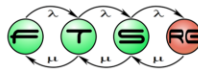


# Virtualizáció

Micskei Zoltán

<http://www.mit.bme.hu/~micskeiz>



Utolsó módosítás: 2012. 04. 16.

Az előadás magáncélra szabadon felhasználható. Köz- és felsőoktatásban felhasználható, csak előtte kérlek írj egy emailt nekem.

# Virtualizáció

- Számítástechnika központi fogalma
- **Virtualizáció:** erőforrás tényleges fizikai tulajdonságainak elrejtése a felhasználója előtt, pl.
  - egy erőforrást több logikaiként felajánlani,
  - több fizikai erőforrást összefogni egybe...
- Virtuális memória, virtuális fájlrendszerek...



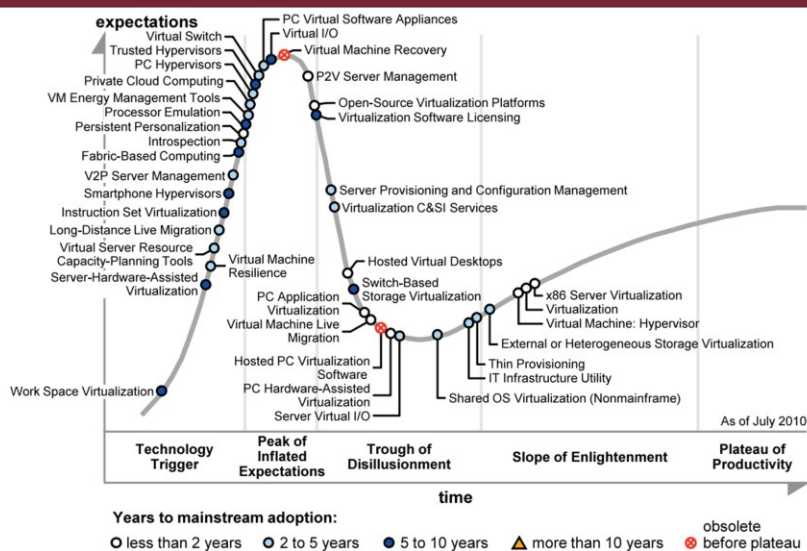
2



Ez egy kellően általános definíció, a virtualizáció lényege az absztrakció, azaz, hogy elfedi az alatta lévő réteg valamilyen jellemzőjét.

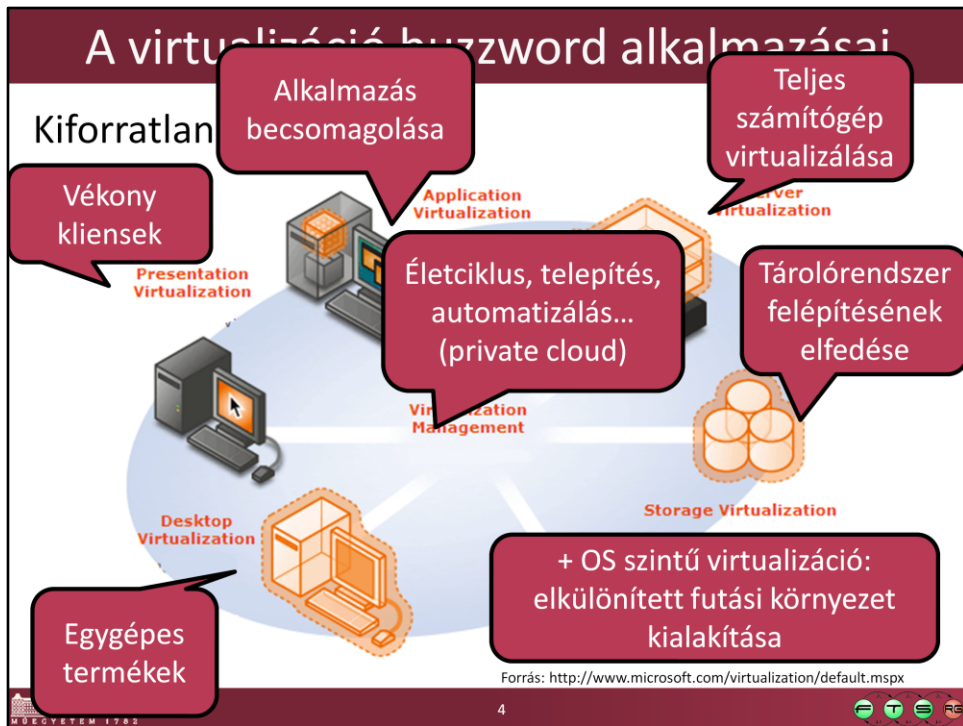
Eddig is találkoztunk már a tárgyban ezzel a fogalommal, de az előadás most a virtualizáció egy speciális fajtájával fog részletesebben foglalkozni.

# Gartner Hype Cycle for Virtualization 2010



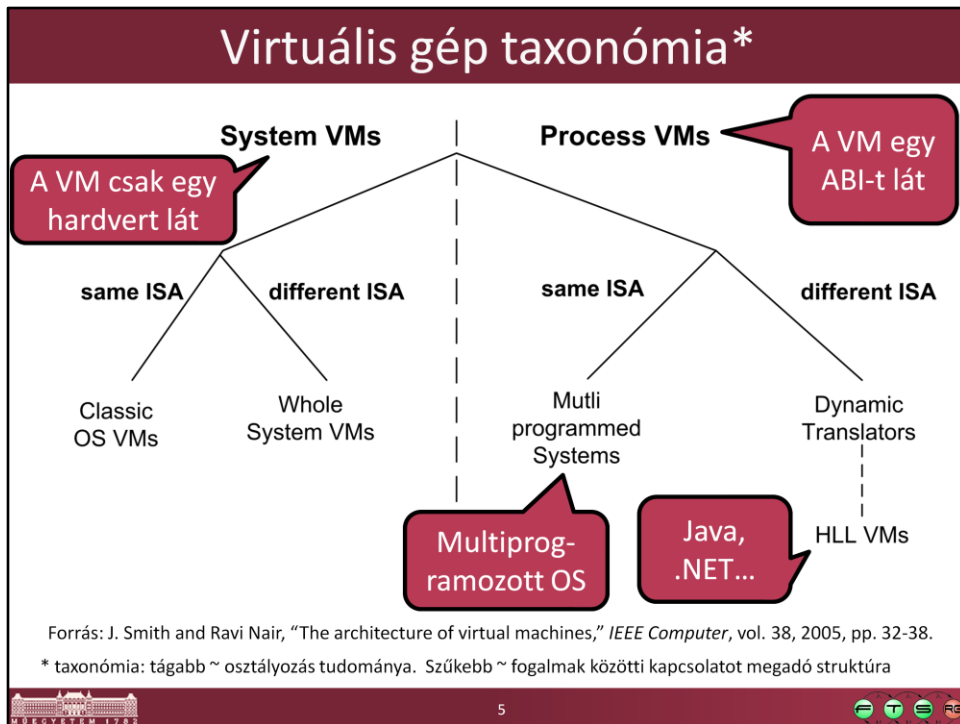
Source: Gartner (July 2010)

Forrás: Gartner Hype Cycle for Virtualization, 2010. Elérhető online: <http://virtualization.info/en/news/2010/09/paper-gartner-hype-cycle-for-virtualization-2010.html>



Ez a csoportosítás nagyon esetleges, minden gyártó máshogy állítja be az összképet. (Pl. Ez a kép a Microsofttól van, nekik nincsen operációs rendszer szintű virtualizációs megoldásuk, így ez nem is került rá a képre 😊.)

