

Virtualizációs Technológiák Mérési Útmutató

Informatikai technológiák laboratórium I.
összeállította: Tóth Dániel
Méréstechnika és Információs Rendszerek Tanszék

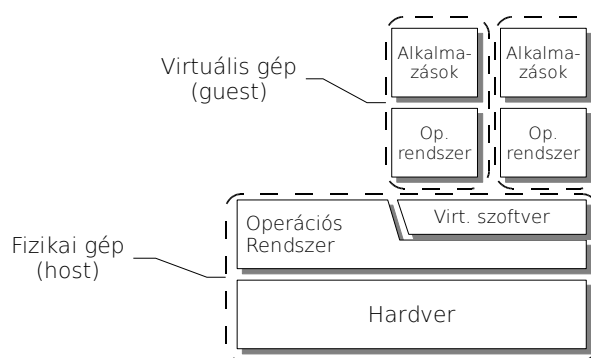
2009. március 25.

A mérés célja

A mérés célja, hogy a hallgatók megismerkedjenek egy nagyvállalati szerver virtualizációs megoldással, annak használatával és alapvető karbantartásával. A mérés során egy előre telepített VMware ESX Server virtualizációs környezetben kell feladatokat végezni. A feladatok részben a virtualizációs rendszer konfigurálásához, részben pedig a virtuális gépek létrehozásához, módosításához kapcsolódnak, egy rövid reprezentatív áttekintést adva egy virtualizációs rendszer mindennapi használatáról. A mérés során lehetőség nyílik a virtualizáció központi menedzserjének megismerésére is a VMware VirtualCenter Server segítségével. Ilyen központi menedzserhez kapcsolódó feladat például a virtuális gépek működés közbeni mozgatása, hibatűrő fürt kiépítése, automatikus terheléelosztás, vagy virtuális gépek automatizált üzembehelyezése.

1. Bevezető

Virtualizációnak nevezzük az olyan technológiákat, melyek lehetővé teszik, hogy egy fizikai számítógép vagy annak valamilyen hardver erőforrása több, esetleg eltérő típusú virtuális számítógépek, illetve erőforrásnak látszódjon a gépen futó szoftverek számára.



1. ábra.

Az első virtualizációs megoldásokat az IBM fejlesztette ki a 60-as években a CP-40-re épülő nagygépes rendszereiben. A modern operációs rendszerek is egyfajta *erőforrás virtualizációt* valósítanak meg azzal, hogy elfedik a hardvert a gépen futó folyamatok elől és egy egységes szoftveres interfészt biztosítanak az erőforrások dinamikus lefoglalására és elérésére. Teljes számítógép virtualizálása (ún.: *platform virtualizáció*) oly módon, hogy a gépen több operációs rendszer is futtasson, sokáig csak nagygépes (mainframe) környezetben volt elérhető. Ennek egyik fő

oka, hogy a virtualizáció különleges követelményeket¹, támaszt a processzor utasításkészlete, a memória kezelő egység (MMU, Memory Management Unit) illetve a perifériák programozói felületével szemben. A hagyományos PC-kben található x86-os processzor architektúra tervezésekor ezeket a követelményeket nem vették figyelembe, ezért PC-re platform virtualizáció csak a közelmúltban különleges szoftveres megoldásokkal (futás közbeni kód módosítás, *binary translation*) vált elérhetővé. Ilyen megoldások egyik úttörője és sokáig a piacon egyedüli szállítója a VMware cég volt. A virtualizációs megoldások rohamosan bővülő piacán időközben számos más szereplő és termék is megjelent: x86 platformra: Citrix XenServer, Parallels, Sun VirtualBox, Microsoft VirtualPC és Hyper-V, míg pl. Power platformra az IBM Logical Partitions (LPAR).

A virtualizáció külön ágát képviselik az *alkalmazás szintű futtatókörnyezetek* ami szoftverek futtatását teszi lehetővé eltérő architektúrájú processzorokon, szintén futási idejű újrafordítással. Ennek legismertebb példái a JavaVM (Java Virtual Machine) és a Microsoft .NET CLR (Common Language Runtime). A fentebb ismertetett platform virtualizáció többnyire a hardverrel azonos architektúrájú processzort biztosít a virtuális gépek számára, a JavaVM és a .NET CLR azonban a hardvertől teljesen eltérő utasításkészletet biztosít az alkalmazások számára.

Léteznek még a virtualizációnak egyéb fajtái, melyek valamilyen korlátozott formáját valósítják meg és leginkább az *erőforrás virtualizáció* körébe sorolhatóak be. Legfontosabb az *operációs rendszer szintű virtualizáció*, vagy *konténer alapú virtualizáció* melynek lényege, hogy egyazon operációs rendszeren belül elkülönített végrehajtási környezeteket (*container*, *jail*) alakítunk ki, amik nem tudnak egymás létezéséről, elkülönített erőforráskészlettel (pl. fájlrendszer, hálózati portok) rendelkeznek, úgy viselkednek, mintha külön operációs rendszereken futnának, ám a kernel valójában közös. Ilyen megoldások például, az OpenVZ, Linux VServer, Solaris Containers, IBM AIX Workload Partitions (WPAR). Operációs rendszer szintű virtualizációt leginkább az különbözteti meg a platform virtualizációtól, hogy az előbbi az operációs rendszer erőforrásainak virtualizálását jelenti, míg az utóbbi hardver erőforrásokét. Természetesen vannak a kettőt kombináló megoldások is.

*Alkalmazás virtualizációnak*² nevezik az olyan operációs rendszer szintű virtualizációt megvalósító termékeket, amik kifejezetten azt a célt szolgálják, hogy nagy és komplex alkalmazásokat telepítés nélkül lehessen futtatni egy operációs rendszer felett. Az izolált környezetek közösen használják az operációs rendszer kerneljét, ám elkülönített programkönyvtárakat és konfigurációs fájlokat (Windows esetén Registry-t) kapnak, ami egy önálló, egyszerűen elindítható csomagban terjeszthető. Ilyen termékek például a VMware ThinInstall vagy a Microsoft Softgrid és bizonyos mértékig a Citrix XenApp.

2. A virtualizáció szerepe az IT-ben

A virtualizáció nagyvállalati környezetben megfigyelhető nagyszabású terjedéséért elsősorban három fő szempont felelős:

- Erőforrás konszolidáció
- Szeparáció (hibatűrés és biztonság)
- Távoli menedzselhetőség

Gyakori, hogy a hardverek teljesítményét, erőforrásait nem lehet optimálisan kihasználni, mert az alkalmazások *funkcionális* és *„nemfunkcionális”*³ (más szóval *extrafunkcionális*) követelményei megkötéseket támasztanak azzal kapcsolatban, hogy miket telepíthetünk egy gépre.

