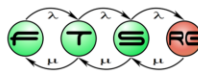


Bevezető, Virtualizációs technológiák áttekintése

Micskei Zoltán, Tóth Dániel



Bemutakozás – előadók

- Micskei Zoltán
(IB417, <http://mit.bme.hu/~micskeiz/>)



- Kövi András
(<http://mit.bme.hu/~kovi/>)



- Szatmári Zoltán
(<http://mit.bme.hu/~szatmari/>)



- Tóth Dániel (Nokia Siemens Networks)



A tantárggyal kapcsolatos adminisztratív kérdésekkel Micskei Zoltánt keressétek.

Bemutakozás – kutatócsoport



(Fault-Tolerant Systems Research Group, FTSRG)

<http://www.inf.mit.bme.hu>

- **BSc** - *Informatikai technológiák szakirány, Rendszertervezés ágazat*
- **MSc** - *Szolgáltatásbiztos rendszertervezés szakirány*
- Főbb témák:
 - Modell alapú szoftverfejlesztés
 - Informatikai rendszerek verifikációja és validációja,
 - Informatikai infrastruktúra elemzése és tervezése,
 - Alkalmazás: beágyazott rendszerek, SOA, IT rendszerek...

Bemutakozás – virtualizáció

- (2007) Intel Virtualizációs és Platformmenedzsment Laboratórium
 - IB413, az itteni gépek egy része☺
- (2009) VMware Academic Program tagság
 - legújabb VMware szoftverek
- Saját infrastruktúra
 - mérések és gyakorlatok mind virtuális gépeken
 - 6-8 ESXi szerver szakirányos hallgatóknak és kutatáshoz
- Kutatási és ipari projektek (virtualizáció, cloud...)

Virtualizációs technológiák és alkalmazásaik

- Tárgy honlapja:
 - <http://www.inf.mit.bme.hu/edu/courses/virttech>
- Időpont
 - minden héten csütörtök 12:15-14:00
- Helyszín
 - IB413 labor
- Előadás és gyakorlat vegyesen

Követelmények

- Egy félévközi önálló munka
 - Házi feladat lehetőségek:
 - Valamely technológia telepítése, egyszerű feladatok kipróbálása
 - Egyszerű programozási, scriptelési feladat valamelyik virtualizációs technológiával
 - Tudományos cikkek feldolgozása, ~5-10 oldalas tanulmányban összefoglalása
 - Egy rövid (~5 perc) szóbeli beszámoló a házi feladatban megismert technológiáról
 - Elkészült tanulmányok, dokumentációk, programok, scriptek feltöltése a 13. hét végéig
- Vizsga
 - Nincs 😊
- Félév végi jegy a házi feladat és előadás alapján

Követelmények (dátumok)

- HF téma választás: 2012. 09. 14-ig (2. hét vége)
(a megadott témákon kívül természetesen saját ötletet is szívesen fogadunk)
- Szóbeli beszámoló a témáról: 2012. 10. 25. (8. hét)
- Elkészült anyagok feltöltése: 2012. 12. 02. (13. hét)
- Javítás:
 - Beadott dolgozatok megbeszélése a pótlási hét elején
 - Javított verzió leadása pótlási hét végéig
- Pótlás:
 - Szóbeli beszámoló: csak különleges esetben, aki a tárgyat tényleg el szeretné végezni mindenképpen jöjjön el a beszámolóra!
 - Anyagok feltöltése: 2012. 12. 16. (pótlási hét vége)

Korábbi dolgozatok

- Válogatás a korábbi HF-ekből elérhető a weben (<https://www.inf.mit.bme.hu/edu/courses/virttech/homework/public>)
- Cél: érdekes dolgozatok ne vesszenek el



Az első rész a virtuális gépek listázása. Itt egy táblában láthatjuk a virtuális gépek nevét, állapotát, és a hozzá tartozó erőforrásokat. A táblában a következő oszlopok vannak: VM név, VM státusza, VM erőforrásai, VM leírása, VM létrehozásának dátuma, VM frissítésének dátuma, VM törlésének dátuma, VM frissítésének dátuma, VM törlésének dátuma.

Művelet eredmények grafikus ábrázolása

A projekt jelenlegi állása szerint van egy háttérben futó alkalmazásunk, ami folyamatosan gyűjt a virtuális gépek működéséről, illetve az adatbázisunk egy táblájában tárolja a számként egy állapota, ami az adatbázisunkból a táblájában tárolt adatok alapján a virtuális gépek működését vizsgálja. Ez a táblázat a virtuális gépek működését vizsgálja.