Nem egyértelmű még a terminológia, sok gyártó (akár szándékosan) máshogy használja az egyes elnevezéseket, mint a többiek.



Forrás: J. Smith and Ravi Nair, "The architecture of virtual machines," *IEEE Computer*, vol. 38, 2005, pp. 32-38. DOI: 10.1109/MC.2005.173, URL: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1430629](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1430629) (Egy kibővített változata elérhető itt: <http://ies.ece.wisc.edu/papers/vm.pdf>)

**Process VM:** „A process VM is a virtual platform that executes an individual process. This type of VM exists solely to support the process; it is created when the process is created and terminates when the process terminates.”

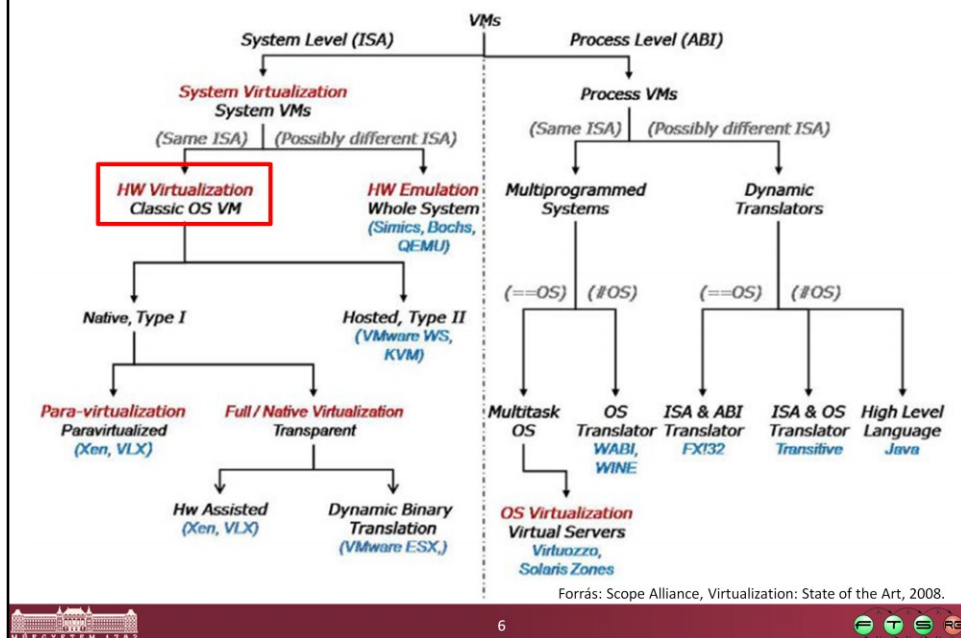
**System VM:** „A system VM provides a complete, persistent system environment that supports an operating system along with its many user processes. It provides the guest operating system with access to virtual hardware resources, including networking, I/O, and perhaps a graphical user interface along with a processor and memory.”

ISA: Instruction Set Architecture

ABI: Application Binary Interface

API: Application Programming Interface

# Virtuális gép taxonómia (részletesebb)



Forrás: Scope Alliance, Virtualization: State of the Art, 2008.

<http://scope-alliance.org/sites/default/files/documents/SCOPE-Virtualization-StateofTheArt-Version-1.0.pdf>

Az előző felosztás folytatása, több típussal. Itt már szerepelnek konkrét termékek is példának (kék színű feliratok).

## Platform virtualizáció

- **Platform virtualizáció:** teljes számítógép virtualizálása, egy gépen több OS futtatása
  - Elnevezés még: szerver, számítógép, hardver virtualizáció..
- **Elemek:**
  - *Gazda gép* (host machine) = fizikai gép
  - *Vendég gép* (guest machine) = virtuális gép
  - *Virtual Machine Monitor* (VMM): a virtuális gépeket kezelő program

## Platform virtualizáció története

- ~1960 - IBM CP-40 rendszere
  - virtualizáció a mainframe termékvonallal
- x86 virtualizáció
  - Sokáig lehetetlennek tűnt
  - 1997: Stanford, Disco projekt
  - 1998: VMware megoldás
  - 2000- További megoldások
- Jelenleg: külön iparág alakul



Kép forrása: IBM Mainframes reference room

[http://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe\\_room.html](http://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe_room.html)

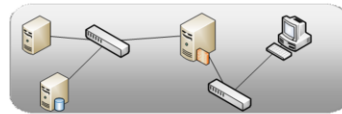
Az IBM CP-40 egy kísérleti rendszer volt, de aztán átkerült a technológia az éles rendszerekbe is. Mainframe-ek esetén tipikus volt, hogy 5-10 különböző operációs rendszert is futtatott egy gép, így magas volt a gépek kihasználtsága.

Az x86-os architektúra esetén hasonló módszerek alkalmazása sokáig lehetetlennek tűnt, lásd később.



## Miért jó a platform virtualizáció?

- Tesztrendszer kiépítése
- HW konszolidáció
- Régi rendszerek (legacy systems)
- On-demand architektúra
- Rendelkezésre állás, katasztrófa védelem
- Hordozható alkalmazások
- ...



9



- Tesztrendszer: fejlesztés esetén az éleshez hasonló rendszerben lehet a szoftvert tesztelni. Akár egy 2 GB-bal rendelkező asztali gépen is el lehet futtatni 2 szervert és 1-2 klienst.

- HW konszolidáció: alacsony kihasználtságú szervereket, amiket a rajtuk lévő egymással nem kompatibilis alkalmazások miatt nem lehet egy fizikai gépre rakni, összevonjuk egy gépre.