¹Gerald J. Popek and Robert P. Goldberg (1974). "Formal Requirements for Virtualizable Third Generation Architectures". Communications of the ACM 17 (7): 412 -421

²Nem összetévesztendő az alkalmazás szintű futtatókörnyezetekkel. Sajnos az elnevezések nem egyértelműek, mert gyakran „application virtualization” néven említik mindkét féle megoldást. Ennek oka részben abban keresendő, hogy számos futtatókörnyezet mellékesen nyújt támogatást alkalmazás izolációra illetve telepítés nélküli önálló csomagból indítható alkalmazásokra is (pl. Java WebStart).

³A teljesítmény is egy jellegzetesen nemfunkcionális követelmény, itt most a teljesítménynél fontosabbnak ítélt nemfunkcionális követelményekről van szó, pl. biztonságról

Például különböző operációs rendszeren futó szolgáltatásokat nem telepíthetünk egyazon gépre. Ez nemcsak szerverek, hanem asztali gépek, munkaállomások esetén is gyakori probléma, például többplatformos szoftverfejlesztésnél. Biztonsági szempontból kritikus alkalmazást sem célszerű telepíteni olyan gépre, ami az Internet irányába nyitott szolgáltatást nyújt. Számos nagyvállalati szoftver olyan követelményeket támaszt az operációs rendszer környezetére, ami megakadályozza, hogy más szoftverekkel együtt fusson egy gépen. Olyan operációs rendszerek menedzsmentje eleve nehézkessé válik, amin túlságosan sok heterogén szolgáltatás fut egyszerre. Ezen követelmények miatt gyakran kell külön gépre telepíteni olyan alkalmazásokat, amik nem használják ki, vagy csak az idő kis részében használják ki a hardver kapacitását.

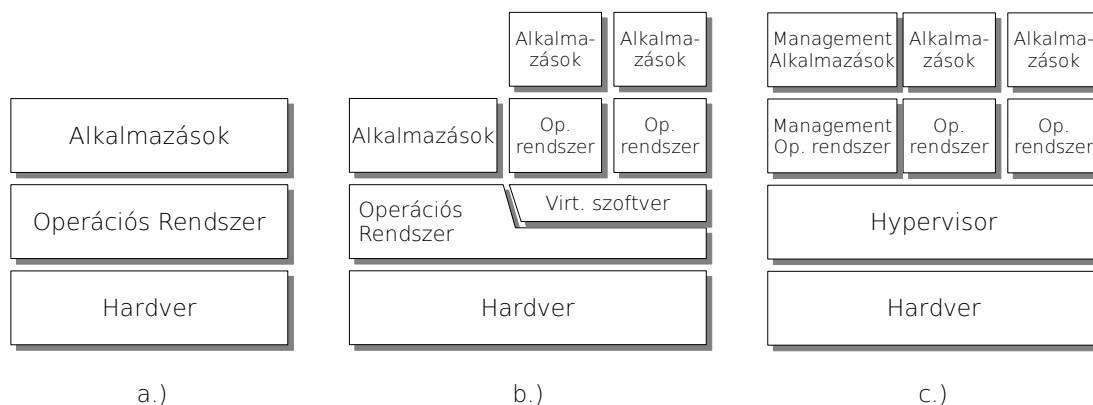
A virtualizáció lehetővé teszi, hogy szeparált környezeteket alakítsunk ki, így egy gépre összehajthatunk olyan szolgáltatásokat, amiket a fent felsorolt vagy ehhez hasonló okok miatt dedikált gépre kellene telepíteni. Ezzel a hardver kihasználtsága javítható, ezáltal a szolgáltatás kiépítési és üzemeltetési költsége (jellemző mérőszám a *Total Cost of Ownership, TCO*) jelentősen csökkenthető. A rendelkezésre állás javítható, hiszen egy meghibásodás a virtuális gépen belülről elszigetelődik, a fizikai gépen futó egyéb szolgáltatásokat nem érinti. Hasonlóképpen a biztonsági kockázat is csökken, mert egy esetleges illetéktelen hozzáférés is virtuális gépen belülről korlátozódik.

Nagy számú gép karbantartása és a változó igények esetén új szerverek üzembeállítása, ideiglenes tesztkörnyezetek kialakítása gyakori feladatok nagyvállalati, de sok esetben kis- és középvállalkozások informatikai rendszereiben is. A virtualizáció lehetővé teszi dinamikus erőforrás allokációt, ami új gépek üzembeállítását könnyíti meg. További fontos szempont a távmenedzsment. Léteznek megoldások számítógépekhez való távoli hozzáférésre, ám ezek sokáig csak drága felső kategóriás szervergépekben voltak megtalálhatóak. Ezen megoldások jó része is korlátozott, vannak olyan feladatok, amik nem végezhetők el fizikai hozzáférés nélkül (konfiguráció megváltoztatása). A virtualizáció, lévén, hogy szoftveresen hozza létre a virtuális környezetet, lehetőséget biztosít arra, hogy a virtuális gépeken olyan műveleteket is elvégezzünk távolról, amiket fizikai gépen csak fizikai hozzáféréssel tehetünk meg.

Fontos megjegyezni, hogy a virtualizáció minden esetben teljesítményvesztéssel (*overhead*) jár, ez bizonyos esetekben elenyésző (néhány %), ám vannak olyan szélsőséges esetek is, ahol csak 10% a virtualizált rendszer teljesítménye a közvetlen hardveren futtatáshoz képest.

3. Platform virtualizációs megközelítések

Teljes számítógép virtualizálására két megközelítés terjedt el.



2. ábra.

Az ún. *hosted* vagy 2. típusú virtualizáció esetén a hardverre közvetlenül telepítve van egy operációs rendszer (a továbbiakban *host* operációs rendszer) és efelett fut egy *virtualizációs szoftver* ami futtatja a virtuális gépeket (a továbbiakban vendég, *guest* gépek) és a virtuális hard-

ver környezetet biztosítja. Ilyenek például a VMware Workstation, Player és Server termékei illetve a VirtualBox és a Microsoft Virtual PC. Ennek előnye, hogy viszonylag könnyen telepíthető meglévő rendszerekre. A hardver eléréséhez a host operációs rendszer meghajtóprogramjait (*driver*) veszi igénybe, mint bármely más alkalmazás. Számos operációs rendszer szintű erőforrás biztosítható a virtuális hardver számára, így egyszerűen megoldott a virtuális gépek kezelése, mert a host operációs rendszeren ablakban megjeleníthető a képernyőkép, hozzáférhet a host gép fájlrendszeréhez stb. Létezik olyan megoldás is (Paralells Workstation, VMware Unity) amely a guest gépen futó alkalmazások ablakait a host gép grafikus felületén natív ablakokba helyezi (seamless windowing), így a felhasználó elől elrejti, hogy valójában az alkalmazás virtualizált környezetben fut. Ezen kényelmi szolgáltatások miatt főleg asztali alkalmazások vagy munkaállomások esetén alkalmazzák az operációs rendszer feletti virtualizációs megoldást. Hátránya viszont, hogy a virtualizált környezetet a host operációs rendszer felett építi fel, így nagyobb az overhead. Nem skálázódik jól, ha sok virtuális gép fut egyazon hoston.

