés:



Hosted

Jellemzően desktop megoldások:
VMware Workstation, Player,
Sun/Oracle VirtualBox,
MS VirtualPC, KVM



Bare-metal

Jellemzően szerver megoldások:
VMware ESXi, Xen,
MS Hyper-V

Emlékeztető – a két fő architektúra

Hosted

- Interaktív alkalmazásnál előnyös
 - helyi hozzáférés, gyors grafika, hang stb.
 - OS szintű erőforrások biztosítása
 - Integráció asztali munkakörnyezetbe
 - Hardver meghajtókat a hoszt OS kezeli
- Jellemzően kevés virtuális gép fut egyszerre
 - sok virtuális gépnél már rossz skálázódás
 - Jellemzően az OS alkalmazásokhoz kitalált ütemezőit próbálja VM-ek erőforrásgazdálkodására használni

Bare-metal

- Interaktív alkalmazások nehézkesek
 - távoli hozzáférés kell (lokális gépen megjelenítésnél is!)
 - teljesen speciális saját környezet, csak virtuális gépek futtatására
 - nincsenek finomított OS erőforrások
 - Hardver támogatást külön meg kell oldani
- Jó skálázhatóság
 - Nincs hoszt OS, ami erőforrást fogyasztana
 - Saját, VM-ek számára optimalizált ütemezők, erőforrás-elosztók



Szerver virtualizációs megoldások

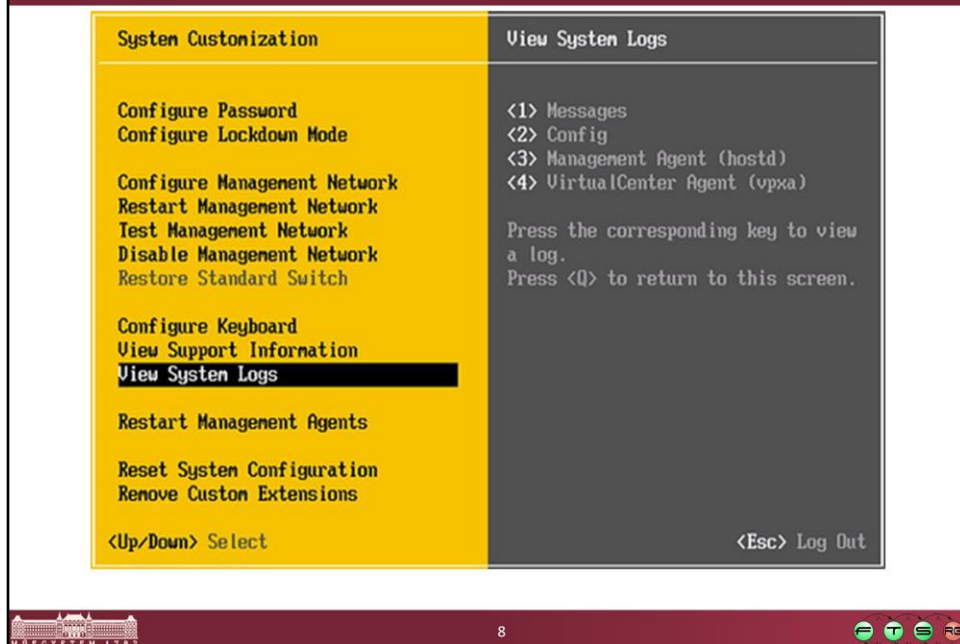
- VMware ESXi
- Xen
 - Xen.org
 - Citrix XenServer
 - Oracle VM Server (Xen alapú)
- Microsoft Hyper-V
- IBM LPAR, DLPAR
- ...



Tartalom

- Jellegzetességek
- **Távoli hozzáférés megvalósítása**
- Háttértárak virtualizációja
- VMware ESXi architektúra
- Microsoft Hyper-V architektúra
- Xen architektúra
- Erőforrás-gazdálkodás

Minimális konzolos beállítási lehetőségek

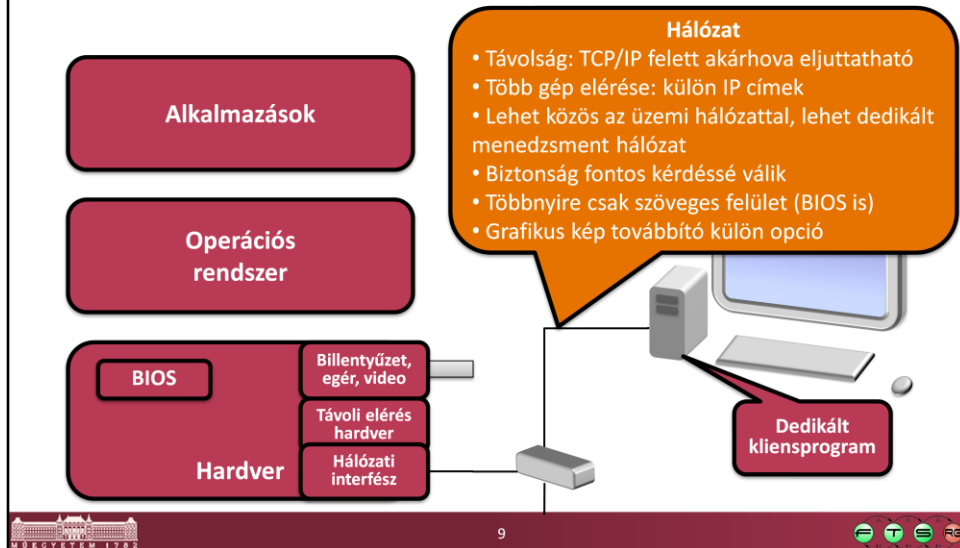


A képen a VMware ESXi konzol felülete látható. Ezen tényleg csak pár kezdeti beállítást lehet megtenni (mi legyen a gép jelszava, hálózati beállításai), amik ahhoz kellenek, hogy távolról utána lehessen menedzselni. Viszont emiatt fontos, hogy a távoli elérés megfelelő módon legyen biztosítva.

