

Virtualizáció típusai, paravirtualizáció (UML), natív virtualizáció (KVM, vmware)

Architektúrák speci
Kiselőadás 2007. 03.07.

I. Bevezetés

A virtualizáció egy absztrakciós réteg, ami elválasztja a fizikai hardvert az operációs rendszertől, és nagyobb erőforráskihasználást és rugalmasságot biztosít.

A virtualizáció több virtuális gép használatát is lehetővé teszi, különböző operációs rendszerek futhatnak egyidőben, egymástól teljesen függetlenül, egymás „mellett” ugyanazon a számítógépen. Minden egyes virtuális gép saját virtuális hardver-kiépítéssel rendelkezik (pl. RAM, CPU, stb.), a vendég operációs rendszerek ezt látják, ezzel dolgoznak. Az operációs rendszer tehát egy teljes értékű hardverkészletet lát, teljesen függetlenül a számítógép valódi hardver-kiépítésétől.

A virtuális gépeket fájlokban tárolja a gazdarendszer, ezáltal lehetővé teszi azt, hogy gyorsan lementhető, másolható és kezelhető legyenek. Teljes rendszerek (teljesen konfigurált alkalmazások, operációs rendszerek, BIOS és virtuális hardver) mozgatható át rövid idő alatt egyik fizikai szerverről a másikra, miközben a külső szemlélők (felhasználók) csupán néhány másodperces leállást tapasztalnak.

A virtualizáció gyökerei az 1960-as évekig nyúlnak vissza, amikor is a nagy (és drága!) mainframe számítógépek feladatait próbálták „szétosztani” többfelé. Idővel, a kisméretű gépek megjelenésével hatékonyabb, olcsóbb megoldás kínálkozott a számítási teljesítmény elosztására, így az 1980-as évekre a virtualizáció már nem volt széles körben alkalmazott technológia.

Az 1990-es években kezdték észrevenni a kutatók, hogy a virtualizáció miként képes megoldani néhány aktuális problémát, amelyek az olcsó hardverek elterjedésével bukkantak felszínre: az alacsony kihasználtságot, a növekedő fenntartási költségeket és a sebezhetőséget.

Ma a virtualizációnak igen fontos szerepe van az informatikában, rugalmasságával, biztonságával segíti a vállalkozásokat.

II. A virtualizáció előnyei

- Több különböző operációs rendszerre írt alkalmazások futtathatók egyazon fizikai számítógépen
- A virtuális gépek teljesen elszigetelten futnak a gazdarendszertől, így ha a virtuális gép összeomlik, vagy meghibásodik, a többi virtuális gép működését nem befolyásolja, és a gazdarendszer sem omlik össze.
- Az adatok csak konfigurált hálózati kapcsolatokon keresztül áramolhatnak a virtuális gépek között.
- A teljes virtuális környezet egy fájlban tárolható, így könnyen mozgatható, másolható
- standard virtuális hardverek használatával biztosítható a kompatibilitás

III. Virtualizációs technikák

III/1. Bevezető

Több [tanulmány](#) is rámutatott arra, hogy míg régebben a vállalatok saját informatikai infrastruktúrájuk fejlesztésére rendkívül sokat költöttek, addig mára egyre nagyobb hangsúlyt kap a már meglévő eszközök mind hatékonyabb módon történő kihasználása. A virtualizációs technikák és a grid technológia egy-egy lehetséges választ nyújt erre az igényre. Alábbiakban röviden áttekintjük a PC architektúrára jelenleg elérhető szoftveres virtualizációs technológiákat, különös tekintettel a nyílt forráskódú megoldásokra.

A virtualizáció kifejezést minden olyan technológiára használjuk általában, ami lehetőséget nyújt arra, hogy az informatikai erőforrásainkat (számítási, adattárolási, hálózati stb.) absztraktnak kezeljük és rendszereink tervezésénél elrugaszkodhassunk azok fizikai megvalósításától. Így a fizikai erőforrások felett megjelennek a virtuális erőforrások, melyek több részre oszthatják vagy össze is vonhatják a fizikai erőforrásokat.

A szerver rendszerek virtualizációja többek között az alábbi előnyöket nyújtja:

- a fizikai szerver kiszolgálók kihasználtsága 10-20%-ról 85-90%-ra növekszik: csökken a hardver költség,
- funkcionális elemek szétválasztásával növelhető a biztonság,
- könnyebb kezelhetőség (management) ("egy kattintással új szerver hozható létre"),
- dinamikus erőforrás (memória, tároló, hálózati sávszélesség, cpu idő) hozzárendelést tesz lehetővé különféle funkciót ellátó kiszolgálókhoz, és alkalmazásokhoz,
- megnövelt rendelkezésreállás biztosítható,
- különféle operációs rendszerek futtatása is lehetséges egyazon fizikai eszközön.

Míg az emuláció a hardveres és szoftveres erőforrásokat osztja meg az operációs rendszerben, addig a virtualizáció logikailag szétosztja az erőforrásokat, több különálló rendszerre. A virtualizáció szoftveres úton is történhet úgy, hogy a gazda és a vendég operációs rendszert nem kell módosítani, de hardveres úton is megoldható speciális processzorral; ilyen az AMD Pacifica technológiája az Opteronokban, illetve az Intel 65 nm-es termékeiben.

Hasonlóan, többféle virtualizációs megoldást különböztetünk meg. A teljes virtualizáció egy teljes hardverkörnyezetet szimulál a módosíthatatlan OS számára (Bochs); a paravirtualizáció egy speciális API-t jelent, ami az OS módosítását is igényli (Xen); a natív virtualizáció pedig részlegesen szimulálja a hardvert, miközben izolálja a vendég operációs rendszereket (VMware). Van azonban egy OS-szintű virtualizáció is, az alkalmazás-virtualizáció, de hardveres technikákat is ismerünk, amelyek fizikailag osztják fel a hardvert.

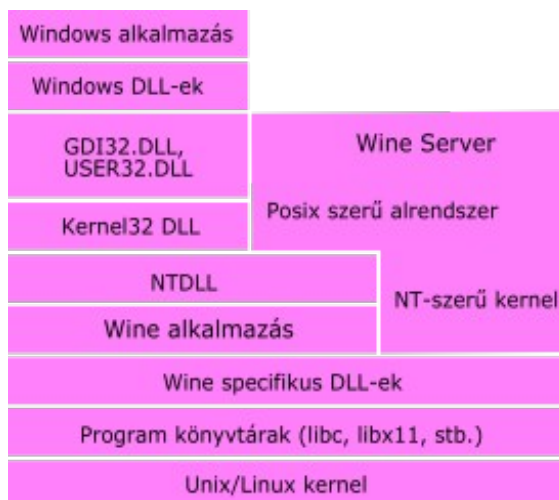
Tehát alapvetően az alábbi kategóriákba sorolhatjuk a módszereket:

- alkalmazás szintű virtualizáció,
- operációs rendszer szintű virtualizáció,
- emuláció,
- paravirtualizáció,
- natív virtualizáció.