A másik megoldás a ún. *bare-metal* vagy 1. típusú virtualizáció, melyben egy *hypervisor* komponens települ közvetlenül a hardverre. A hypervisor lényegében egy speciális operációs rendszer kernel, ami virtuális hardverkörnyezetet biztosít. Minden virtuális gép efelett fut, még a hypervisor menedzseléséhez használt operációs rendszer is valójában egy virtuális gép. A hypervisor is rendelkezhet saját meghajtóprogramokkal, de létezik olyan megoldás is, amikor a guest operációs rendszer közvetlen hozzáférést kap valamely perifériához. Általában a menedzsmint operációs rendszer speciális abból a szempontból, hogy számos hardverhez nem virtualizált, közvetlen hozzáférése van. Hypervisor architektúrát használ pl. a Xen, a VMware ESX Server és microsoft Hyper-V, valamint Power platformon az LPAR.

3.1. Emuláció, natív- és paravirtualizációs megoldások

Ahhoz, hogy a guest operációs rendszert ne befolyásolja, hogy valódi hardver helyett virtualizált környezetben fut, a futtatórendszernek a speciális (*privilegizált*, csak operációs rendszer kernel által használható) utasításokat el kell fognia (*trap*) és ki kell cserélnie saját rendszerhívásaira. Az x86-os architektúra jónéhány olyan privilegizált utasítással rendelkezik, amire nem lehet hardveres elfogást beállítani. Hasonló a probléma a memóriakezeléssel, hiszen az operációs rendszerek a virtuális memória lapok (*page*) hatékony kezelésekor hardveres megoldásra támaszkodnak (MMU), a virtuális gépek számára azonban egy indirekciós réteget kell beiktatni. Ennek a problémának a megoldására három fő megoldási irány létezik:

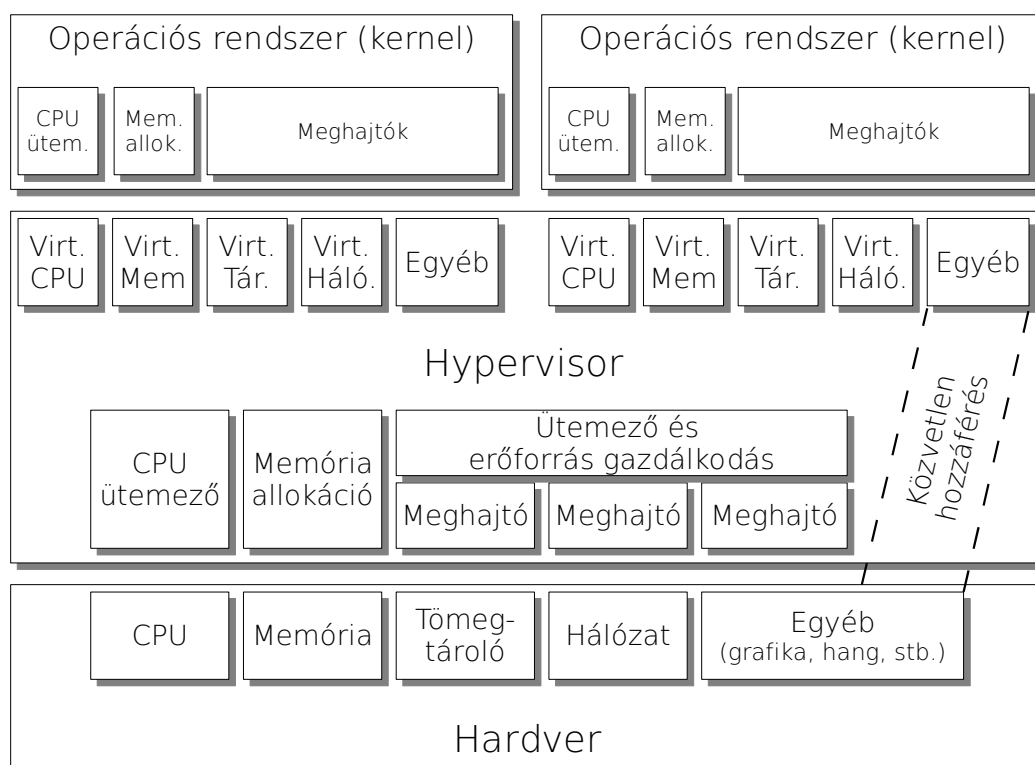
- **Emuláció** - egy futtató szoftver vizsgálja és átalakítja a guest operációs rendszer által végrehajtott utasításokat. Saját maga tart karban egy legfelső szintű page táblát a memóriaelérések kezelésére. Ez a leglassabb, de legflexibilisebb megoldás, mert ez lehetővé teszi, hogy a fizikai processzorétól eltérő utasításkészletet használjon a virtuális gép. Ilyen megoldás a QEMU, és a Java illetve .NET Runtime környezetek. Az emuláció megvalósítható futás közben interpretált módon vagy dinamikus újrafordítással (ún. just-in-time, JIT fordítás). A JIT fordítók általában az interpretációnál lényegesen jobb futási teljesítményt biztosítanak. Platform virtualizációnál a nagy overhead miatt kívánatos elkerülni a tiszta emuláció alkalmazását. *Szoftveres virtualizációnak* nevezzük az emulációnak azon speciális formáját, amikor azonos a fizikai és a virtualizált platform, és csak bizonyos utasításokat kell újrafordítani, más utasítások változtatás nélkül végrehajthatóak. Ez a megoldás a tiszta emulációnál lényegesen jobb teljesítményt nyújthat.
- **Paravirtualizáció** - a guest operációs rendszert módosítjuk, hogy ne hajtson végre olyan utasításokat, amik problémát okozhatnak, helyettük használjon egy-egy speciális rendszerhívást. Hasonlóképpen a memória lap táblák kezelésére is speciális rendszerhívásokat használ, a közvetlen MMU elérés helyett. A guest operációs rendszer CPU ütemezése is kooperálhat a virtualizációs keretrendszer ütemezőjével. Jelenleg ez a leggyorsabb megoldás, ám sok esetben nem használható, mert a guest operációs rendszerben nem lehet módosításokat végezni. A nyílt forrású (mindenekelőtt a Linux alapú) rendszerekben ez

megoldott, a legújabb kernel kiadások forráskódjában már eleve benne vannak a paravirtualizációt támogató részek. Windows esetén általában nem lehet tisztán paravirtualizációt megvalósítani, mert a kernel nem módosítható, ilyenkor az emulációnak és a paravirtualizációnak kombinációját alkalmazzák, ami szintén viszonylag jó teljesítményt biztosíthat.