Távoli hozzáférés lehetőségek (1)

Beépített távoli menedzsment hardver

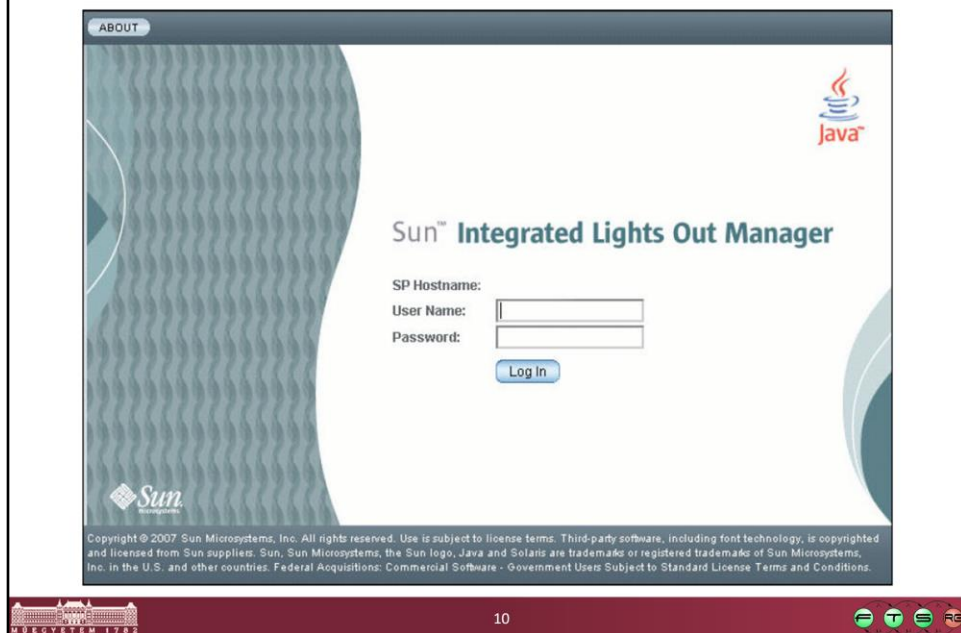
Távoli elérés Ethernet hálózaton keresztül (IPMI, Intel AMT)



Szervereknél gyakori megoldás, hogy valamilyen dedikált hardver elemmel hálózatról elvégezhetővé tesznek olyan műveleteket, amik amúgy manuális hozzáférést igényelnének. Ez az igény a virtualizált környezetekkel szemben is megjelenik.

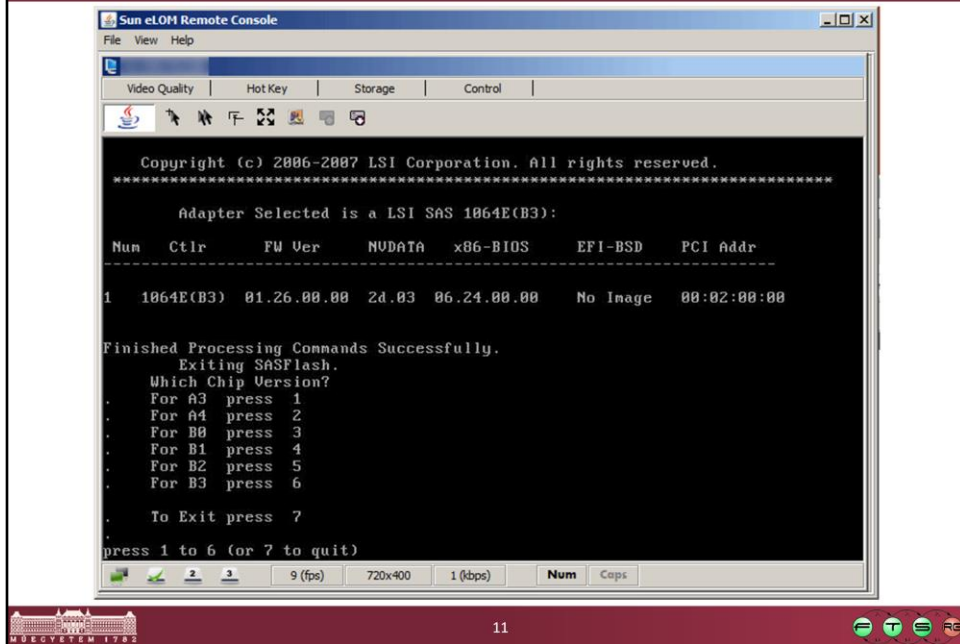
BMC (Baseboard Management Controller)

Példa: SUN ILO



A SUN dedikált hardveres távoli menedzsment felületének képe.

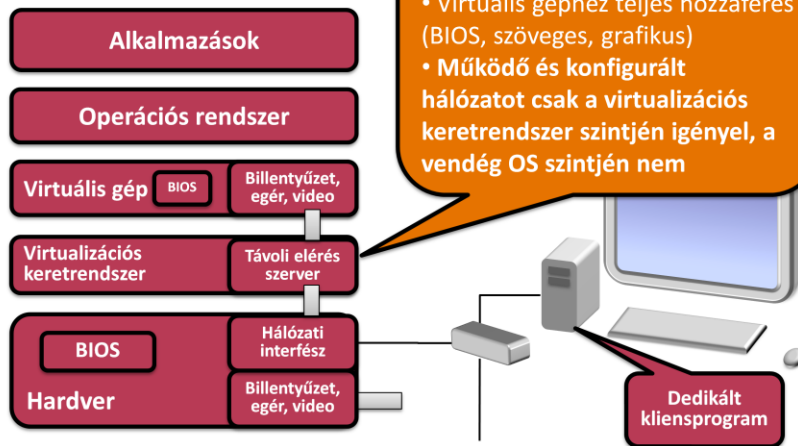
SUN ILO működés közben



SAS vezérlő beállításainak módosítása távolról

Távoli hozzáférés lehetőségek (2)