A továbbiakban a virtuális környezetet biztosító rendszert gazda vagy host rendszernek fogjuk nevezni.

III/2. Alkalmazás szintű virtualizáció

Az alkalmazás szintű virtualizációt nem az operációs rendszer szintjén valósítjuk meg, hanem az alkalmazások által használt alapvető program könyvtárak (library) ill. alkalmazások szintjén. Egyik legjellemzőbb példája ennek a technológiának a használatára a Microsoft Windows operációs rendszeren megtalálható [Cygwin](#). Ez a rendszer szabványos Posix/GNU programozási környezetet igyekszik biztosítani Microsoft Windows környezetben, de nem teszi lehetővé a UNIX/Linux binárisok módosítás nélküli futtatását. Ezt a rendszert elsősorban a UNIX/Linux környezetben fejlesztett alkalmazások Microsoft Windows platformra történő portolásának megkönnyítésére használhatjuk. A másik népszerű alkalmazás a [Wine](#) (1. ábra), ami a Windows API-k és egyes kernel funkciók megvalósítását tűzi ki célul Linux és X11 rendszerek felett. Ez megoldás már lehetőséget nyújt számos esetben a Microsoft Windows alkalmazások módosítás nélküli futtatására is. A Wine kód bázisra több kereskedelmi alkalmazás is épít, amik közül itt most a CodeWeavers nevű cég [Crossover Office](#) termékét emelnénk ki.

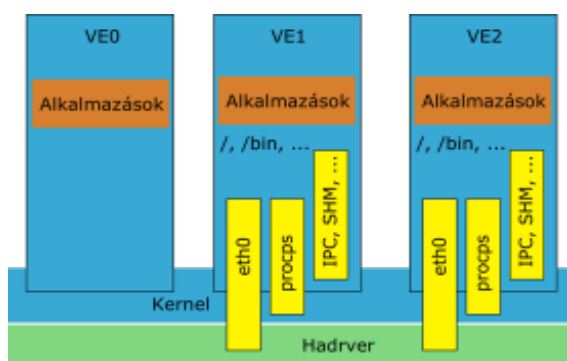


1. ábra: Wine architektúra

Mint látható ez a virtualizációs technika nem minden esetben transzparens az alkalmazás készítő számára, ugyanakkor jelentősen megkönnyítheti az alkalmazások adaptálását. Számos esetben pedig elérhető, hogy az alkalmazások módosítás nélkül futtathatók legyenek a virtualizált platformon.

Ebbe a kategóriába sorolhatóak a Microsoft Windows alapú rendszereken egyre népszerűbb alkalmazás virtualizációs megoldások is. Ezek elsősorban az ún. [DLL-pokol](#) problémára igyekeznek megoldás nyújtani. A problémát például a [Softricity](#) termékei úgy oldják meg, hogy a DLL-ek egy részét az alkalmazás első indításakor ideiglenesen helyezik el a rendszeren, más részüket pedig hálózati rendszerhívásokon keresztül teszik elérhetővé. Mindehhez egy Active Directory alapú hozzáférés szabályozást is biztosítanak.

III/3. Operációs rendszer szintű virtualizáció



2. ábra: Linux-VServer architektúra

úgy egészítik ki az alap operációs rendszert, hogy az önmagán belül képes legyen létrehozni több

Ez a virtualizációs technika lehetőséget biztosít arra, hogy egy operációs rendszeren belül több egymástól izolált, önálló programvégrehajtási környezetet hozzunk létre. A különféle megvalósítások más és más szintű izolációt valósítanak meg, valamint eltérő szinten szabályozható bennük a fizikai, és az operációs rendszer szintű erőforrásokhoz való hozzáférés is. Ezt a technológiát szokás több különböző néven is emlegetni: sandbox, jail, particionálás stb. Számos operációs rendszer szintű virtualizációs technológia létezik. Egyesek az elterjedt operációs rendszerek szerves részét képezik, mások azok kiegészítéseként érhetőek el. Közös jellemzőjükként,

virtuális környezetet (Virtual Environment - VE). Ezek a VE-k valójában ugyanannak az operációs rendszernek egy-egy virtuális példányai. Ezért ez a technika nem képes az alaprendszer operációs rendszer változatától eltérő rendszert futtatni (2. ábra egy mintaimplementáció, a Linux-VServer architektúráját mutatja). Cserébe a virtualizációból adódó teljesítmény-csökkenés 1-3 % környékén tartható mind számítási, mind pedig I/O hozzáférés esetén valamint akár több száz virtuális környezetet is ki lehet alakítani ugyanazon a fizikai eszközön. Ebből adódóan ennek a technikának a legjelentősebb felhasználói a tárhely (hosting) szolgáltatók, akik Virtuális Privát Szerver (VPS) szolgáltatásokat nyújtanak e környezet segítségével.

A legelterjedtebb operációs rendszerszintű virtualizációs megoldások

- [chroot](#): a UNIX/Linux rendszerek beépített rendszerhívása, ami lehetővé teszi, hogy megváltoztassuk a gyöker könyvtár helyét az állomány rendszerben. A chroot hívás után indított alkalmazások és gyermekeik számára az új könyvtár lesz a gyöker (root). Ez a technika kizárólag gyenge állomány rendszer szintű izolációt tud megvalósítani, azonban oda kell figyelni arra, hogy számos egyszerű módszer létezik ebből az egyszerű virtuális környezetből való kitörésre, ami biztonsági kockázatot jelenthet. Ez a megoldás jól használható teszt környezetek, például különféle LINUX terjesztések kipróbálására, valamint hasznos eszköz a 32 bites környezetről 64 bites környezetre való váltás során is.
- [FreeBSD Jail](#): a [FreeBSD](#) operációs rendszer fejlesztői a chroot rendszerhívást fejlesztették tovább a jail rendszerhívás készítésekor. Egyfelől az állomány rendszer szintű szeparáció mellett a jailek önálló hálózati alrendszert is kaptak, így lehetőség van az egyes jaileket többek között különböző hálózati címekkel és szűrési szabályokkal is ellátni. A jail rendszerhívás biztonsági szempontból is jobb, mint a chroot, mivel számos alapértelmezett korlátozást tartalmaz az alap operációs rendszer kritikus erőforrásainak hozzáférése, valamint jól együttműködik a FreeBSD rendszer hozzáférés szabályozási alrendszerével.
- [Solaris Containers](#): az [OpenSolaris 10](#) operációs rendszerben található virtualizációs megoldás még tovább növeli a virtuális környezetek (zónák) izolációs szintjét. Az állomány rendszer szintű és hálózati szeparáció mellett, a virtuális környezetek által felhasználható CPU idő, virtuális memória, hálózati sávszélesség, I/O teljesítmény stb. is szabályozható. Szerencsés konstrukciójának köszönhetően a [BrandZ](#) nevű továbbfejlesztéssel lehetőség nyílik Solaris operációs rendszertől különböző rendszerek futtatására is. Jelenleg a Linux futtatására szolgáló megoldás van kísérleti stádiumban.
- Linux alapú rendszereken is elérhető az a szintű rugalmasság a virtuális környezetek erőforrás szabályozásában amit a Solaris Container nyújt. A nyílt forráskódú kernel módosítások közül a [Linux-VServer](#) megoldás a legnépszerűbb, melyet több különböző irányba is továbbfejlesztettek. Egy korai elágazása a [FreeVPS](#) projekt elsősorban a tárhely szolgáltató körében népszerű. A VPS rendszerek legnagyobb számban ugyanakkor a kereskedelmi forgalomban kapható [Virtuozzo](#) rendszerre épülnek, melynek nemrégiben jelent meg nyílt forrású változata az [OpenVZ](#). A Virtuozzo népszerűségét elsősorban a kifinomult kezelő rendszerének köszönheti.