A program PHP nyelven íródott és a Google Charts API-t használja a vizualizációk készítésére. A Google Charts API-t az alábbi módon lehet használni: <https://developers.google.com/chart/interactive/docs/basic-learn>

Az alábbiakban a program a legutóbbi eredmények grafikus ábrázolását tud generálni, a következő paraméterekkel: <https://developers.google.com/chart/interactive/docs/basic-learn>



30 napos időtartamú adatok alapján a grafikon megmutatja a havi átlag teljesítményt (havi átlag teljesítmény), ami alapján lehet megállapítani a havi átlag teljesítményt. Ez a havi átlag teljesítmény.

Az RTT mértékének vizsgálata az alábbiakban.



A technológiai hálózata a hálózati operációs rendszerek működését vizsgálja a hálózati operációs rendszerek működését vizsgálja.

A hálózati operációs rendszerek működését vizsgálja a hálózati operációs rendszerek működését vizsgálja.

A hálózati operációs rendszerek működését vizsgálja a hálózati operációs rendszerek működését vizsgálja.

A hálózati operációs rendszerek működését vizsgálja a hálózati operációs rendszerek működését vizsgálja.

A hálózati operációs rendszerek működését vizsgálja a hálózati operációs rendszerek működését vizsgálja.

A hálózati operációs rendszerek működését vizsgálja a hálózati operációs rendszerek működését vizsgálja.

A hálózati operációs rendszerek működését vizsgálja a hálózati operációs rendszerek működését vizsgálja.

A hálózati operációs rendszerek működését vizsgálja a hálózati operációs rendszerek működését vizsgálja.

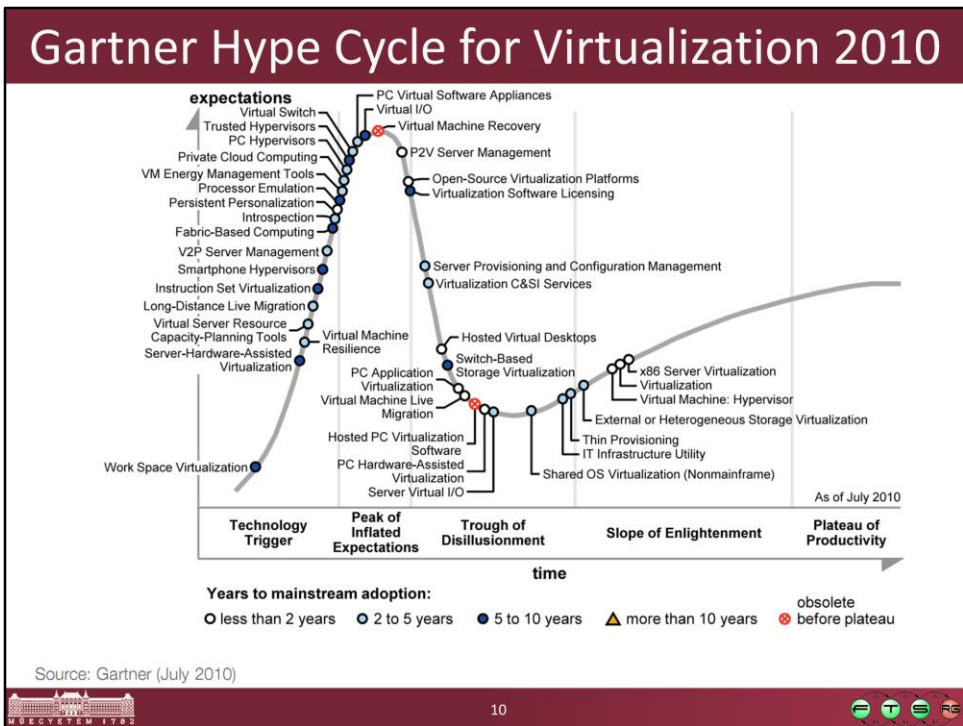
A hálózati operációs rendszerek működését vizsgálja a hálózati operációs rendszerek működését vizsgálja.