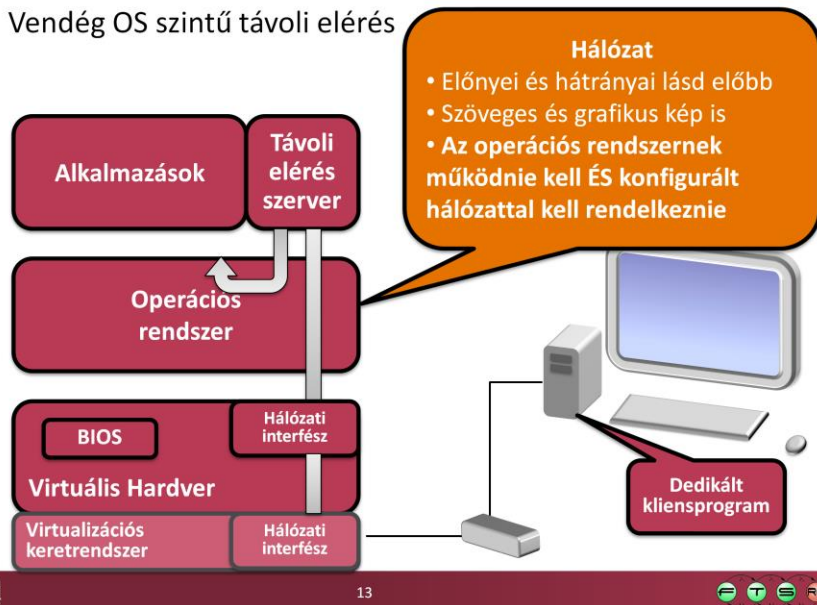
Virtuális gép virtuális konzolja



- Ebben az esetben már a virtualizációs keretrendszer biztosítja azt a funkciót, hogy a virtuális gépek „konzoljához” távolról hozzá tudjunk férni.
- Bare-metal és hosted esetben is gyakran van valamilyen távoli elérés szerver.

Távoli hozzáférés lehetőségek (3)

Vendég OS szintű távoli elérés



- A harmadik lehetőség, hogy ha már a vendég OS működik és konfigurált hálózata van, akkor arra lehet belépni SSH-val, VNC-vel, RDP-vel.
- Az ábrát úgy kell nézni, hogy ez egy vendég gép, ami magának biztosítja hozzáférést a beépített távoli elérés szerverével.

Tartalom

- Jellegzetességek
- Távoli hozzáférés megvalósítása
- **Háttértárak virtualizációja**
- VMware ESXi architektúra
- Microsoft Hyper-V architektúra
- Xen architektúra
- Erőforrás-gazdálkodás

Nagy integrált rendszerek kiépítése

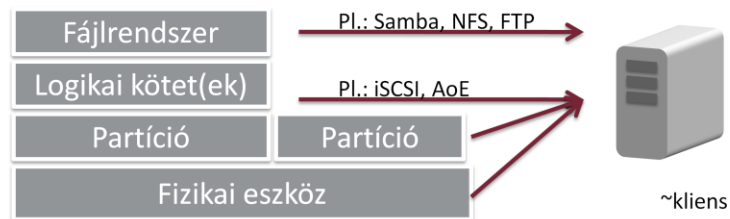
- Szerver virtualizációnál egy fizikai gép ritkán elég
- Jó, ha egyes erőforrások:
 - központilag kezeltek
 - átcsoportosíthatóak, hozzárendelések megváltoztathatóak
 - megoszthatóak, több helyről is egyformán elérhetőek
- Példák
 - háttértár: sok kicsi, gépekben elszórt merevlemez helyett egy-egy nagy tároló alrendszer (storage system/disk array)
 - hálózatok: minden hoszton egyformán elérhetőek legyenek
 - magasabb szinten: jogosultságkezelés, felhasználói fiókok címtára



Erről még később fogunk több szót ejteni.

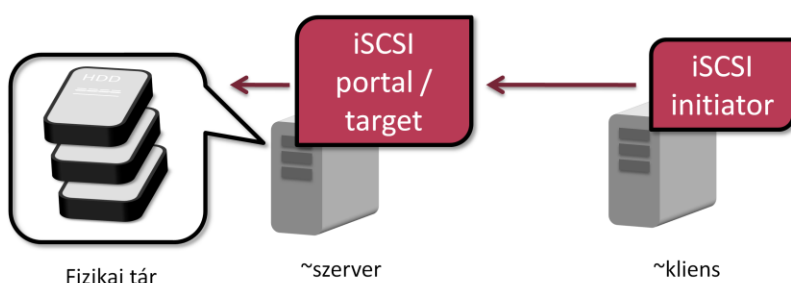
Háttértár virtualizáció

- Elterjedt megoldások:
 - SAN – Storage Area Network, blokkos eszközt biztosít
 - NAS – Network Attached Storage, fájlrendszert biztosít
- Példa: Egyszerűsített háttértár stack



iSCSI – SAN megoldás

- Blokkos eszköz biztosítása TCP/IP hálózatok felett
- Fájlrendszernek támogatnia kell a több gépről történő elérést
 - Pl. VMFS, GFS fájlrendszer



Fürtözhető fájlrendszerek példákat lásd még:

http://en.wikipedia.org/wiki/Shared_disk_file_system

- Az iSCSI initatort futtató gép ugyanúgy tudja kezelni a virtuális tárat, mintha az egy lokális fizikai eszköz lenne.
- Szerver oldalon a hierarchia:
 - Egy szerver egy portal-t szolgáltat (ez azonosítható a szerver IP címével / nevével)
 - Egy portalon belül lehet több target (ezek különböző jogosultságokkal szétválaszthatóak)
 - Egy target több különálló kötetet (volume) is szolgáltat, és egyszerre több initatort is ki tud szolgálni (akár egy kötetet is többen használhatnak).
- iSCSI initiatorból van szoftveres implementáció és van hardveres is. Ez utóbbi bootolni is tud távoli eszközről.