- Régi rendszerek: Csak DOS-on vagy valami régi UNIX-on futó alkalmazásnak se kell már 1 külön gépet fenntartani

- On-demand/dinamikus adatközpont: a szervereket egy nagy, közös erőforráskészletként kezeljük, és a virtuális gépeket az éppen aktuális terheltségüknek megfelelően osztjuk szét. Ha valakinek ideiglenesen több memória kell, akkor megkapja, és később el lehet venni tőle.

- Rendelkezésre állás: virtuális gépekből könnyű egy másodlagos rendszert összerakni, ami hiba esetén egyből átveheti az éles rendszer funkcióit

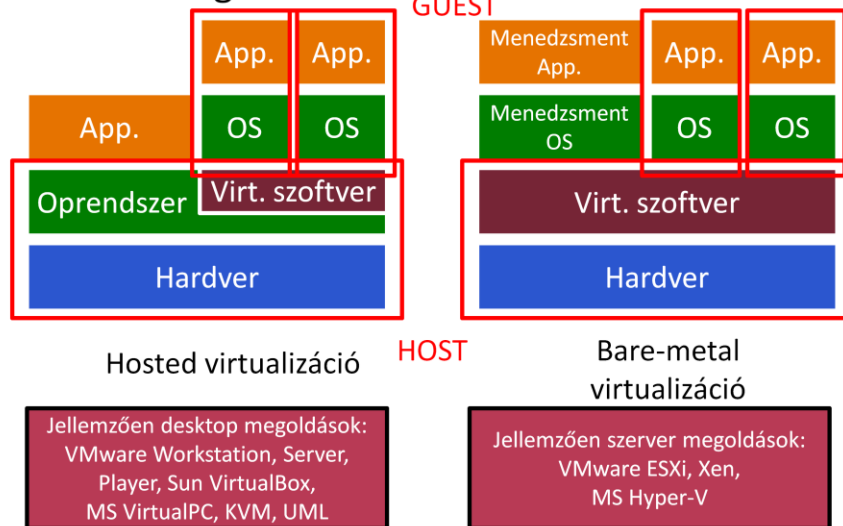
- Hordozható alkalmazás: az alkalmazást és az összes hozzá szükséges komponenst egy virtuális gépbe rakjuk, és azokat együtt tudjuk mozgatni.

-... van még egy csomó egyéb hasznos tulajdonsága is a virtualizációnak.

Azonban érdemes a negatívumokat is figyelembe venni, pl. így egy gép kiesése több virtuális számítógép hibáját okozza, az ilyen rendszerek biztonságával kapcsolatban mostanában egyre több kérdés merül fel, újra kell tanulni az ilyen rendszerek méretezését...

# Platform virtualizáció architektúrái

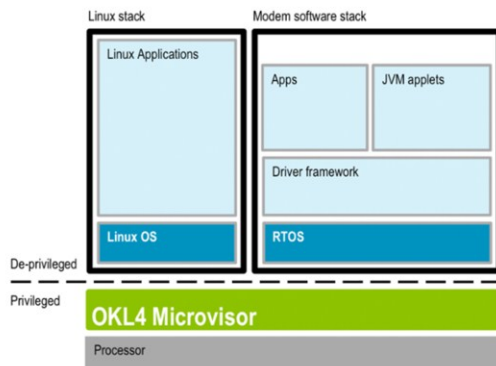
## ▪ Kétféle megközelítés:



- **FONTOS:** Bare-metal esetén a VMM kezeli a HW erőforrásokat, míg hosted típusú esetén ezt a host OS végzi.
- Azt a megkülönböztetést érdemes kerülni, hogy a VMM a hardveren fut vagy hogy hosted esetben van a gépen operációs rendszer is (mert tulajdonképpen bare-metal esetben a virtualizációs szoftver is egy speciális operációs rendszer).
- A Bare-metal esetben szokták a VMM-et hypervisornak nevezni, de itt is szoktak eltérések lenni az elnevezésben.
- Szokták még a Type I és Type II megkülönböztetést is használni, de ez nem annyira egyértelmű, a hosted és bare-metal felosztás szerintem szerencsésebb (lásd: Micskeiz Zoltán. Type I vs. Type II VMM. URL: <https://micskeiz.wordpress.com/2011/11/13/type-i-vs-type-ii-vmm/>).

# Új terület: mobil virtualizáció

## Funkciók elkülönítése



## Bring Your Own Device (BYOD)



## Képek forrása:

- Bal oldal: Open Kernel Labs. What is Mobile Virtualization? URL: <http://www.ok-labs.com/solutions/what-is-mobile-phone-virtualization>
- Jobb oldal: Mobiputing. VMware and LG introduce mobile virtualization tech for smartphones. URL: <http://mobiputing.com/2010/12/vmware/>

## Virtualizáció felhasználása mobiltelefonok esetén:

- (Bal) Például két független virtuális gépet futtatunk a telefonon, az egyikben futhat egy általános célú OS, amiben könnyű GUI alkalmazásokat készíteni, a másikban pedig futhat egy RTOS a telefonálás funkció ellátására.
- (Jobb) Bring Your Own Device: felhasználók használhatják a saját személyes telefonjukat, és azon a céges alkalmazások egy elkülönített, központilag menedzselt virtuális gépben futnak.

- Seamless window mód
- Állapot elmentése
- P2V konverzió

# Elméleti alapok



## Elméleti alapok - Követelmények

Követelmények egy virtualizációs megoldástól:

- **Azonosság:** a virtuális gépen futtatott programok ugyanazt az eredményt adják
- **Biztonságosság:** a VMM kezeli az összes hardver erőforrást
- **Hatékonyság:** a vendég gép utasításainak nagy része beavatkozás nélkül fut

*Gerald J. Popek, Robert P. Goldberg: Formal Requirements for Virtualizable Third Generation Architectures. Commun. ACM 17(7): 412-421 (1974)*



Ezt a három követelményt a 70-es években fogalmazták meg az akkori megoldások kapcsán, de még nagyrészt ma is érvényesek, és sokan erre hivatkoznak.