- Érdekes jó HF-et készíteni, jó referencia később



Tartalom

- Adminisztrációs tudnivalók
- **A virtualizáció fajtái**
- A félév menete

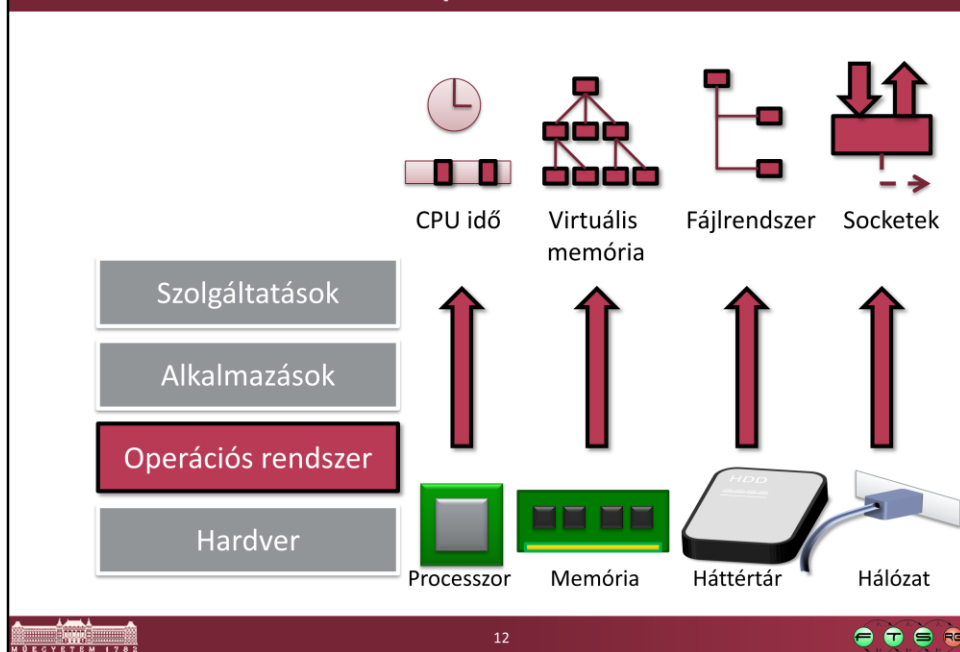


Forrás: **Gartner Hype Cycle for Virtualization, 2010**,
<http://premierit.intel.com/docs/DOC-5768>

Mi is az a virtualizáció?

- „Az erőforrások elvonatkoztatása az erőforrást nyújtó elemektől”
 - kellemesen sejtelmes általános definíció ☺
- Jellemzően:
 - fizikai erőforrásokból logikai erőforrások képzése, amik függetlenek a tényleges fizikai elemektől
 - korlátos erőforrások szétosztása több részre
- Ez egy új ötlet?
 - Korántsem – az oprendszerek is ezt csinálják...

Mit csinál az operációs rendszer?



- **CPU:** tekinthetjük úgy, hogy az OS rendszerhívásai (SYSCALL vagy INT utasításon keresztül) mintegy „kibővítései” a CPU ISA-nak, a kernel olyan funkciókat tesz elérhetőek ezen keresztül, amik valójában a hardverben nem léteznek. Ez egy nem feltétlenül 100%-ig technikailag korrekt magyarázat, inkább csak szemléletmód.
- **Memória:** minden alkalmazás saját 0..N-ig terjedő (akár összefüggő, de nem feltétlenül) címtartományt kap, ami a fizikai memóriába más és más címre képződik le, akár futás közben is változhat. Háttértárra ki is menthető a memória egy része, így több memória lehet szétszotva, mint amennyi fizikailag van. További képességekkel is kiterjeszti az OS a memóriát: pl. folyamatok között megosztott memóriaterület vagy fájl tartalmának memóriába leképezése, stb.
- **Háttértár:** Itt nagy az absztrakció, az egységes fix méretű adattároló tömbön tetszőlegesen sok, tetszőleges méretű, névvel, és egyéb attribútumokkal ellátott, hierarchikus névtérbe szervezett perzisztens adattároló tömböket lehet létrehozni. A fájlok minőségileg is nagyban különböznek a mögötte lévő háttértártól.
- **Hálózat:** a fizikai hálózaton több protokollrétegen keresztül absztrakt, kapcsolat alapú vagy anélküli socket hozható létre, aminek életciklusa kezelhető, hibakezeléssel is rendelkezik, több is mehet párhuzamosan, a fizikai kapcsolatot igény szerint használják (nem tartja fogva üresjáratban)... Ez is nagy absztrakció a fizikai hálózat szolgáltatásához képest.

Mit csinál az operációs rendszer?

Az operációs rendszer a fizikai hálózati kapcsolatból egy szoftveres socketet hoz létre, amiből több is lehet egyidejűleg, életciklussal hibakezeléssel.