Tartalom

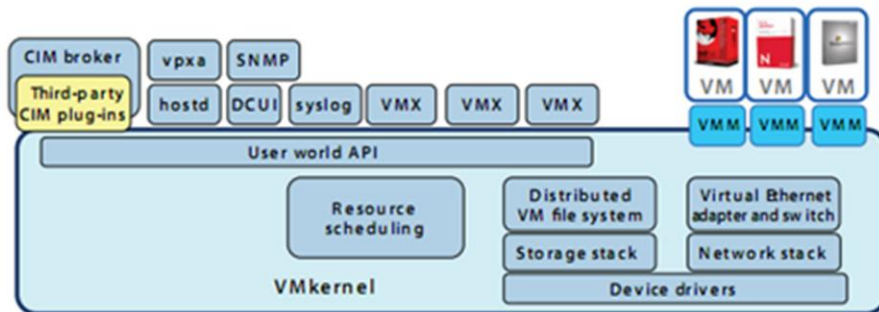
- Követelmények
- Szerver virtualizációs architektúrák
- Háttértárak virtualizációja
- **VMware ESXi architektúra**
- Microsoft Hyper-V architektúra
- Xen architektúra
- Erőforrás-gazdálkodás

VMware ESXi

- VMware bare-metal megoldása
 - Követelmény: 64 bites CPU
 - (Van ingyenes verziója is)
- VMware ESX utódja (új architektúra)
- vSphere 5 család része
 - vCenter Server, vMotion, DRS, HA, FT...
 - lásd a későbbi előadást

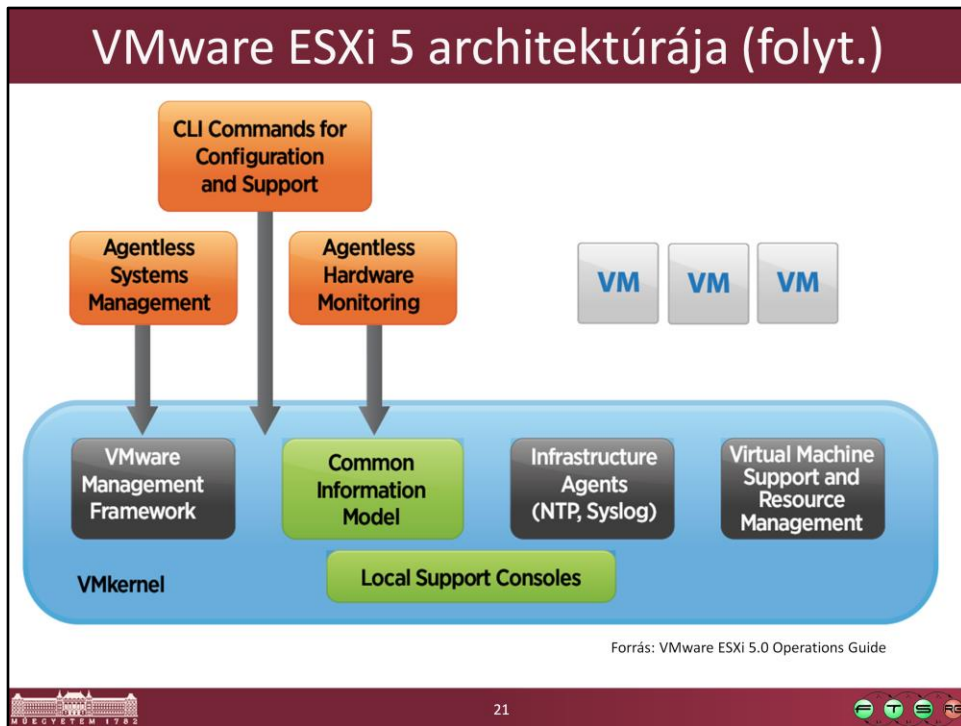


VMware ESXi 5 architektúrája

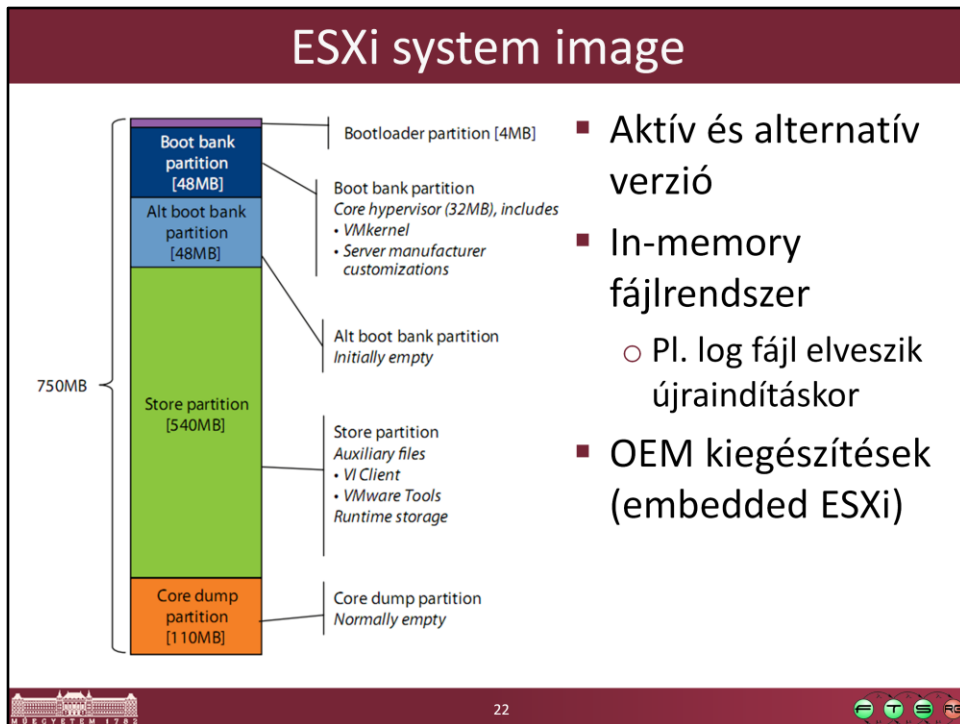


- VMkernel
- World
 - VM world
 - Ágensek, shell...

- Az ábrán látszik részletesebben, hogy milyen funkciókat kell a VMkernelnek ellátnia.
- A VMkernel úgynevezett worldöket futtat (~ ez felel meg kb. a folyamatnak). Ezek lehetnek ágensek, helyi shell, menedzsment eszközök vagy pedig virtuális gépek és segédfolyamataik.



- Forrás: VMware, **VMware ESXi 5.0 Operations Guide**, Technical white paper, 2011.
URL: <http://www.vmware.com/resources/techresources/10215>
- Menedzsment APIk
 - helyi és távoli parancssoros felület (CLI)
 - Web Service alapú API programozott eléréshez (Java, .NET, PowerShell...)
 - CIM alapú HW felügyelet



Forrás: VMware, The Architecture of VMware ESXi, White Paper, URL: <http://www.vmware.com/resources/techresources/1009>

Fontos ötletek:

- Dual boot bank: hibás frissítés esetén csak újra kell indítani, és visszaállhatunk a régihez
- Egyben frissíthető az egész, nem kell frissítések függőségeit keresni, „firmware”-szerűen cserélhető az egész
- Ehhez kell az in-memory fájlrendszer is, a konkrét beállítások egy state.tgz fájlban tárolódnak.