## Alapvető probléma

- Vendég gépektől védeni kell a rendszert
- Pl.: HLT (Halt) utasítás kiadása
  - Elvárt: csak a vendég álljon le
  - Ha végrehajtanánk: mindenki leáll
- Megoldás: VMM felügyelje a vendég utasításait
  - Privilegizált utasítások kezelése



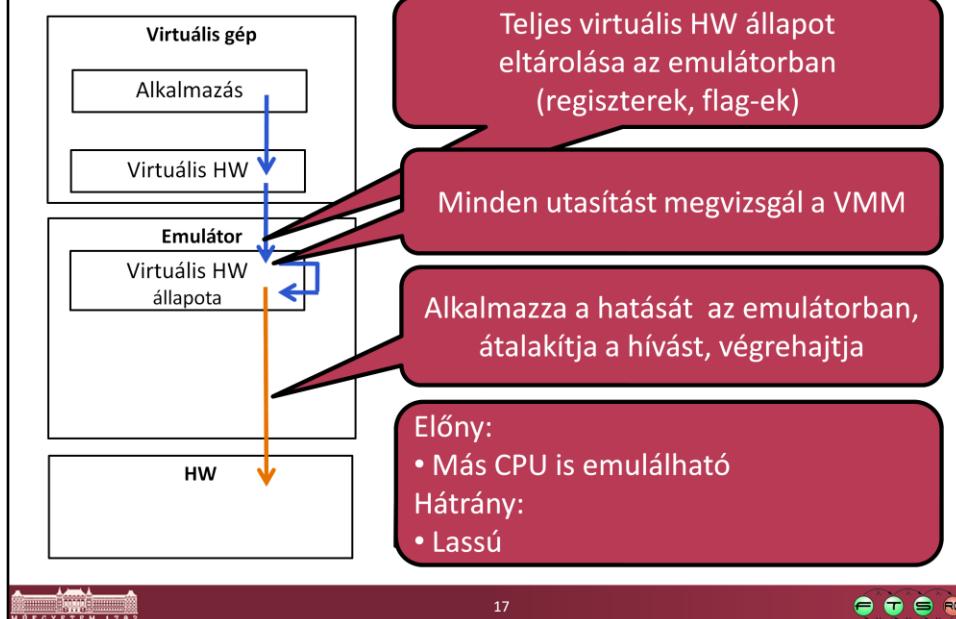
Privilegizált utasítás lehet olyan is, ami valami érzékeny regisztert vagy flaget módosít.

## Elméleti alapok

- CPU virtualizáció
- Memória virtualizáció
- I/O virtualizáció

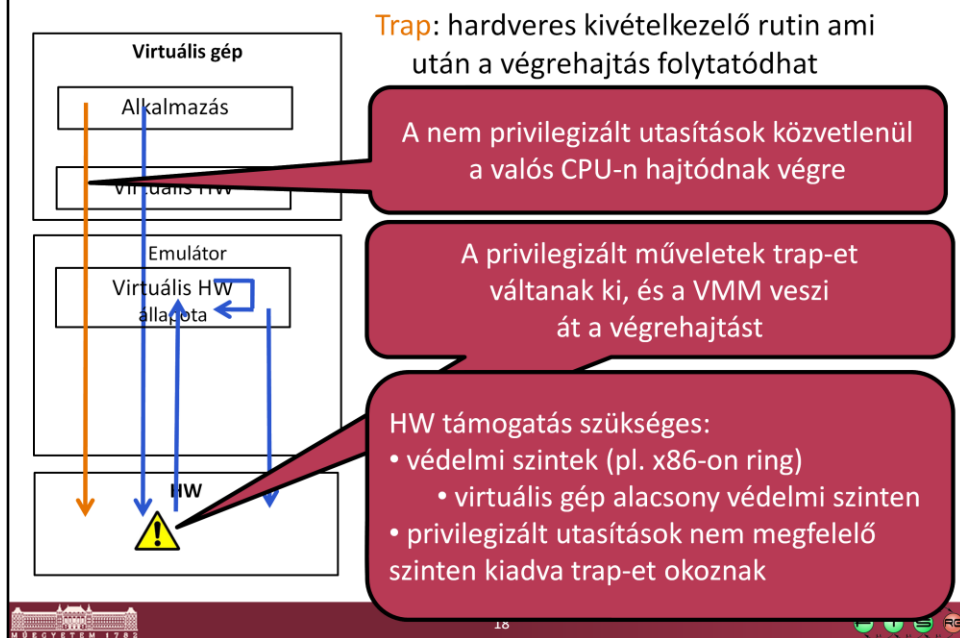


## Alapvető módszerek – Tiszta emuláció



- Vigyázat: Az még emuláció és nem virtualizáció! A kettő közötti egyik különbség pont az, hogy virtualizáció esetén az utasítások nagy része beavatkozás nélkül futhat.
- (Vigyázat: Az ábra elnagyolt, pl. a vendég gépben vagy a fizikai gépen futó OS nem szerepel rajta.)

## Alapvető módszerek – Trap and emulate



A módszer elve nagy vonalakban:

- Non-sensitive, unprivileged application instructions can be executed directly on the processor with no VMM intervention.
- Sensitive, privileged instructions will be detected when they trap after being executed in user mode. The trap should be delivered to the VMM that will emulate the expected behavior of the instruction in software.
- Sensitive, unprivileged instructions must be detected so that control can be transferred to the VMM.

Ehhez szükséges:

- Virtuális gép alacsonyabb védelmi szinten futtatása. Ezt nevezik deprivilegizálásnak (de-privileging).
- Különböző védelmi szintek a CPU-ban. x86 esetén ez az úgynevezett ring: 4 darab van, 0 a legmagasabb prioritású (itt futnak általában az OS-ek kerneljei), 3 a legalacsonyabb (itt futnak az alkalmazások).

## x86 virtualizáció korlátai

- Egyes architektúrák könnyen virtualizálhatóak, az x86 nem ilyen volt
- ~250 utasításból 17 megsérti a klasszikus feltételeket, pl.
- POPF utasítás: EFLAGS regisztert módosítja
  - Ha nem ring 0-n adjuk ki, akkor nem ír felül bizonyos biteket, és nem is dob kivételt
- Privilegizált állapot kiolvasható
  - Virtuális megtudhatja, hogy virtualizált

**Következmény:** nem használható a trap & emulate módszer a klasszikus x86 esetében

Lásd még: J. S. Robin and C. E. Irvine. Analysis of the Intel Pentium's ability to support a secure virtual machine monitor. In *Proceedings of the 9<sup>th</sup> USENIX Security Symposium, Denver, CO, USA, pages 129.144, Aug. 2000.*

## Megoldások az x86 CPU virtualizációra

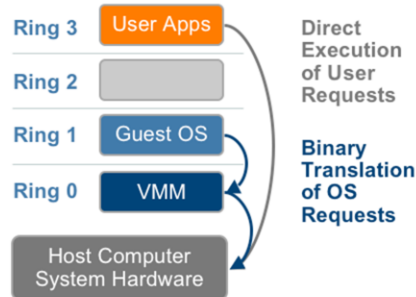
- Binary translation (szoftveres)
- Paravirtualizáció
- Hardveres virtualizáció



Ez egyben időbeli sorrend is, így jelentek meg ezek a módszerek.