- **Natív virtualizáció** - a CPU utasításkészletének és üzemmódjainak olyan kiegészítése, ami lehetővé teszi a guest operációs rendszer kódjának módosítás nélküli futtatását. Az Intel és az AMD legújabb processzorai tartalmazznak ilyen kiegészítéseket (Intel VT, AMD-V), ám régebbi processzorokkal szerelt gépek esetén ez a megoldás nem alkalmazható. Teljesítménye jelenleg elmarad a paravirtualizációhoz, illetve a paravirtualizációval kombinált emulációhoz képest, ám az elmaradás az egyes processzorgenerációkkal folyamatosan csökken. Nem minden virtualizációs környezet képes kihasználni ezt a lehetőséget.

A VMware termékei a már említett szoftveres virtualizációra alapulnak, ami kiegészülhet paravirtualizációval, és speciális esetekben (64 bites guestek futtatása) hardveres virtualizációt is igénybe vesz. Más szoftverek (pl. Xen Server) tisztán paravirtualizációt használnak, ha a guest operációs rendszer ezt lehetővé teszi, egyébként hardveres virtualizációra hagyatkoznak. Ha processzorból hiányzik az ehhez szükséges kiegészítés, teljes emulációra lépnek vissza.

3.2. Perifériák virtualizálása



3. ábra.

A guest operációs rendszerek számára biztosított hardver a processzoron és az allokált memórián kívül tartalmaz virtuális merevlemez és hálózati interfészt, grafikus megjelenítőt, opcionálisan egyéb perifériát is. Ez leggyakrabban soros, párhuzamos vagy USB port, néhány rendszer esetén hangkártya vagy akár grafikus gyorsító is lehet. A virtuális hardver többféleképpen lehet megvalósítva. A főbb lehetőségek hasonlóak az alapvető virtualizációs megoldásoknál tárgyaltakhoz:

- **Emulált periféria** - valamilyen létező hardver programozói felületét (regiszterkészletét, interrupt és DMA viselkedését) egy szoftveres implementáció valósítja meg, a valódi hardvert a lehető legpontosabban utánozva. A guest kernel driverai pontosan úgy működnek, mintha valódi hardvert kezelnének, miközben valójában a hardvert emuláló szoftverrel kommunikálnak. Gyakorlatban ez a legkevésbé problémás megoldás, mert ha a virtualizációs keretrendszer egy kellően elterjedt hardvert emulál, akkor a guest operációs rendszer azt kezelni fogja meghajtó telepítése nélkül is. Ez főleg a guest operációs rendszer telepítése során fontos, amikor még nincs lehetőség saját drivereket betölteni. Ennek a megoldásnak meglehetősen nagy az overheadje, viszont az emulált hardver eltérő lehet a fizikaitól, akár operációs rendszer szintű erőforrás is könnyen használható virtuális hardverként.
- **Paravirtualizált perifériák** - leegyszerűsített programozói interfész használata a hardver emulátorban, az egyes I/O műveletek helyettesítése rendszerhívásokkal. Így gyakorlatilag a regiszterek, direkt memória-hozzáférés és megszakítások emulációja elhagyható, az interfészen keresztül a magas szintű műveletek lényegében függvényhívás-szerűen zajlanak le. Ez a megoldás nagyon gyors, hátránya azonban, hogy saját meghajtóprogramok telepítését igényli, hiszen valóságban nem létezik olyan hardver, amelyiknek a programozói felülete hasonlítana a paravirtualizált hardverekéhez. Ennél a megoldásnál van a legtöbbféle lehetőség operációs rendszer szintű erőforrások virtuális hardverként való kiajánlására (pl. fájlrendszer is kiajánlható, nemcsak blokkos eszköz).
- **Közvetlen hardver elérés** - egyes hardver elemek dedikáltan hozzárendelhetőek egy-egy virtuális géphez, aminek az operációs rendszere közvetlenül vezérli a hardvert. Ez egy rendkívül gyors és hatékony módja a virtuális perifériák megvalósításának, ám sok szempontból kompromisszumot jelent. Jelenleg ez leginkább azt jelenti, hogy egy hardvert legfeljebb egy virtuális gép vezérelhet egyidejűleg, a hozzárendelés statikus. Csak a tényleges fizikai hardverfajta virtualizálható, a guestnek ezt kell támogatnia. A szeparációt is aláaknázhatja ez a megoldás, mert sok olyan hardverfajta van, ami felprogramozható, hogy tetszőleges *fizikai* címen végezzen DMA műveleteket. DMA segítségével (akár nem szándékosan, meghibásodásból kifolyólag is) hozzáférhet egy virtuális gép olyan memóriaterülethez, ami más virtuális géphez, a hosthoz vagy esetleg a hypervisorhoz tartozik. Ezt a megoldást hátrányai miatt általában kerülni szokták, csak speciális esetekben alkalmazzák, különösen akkor, ha olyan hardvert kell elérhetővé tenni, amihez a virtualizációs keretrendszernek nincs driverre (pl. 3D grafikus gyorsítók). A közeljövőben várható IOMMU-val szerelt alaplapi chipsetek és többszörös konkurens hozzáférésre felkészített perifériák megjelenése, mely kiküszöböli számos hátrányát ennek a megoldásnak.

4. A VMware ESX Server bemutatása

A VMware cég nagyvállalati szerver virtualizációs megoldását forgalmazza a *Virtual Infrastructure* néven. Ennek alap komponense az **ESX Server**, mely egy **hypervisor** alapú virtuális gép menedzselő és futtatókörnyezet. Ez azt jelenti, hogy közvetlenül a hardverre kell telepíteni, magában tartalmazza a hypervisort és a menedzsment operációs rendszert, ami egy módosított RedHat Enterprise Linux 3. A közelmúltban megjelent egy ehhez hasonló, ám egyszerűsített felépítésű (és ezáltal kisebb erőforrásigényű) változat az **ESXi Server**. Ennek jelenleg egy ingyenes változata is elérhető. Eredetileg beágyazott (tehát a szerver gépbe beépített, flashben tárolt) firmware célokra fejlesztették ki, ám várhatóan idővel teljesen le fogja váltani az eredeti ESX Servert. A hardver támogatottság már most is szélesebb körű az ESXi esetén, mint az eredeti ESX-nél. Az ESXi architektúráját most részletesen nem ismertetjük, a mérés során a régebbi architektúrájú ESX Servert fogjuk használni.