Tanulság:

Az operációs rendszer kétféle módon vonatkoztat el a fizikai erőforrásoktól:

- Erőforrás megosztással
- Erőforrás finomítással, saját logikai erőforrásfajták bevezetésével

Szolgáltatás

Alkalmazás

Operációs rendszer

Hardver

Processzor

Memória

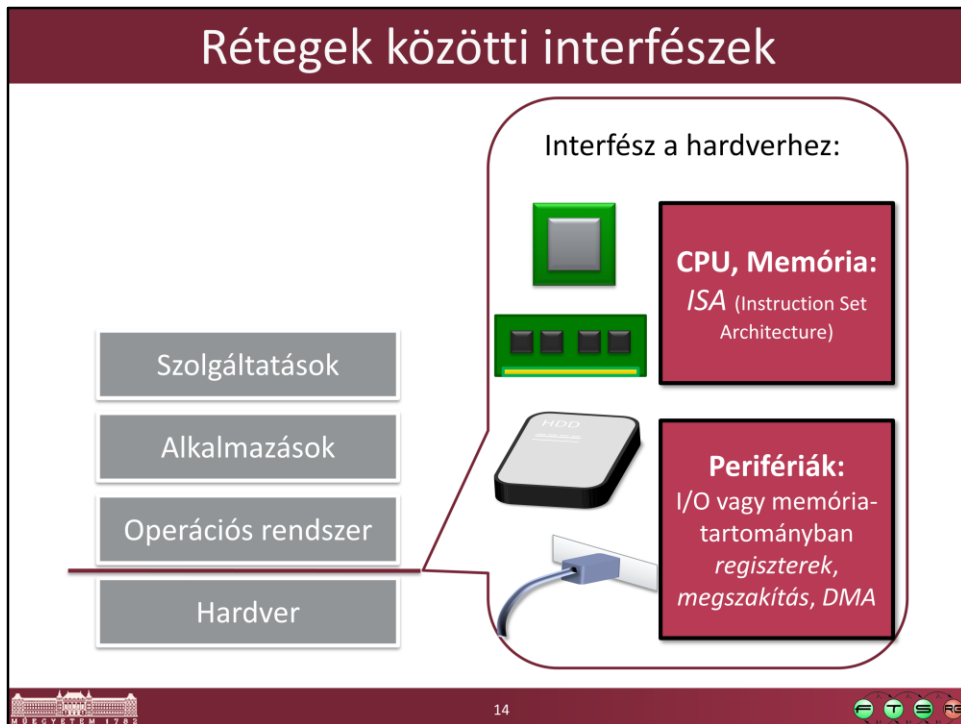
Háttértár

Hálózat



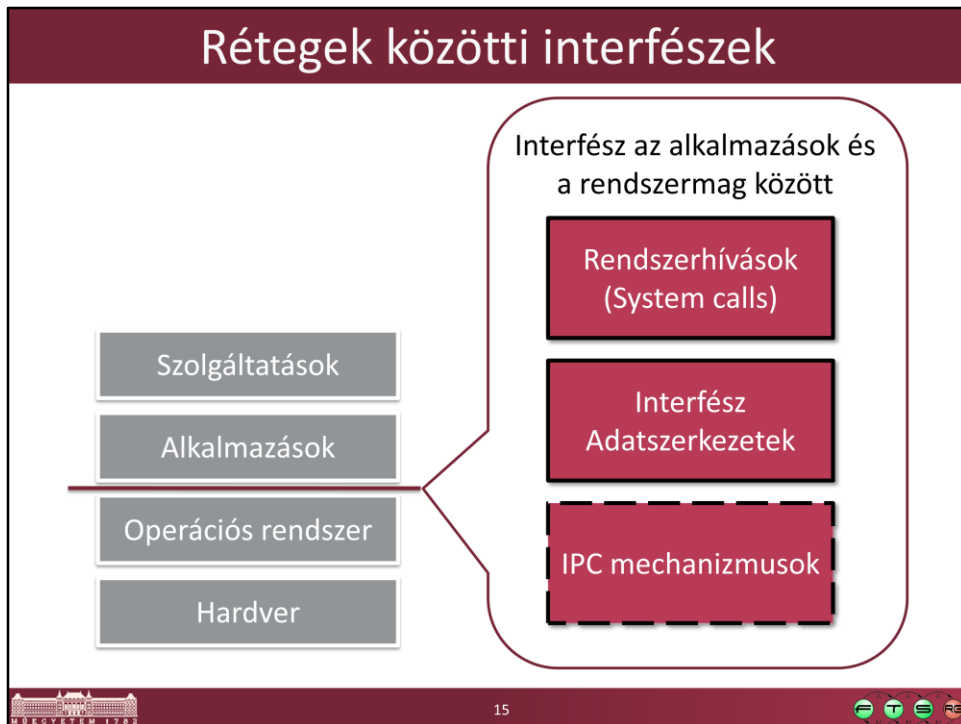
Socketek





„Platform virtualizáció”

Itt a lényeg, hogy a virtuális gépekben egy-egy saját kernel fut valamilyen izolált környezetben, tehát egymástól eltérő fajta operációs rendszerek is futhatnak egymás mellett. Általában (de nem minden esetben, lásd. paravirtualizáció) a virtuális gépben futó OS számára egy látszólag valódi fizikai hardverhez hasonló hardverkörnyezetet biztosít.