III/4. Emuláció

Az emuláció lényege, hogy egy fizikai gépen egy másik gép fizikai felépítését teljes egészében vagy részben szimuláljuk. Általában az emulációs technikák egy a fizikai rendszerhez teljesen hasonló virtuális környezetet nyújtanak (ugyanolyan architektúra esetleg más hardver komponensek), amin ugyanúgy telepíthetünk és futtathatunk, módosítás nélkül, általunk választott operációs rendszert, ahogy azt a fizikai gépen tennénk. Az emulációval lehetőség van teljesen más architektúra szimulálására is, például ezt használják a beépített rendszerek vagy a mobiltelefon

alkalmazások fejlesztői, hogy saját személyi számítógépeiken fejlesszék és teszteljék az alkalmazásaikat, vagy ugyanennek a technikának köszönhetően élhetjük újra például C64 és AMIGA játékaink által nyújtott élményeket PC-nken is.

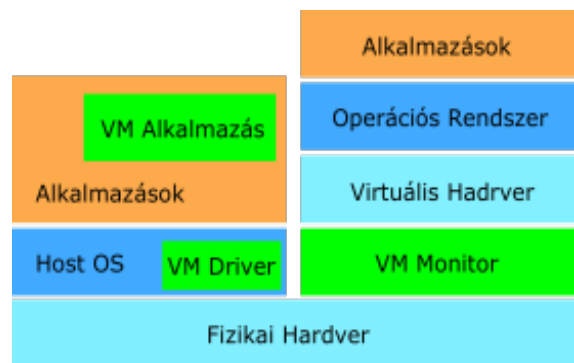
A legtöbb emulációs technika futásidőben teljesen szoftveres módon alakítja át a virtuális környezet utasításait a fizikai környezet utasításaira. Ennek a technikáknak az egyik nagy előnye, hogy sem a telepített, sem pedig az a gazda operációs rendszereken nem igényel módosítást; teljesen alkalmazási szinten valósítható meg. Segítségével szinte tetszőleges hardver elem működését (még el nem készültet is) szimulálni tudjuk. Ez az előny komoly hátránnyá alakul, amikor teljesítményre van szükségünk. Nem ritka, hogy a virtuális környezet a fizikai rendszersebességének 10-20%-át éri csak el.

Számos optimalizációs eljárást fejlesztettek az alap emulációs technikához az idők során. Ezek egyike a "dinamikus újrafordítás" néven ismert megoldás.

Ennek lényege, hogy amikor az emulátor első alkalommal találkozik egy utasítás blokkal (amit a virtuális gép kiad), abból az gazda rendszer számára közvetlenül értelmezhető bináris kódot állít elő. Ezután minden alkalommal ezt használja, amikor a virtuális gép újra ugyanazt az utasításblokkot szeretné végrehajtani. A sebesség még tovább fokozható ha a gazda és a virtuális rendszer architektúrája azonos. Ekkor ugyanis számos utasítás újrafordítás nélkül végrehajtható.

Az emulációs technikák fő felhasználási területe a rendszer fejlesztés és tesztelés ill. az olyan esetek amikor a virtualizálni kívánt operációs rendszer nem módosítható (pl.: Microsoft Windows), de az említett optimalizációk segítségével akár tényleges szolgáltatási platform is kialakítható segítségével egy konszolidációs projekt során.

Több készen megvehető alkalmazás is alapvetően ugyanazt csinálja a piacon - konkrétan, hogy létrehoz egy emulációs környezetet a Linux platformra, ahol más környezetek (többnyire Windows) futhatnak. A lehetőségek és a módszerek változók, így nehéz ezeket korrekt módon összehasonlítani. Vannak "részleges" emulációs környezetek - mint a simple crossover API-k, amelyek lefordítják a Windows-specifikus hívásokat Linux-stílusú hívásokra és fordítva. Ezen kívül vannak olyan teljes megoldások, amelyek sokkal szélesebb körű lehetőségeket kínálnak azzal, hogy egyszerre több OS futtatását is lehetővé teszik a guest/host (vendég/gazda) kapcsolat segítségével.



3. ábra: VMware Workstation architektúra

III/5. API Emuláció /Application Programming Interface/

A TransGaming Technology Cedega nevű terméke például lehetővé teszi a Linux felhasználók számára, hogy Windows alapú játékokat futtassanak. Hogyan lehetséges ez? A Windows-centrikus API-kat (függvénykönyvtárak, amelyek a szoftverek építőköveit tartalmazzák), mint a Direct3D, DirectInput és DirectSound szinte teljesen megfeleltetik más, Linuxos könyvtárakkal, mint az OpenGL, az X11 GUI vagy az Open Sound System (OSS) és az Advanced Linux Sound Architecture (ALSA).

Egy ilyen absztrakciós réteges Win32 API emulációval ellentétben, az olyan teljes emulációs környezetek, mint a Win4Lin, mindent emulálnak. A VMware pedig lehetővé teszi több OS futtatását a gazda OS-en, így ezeket manipulálhatjuk, újraindíthatjuk anélkül, hogy a gazda szoftvereinek futását meg kéne szakítanunk.