4.1. Az ESX Server szolgáltatásai

Az ESX Server célja, hogy kevés kiépítési és karbantartási ráfordítással biztosítson nagyvállalati igényeket is kielégíteni képes szerver virtualizációs környezetet, az ehhez szükséges összes kiegészítő szolgáltatással (pl. távoli elérés, erőforrások dinamikus kiosztása és átkonfigurálása, hozzáférési jogosultságok kezelése stb.). Fontos megjegyezni, hogy a VMware a teljes termékcsaládján kompatibilis virtuális gép formátumot használ. Ez azt jelenti, hogy (néhány megkötéssel) a Workstationnel készített virtuális gépek futtathatóak pl. Playerrel és ESX Serverrel is, és fordítva.

4.2. Az ESX felépítése

Az ESX Server architektúrája tehát a következőkből áll:

- **VMkernel** - ez a hypervisort és a virtualizált hardverek meghajtóprogramjait tartalmazó alap operációs rendszermag.
- **Service Console** - ez a menedzsment felületet biztosító Linux alapú virtuális gép.

A Service Console alpból biztosít egy szöveges módú felületet a host gép konzolján. Ez a szöveges felület elérhető SSH protokollon keresztül is. Ezen kívül van egy webes kezelőfelülete, amivel meg lehet tekinteni a konfigurációt és néhány alapműveletet el lehet végezni a virtuális gépeken. A Service Console webszolgáltatásokon (Web Services) keresztül biztosít kívülről elérhető programozói felületet, továbbá támogatja a WBEM (Web Based Enterprise Management), ezen belül a CIM-XML (Common Information Model over XML), illetve SNMP (Simple Network Management Protocol) szabványokat is.

A Virtual Infrastructure Client egy Windows-on (tehát külön gépen) futó vastagkliens alkalmazás, mellyel a Service Console-hoz kapcsolódhatunk a webszolgáltatásos felületen. Ennek segítségével konfigurálhatjuk a host rendszert illetve kezelhetjük a virtuális gépeket.

Ez kiegészülhet még a **Virtual Center Serverrel**, ami egy Windows Serveren futó komponens, ami számos ESX Servert futtató host kezelését képes központosítottan megoldani és számos fejlettebb menedzsment képességgel terjeszti ki azokat.

4.3. Főbb alrendszerek

Virtuális tár - a virtuális gépeket leíró konfigurációs fájlok (.vmx, .vmxf) a virtuális gép állapotát tároló fájlok (.nvram, .vswp) és a virtuális gép merevlemezeinek tartalmát tároló fájlok (.vmdk) tárolására szolgál. Az ESX Server egy speciális fájlrendszert (*VMFS3*) használ erre a célra, melyet kifejezetten nagy fájlok tárolására optimalizáltak. Valójában alacsony szinten a logikai kötetkezelőkhöz hasonló elven működik, ám egy szabályos fájlrendszer felületet biztosít, ami a kezelését egyszerűsíti. Fontos tulajdonsága, hogy fürtözött (clustered) elérést biztosít, tehát egy hálózati tárrendszeren (SAN) elhelyezve egyszerre több különálló ESX Server is használhat egy közös tárat.

Hálózati tárolást a VMkernel biztosít, ami egy teljesen független hálózati stack és hálózati tárolási protokoll implementációval rendelkezik.

Virtuális hálózat - a virtuális gépek hálózati kapcsolatát virtuális switch-ek (gyakorlatilag szoftveres Ethernet hidak) biztosítják. A virtuális switch-ek egy közbelső réteget jelentenek a fizikai interfészek és virtuális gépek hálózati interfészei között, használatukkal flexibilis hozzárendelést lehet kialakítani. A Service Console és a VMkernel hálózati hozzáférését is virtuális switch-eken konfigurálhatjuk, ezeknek is külön MAC címük és IP címük van, pontosan úgy, mintha fizikai hálózati interfésszel rendelkeznének. Ezáltal megoldható a fizikai hálózati interfészen pl a hálózat tárrendszer forgalmának multiplexelése a virtuális gépek forgalmával. Lehetséges zárt (hoston belüli, fizikai interfész nélküli) virtuális hálózatokat is definiálni, ami csak egy host gépen futó guest gépek között működik.

CPU ütemező - a host CPU idejét osztja szét a guest gépek, a Service Console és a VMkernel belső szolgáltatásai között. Lehetséges prioritásokat és minimálisan garantált illetve maximális kiosztható CPU időt definiálni. Az ütemező felismeri a processzorok topológiáját, a többmagos processzorokat, a Hyperthreading technológiát. Figyelembe veszi az ütemezéskor, hogy mely logikai processzorok függetlenek egymástól, mennyi a költséggel jár egy virtuális gépet ami idáig egy processzoron futott átütemezni egy másik processzorra, ami különösen nagyméretű, nem egységes memóriaelérést használó rendszereknél (NUMA, Non Uniform Memory Access) számít teljesítményben sokat. A CPU virtualizáló rendszer működhet emulációs és paravirtualizált üzemmódban. Tisztán emulációs mód többnyire csak a guest operációs rendszer telepítésének idején aktív, ha lehetséges paravirtualizációs vagy szoftveres emulációval kombinált virtualizációs üzemmódba lép át.

Memóriakezelő - a CPU ütemezőhöz hasonlóan ez is kezel minimális és maximális foglalási beállításokat, valamint prioritásokat. Dinamikus memóriakiosztást használ, ami virtualizáció esetén különösen nehéz feladat, mert a virtuális gépeknek adott memóriaméretük van, amiről feltételezik, hogy számunkra rendelkezésre áll. A VMkernel memóriakezelője figyeli az egyes virtuális gépek aktivitását és memórialap-használatát, ennek megfelelően képes késleltetni a memória tényleges allokálását egy virtuális gépnek. Ilyen módon lehetséges akár túlfoglalni (overbooking) is a host gépet, bár ez erősen ellenjavallt. Ha a host memóriája nem elég, akkor képes lapozófájlba (swap) írni a virtuális gépek memóriatartalmát. Kifejezetten káros lehet, ha a virtuális gép operációs rendszere lapoz (belső swap), miközben a virtualizációs keretrendszer is lapozza memóriáját (külső swap), mert a guest operációs rendszer nem tudja, hogy az ő általa gyors memóriának gondolt címtartomány egy része is diszken van, így előfordulhat, hogy a két swap terület között ide-oda másolgatják a memórialapokat ami vergődést (trashing) azaz a diszka alrendszer nagy terhelése mellett rendkívül alacsony teljesítményt okoz. Ennek elkerülésére létezik paravirtualizációra alapuló kooperációs technika a vmkernel és a guest operációs rendszer között (*memory ballooning*).