A felépítésről részletesen lásd:

- Micskei Zoltán. ESXi 4.1 felépítése és partíciói. URL: <https://micskeiz.wordpress.com/2010/11/25/esxi-4-1-felptse-s-partcii/>
- Technikai részletek: Olivier Cremel. VisorFS: A Special-purpose File System for Efficient Handling of System Images. VMware Technical Journal, Issue 1, 2012. URL: <http://labs.vmware.com/publications/cremel-vmtj-spring2012>

Tartalom

- Követelmények
- Szerver virtualizációs architektúrák
- Háttértárak virtualizációja
- VMware ESX és ESXi szerver architektúrája
- **Microsoft Hyper-V architektúra**
- Xen architektúra
- Erőforrás-gazdálkodás

Microsoft Hyper-V

- Microsoft bare-metal virtualizációs megoldása
 - Jelenleg: 3. verzió béta (Windows Server 2012)
- Két változat:
 - Windows Serverben a Hyper-V szerep
 - MS Hyper-V Server (különálló, ingyenes, csak Hyper-V)
- HW igény:
 - CPU: 64 bites, HW-es virtualizációs támogatás



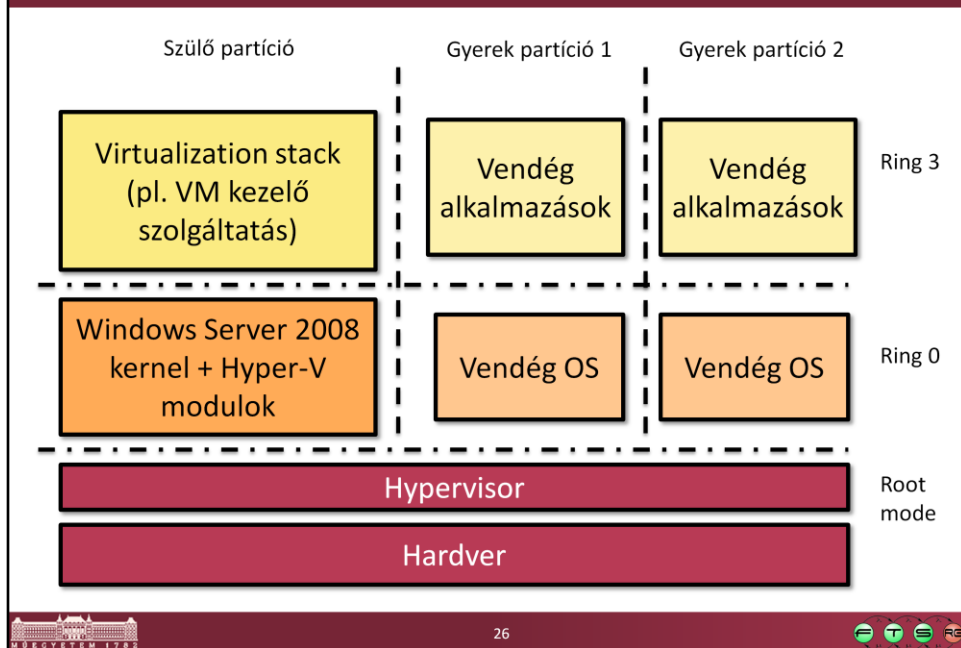
Microsoft Hyper-V v3 újdonságai

- Több párhuzamos „Live migration” folyamat
- „Live migration” osztott tárhely nélkül
- Resource pool kezelés
- Adat de-duplikáció támogatása
- Virtuális gépeknek 64 CPU és 1 TB RAM
- Új virtuális lemezkép típus támogatása (16 TB-ig skálázható)

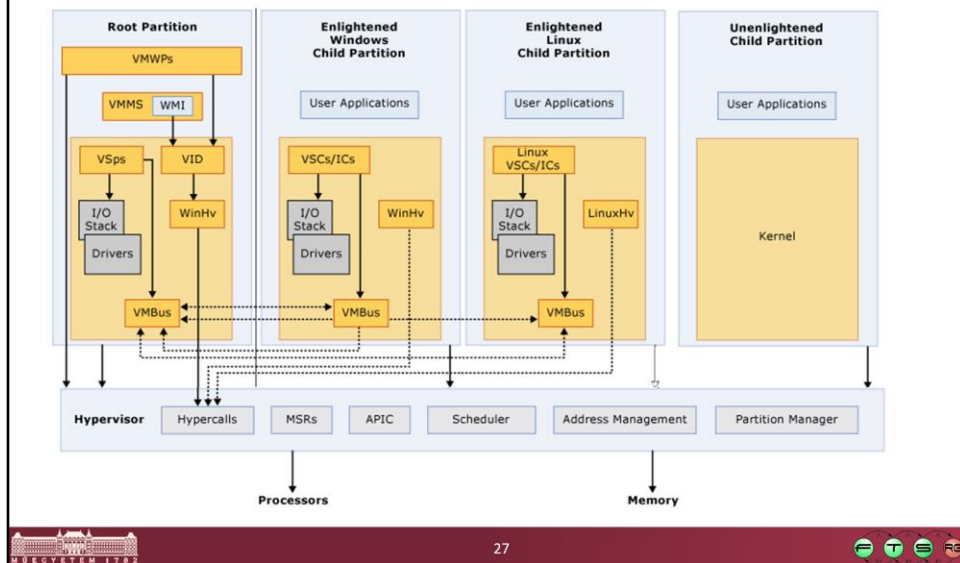


Részleteket lásd: Microsoft. What's New in Hyper-V, URL:
<http://technet.microsoft.com/en-us/library/hh831410.aspx>

Hyper-V architektúra I



Hyper-V architektúra II



Forrás: Appendix B: Hyper-V Architecture and Feature Overview
[http://msdn.microsoft.com/en-us/library/dd722833\(BTS.10\).aspx](http://msdn.microsoft.com/en-us/library/dd722833(BTS.10).aspx)