A legelterjedtebb emulációs megoldások

- [VMware](#): kétségtelen, hogy az x86-os rendszerek emulációjában a VMware termékei az úttörők, még ha napjainkra több rivális is felzárkózott hozzá. A kereskedelmi [Workstation](#) (3. ábra) és annak ingyenes, lebutított kistestvére a [Player](#) elsősorban desktop rendszereken használhatóak teszt és fejlesztési feladatokra. A szerver rendszereken, a nemrégiben ingyenessé vált [VMware Server](#) (korábban GSX), nyújt konszolidációs lehetőséget teljesen emulációs technikára alapozva. A VMware termékcsalád zászlóshajója azonban a [VMware ESX](#) és a köré épített komplett [Virtuális Infrastruktúra](#) termékcsalád. A VMware ESX lényegesen eltér a cég többi termékétől abban, hogy annak a virtualizációt végző kernel közvetlenül a fizikai gépen fut és csak a betöltéshez és a managementhez használ egy Red Hat Linux alapú rendszert.
- [MS Virtual PC és Szerver 2005](#): a Microsoft, a Connectix cég Virtual PC termékének felvásárlásával lépett be erre a piacra. Nemrégiben a Szerver megoldást ingyenesen elérhetővé tette (bizonyos megszorításokkal) és a virtualizációs megoldások beépített részét fogják képezni a következő generációs Windows szerver és desktop operációs rendszereknek.
- [QEmu](#): talán a legnépszerűbb, és leghasználhatóbb nyílt forrású emulációt végző általános rendszer. Bár a sebessége igen szerény (eltérő architektúrák között akár 10% -ra is lecsökkenhet) nagy erőssége a nyíltsága és az hogy x86-ostól eltérő architektúrát is képes emulálni. A [Win4Lin](#) termék is alapoz az eredményeire, de komoly fejlesztéseket végeztek a teljesítmény növelése érdekében.
- [Bochs](#): szintén nyílt forráskódú projekt amit jelenleg is aktívan fejlesztenek. Számos más projektnek képezte, képezi jelenleg is alapját. Sebessége még a QEmu-tól is jelentősen elmarad.
- Számos egyéb elképzelés, tanúságos próbálkozás létezik. Említés szintjén álljon itt most néhány nyílt forrású projekt: [Plex86](#), a kifejezetten kutatási, oktatási céllal fejlesztett [FAUmachine](#), [OzVM](#), [LilyVM](#) FreeBSD és NetBSD-re, [FVM](#), [KrySyS](#), [VMBear](#).

III/6. Paravirtualizáció

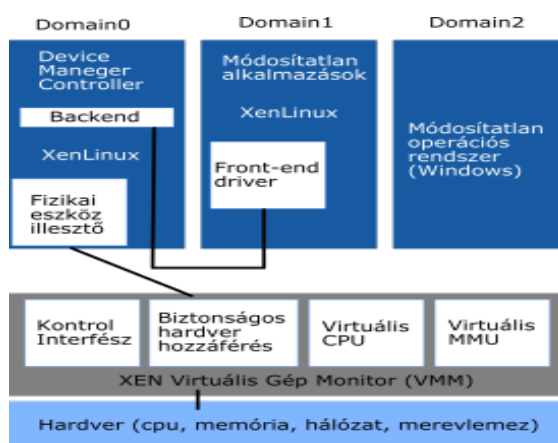
Az emulációval szemben jobb teljesítményt érhetünk el, ha lehetővé tesszük a virtuális gépnek, hogy néhány esetben közvetlenül a hardverrel kommunikáljanak.

Ebben az esetben a gazda operációs rendszer a közvetlen hardver hozzáférést egy API-n keresztül biztosítja. Ennek az API-nak a használatára a virtuális operációs rendszert fel kell készíteni, lényegében portolni kell a virtuális operációs rendszer kernelét (csak azt és semmi mást) erre az új architektúrára. Ez a módosítás szerencsére a nyílt forráskódú operációs rendszerek esetén könnyen kivitelezhető.

Ennek a megoldásnak nagy előnye az emulációval szemben, hogy sokkal jobb teljesítmény érhető el vele: gyakran a fizikai gépek sebességéhez képest 5% veszteség van.

Legelterjedtebb megoldások:

- [XEN](#) (4. ábra), amit a [XenSource](#) cég támogat. Jelenleg ez a legintenzívebben fejlődő nyílt forráskódú megoldás köszönhetően a kiváló teljesítményének és rugalmas felépítésének. A fejlesztés során kifejezetten az ipari igények kielégítésére koncentrálnak, így



4. ábra: XEN architektúra

a projekt élvezzi a legnagyobb gyártók (HP, IBM, Intel, AMD, Novell stb.) támogatását is. A nyílt forráskódú operációs rendszerek közül a Linux, FreeBSD, NetBSD, Plan9 rendszerek egyes változatainak érhetőek el a XEN platformra portolt változatai. A XEN már jelenleg is képes, hardveres támogatással futtatni módosítatlan virtuális operációs rendszereket, mint például a Microsoft Windows XP vagy 2003 szerver változatokat.

- [User Mode Linux](#). A XEN elterjedése előtt széles körben ezt a nyílt forráskódú technikát használták/használgák Linux rendszerek Linuxon történő virtualizációja. Mára a standard Linux kernel fa részévé is vált. Az UML által alkalmazott megoldás valójában egyfajta keveréke a paravirtualizációnak és az emulációnak, de az előbbihez áll közelebb. Habár elsődleges cél a fejlesztésnél a Linux kernel fejlesztés elősegítése volt, számos vállalkozás alapult az erre a technológiára alapozott hoszting szolgáltatás nyújtására is. Továbbá kiválóan alkalmazható oktatási környezetben is.
- [CooperativeLinux](#). A User Mode Linux rendszerből nőtt ki, majd alakult át. Először kifejezetten Microsoft Windows rendszeren tette lehetővé Linux operációs rendszer virtuális futtatását, mára azonban a Linuxon futó változata is használható.
- Természetesen kereskedelmi szoftverek is használják a paravirtualizációs technikát. Itt elsősorban a [Denali](#) és [Trango](#) rendszereket kell kiemelnünk. Ugyanakkor ezek elterjedtsége sokkal kisebb, mint a nyílt forrású megoldásoké.

III/7. Virtual SMP rendszerek

A virtualizációs technikák fejlődésében egy jól látható irányt képviselnek az ún. Virtual SMP rendszerek (nem összekeverendő a VMware hasonló elnevezésű technológiájával, aminek az a lényege, hogy maga a virtuális gép lehet SMP rendszer, vagyis az alaprendszer több processzort tud emulálni). Lényegük, hogy független hardver elemeket a klaszter és a virtualizációs technikák együttes használatával kössenek össze és ezáltal egy hatékony és jól skálázható platformot teremtsenek. Ez a technológia, mivel a memória és az I/O műveletek megosztását is megcélozta, igen érzékeny az önálló fizikai egységeket összekötő hálózati technológiára, különösen a hálózati késleltetésre. Jelenlegi trendek azt mutatják, hogy az [Infiniband](#) alapú hálózati elemek teszik lehetővé leginkább azt, hogy a virtuális SMP rendszerek által támasztott követelményeket kielégítse. Számos szoftvergyártó próbálkozott ilyen rendszerrel, de kereskedelmi forgalomba egy sem került. A nyílt forráskódú projektek közül a [Kerrighed](#) névre keresztelt Linux kernel módosítás foglalkozik a Virtual SMP rendszer megvalósításával, de jelenleg még csak kezdeti stádiumban tartanak.

III/8. Natív virtualizáció

A szoftveres megoldások mellett a két nagy processzor gyártó ([INTEL VT](#), [AMD SVM](#)) egyaránt bejelentette, hogy 2006-ban elérhetővé teszi azokat a processzorait és alaplapi chipjeit, melyekkel egyes virtualizációs technikákat hardver szinten is támogatnak. Jelenleg leginkább a XEN és a VMware ESX tudja kihasználni a processzor ilyen képességét, de a többi fejlesztő csapat intenzíven dolgozik ezeknek a processzoroknak a támogatásán.