Ezen túl még van számos, nem kevésbé jelentős alrendszer is, pl a guest gépek grafikus felületéhez való távoli hozzáférést biztosító virtuális framebuffer, ami itt nem kerül részletes tárgyalásra.

4.4. A virtuális hardver felépítése

A virtuális gépek számára biztosított hardverkörnyezet tartalmaz:

- **CPU** - 1, 2 vagy 4 db lehet (de nem több, mint a host gépben lévő processzorok száma), típusa azonos a host CPU-éval, ám opcionálisan letilthatóak speciális kiegészítések (SSE, 64bit, Nx bit, stb). Ez utóbbira akkor van szükség, ha működő (vagy suspendelt) virtuális gépet mozgatni akarunk eltérő CPU-val szerelt hostok között. Az ilyen letiltás teljesítményvesztéssel járhat és még a virtuális gép elindítás előtt kell beállítani.
- **Memória** - maximum 64 GB memória osztható ki, akár 32 bites guesteknek is, emulált PAE (Paging Address Extension) kiterjesztéssel. A memóriaméret is csak leállított virtuális gépen módosítható.

- **Chipset** - Intel 440BX chipsetet emulál. Csak interrupt és periféria erőforrás konfigurációs szerepe van.
- **BIOS, NVRAM** - Egy saját BIOS-t biztosít. A BIOS beállításokat tároló nem felejtő memória tartalmát külön fájlban (.nvram) tárolja.
- **Grafikus vezérlő** - Alap VGA, VESA üzemmódokat emulál, nagyon lassú. Létezik paravirtualizált üzemmódja is, mely jól használható sebességet biztosít.
- **IDE vezérlő** - ESX Server csak CD/DVD meghajtót emulál IDE felett, a Workstation merevlemez is támogat.
- **SCSI vezérlő** - LSI Logic vagy Buslogic típusú emulált hardver, külön meghajtóprogram van hozzá, amit telepítésnél pl. Windows XP-nek meg kell adni.
- **Floppy vezérlő** - floppy lemez image-et tartalmazó fájlból szolgáltat tartalmat vagy a host fizikai meghajtóját kapcsolja hozzá a virtuális géphez. Lehetőség van a Virtual Infrastructure klienst futtató gép meghajtóját is csatlakoztatni hálózaton keresztül.
- **CD/DVD** - CD vagy DVD image-et (.iso) tartalmazó fájlból szolgáltat tartalmat vagy a host fizikai meghajtóját kapcsolja hozzá a virtuális géphez. Lehetőség van a Virtual Infrastructure klienst futtató gép meghajtóját is csatlakoztatni hálózaton keresztül.
- **Merevlemez** - ESX alatt csak a SCSI vezérlőre csatlakoztatható. Tetszőleges méretű fájl (.vmdk) hozható létre az adatok tárolására. Fontos, hogy az ESX mindig a teljes diszket allokalja (*flat* formátum), míg a Workstation támogat olyan formátumot is, ahol dinamikus a helyfoglalás (*sparse* formátum).
- **Hálózati interfész** - Lehetőség van AMD PCNet32 vagy Intel E1000 (Intel PRO/1000 termékcsalád) kártyák emulálására vagy VMXNet meghajtóval paravirtualizált Ethernet interfész kialakítására. Az éppen használt típust többnyire automatikusan választja ki („flexible”), azonban ez manuálisan felülbíráható.
- **Soros port** - a host gép soros portjára, egy fájl tartalma vagy hálózaton keresztül a klienst futtató gép portjára kapcsolható.
- **Párhuzamos port** - a host gép párhuzamos portjára, egy fájl tartalma vagy hálózaton keresztül a klienst futtató gép portjára kapcsolható
- **Egyebek** - Csak Workstation alatt támogatott: **Audio vezérlő, USB port, Megosztott könyvtár** (fájlrendszer megosztás kiajánlása)

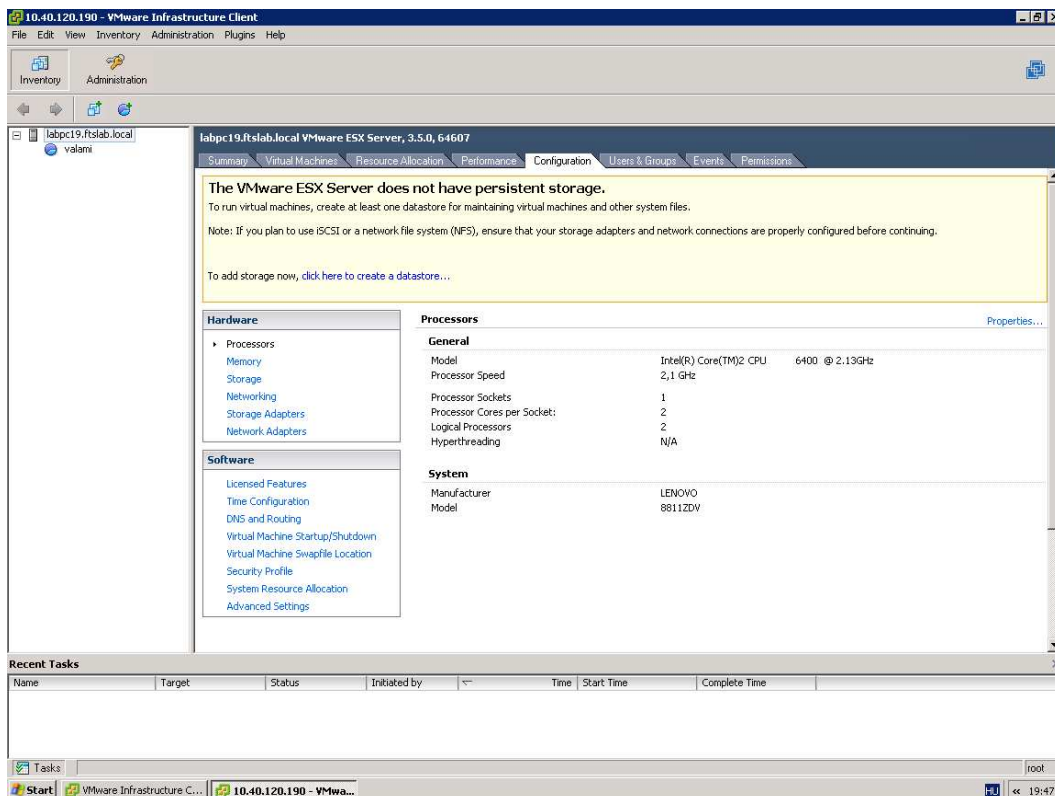
Mint már említettük, a jó teljesítmény elérése érdekében a paravirtualizáció használata a kívánatos. Erre van a **VMware Tools** csomag, ami paravirtualizált meghajtóprogramokat tartalmaz a guest operációs rendszer részére. Leginkább a grafikus vezérlő sebességén és az egér transzparens kezelésén (simább mozgás, ablak szélén be- és kilépés) vehető észre a Tools hatása, de általában minden virtuális hardverkomponens teljesítményén jelentősen javít.