„Acronyms and terms used in the diagram above are described below:

- **APIC** – Advanced Programmable Interrupt Controller. A device which allows priority levels to be assigned to its interrupt outputs.
- **Child Partition** – Partition that hosts a guest operating system. All access to physical memory and devices by a child partition is provided via the Virtual Machine Bus (VMBus) or the hypervisor.
- **Hypercall** – Interface for communication with the hypervisor. The hypercall interface accommodates access to the optimizations provided by the hypervisor.
- **Hypervisor** – A layer of software that sits between the hardware and one or more operating systems. Its primary job is to provide isolated execution environments called partitions. The hypervisor controls and arbitrates access to the underlying hardware.
- **IC** – Integration component. Component that allows child partitions to communicate with other partitions and the hypervisor.
- **I/O stack** – Input/output stack.
- **MSR** – Memory Service Routine.
- **Root Partition** – Manages machine-level functions such as device drivers, power management, and device hot addition/removal. The root (or parent) partition is the only partition that has direct access to physical memory and devices.
- **VID** – Virtualization Infrastructure Driver. Provides partition management services, virtual processor management services, and memory management services for partitions.
- **VMBus** – Channel-based communication mechanism used for inter-partition communication and device enumeration on systems with multiple active virtualized partitions. The VMBus is installed with Hyper-V Integration Services.
- **VMMS** – Virtual Machine Management Service. Responsible for managing the state of all virtual machines in child partitions.
- **VMWP** – Virtual Machine Worker Process. A user mode component of the virtualization stack. The worker process provides virtual machine management services from the Windows Server 2008 instance in the parent partition to the guest operating systems in the child partitions. The Virtual Machine Management Service spawns a separate worker process for each running virtual machine.
- **VSC** – Virtualization Service Client. A synthetic device instance that resides in a child partition. VSCs utilize hardware resources that are provided by Virtualization Service Providers (VSPs) in the parent partition. They communicate with the corresponding VSPs in the parent partition over the VMBus to satisfy a child partitions device I/O requests.
- **VSP** – Virtualization Service Provider. Resides in the root partition and provide synthetic device support to child partitions over the Virtual Machine Bus (VMBus).
- **WinHv** – Windows Hypervisor Interface Library. WinHv is essentially a bridge between a partitioned operating system's drivers and the hypervisor which allows drivers to call the hypervisor using standard Windows calling conventions.
- **WMI** – The Virtual Machine Management Service exposes a set of Windows Management Instrumentation (WMI)-based APIs for managing and controlling virtual machines.”

Tartalom

- Követelmények
- Szerver virtualizációs architektúrák
- Háttértárak virtualizációja
- VMware ESX és ESXi szerver architektúrája
- Microsoft Hyper-V architektúra
- **Xen architektúra**
- Erőforrás-gazdálkodás

Xen

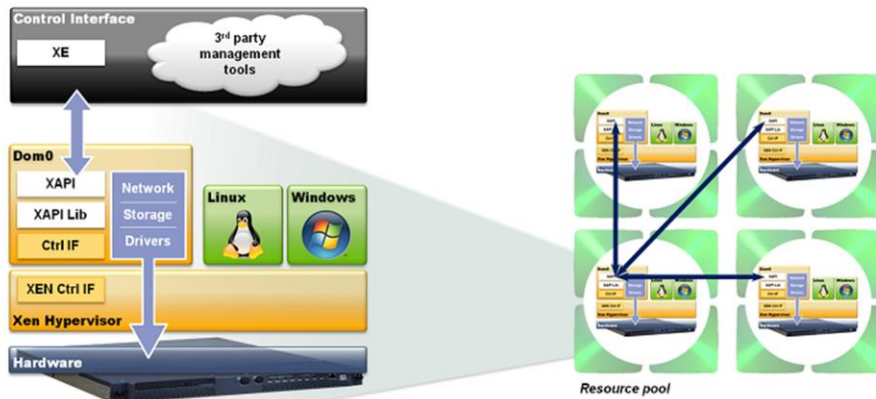
- University of Cambridge kutatási projekt 
- Jelenleg:
 - Xen.org: nyílt forráskód, sok disztribúcióban elérhető
 - Citrix XenServer: plusz funkciók, fizetős (is)
 - Xen Cloud Platform (XCP): XenServer nyílt változata
 - Oracle VM, HUAWEI UV...
- Követelmény:
 - Paravirtualizációs kiegészítés része a Linux kernelnek
 - Windows vendéghez HW-es virtualizáció kell



- Lásd: Paul Barham et al. (2003) Xen and the art of virtualization. In *Proceedings of the nineteenth ACM symposium on Operating systems principles* (SOSP '03). ACM, New York, NY, USA, 164-177. URL: <http://doi.acm.org/10.1145/945445.945462>
- Wim Coekaerts Blog. Linux mainline contains all the Xen code bits for Dom0 and DomU support, May 26, 2011, URL: https://blogs.oracle.com/wim/entry/linux_mainline_contains_all_the

XCP architektúra

- Dom0: menedzsment OS
- DomU: virtuális gépek



Forrás: <http://xen.org/products/cloudxen.html>

XCP tervek: Domain 0 Disaggregation

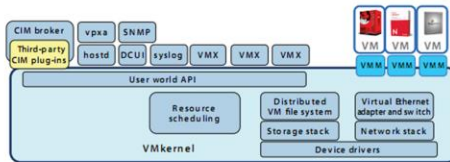
- Domain 0 szétválasztása
 - Külön Driver, Stub és Service Domain
 - Külön-külön újraindíthatók
 - Privilegiumok szűkebb kiosztása, jobb biztonság
- XCP 2.0-ra ígérik (2013)
- Qubes OS hasonlót használ a hálózatra
(lásd <http://qubes-os.org/Architecture.html>)

Forrás: John Garbutt, Xen Cloud Platform Update, XenSummit 2012



Bare metal megoldások architektúrái

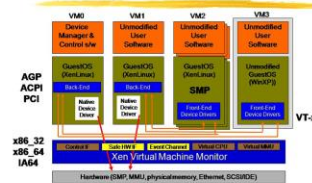
ESXi



- I/O eszközöket is a hypervisor kezeli
- Meghajtókat (driver) a VMware szállítja
- Extra kis méret: ESXi (32 MB)