III/9. PC architektúrán túl

Érdekes és tanúságos tény, hogy a virtualizáció technológiája a 60-as évek közepe óta rendelkezésre áll a IT infrastruktúra építőinek, mégis mára vált általánosan ismerté és még egy-két év kell, hogy általánosan elfogadottá váljon. A fő oka talán ennek a késlekedésnek, hogy virtualizációs megoldások korábban döntően csak a mainframe és a szuperszámítógép kategóriájú gépeken voltak elérhetőek. Ez a helyzet mit sem változott. Ezek a "nagykategóriájú" rendszerek többsége alapértelmezésben tartalmaz vagy a partícionálás vagy pedig a hardveresen támogatott

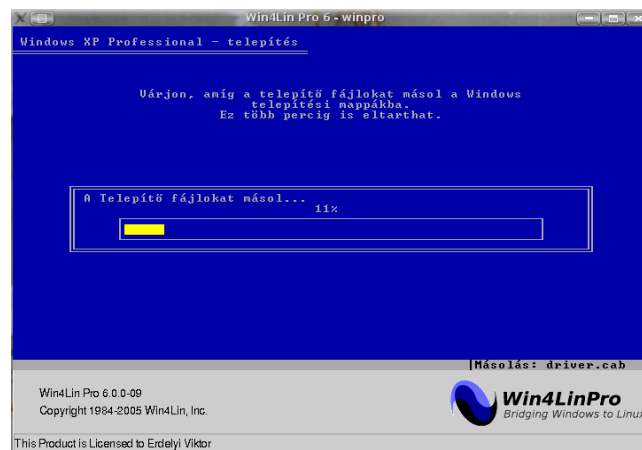
hypervisor alapú megoldásokat. Az utóbbi megoldásra jó példa, amely még elérhető e kategóriából, az [IBM Power 5](#) processzoraira épülő rendszerek.

IV. Virtualizációs megoldások

IV/1. Win4Lin vagy VMware?

A különbség röviden az a Win4Lin és a VMware között az, hogy a Win4Lin egy speciális környezet, ami Windowst emulál Linux alatt. Másrészt viszont a VMware célja az, hogy számos operációs rendszert legyen képes futtatni több platformon. Ezért van az, hogy a Win4Lin kevésbé terheli le a rendszert, feláldozva ezzel a rugalmasságot (megj.: a tapasztalat mást mutat...).

A felszín alatt a két rendszer teljesen különböző. A Win4Lin az ingyenes és népszerű QEMU platformon alapul, ami a felhasználói módú emulációt alkalmazza a keresztplatformos futtatás és a teljes hardver-emuláció megvalósítására. A VMware a saját, egyedi megoldásait alkalmazza ugyanerre.



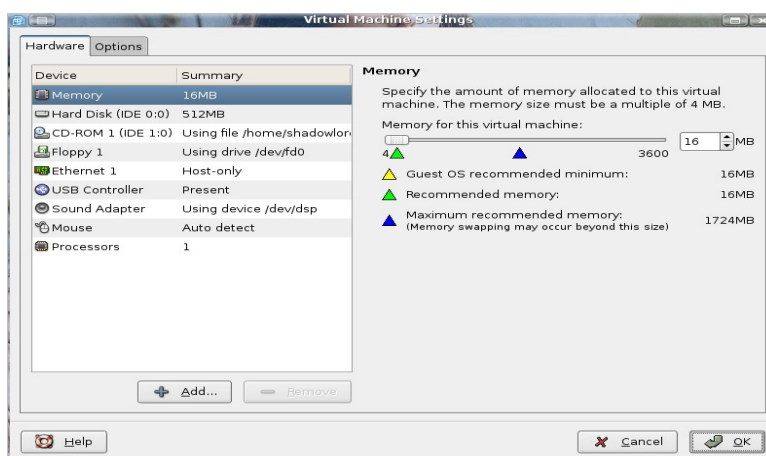
Win4Lin - XP telepítés



IV/2. VMware

Fejlesztő:	VMware Inc.
Legfrissebb stabil kiadás:	5.5.3
Operációs rendszer:	Windows, Linux, BSD
Kategória:	emuláció, virtualizálás
Licenc:	VMware Inc.
Weboldal:	http://www.vmware.com

A VMware Inc. egy szoftvercég, amely operációs rendszerek emulációjára és virtualizálásra specializálta magát és az egyik legismertebb terméke a „VMware Workstation”. 2004 elején felvásárolta az EMC Corporation a VMware-t.



1. Termékcsalád

1.1. VMware Workstation

A „VMware Workstation“-nel Linux s Microsoft Windows alatt egy komplett számítógép emulálása lehetséges. Ezeken az emulált gépeken különböző oprendszeret pl: Windows, Linux és sok más oprendszert is lehet emulálni, persze a valódi gépünk erőforrásaitól és oprendszerétől is nagymértékben függ a sebesség. Sajnos egy Windows 2000-es emulált virtuális gép, amely egy öregebb Windows NT 4-en fut, nem tudja támogatni az USB-t. Viszont a VMware 5.5 részben lehetővé teszi a használatát.

Hasznos szolgáltatás a Workstationben a VMware Tools. Ez egyfajta hidat képez a hardver és a virtuális gép között. A virtuális gép által küldött I/O jeleket (azokat, amelyek gyors reagálást kívánnak – lásd videojelek, egér), egyenesen a hardverhez továbbítja, ezzel jelentősen növeli a virtuális gépek használhatóságát (persze csak azokon a rendszereken, amelyekre készült VMware Tools csomag). Nyilván nem nyújt olyan teljesítményt, mintha natív módban futna a rendszer, de a szoftveres emulációhoz képest nagy előrelépés.

Ha virtuális gépeket szeretnénk használni, akkor szükségünk lesz egy erős gépre, hiszen nemcsak a gazdarendszer folyamatainak, hanem a virtuális rendszerprocesszeknek is biztosítanunk kell megfelelő mennyiségű erőforrást (CPU, RAM, merevlemez).

Voltaképpen bármilyen rendszert és programot futtathatunk a Workstationnel, ha az a virtuális hardvert ismeri:

DOS

Technical specifications for the Lamborghini Countach:

Layout:.....mid engine/rear drive	Approximate Price:...	100
Engine type:.....dohc 4-valve V-12\$130,000	90
Displacement:.....51b7cc		80
Compression ratio:.....9.5:1	0-60mph:.....5.2s	70
Bhp @ rpm, SAE net:.....420 @ 7000	0-100mph:.....12.0s	60
Torque @ rpm, lb-ft:.....341 @ 5000	1/4 mile:.....13.7s	50
Transmission:.....5 sp manual	@ 100mph	40
Braking from 80mph:.....252ft	Top speed:.....173mph	30
Tires:.....Pirelli P7R; lb/Bhp:.....8.1	Lateral Accel: 0.88g	20
.....325/50VR-15 (f)		10
.....345/35VR-15 (r)		0

Acceleration graph: X-axis: Time (s) 0-40, Y-axis: Acceleration 0-100. Curve shows rapid acceleration reaching 100 by 10s.