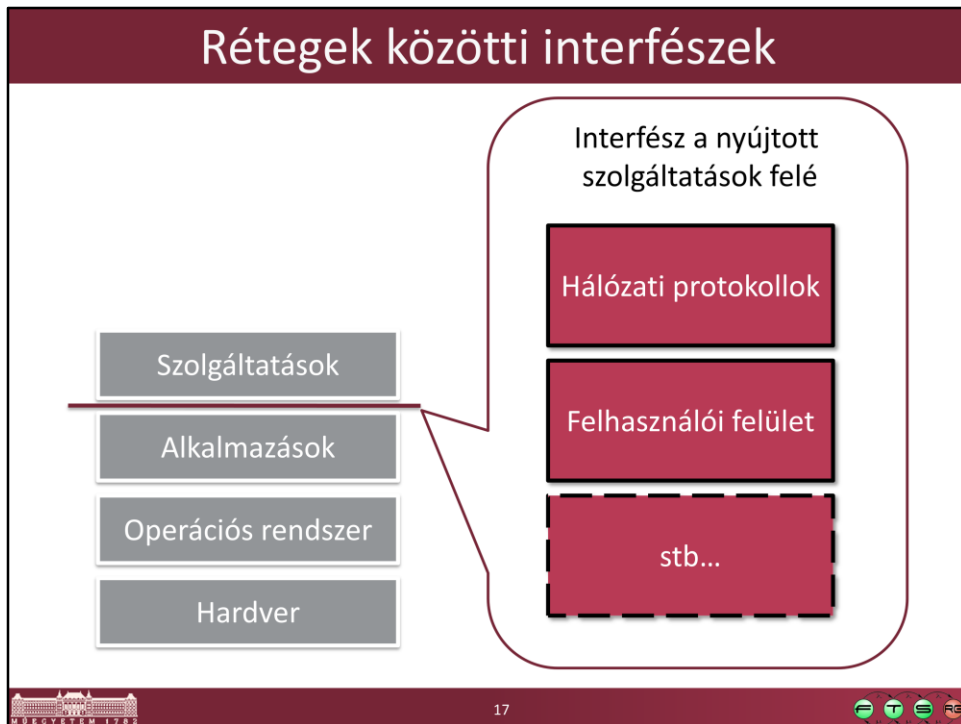
„Operációs rendszer szintű virtualizáció” vagy „Konténer alapú virtualizáció”

Itt az operációs rendszer felett alakítunk ki elkülönített virtuális környezeteket (jail, container), amely közül mindegyik úgy érzékeli, hogy csak ő fut a kernel felett. Ehhez a kernel által biztosított – normális esetben singleton – erőforrásokat kell többszörözni minden környezethez. A megoldás változó mélységű lehet, van olyan, ahol csak a kernel látszik, van, olyan, ahol valamilyen közös magas szintű felhasználói-módú erőforrásokat is elérhetővé tesz minden izolált környezetben.



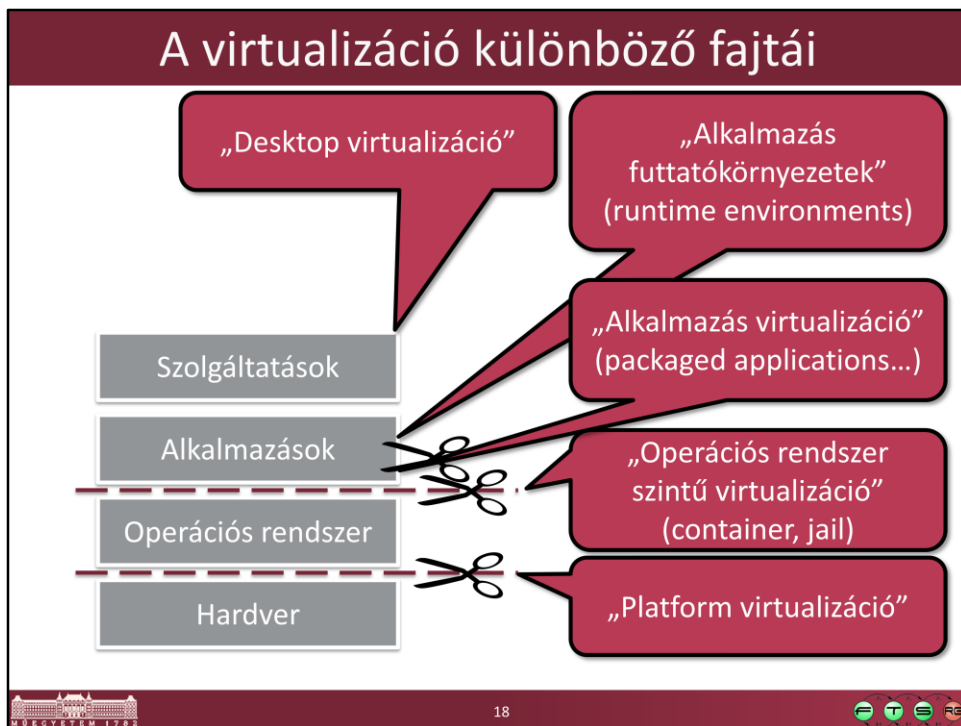
Igazából hagyományosan az OS-hez soroljuk, de nem a kernel, hanem felhasználói módban futó könyvtárak, folyamatok szolgáltatják az adott operációs rendszer magas szintű programkörnyezetét.