Xen / Hyper-V

Xen 3.0 Architecture



- I/O eszközök kezelése a szülő partícióban
- Meghajtókat a HW gyártók szállítják

Tartalom

- Követelmények
- Szerver virtualizációs architektúrák
- Háttértárak virtualizációja
- VMware ESX és ESXi szerver architektúra
- Microsoft Hyper-V architektúra
- Xen architektúra
- **Erőforrás-gazdálkodás**

Erőforrás-gazdálkodás

- A virtuális gépek közös erőforráson osztoznak
- Jellemző példák:
 - CPU: gyakran (összesen több vCPU, mint fizikai)
 - Memória: ritkábban (memory overcommit)
 - Háttértár I/O műveletek: itt jellegzetesen osztozás van!
 - Hálózati áteresztőképesség: itt is osztozás van



Versengés az erőforrásokért

- Erőforrás szűk keresztmetszet lesz
 - Kis terheléseknél ritka
 - De szerverkörnyezetben gyakran előfordul
- Hogyan osszuk el ilyenkor az erőforrásokat?



Feladatok erőforrások kezelésénél

- Tipikus igények:
 - Korlátozni valakinek a felhasználást
 - Garantálni valakinek a minimumot
 - Prioritás versenyhelyzet esetén
- Megoldások:
 - Kemény korlátozások, „lággy” korlátok, részesedés
 - „Proportional-Share Based Scheduler”

Szabályozási lehetőségek (VMware)

- **Resource Limit** – kemény felső korlát az erőforrás igénybevételére
 - Akkor is érvényes, ha egyébként van szabad erőforrás
- **Resource Reservation** – garantált rendelkezésre álló erőforrás mennyiség
 - Nem feltétlenül használja ki, csak verseny esetén érvényesül, egyébként a keretet más használhatja
- **Resource Shares** – prioritás
 - Verseny esetén az alapértelmezett „igazságos” elosztás módosítható ezzel



További információ (ütemező leírása, co-scheduling, „CPU topology aware load-balancing”, stb.):

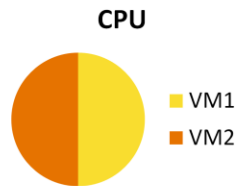
VMware. „VMware vSphere: The CPU Scheduler in VMware ESX 4.1”,
<http://www.vmware.com/resources/techresources/10131>

Példa a share használatára

- Több VM-et futtató gép esetén a CPU share értékek a következők

- VM1: 1000

- VM2: 1000



Forrás: Carl Waldspurger, Resource Management for Virtualized Systems



38



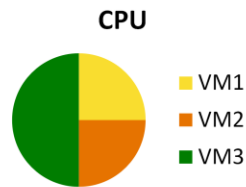
A példában most mindegyik virtuális gép egyprocesszoros, egyébként szorozni kéne a processzoraik számával.

A következő pár fólia VMware példáinak forrása: Forrás:
<http://labs.vmware.com/academic/basic-virtualization>

Példa a share használatára

- Több VM-et futtató gép esetén a CPU share értékek a következők

- VM1: 1000
- VM2: 1000
- VM3: 2000



- Versenyhelyzet esetén mennyi a VM1 részére biztosított erőforrás-mennyiség?
 - $1000 / (1000+1000+2000) = 1/4$ CPU idő

Erőforrás gazdálkodás



VM share módosítása
dinamikus újrakonfigurálás



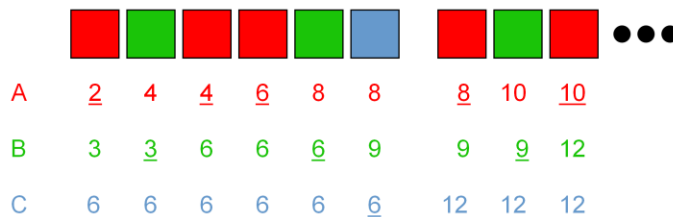
Új **VM** felvétele
Graceful degradation



VM eltávolítása
Extra erőforrások kihasználása

CPU ütemezés

- Simplified virtual-time algorithm
 - „Virtuális idő” = felhasználás / share
 - Ütemezés: Legkisebb „virtuális idejű” VM
- Példa: 3 VM A, B, C (3 : 2 : 1 share aránnyal)

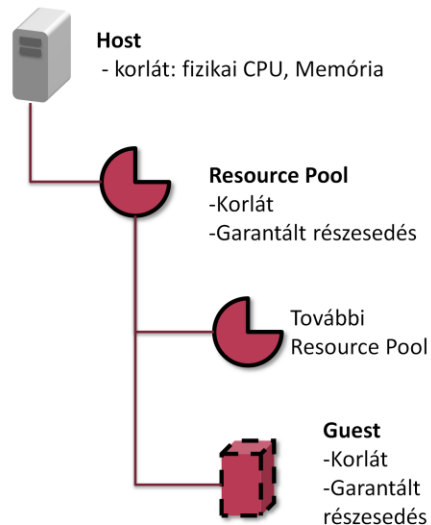


A „virtuális idő” (virtual time) bevezetésének az a lényege, hogy akinek több share-e van, annak ugyanannyi CPU-használatot kevesebbnek számol el.

Például az ábrán ha a C dolgozik 1 kockányit, akkor az neki 6 egységnek számít. Ha ugyanennyit dolgozik az A, akkor az neki csak harmadannyinak, azaz 2 egységnek számít (mert neki 3-szor annyi a share-je, mint C-nek). Hasonlóan B-nek ugyanennyi futást csak 3 egységnek számol el az ütemező. Így garantálható, hogy ha mindenki folyamatosan futni szeretne, akkor a részesedésüknek megfelelően kapnak az erőforrásból.