DOS File Directories:

```

C:\ Name      C:\GAMES
      Name      Name
      .
      Z400AD
      CUG1TOYS
      DVA
      F29
      IGNITION
      KEEN
      MAR10
      PAPERBOY
      SUPAPLEX
      TD1
      TD2
      TD3
      WOLF3D
      cat      exe

C:\ Name      C:\
      Name      Name
      .
      DOS
      GAMES
      INSTALL
      MC
      MORTON
      PROGRAMS
      RETAL
      TEMP
      WINDOWS
      win20  386
      autoexec  bat
      command  com
      treeinfo  ncd
      config   sys
      io       sys
      msdos    sys
  
```

Win95, WinME, WinXP:

Windows 95 Desktop Environment:

- Desktop icons: Saját gép, Az Internet, Lomtár, Posta, Ignition.
- Open windows:
 - Intéző - Win95 (C:): File Explorer showing drive contents.
 - Total Commander 6.53 - NOT REGISTERED: File manager showing file lists for 'winme' and 'c:*'.

Name	Ext	Size	Date	Attr
[Dokumentumok]	<DIR>		2006.11.11 11:53	----
[Program Files]	<DIR>		2006.11.11 11:36	----
[Windows]	<DIR>		2006.11.11 11:36	----
Autoexec	bak	194	2006.12.21 12:41	-a--
Config	bak	0	2006.12.21 12:41	----
NERO_B_	exe	70 144	2004.11.25 18:36	-a--
Config	sys	0	2006.12.21 12:41	-a--
 - Rendszer tulajdonságai: System Properties dialog box showing 'Microsoft Windows Millennium Edition 4.90.3000'.

Fontos, és egyben nagyon hasznos tulajdonsága a Workstationnek az úgynevezett Saved State. Ez egyfajta hibernálást jelent, egy adott állapotában megállítjuk a virtuális gép működését, a memória tartalmát lemezzre írjuk, és bármikor folytathatjuk a használatát, egy „Play” gomb megnyomásával.

Az alsó ábrán láthatjuk, hogy a jobb alsó sarokban még egy screenshotot is kapunk a virtuális gép aktuális állapotáról.

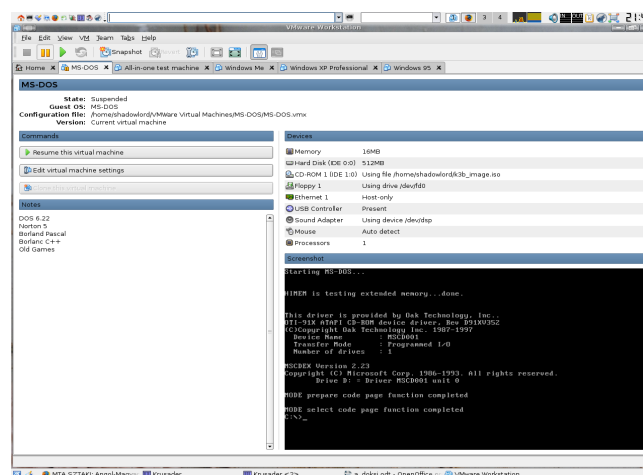


1.2. VMware GSX, VMware Server és ESX

Szerver üzemeltetésére van pl. a VMware GSX, s a Professzionális megoldás az ESX Server. A VMware GSX utódja a VMware Server, amely szabadon letölthető, s így lehetővé teszi a virtuális világba való belépést s kipróbálását. A GSX-re épül fel. Az előállító kínál supportot, s lehetséges a Virtual Centerhez való kapcsolódás is. VMware GSX s VMware Server úgynevezett Hosted-rendszerek, ez azt jelenti hogy szükségük van egy létező operációs rendszerre, amely lehet Windows vagy Linux.

VMware ESX Server egy saját VMware-magra/Kernel/ épül amelynek e miatt nincs szüksége egy leendő operációs rendszerre. Ez a verzió a Professzionális felhasználók, s cégek felé irányul, mivel a software segítségével egy cég kiadásait jelentősen csökkenteni lehet. A 3-as verziójú ESX server már támogatja az automatikus Loadbalancing -et.

Most már a VirtualCenter-rel teljes virtuális infrastruktúrákat lehet megvalósítani. Ez azt jelenti hogy pl.: egy hálózatban 40 szervert látunk, s igazából csak 2 fizikai szerver van, a többi virtuális szerver.



1.2.1. VirtualCenter

A VirtualCenter megvalósítja a GSX és ESX-Szerverek, s azoknak a virtuális gépek központi adminisztrációját. Így nem kell mindegyik szerverre, s azok virtuális gépeinek az adminisztrációs konzoljára bejelentkezni. az ESX/GSX szerverek és az összes virtuális gép központi adminisztrációja teljesítménymérés az ESX/GSX szervereken s mindegyik VM-en Vmotion követelménye szerverek megfigyelése s riasztás

A VirtualCenter több szerverkomponensből áll: adatbázis /Access/, Oracle, MYSQL Server, egy vagy több clientekből. A Client és a Server közötti kommunikáció Web Service -el valósul meg.

1.2.2. Vmotion

A VMotion megvalósítja a VirtualCenter-rel együtt, a futó virtuális gépek áttelepítését ESX-Server-ek közt. A Storage Area Network amelyben a virtuális merevlemezek /rövidítve: VMDK/ egy Shared Storage -on fekszenek, kötelező követelmény.

A VMotion Agent-ek elosztását a Hostokra a VirtualCenter intézi. Így minden Hostnak egy csakis erre a célra "épített" virtuális Switchhez kell kapcsolódnia s ez egy fizikai hálózati kártyához van

kötvé. Így költöztetjük át az adatokat a létrehozott "Managed-Hálózaton".

Egy futó virtuális gép átköltöztetése egy másik ESX szerverre a következőképpen működik: először a virtuális gép merevlemeze írásvédetté változik meg, minden ettől a pillanattól való változás egy Redo-Log -ba íródik. Ugyanez történik a virtuális memóriában és virtuális processzorban lévő állapotadatokkal. Mindezek közben a virtuális gép az első ESX-en tovább fut. Ha a merevlemez(ek) tartalma és utána a vezérlés a második ESX-en megérkeztek s feladatát felvette akkor az a VMotion elkezd a Redo-Log -ot az első ESX-ről a másodikra integrálni. Csak ha majdnem 100%-os a integrálás csak akkor állítja meg az első ESX a virtuális gépet. A maradék a Redo-Log -ból átvitelre kerül s a virtuális memória és processzor tartalma is integrálódik, így a második gépen teljes mértékben áll a rendszer, s tovább folytatja a feladatait. Így a felhasználó által észrevett kiesés csupán egy-két másodperc/a Redo-Log és memória átvitelére igénybe vett idő/. Mivel a memóriatartalom kerül átvitelre ezért a processzor típusa a két gépben ugyanazkell hogy legyen.