5. ESX Server kezelése a Virtual Infrastructure Clienttel

Az ESX Server 3.5-ös verziójához a Virtual Infrastructure Client 2.5-ös verziójával lehet kapcsolódni. A VI Client telepítőkészlete letölthető a ESX Server webes felületéről.

Egy frissen telepített ESX Serverre csak a **root** felhasználóval lehet belépni, a telepítés során megadott jelszóval. Belépés után a 10. ábrán látható kezelőfelület fogad. Jobb oldalt egy erőforrás fa található, legfelül a host géppel. Bal oldalon az aktuális kiválasztott objektum (Host, erőforrás csoport, virtuális gép) tulajdonságait tekinthetjük meg, illetve állíthatunk be különböző füleken.

5.1. Alap konfiguráció beállítása



4. ábra.

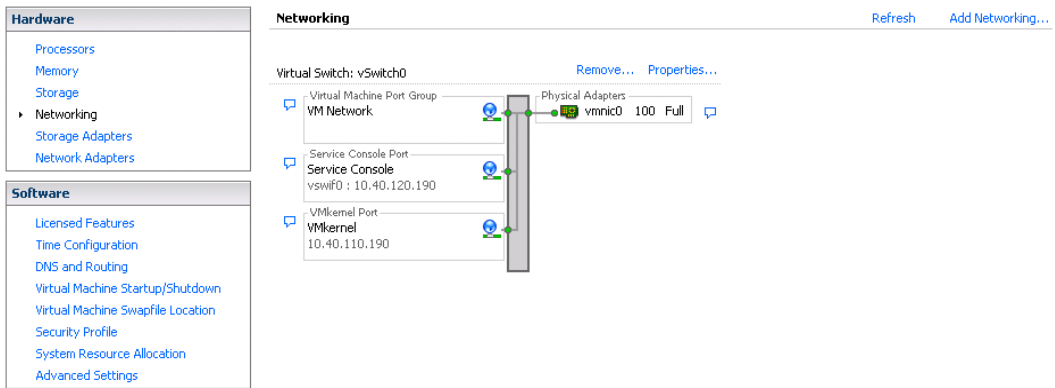
A host gépet kiválasztva a *configuration* fülön végezhetjük el a beállításokat (4. ábra).

Fontos megjegyezni, hogy az ESX Server virtuális gép adattárnak csak SCSI merevlemezeket, iSCSI célpontokat vagy NFS megosztásokat tud kezelni. Mivel a laborgépekben nincs SCSI merevlemez, ezért a mérés során iSCSI célpontot kell konfigurálni. Első lépésként meg kell győződni róla, hogy a *VMkernel* és *Service Console* hálózati interfészei fel vannak véve a *Networking* alatt (5. ábra).

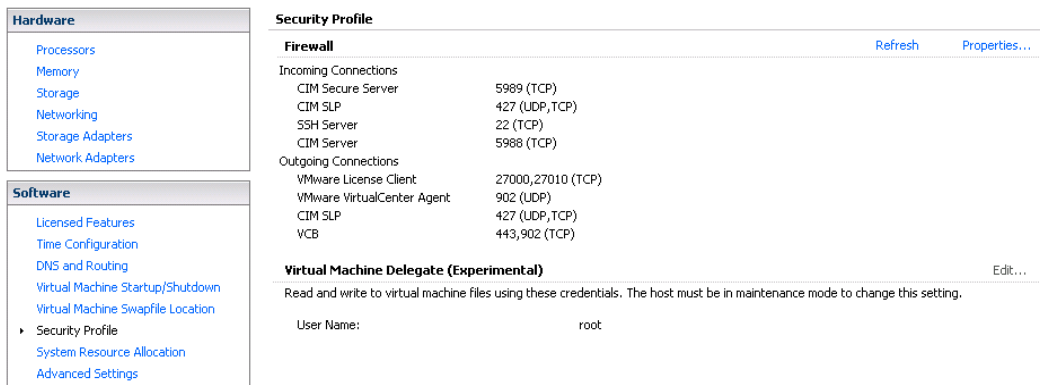
Ezután a *Security Profile* alatt ellenőrizni kell, hogy a Service Console tűzfala kiengedi az iSCSI kliens forgalmát (6. ábra), ha nem, akkor a tulajdonságok alatt engedélyezni kell. A *storage adapters* alatt az *iSCSI Software Adapter* tulajdonságainál állíthatjuk be a célpontot. Az iSCSI-ről részletesebb ismertetés a „Háttértár rendszerek” mérés segédletében olvasható. *General* fülön a *Configure...* alatt engedélyezni kell az iSCSI kezdeményezőt (7. ábra).

Az engedélyezés után visszatérve a *Configure...*-ra adhatunk iSCSI állomásnevet a gépnek, ám az alapértelmezett állomásnév megváltoztatása újraindítást igényel.

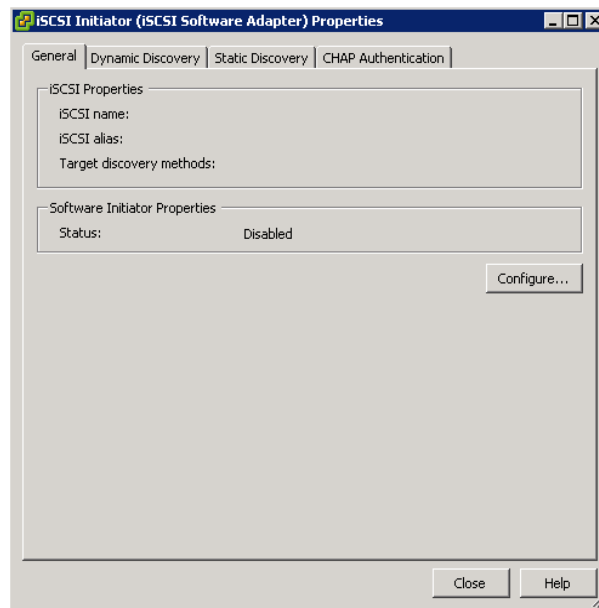
Az iSCSI célpont megadása a *Dynamic Discovery* fülön lehetséges. Új célpont hozzáadása után



5. ábra.