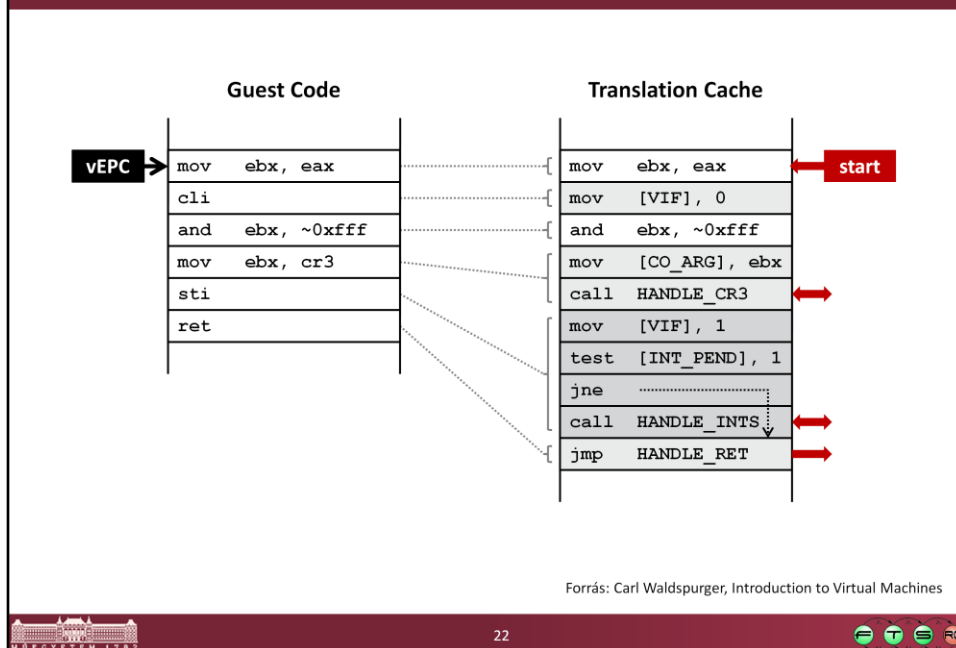
# Binary translation

- utasítások nagy része közvetlenül fut
- privilegizált utasítások átírása **futás közben**
- nem igényel forráskódot
- átírt változatot eltárolja
- vendég OS nem tud arról, hogy virtualizált



- Kép forrása: VMware. „Understanding Full Virtualization, Paravirtualization, and Hardware Assisted Virtualization”, URL: [http://www.vmware.com/files/pdf/VMware\\_paravirtualization.pdf](http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf)
- Binary translation: A VMM megvizsgálja az utasításokat, és amivel probléma lehet, azt átalakítja, így a CPU-hoz már csak a biztonságos utasítások jutnak el.
- Ez alapvetően a VMware megoldásának működése, a Microsoft VirtualPC-nél bár hasonlóak az elvek, de kicsit más a megvalósítás.
  - Lásd: Lepénye Tamás. „Virtual PC / Virtual Server virtualizációs technikák”, URL: <http://lepenyet.wordpress.com/2008/05/20/virtual-pc-virtual-server-virtualizacios-technikak/>

## Binary translation – példa



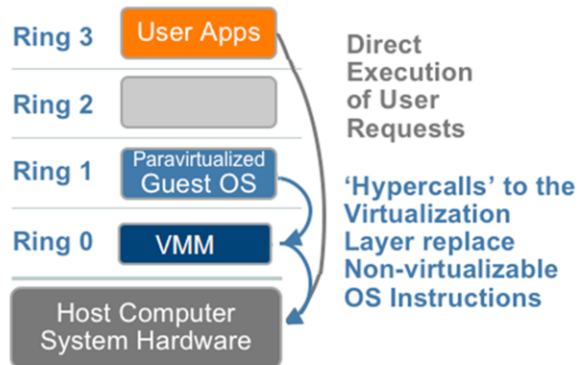
(Példa forrása: Carl Waldspurger, Kenn Bar, & Ravi Soundararajan. Introduction to Virtual Machines, URL: <http://labs.vmware.com/academic/mit-iap-2010>)

A vendégben futó kód:

- Az első utasítás egyik regisztert mozgat a másikba, ezzel nincs gond
- cli (clear interrupt flag): nem engedhetjük meg, hogy a fizikai gépen tiltsa le a megszakításokat, így ehelyett egy más utasítást hajtunk végre. Itt csak in-line módon lecseréljük egy másik utasításra.
- A cr3 regiszter a laptábla kezdőcímét jelzi, így a virtuális gépben futó kód ezt sem módosíthatja. Itt az utasítás lecserélésén túl a VMM egy külön rutinját is meghívjuk, ami ezt lekezeli (HANDLE\_CR3).
- sti (set interrupt): szintén nem engedhető meg, úgyhogy ehelyett is mást kell végrehajtani, de itt már le kell kezelni az esetleges függő IT-eket is.

# Paravirtualizáció

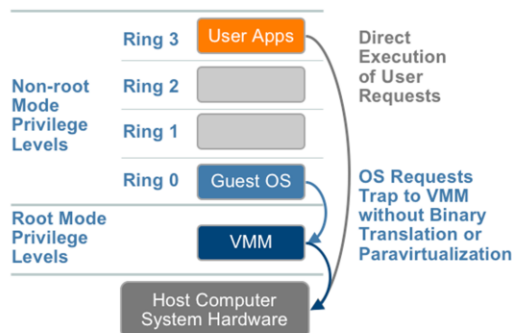
- Vendég OS forrásának módosítása
- Problémás utasítások lecserélése
- Hypercall: VMM-et hívja közvetlen



Itt módosítjuk a vendég OS-t, hogy ne is akarjon problémás utasításokat hívni, hanem azok helyett azoknak megfelelő függvényeket hívjon a hypervisorban. Ezzel egyszerűsödik a hypervisor implementálása, viszont csak úgy működik a módszer, ha módosítják a vendég OS forrását (a Linux kernelbe már bekerültek ilyen módosítások, és a Vista/Windows Server 2008-ba is került hasonló funkcionalitás).

## Hardveres virtualizáció

- ~2005: Intel Virtualization Technology (VT-x) és AMD AMD-V
- HW-es támogatás: root mode, VMCS
  - Utasítások, pl.: VMCALL, VMLAUNCH
- Működik a trap & emulate módszer



Intel VT-x elemei például:

- VMCS (Virtual Machine Control Structure): CPU állapotát tartalmazó struktúra, minden egyes új virtuális géphez készül egy ilyen. VM entry esetén a megfelelő VMCS visszatöltődik.
- VMLAUNCH Launches a virtual machine managed by the VMCS. A VM entry occurs, transferring control to the VM.
- VMCALL Allows a guest in VMX non-root operation to call the VMM for service. A VM exit occurs, transferring control to the VMM.