1.3. VMware Player

VMware a VMware Player -rel egy szabad software-t kínál amellyel már a készen konfigurált virtuális gépeket "le tudjuk játszani". A Virtual Machine Center oldalon le lehet tölteni kész Image-eket amelyeket le tudunk játszani. S így lehetővé válik minden befektetés nélkül például egy Linux környezet letöltése amellyel Windows alatt relatív biztonságosan tudunk netezni.

1.4. VMware P2V Assistant

VMware P2V (physical-to-virtual) Assistant egy Migrálási eszköz Windows OS-alá amely fizikai rendszerekből virtuális rendszereket generál VMware-hez.

2. Felhasználhatósága az informatika környezetben

A softwarefejlesztés terén megkönnyítik a munkát, mivel több különböző OS folyamat egyszerre futhat. Ezzel különböző Release-eket könnyedén lehet tesztelni. A Snapshot segítségével úgymond visszaállítási pontokat lehet menteni, amelyekhez bármikor vissza lehet térni. Az installációk Image-be való elmentése miatt ezeket hálózatunkon el tudjuk küldeni fejlesztőknek.

Mivel régebbi OS-eket is az aktuális rendszerekbe tudjuk integrálni, ezért az öregebb OS-t korlátlanul tudjuk használni, amíg nem lép fel problémás hardwarehozzáférés (gondoljunk itt az újabb videovágókra és az igényes grafikus alkalmazásokra).

Továbbá a VMware "híd"-ként szolgál Windowstól Linuxhoz, mivel minden Windows OS minden Linux változatba integrálható. Így minden Windows programot tudunk futtatni Linux alatt anélkül hogy párhuzamosan egy merevlemezünkön kép partíción a Windows és a Linux installálva van. A virtuális Windows OS egyszerűen egy ablakként jelenik meg Linuxba, persze mellette Linuxba tovább dolgozunk. Természetesen Windows alatt is lehet ezt csinálni Linuxszal vagy más OS-sel.

Nagy előny a Disaster Recovery. Mivel egy virtuális server olyan könnyen másolható, mint egy könyvtár (minden beállítással együtt), ezért elmenthető egy külső adathordozóra az Image, pl: USB HDD. Szükség esetén akár serverparkokat tudunk legrövidebb időn belül "újraéleszteni", s mindezt az összes beállítással. Az időigényes és persze nem olcsó rekonstrukció, így már csak történelem.

Mivel a virtuális gépek hardware-je mindig ugyanaz, ezért egy átköltözés egyik fizikai gépről a másikra teljesen problémamentes és elhanyagolható, azaz egy könyvtár másolása. Ez persze nagyon előnyös Windows rendszereken.

IV/3. UNIX - 2.6.20 kernel újdonságok: kernel alapú virtualizáció (KVM), szervizelhetőséget javító funkciók (kdump, SystemTrap)

Az egyik figyelemre méltó újdonság a KVM lesz. A KVM jelentése "Kernel-based Virtual Machine". Ahogy azt a neve mutatja, a patch a Linux kernel virtualizációs képességeit fogja javítani. A KVM a honlapja szerint teljes virtualizációs megoldást nyújt x86 architektúrájú gépeken futó Linuxra. A KVM egy betölthető kernel modulból (kvm.ko) és user space komponensekből áll.

A KVM segítségével felhasználója módosítás nélküli Windows és Linux image-eket futtathat egymástól függetlenül. Mindegyik virtuális gépnek saját virtualizált "hardvere" van, azaz szeparált hálózati kártyával, lemezzel, grafikus adapterrel, stb. rendelkezik. A projekt honlapja szerint a KVM [stabilnak mondható és elég gyors desktop feladatok ellátására](#) a mai modern processzorokon. Szerver funkciók ellátására kicsit másképpen optimalizált virtualizáció szükséges. Ennek elkészítése folyamatban van.

Mi az ami jelenleg működik?

- Intel-based hosts (requires VT capable processors)
- AMD-based hosts (requires SVM capable processors)
- Linux guests (32-bit and 64-bit)
- Windows guests (32-bit)
- SMP hosts

Folyamatban levő munkák

- Optimized MMU virtualization
- Live migration

Tervezett funkciók

- SMP guests

A KVM-ről bővebben a [honlapján](#).

A 2.6.20-as Linux kernel a virtualizációs képességek javítása mellett egyéb fejlesztéseket is hoz majd. Érdemes megemlíteni a szervizelhetőséget, hibakeresést megkönnyítő kdump vagy a SystemTrap körüli munkákat.

A kdump körüli fejlesztések segítik, hogy minél megbízhatóbban és gyorsabban készítsünk crash dump-okat annak érdekében, hogy offline analizálhassuk a rendszer működését.

Az új SystemTrap funkció ezzel ellentétben lehetővé teszi a fejlesztők és rendszermérnökök számára, hogy a rendszert online állapotban, működés közben analizálhassák teljesítmény csökkenés vagy újrafordítás szükségessége nélkül.

Andrew Morton különösen jelentősnek nevezte a kdump-pal kapcsolatos munkákat, hiszen ez az egyetlen olyan crash dump eszköz, amely elfogadásra került a mainline Linux kernelbe. A főkarbantartó elmondta, hogy várakozásaik szerint az eszköz igazán hasznos lesz a kernelfejlesztői csapat számára, mert használatával részletes, kernel bugokkal kapcsolatos információkhoz juthatnak.

Bővebben [itt](#).

V. A két nagy processzorgyártó és a virtualizáció

V/1. INTEL

A virtualizáció segítségével a platformok képesek több operációs rendszert vagy alkalmazást futtatni független partíciókban vagy „konténerekben”, amelyek olyan specifikus igények kielégítésére is alkalmasak, mit az IT menedzsment szolgáltatások és a hálózatba kapcsolt eszközök védelme.

“Az Intel® Virtualizációs Technológiájához (Intel® Virtualization Technology) hasonló innovációkkal az Intel élen jár az üzleti szervezeteket ma érintő kihívások leküzdésében., – nyilatkozta Robert B. Crooke, az Intel Üzleti Ügyfél Csoportjának (Business Client Group). „Iparági elemzők egyöntetű megítélése szerint a virtualizáció a személyi számítógépekhez kapcsolódó legforradalmibb technológia ebben az évtizedben, és a technológiánkat az iparági erőfeszítésekkel ötvözve segíteni fog abban, hogy a jövőben ez legyen a fejlesztések fő csapásiránya.”

A virtualizációs támogatás segítségével az üzleti szervezetek teljes ellenőrzésük alá vehetik a PC-k egy részét biztonsági vagy menedzsment szolgáltatások futtatására anélkül, hogy akadályoznák a felhasználó munkáját, vagy akár a kritikus alkalmazásokat is megváltoztathatják. Az üzleti szervezetek tehát magasabb szinten lesznek képesek megvédeni magukat rosszindulatú kódok vagy vírusok ellen oly módon, hogy a hálózati forgalmat megszűrik egy elkülönített IT partíción keresztül, mielőtt az elérne a felhasználóhoz.