Az „alkalmazás szintű virtualizáció” igazából a konténer alapú virtualizációk egy olyan esete, ahol a használati eset specifikusan csak egy-egy alkalmazás „becsomagolása”, az OS környezet összes többi része nincs izolált konténerben. Pl. MS Office használata telepítés, registry módosítás, stb. nélkül.



„Desktop virtualizáció”-nak szokták hívni azt az esetet, amikor a felhasználó számára nyújtott szolgáltatást (elsődlegesen grafikus felületen) távolítjuk el szolgáltatási igénybevételi ponttól. Jellemző eset, hogy a felhasználó desktop környezete valami szervergépen fut (akár valamilyen más, platform vagy OS szintű virtualizációs formával megvalósítva), a kezelőfelület távoli elérésére azonban vékonykliens gépeket használnak.

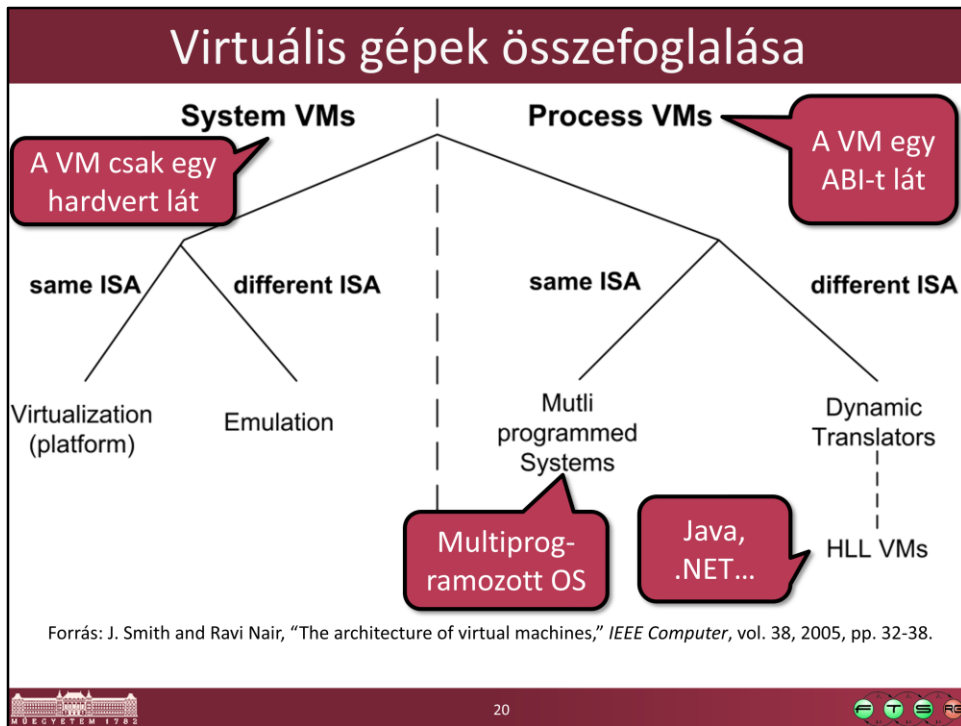
Más esetben a szolgáltatás valamilyen hálózaton elérhető technikai szolgáltatás, webes felület stb. A manapság oly divatos „cloud computing” alapötlete, hogy a hálózati szolgáltatásoknál könnyen elfedhető, hogy valójában mely komponens biztosítja, ezt is egyfajta „virtualizációnak” szokták nevezni.



Természetesen ettől különböző felosztások is szoktak lenni, sok mindent sorolnak még be az egyes gyártók vagy emberek a virtualizáció fogalma alá.

A virtualizáció különböző fajtái

- Tehát minden lehet „virtuális”
- „De eddig virtuális gép az OS volt meg a Java...”
- Azok hogyan kapcsolódnak ide?