További információ:

- Intel® Virtualization Technology: Hardware Support for Efficient Processor Virtualization, Intel Technology Journal, Volume 10, Issue 03, <http://www.intel.com/technology/itj/2006/v10i3/1-hardware/1-abstract.htm>
- VT-x referencia: Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 1: Basic Architecture, <http://www.intel.com/products/processor/manuals/index.htm>
- Darvas Dániel, Horányi Gergő: Intel és AMD technológiák a hardveres virtualizáció megvalósítására, URL: <http://www.inf.mit.bme.hu/edu/homework/public/2010-%C5%91sz/intel-%C3%A9s-amd-technol%C3%B3gi%C3%A1k-hardveres-virtualiz%C3%A1ci%C3%B3-megval%C3%B3s%C3%ADt%C3%A1s%C3%A1ra>



## Melyik a legjobb/leggyorsabb módszer?

- Folyamatosan változik a válasz 😊
  - környezettől, terheléstől is függ
  - HW virtualizáció kezdetben kiforratlanabb, mint a BT
- Megoldások több módszert használnak vegyesen

	VMware ESX/ESXi	Microsoft Hyper-V	Xen
Szoftveres (BT)	+	-	-
Paravirtualizáció	- (már nem)	+ (részben)	+
Hardveres (Intel VT-x, AMD-V)	+	+	+



A kérdésre a válasz folyamatosan változik. Kezdetben csak BT volt, most nagyjából az még mindig felveszi a versenyt a többi megoldással, mert lassan 10 éve optimalizálják. Valamint ha régi OS-t akarunk futtatni, aminél nincs paravirtualizáció beépítve és nincs a fizikai gépben HW-es támogatás, akkor csak ez a módszer használható. Az is megfigyelhető, hogy lassan mindenki elkezdte használni a HW-es virtualizációt, mert egy idő után biztos gyorsabb lesz ez a megoldás, hisz 1 utasításból megcsinálja azt, ami szoftverből 50-100 kell.

Leírás, hogy melyik gyártó mikor mit használ: Micskei Zoltán. „Virtualizációs technológiák összehasonlítása”, 2011. november, URL:

<http://micskeiz.wordpress.com/2011/11/13/virtualizacios-technologiak-osszehasonlitasa/>

## Elméleti alapok

- CPU virtualizáció
- **Memória virtualizáció**
- I/O virtualizáció

## Ismétlés

Hogy néz ki a virtuális memória -> fizikai memória leképezés lapszervezés esetén?

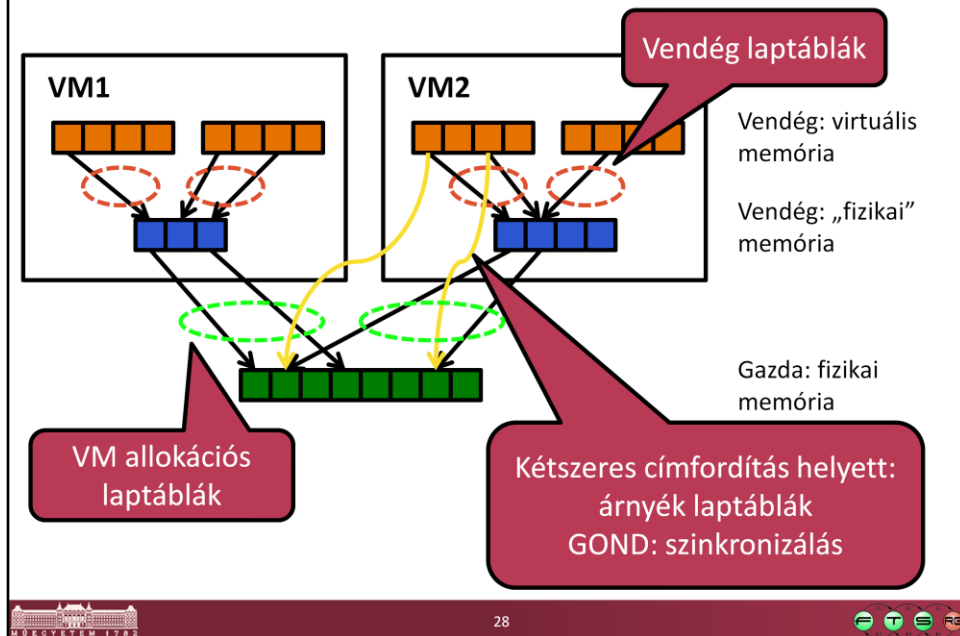


27



virtuális memória, lapozás, laptáblák, TLB

## Memória virtualizálása - szoftveres



Kétszeres címfordítás kell, így két féle laptáblákat kell fenntartani.

A gond az, hogy a régebbi hardverekben csak egy leképezéshez van támogatás, így fenntartanak egy harmadik féle struktúrát is (shadow page tables), ami közvetlenül vendég virtuális címeket képez le gazda fizikai címekké, és a TLB ennek az elemeit cache-eli. Így viszont a VMM-nek gondoskodnia kell arról, hogy az árnyék laptáblák szinkronban legyenek a vendég laptáblákkal. Figyelni kell a vendég OS minden memórialeképezés módosítási műveletét, ami költséges.

Bővebb információ a VMware-s megvalósításról: C.A. Waldspurger, "Memory resource management in VMware ESX server," *SIGOPS Oper. Syst. Rev.*, vol. 36, 2002, pp. 181-194. , URL: <http://www.waldspurger.org/carl/papers/esx-mem-osdi02.pdf>

## Memória virtualizálása - paravirtualizáció

- Árnyék laptáblák használata itt is
- Vendég OS forrásának módosítása
- Ha a vendég módosítja a laptábláit, akkor értesítse a VMM-et is erről

## Memória virtualizálása - hardveres

- HW támogatás az újabb CPU-kban
  - AMD Rapid Virtualization Indexing , Intel Extended Page Tables
- Beágyazott laptábla (Nested page table)
  - vendég fizikai -> gazda fizikai leképezés eltárolása
  - cím leképezési rutin ezt is bejárja
- TLB bejegyzések azonosítóval ellátása
  
- Nagy teljesítménynövekedés:
  - 2008. 04., KVM: [MMU paravirtualization is dead](#)
  - 2009., VMware: [Performance Evaluation of AMD RVI Hardware Assist](#), akár 42%-os növekedés is