Hierarchikus erőforráskezelés

- Nemcsak virtuális gépek szintjén lehet korlátozni
- Pool-okba szervezhetők a VM-ek
- Használati eset példák:
 - Egy felhasználó összes gépére egy közös korlátozás
 - Egy feladatot ellátó gépek csoportjára korlát
 - Kritikus/nem kritikus alkalmazások csoportosítása



Hierarchikus erőforráskezelés

- Nemcsak virtuális gépek szintjén lehet korlátozni
- Pool-okba szervezhetők a VM-ek

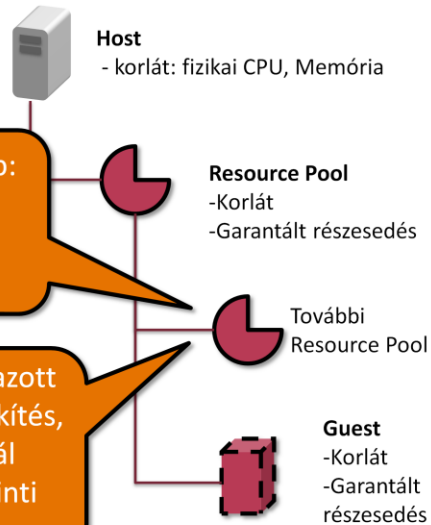
- Használati eset

- Egy felhasználó gépére egy közös
- Egy feladatot ellátó gépek csoportjára korlátozás
- Kritikus/nem kritikus alkalmazások csoportosítása

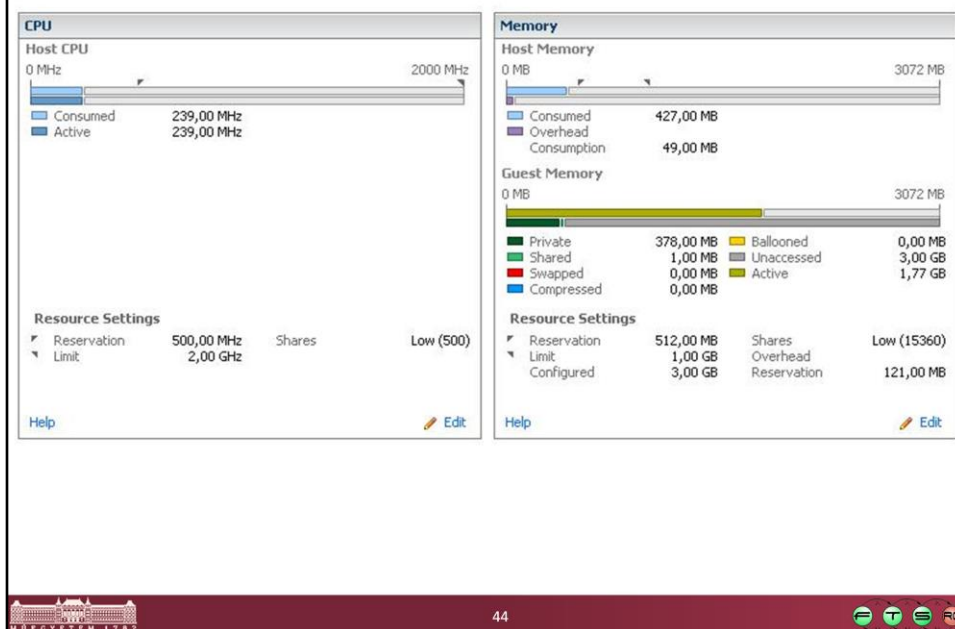
Korlátokat szab:

-Host
-Resource Pool
-Guest

Egymásba ágyazott korlátozásoknál szűkítés, konfliktusnál prioritás szerinti feloldás

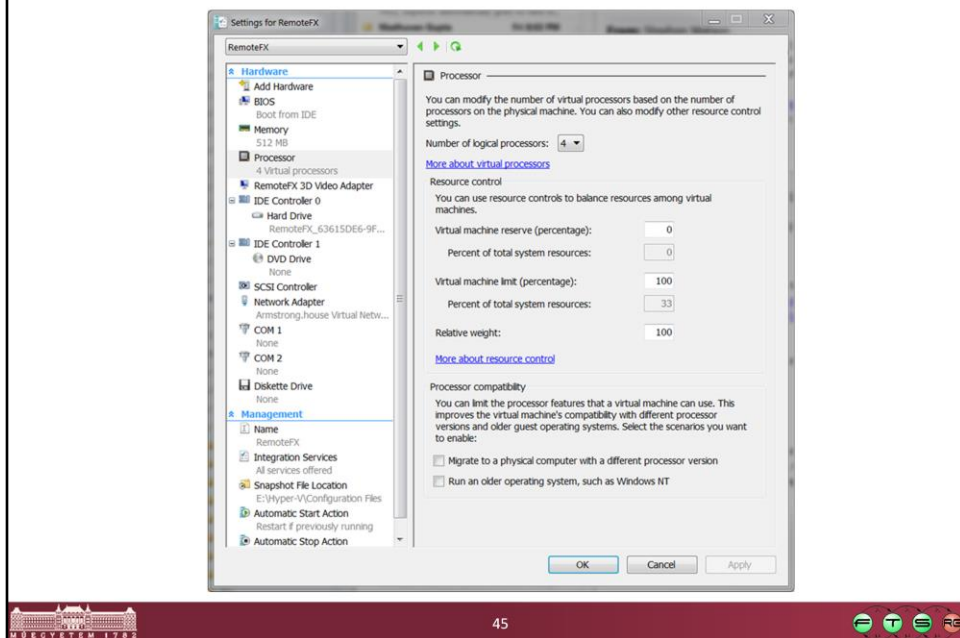


Erőforrás gazdálkodás: GUI felület



Az ábrán az ESXi felületén látszanak egy konkrét virtuális gépre beállított korlátozások.

Hyper-V: hasonló erőforrás-gazdálkodás



Pl. CPU esetén itt is Reserve / Limit / weight (share) van

Összefoglalás

- Követelmények
 - Sok guest, jó skálázhatóság, távoli elérés
- Szerver virtualizációs architektúrák
 - Dominánsan bare-metal
- Háttértárak virtualizációja
 - iSCSI SAN
- VMware ESX és ESXi szerver architektúrája
 - Mikrokernel, eltérés a Service Console megvalósításában
- Hyper-V architektúra

DEMO Gyakorlat: ESXi használata

- Indítás: alap konfiguráció (IP, jelszó) beállítása
- Csatlakozás VI Clienttel
- iSCSI storage beállítása datastore-nak