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
 (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

Platform virtualizáció

- Amikor a „virtualizáció” buzzword elhangzik leggyakrabban erről van szó
 - „Szerver virtualizáció”, „Hardver virtualizáció”, „Számítógép virtualizáció” szinonim fogalmak
 - De nem összekeverendő a „hardveres” virtualizációval!
- Cél: megosztani a hardver erőforrásokat:
 - Nem végzünk finomítást, az eredeti(hez hasonló) interfészen maradnak elérhetőek
 - Izolált környezeteket („sandbox”) biztosítunk
- Célok gyakorlatiasabban megfogalmazva:
 - Több operációs rendszer példányt futtatni egyazon gépen



A hardveres virtualizáció csak egy lehetséges technika a platform virtualizáció megvalósítására, kb. rész-egész viszonyban vannak. Lásd később...

A hardver erőforrások megosztásának fogalmába általában belefér az, hogy nem pont azonos interfészen osztjuk meg az erőforrást, tehát lehet más CPU architektúra, más típusú periféria, de fontos megkülönböztetés, hogy nem képezünk belőle magasszintű logikai szolgáltatásokat, mint egy hagyományos OS teszi. Persze ez alól is lesznek kivételek, de nagyvonalakban ez igaz marad.

Mire lesz ez jó nekünk?

- Allow me to demonstrate! 😊

Vegyünk több vasat!

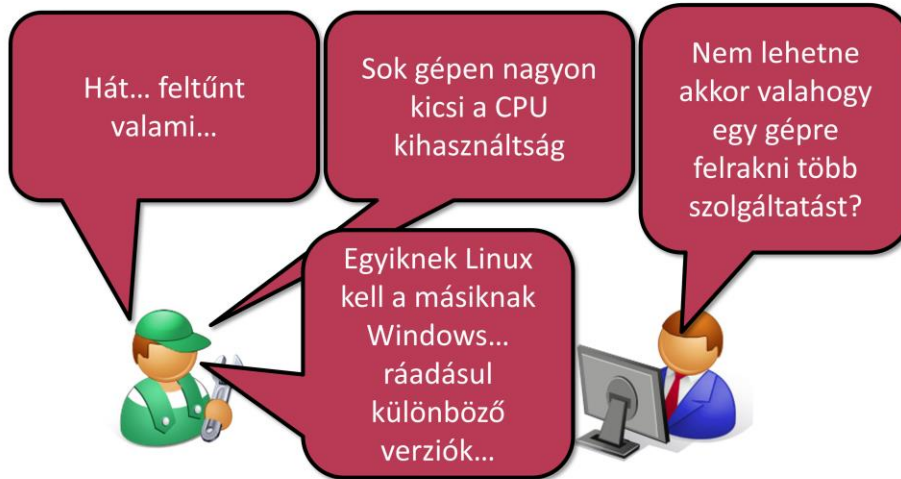
Új üzleti szolgáltatást
akarok beindítani

Biztos, hogy ez
segít?
Biztos, hogy ez a
költséghatékony
megoldás?



Mire lesz ez jó nekünk?

- Allow me to demonstrate! 😊



Mire lesz ez jó nekünk?



(Ő a biztonsági felelős a cégnél)

Biztonsági okokból nem szabad egy gépre rakni őket!



Egyiknek Linux kell a másiknak Windows... ráadásul különböző verziók...



Nem lehetne akkor valahogy egy gépre felrakni több szolgáltatást?

Mire lesz ez jó nekünk?

Több platformon kell fejlesztenem, tesztelnem... az időm nagy része az ide-oda váltogatással megy el. Ráadásul folyton széthomokozom az oprendszer



(szoftverfejlesztő avatarja)



Szóval nekem is mindenféle sokgépés bonyolult tesztkörnyezetet kell csinálnom a ti cuccaitokhoz

Mire lesz ez jó nekünk – alkalmazási területek

Szoftver-fejlesztés

fejlesztés, tesztelés sok különböző környezetben, könnyen

„homokozó” aka sandbox – a tesztelés nem veszélyezteti az éles munkakörnyezetet

különleges hibakeresési lehetőségek (debugging)

Desktop alkalmazások

különböző környezetet igénylő alkalmazások együtt futtatása

biztonsági szempontból jó (vagy legalábbis *jobb*) szétválasztás

sandbox – alkalmazások telepítés, kipróbálása éles munkakörnyezet veszélyeztetése nélkül

Nehezen/kényelmetlenül telepíthető környezetek egyszerű, gyors cserélése (pl.: mérés laborok, tanfolyamok)

Mire lesz ez jó nekünk – alkalmazási területek

Szerverek

Erőforrás megosztás, hardverköltség csökkentés

Energiatakarékosság, terhelésfüggő átcsoportosítás

Könnyű erőforrás átrendezés („kéne nekem most gyorsan még 4GB RAM és még 2 processzormag a gépembe...”)