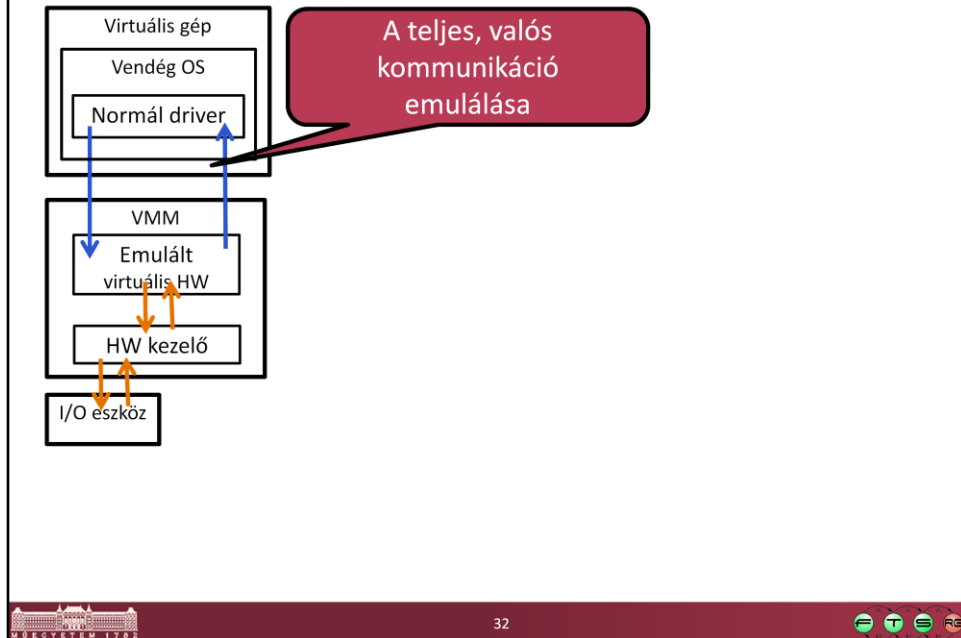


Az AMD RVI-ről leírás: AMD. „AMD-V™ Nested Paging”, July 2008. URL: <http://developer.amd.com/assets/NPT-WP-1%201-final-TM.pdf> (ebben van a szoftveres memória virtualizálásról is egy jó összefoglaló)

## Elméleti alapok

- CPU virtualizáció
- Memória virtualizáció
- **I/O virtualizáció**

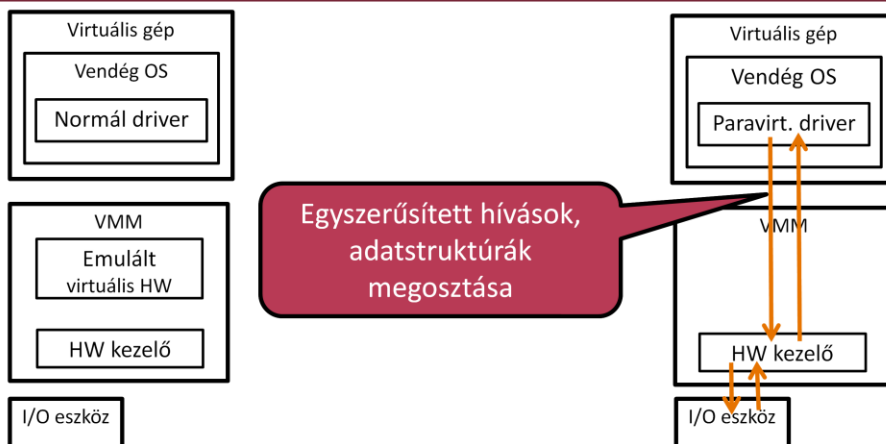
## I/O eszközök kezelése (szoftveres)



- Ilyenkor egy létező I/O eszközt emulál a VMM, a vendég az ehhez való normál eszközmeghajtót használja. Valami nagyon elterjedt eszközt szoktak mutatni a VMM-ek, amikhez biztos minden OS-ben lesz beépített eszközmeghajtó (például S3 Trio videokártya vagy Intel E1000 hálózati kártya).
- Így beavatkozás nélkül tudja használni a vendég OS az I/O eszközt, akár a telepítés során is. Viszont lassú lesz, mert mindent úgy kell csinálni, mintha az tényleg egy lassú buszra csatolt fizikai eszköz lenne.



## I/O eszközök kezelése (paravirt.)



- Speciális csomag telepítése szükséges a vendégben
  - VMware Tools, Virtual PC Additions
  - **Mindig telepítsük a vendég gépen!**



- Ebben az esetben már egy nem létező eszközt lát a vendég gép, pl. VMware SVGA II. Ilyenkor a kommunikációt le lehet egyszerűsíteni, és például egy kép kirajzolása vagy egy csomag elküldése lehet egy egyszerű megosztott memóriába írás is.
- Viszont ehhez telepíteni kell a VMM speciális eszközmeghajtóit a vendég OS-be.

## I/O eszközök kezelése (hardveres)

- Hardveres támogatás
  - Intel VT-d, AMD IOMMU
  - PCI szabvány kiegészítése: I/O Virtualization (IOV)
  
- I/O eszközök
  - megosztása virtuális gépek között
  - közvetlen hozzárendelése egy virtuális géphez



További információ I/O eszközök virtualizációjáról:

- Carl Waldspurger and Mendel Rosenblum. 2012. I/O virtualization. *Commun. ACM* 55, 1 (January 2012), 66-73. DOI=10.1145/2063176.2063194 (<http://doi.acm.org/10.1145/2063176.2063194>)

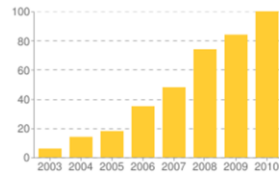
# Megoldások, cégek, termékek



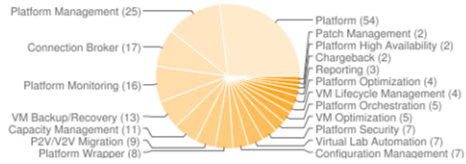
# Játékosok

<http://www.virtualization.info/radar/>

TRACKED COMPANIES



CATEGORIES










Application Virtualization	OS Virtualization	Hardware Virtualization		
Altiris	iCore Software	Snine	Invirtus	Skytap
AppStream	Oracle	Akimi	Kaviza	SolarWinds
AppZero	Parallels	Altir Networks	Kidaro	StackSafe
Ardence	RingCube	CI2G Labs	Lanamark	SteelEye
Ceedo	Sun	CA	Leostream	Sun
Citrix		Catbird	Liquidware Labs	Surgent
Doegel IT-Management		CIRBA	ManageIQ	Symantec
Endeavors		Citrix	Microsoft	Third Brigade
Technologies		cloud.com	MokaFive	ToutVirtual
FastScale		CloudShare	Neocleus	Tresys Technology
GreenBorder		Configuresoft	Neverfail	Trilead
Technologies		Connectix	Nicira	VDIworks
InstallFree		Convirture	Nimbula	Vecam
KACE		Desknote	Novell	Virto
Microsoft		Dunes Technologies	Oracle	Virtual Bridges
Softricity		DynamicOps	Pancetera	Virtual Computer
Spoon		eG Innovations	Pano Logic	Virtual Iron
Symantec		Embotics	Parallels	Virtugo
Systancia		Enomaly	PHD Virtual	Vizioncore
Thininstall		Ericom	Phoenix Technologies	Vkernel
Trustware		Eucalyptus Systems	PlateSpin	VMG Software
Unidesk		Fortisphere	Propero	VMLogix
VMware		HelperApps	Provision Networks	vmSight
		HP	Proxmox Server	VMTurbo
		Hyper9	Solutions	VMware
		HyTrust	Quest	XenSource
		IBM	Qumranet	
		icomsoft	Reflex Systems	
		innotek	Replicate Technologies	
			Sentillion	