Több partíció létrehozása a PC-ken növeli a kezelhetőséget azáltal, hogy támogatást biztosít a régi alkalmazásokhoz úgy, hogy közben kipróbálható és bevezethető egy új működési környezet.

Intel® Pentium® 4 –es processzoron alapuló asztali számítógépeket jelenleg az Acer, a Founder, a Lenovo és a TongFang kínál. A Lenovo már kínál Intel® Virtualizációs Technológián alapuló tesztrendszereket egyes kiválasztott ügyfelek számára, valamint 2006 első negyedévéől várhatóan számos neves rendszert integrátor nyújt majd széles körű támogatást asztali PC megoldások számára. Emellett az iparág vezető, virtualizációs technológiát kínáló szereplői, mint például a VMware, a Microsoft és a Xen is csatlakoztak az Intel jövőbeli, Intel processzorokon alapuló PC-k és szerverek Intel Virtualizációs Technológiával történő támogatásához.

A beépített, világszínvonalú IT biztonságot és menedzsment képességeket tovább fejlesztve az Intel egy másik prémium platform fejlesztést is kínál Intel® Aktív Menedzsment Technológia (Intel® Active Management Technology) néven, az Averill kódnevű, 2006 folyamán piacra dobandó kétfagos, Intel® Professional Business Platform virtualizációs támogatásával együtt.

A virtualizációs képességeket noteszgépekre is kiterjesztik, amelyet először az Intel® Centrino® Mobil Technológián alapuló platformokkal vezetnek be 2006-ban.

Az üzleti vállalkozások további előnyökre tehetnek szert a virtualizáción keresztül szerveralkalmazásaik számára is megnövekedett funkcionalitás és munkafolyamat- támogatás segítségével. Az Intel az elmúlt héten jelentette be, hogy megkezdte a virtualizációs technológiát hardveresen támogató Intel® Xeon® MP processzorainak szállítását. A cég Intel® Itanium® processzoron alapuló, ugyanezen technológiát alkalmazó platformjai 2006 közepétől érhetőek majd el.

A virtualizáció nyújtotta előnyök kiterjeszthetők az otthoni tevékenységekre és szórakozásra is. A jövőben az otthoni számítógép-felhasználók képesek lesznek virtuális „partíciók” létrehozására és ezáltal számos felhasználói környezet elkülönítésére, mint például a dedikált böngészők, produktivitás szoftver alkalmazások, személyi videó felvevők, és a virtualizációs technológia

lehetőséget teremt a szolgáltatóknak arra, hogy távolból is diagnosztizálják és megoldják a felmerülő problémákat.

V/2. AMD

Az AMD az In-Stat elemzői cég által San Joséban megrendezett [Fall Processor Forum](#) nevű rendezvényen beszélt a fejlesztés alatt álló Pacifica technológiáról. Az FPF elsősorban szakmabelieknek, vagyis mérnököknek, vállalati vezetőknek szól, általában a legújabb fejlesztéseket, az azokkal kapcsolatos terveket, illetve a várt eredményeket ismertető előadásokkal, bemutatókkal. A mostani rendezvényen - The Road to Multicore - elsősorban az új ikermagos chipekről, és az ezekkel érkező megoldásokról tartottak beszámolókat.

Az eWeek [tudósításából](#) kiderül, hogy az AMD a Pacifica kódnéven ismert virtualizációs technológia körüli felhajtásra koncentrált, kihangsúlyozva annak forradalmi, korszakalkotó jellegét. "Szeretném, ha mindenki tisztán látná, hogy a virtualizáció a következő nagy dobás az x86 architektúra terén. Az embereknek minél hamarabb meg kell érteniük a technológia lényegét, hogy hatékonyan használhassák fel azt" - jelentette ki Kevin McGrath, az AMD Advanced Architecture csoport igazgatója. Beszédében megemlítette, hogy a [Pacifica](#) révén feleslegessé válnak azok a komplex, bonyolult programozási bűvészmutatványok, amelyek lehetővé tették a virtualizációs szoftverek (pl. VMware) korai megjelenését.

	2005	2006	2007
CPUs	Dual Core Complete 64-bit offering	"Pacifica" Virtualization "Presidio" Security DDR2, Lower Power	New Core Larger Caches DDR3 xGHz HT
Performance (WS/Desktop)	SLI Graphics RAID	DDR2 "Aero Glass" graphics "Pacifica" Virtualization "Presidio" Security TPM	DDR3 xGHz HT PCIe Gen II
Mainstream Stable Platform	HD Audio GbE + WLAN	UMA + PCIe Option	CSIP Managed Platform
Blade PCs, Thin Clients	UMA, sub-30W Low Noise Cooling Custom FF	Sub-10W DDR2 "Pacifica" Virtualization "Presidio" Security, TPM	DDR3 xGHz HT

Az AMD szerint a hardveres eljárás debütálásával elsősorban a szoftverfejlesztők munkáját könnyítik meg, egyszersmind javítani tudnak a sebességen és a számítási igényeken is. Igaz, a már kész termékek esetében ez kevésbé érvényesül, hiszen fejlesztőik már rengeteg munkát beleöltek szoftvereikbe, ám új lehetőségeket nyithatnak meg az újonnan induló projektek előtt. Ez azonban mindössze a kezdeti lépés lesz, és belátható időn belül szeretnék tovább bővíteni a technológiát, például az I/O-műveletek virtualizációjával.

A gyártó a technológia két óriási előnyét hangsúlyozza. Egyrészt a vállalatok minden eddiginél hatékonyabban ki tudják majd használni a rendelkezésükre álló gépparkot (szervereket), másrészt a biztonság terén is új megoldásokat próbálhatunk ki. Szabadon böngészhetünk például az interneten egy külön e célból létrehozott virtuális partícióval, anélkül, hogy bármilyen veszélynek tennénk ki fő operációs rendszerünket - mindezt könnyedén és gyorsan megvalósítva.

Eközben természetesen az Intel is készült saját megoldásával, amely Intel Virtualization Technology, illetve Vanderpool néven ismeretes.

VI. Referenciák, bővebb információ:

<http://www.kernelthread.com/publications/virtualization/>

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

<http://en.wikipedia.org/wiki/Z/VM>

http://www.tomshardware.hu/howto/06q2/060616/run_windows_on_linux-01.html

<http://www.sg.hu/cikkek/40408>

http://www.avaxio.hu/virttech_intro

<http://hu.wikipedia.org/wiki/Vmware>

<http://hup.hu/node/33225>

http://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine

archivum.lme.hu/2005/Konf/Konferencia_kiadvany_v1.0.pdf

www.intel.com/cd/corporatepressroom/emea/hun/247711.htm

www.inf.u-szeged.hu/oktatas/kurzusleirasok/1102.xml

Készítették: Jámbor Dávid, Erdélyi Viktor

2007-03-04