Ideiglenes tesztkörnyezetek gyors kiépítése („...és még két ugyanilyen gép is kéne, de csak a jövő hétig”)

Életciklus kezelés - szerverek könnyű létrehozása, kezelése, mentése, üzembiztos működtetése, nyilvántartása

Desktop gépek kiváltása szerveren futó virtuális gépekkel – nem kell desktop gépeket menedzselni (VDI)

Mobil / beágyazott

Kritikus feladatok szétválasztása, üzembiztosság, biztonság

Mobil eszközök: céges és saját környezet szétválasztása



Történeti áttekintés (részlet)

~1960

- IBM CP-40 rendszere, első kísérletek a virtualizációval

1972

- CP-67 a virtualizáció a mainframe termékvonalon (S/370) része lett
- Paravirtualizációt is tartalmazott (bár senki nem nevezte még így)

1974

- Popek és Goldberg publikációja

1998

- az első x86 alapú platform virtualizáció megoldás (VMware)

1999

- User Mode Linux
- első praktikus paravirtualizáció x86-on

~2002

- kezdve számos más megoldás (Xen, Parallels, Sun, Microsoft...)
- második felétől kezdve a szerver virtualizáció felfutóban

~2005

- beágyazott rendszerek virtualizációja

~2010

- virtualizáció alap építő elem lett mindenhol
- Cloud computing buzzword felfut



Bővebben: <http://www.vm.ibm.com/history/>

Tartalom

- Adminisztrációs tudnivalók
- A virtualizáció fajtái
- **A félév menete**

A félév menete



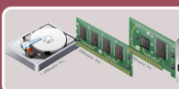
1. hét - Bevezetés

- A virtualizáció kialakulása
- Megközelítések áttekintése



2. hét - A virtualizáció alapjai I.

- CPU virtualizációs megoldások



3. hét - A virtualizáció alapjai II.

- Memória virtualizálása
- Perifériák, I/O rendszerek virtualizációja



4. hét – Kliens oldali virtualizáció

- Alkalmazási lehetőségek, virtuális hardver
- Gyakorlat: alapfunkciók, pillanatképek, klónozás



5. hét - Schönherz QPA

A félév menete



6. hét - Kiszolgáló oldali virtualizáció I.

- Architektúrák összehasonlítása
- VMware ESXi és MS Hyper-V ismertetése



7. hét - Kiszolgáló oldali virtualizáció II.

- Gyakorlat: ESXi telepítése, konfigurálása
- Jellemzőes alapfeladatok



8. hét - Házi feladat szóbeli beszámoló

- 5-5 perc előadás a választott házi feladat témában

9. hét: SZÜNET



10. hét - Operációs rendszer szintű virtualizáció

- Konténer-alapú virtualizáció
- Gyakorlat: OpenVZ

A félév menete



11. hét - Alkalmazás és megjelenítés virtualizáció

- Alkalmazás csomagolás, távoli elérés, vékonykliensek, VDI
- Gyakorlat: VMware ThinApp, MS Remote Desktop Services



12. hét - Menedzsment feladatok I.

- Virtuális gép sablonok, életciklus kezelés, felügyelet
- Katasztrófavédelem, hibatűrés



13. hét – Menedzsment feladatok II.

- Gyakorlat: VMware vCenter
- Virtuális gépek működés közbeni mozgatása, hibatűrés



14. hét – Cloud computing

- Számítási felhők fajtái, céljai, előnyei és hátrányai
- Amazon EC2, Rackspace Cloud, MS Azure...

További információ

Néhány fontosabb blog, hír aggregátor oldalak:

- Hírek, független(ebb) vélemények
 - <http://www.virtualization.info>
 - <http://www.cloudcomputing.info>
- Planet VMware, összes VMware blog
 - <http://www.vmware.com/vmtn/planet/vmware/>
- Microsoft
 - Ben Armstrong (Virtual PC Guy),
http://blogs.msdn.com/virtual_pc_guy/rss.xml
 - Windows Virtualization Team Blog,
<http://blogs.technet.com/b/virtualization/>

Összefoglalás

- A virtualizáció általános definíciója
- A főbb virtualizációs lehetőségek áttekintése
- Történeti áttekintés
- Platform virtualizációs demo

- A következő rész tartalmából
 - Platform virtualizáció részletes áttekintése
 - CPU virtualizáció megoldási lehetőségei, nehézségei
 - Memória virtualizáció lehetőségei, nehézségei