lásd még: [http://en.wikipedia.org/wiki/Comparison\\_of\\_virtual\\_machines](http://en.wikipedia.org/wiki/Comparison_of_virtual_machines)

## Játékosok – 2

	ESXi, vSphere...
	nyílt forrású hypervisor
	XenServer, XenApp
	Virtual PC, Hyper-V, System Center
	Solaris Containers, Oracle VM, VirtualBox
	Kernel based Virtual Machine (KVM)
	mainframe, powerVM
	...

Ez csak egy esetleges válogatás a megoldásokból.

## DEMO Központi menedzsment

- Erőforráskészlet – korlátozások, garanciák
- VM térkép
- Teljesítmény mérés
- Live Migration – virtuális gépet futás közben átrakjuk másik gépre

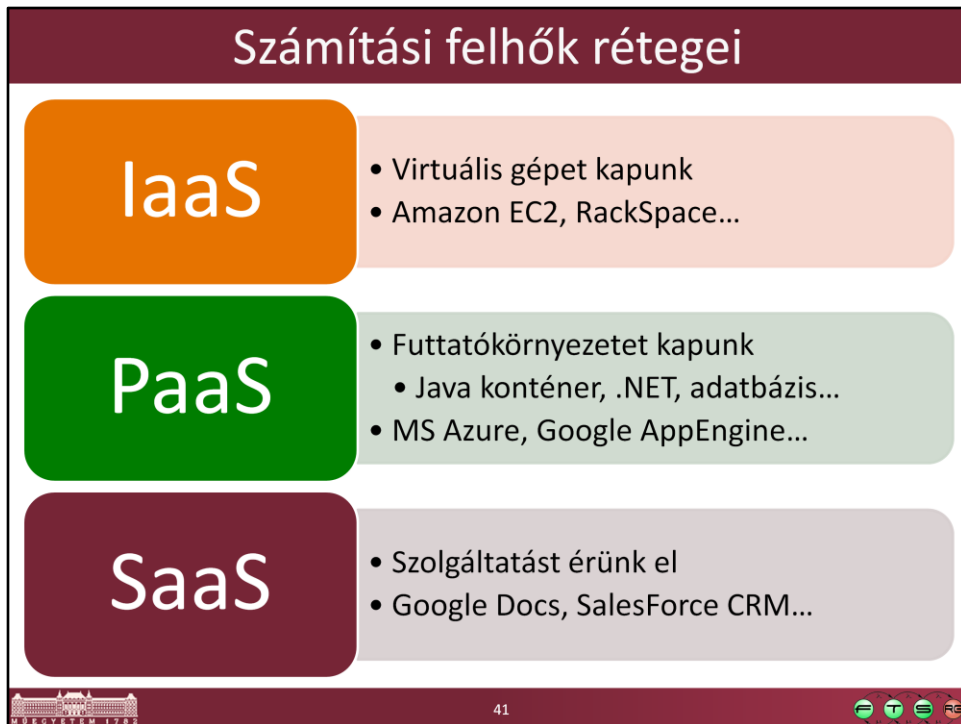
# Cloud computing

???

# Cloud computing







IaaS = Infrastructure as a Service

PaaS = Platform as a Service


SaaS = Software as a Service

A lista természetesen mindegyik elemnél hosszú ideig folytatható, rengeteg szolgáltató és szolgáltatás van már ma.

# Virtual Appliance


- Appliance: célhardver (pl. tűzfal)
- Virtual Appliance:
  - adott célra összeállított virtuális gép
- Előny:
  - nincs telepítés, függőség
  - csak a feltétlen szükséges komponensek vannak telepítve
- JeOS
  - Just enough Operating System
- Operációs rendszerek szerepe?

Search Virtual Appliances  
Enter the name of a company or appliance you are looking for.

User Favorites

## Most Downloaded

Download and deploy our most popular virtual appliances. 

▶ Ubuntu 8.04 b 'Hardy Heron'...

▶ Ubuntu 7.10 (Gutsy Gibbon) Desktop...

▶ Microsoft Windows Server 2003 R2...

▶ Fedora 8 VMware Image



Letölthető virtuális gépek pl.: <http://www.vmware.com/appliances/>

## További információ

### ■ Szakirány:

- Informatikai technológiák szakirány, **Rendszertervezés** ágazat, (pl. [Intelligens rendszerfelügyelet](#) tantárgy)

### ■ Választható tárgy (ősz):

- [Virtualizációs technológiák és alkalmazásai](#)

### ■ Cikk:

- Ole Agesen *et al.*: [The evolution of an x86 virtual machine monitor](#), *SIGOPS Oper. Syst. Rev.* 44, 4 (December 2010)



- A cikkek BME-s hálózaton belülről elérhetők és letölthetők a BME könyvtári előfizetéseknek köszönhetően (lásd: <http://www.omikk.bme.hu/>)!
- Rendszertervezés szakirány: <http://www.inf.mit.bme.hu/edu/specialization>
- Választható tárgy: <http://www.inf.mit.bme.hu/edu/courses/virttech>

## Összefoglalás

- Virtualizáció: növekvő jelentőségű
- Ellentmondásos terminológiák
- Versengő technológiák, rengeteg gyártó
  
- Operációs rendszerek
  - Funkciók megvalósítása a hypervisor-ban
  - Kisebb jelentőség ?



Érdekes lesz, hogy milyenek lesznek a következő OS verziók. A funkciók egy része átkerül/átkerült a hypervisorba. Lehetővé válik, hogy ne széles, mindenféle helyzetre használható OS-eket alkalmazzunk, hanem csak az adott konkrét funkcióhoz legyen benne támogatás, amit az a virtuális gép ellát. Látszik, hogy az OS-ek kezdenek túl nagygyá és komplexszé válni, a virtualizáció talán egy lehetőség, hogy kisebb, de jóval megbízhatóbb rendszereket készítsenek.