6. ábra.



7. ábra.

Storage Adapters Rescan...

Device	Type	SAN Identifier
iSCSI Software Adapter		
vmhba32	iSCSI	iqn.2008-03.local.ftslab:la...

Details

vmhba32 Properties...

Model: iSCSI Software Adapter
 iSCSI Name: iqn.2008-03.local.ftslab:labpc19
 iSCSI Alias: labpc19.ftslab.local

IP Address:
 Discovery Methods: Send Targets
 Targets: 2

SCSI Target 0

iSCSI Name: iqn.2008-02.local.ftslab.tuor:testvol
 iSCSI Alias:
 Target LUNs: 1 Hide LUNs

Path	Canonical Path	Type	Capacity	LUN ID
vmhba32:0:0	vmhba32:0:0	disk	1,00 GB	0

SCSI Target 1

iSCSI Name: iqn.2008-02.local.ftslab.tuor:meresvol
 iSCSI Alias:
 Target LUNs: 1 Hide LUNs

Path	Canonical Path	Type	Capacity	LUN ID
vmhba32:1:0	vmhba32:1:0	disk	31,25 GB	0

8. ábra.

egy hosszabb várakozás következik. Lépünk ki a dialog ablakokból és a *iSCSI Software Adapter*en indítunk egy *Rescan...* műveletet, hagyjuk, hogy mindkétféle felderítést elvégezze. A felderítés folyamatát az ablak alján lévő eseménynaplóban figyelhetjük. A felderítés végeztével látnunk kell a közös iSCSI tárat (8. ábra).

Hardware

- Processors
- Memory
- Storage
- Networking
- Storage Adapters
- Network Adapters

Software

- Licensed Features
- Time Configuration
- DNS and Routing
- Virtual Machine Startup/Shutdown
- Virtual Machine Swapfile Location
- Security Profile
- System Resource Allocation
- Advanced Settings


Storage Refresh Remove Add Storage...

Identification	Device	Capacity	Free	Type
testdatatore4	vmhba32:0:0:1	768,00 MB	710,00 MB	vmfs3
Tuor-MeresVol	vmhba32:1:0:1	31,00 GB	19,93 GB	vmfs3

Details Properties...

testdatatore4

Location: /vmfs/volumes/47d96506-3... 768,00 MB Capacity

58,00 MB Used 
 710,00 MB Free

Path Selection

Fixed

Properties	Extents
Volume Label: testdatatore4	vmhba32:0:0:1 1022,00..
Datatore Name: testdatatore4	Total Formatted Capacity 768,00...

Paths

Total:	1
Broken:	0
Disabled:	0

Formatting

File System:	VMFS 3.31
Block Size:	1 MB

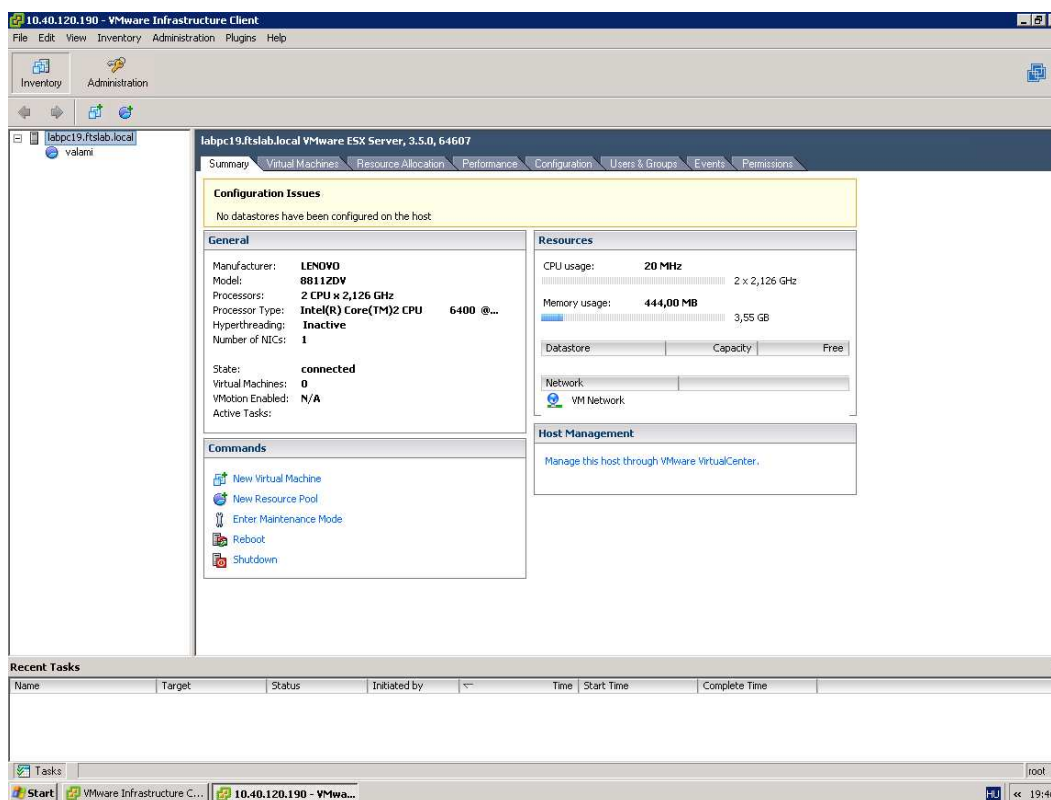
9. ábra.

Ha célponton már létre lett hozva VMFS fájlrendszer, akkor az meg fog jelenni az adattárak listájában a *Storage* fül alatt (9. ábra).

Innentől kezdve az ESX Server rendelkezik adattárral, lehet virtuális gépeket felvenni.

5.2. Virtuális gépek létrehozása

Az ESX Server mindennapi használata során gyakori művelet, hogy új virtuális gépet hozunk létre. Ennek az ESX Server alatt kétféle alapesetével találkozhatunk. Lehetséges előre elkészített feladatspecifikus virtuális gépeket (ún. virtual appliance) letölteni és telepíteni (*File/Virtual Appliance/Import...*). Továbbá lehetőségünk van természetesen saját virtuális gépet létrehozni, melynek lépéssorozata sokban hasonlít a Workstation alatti lépésekhez. Lényeges eltérés, hogy a virtuális gépeket az erőforrás fába (*inventory*) helyezjük el.

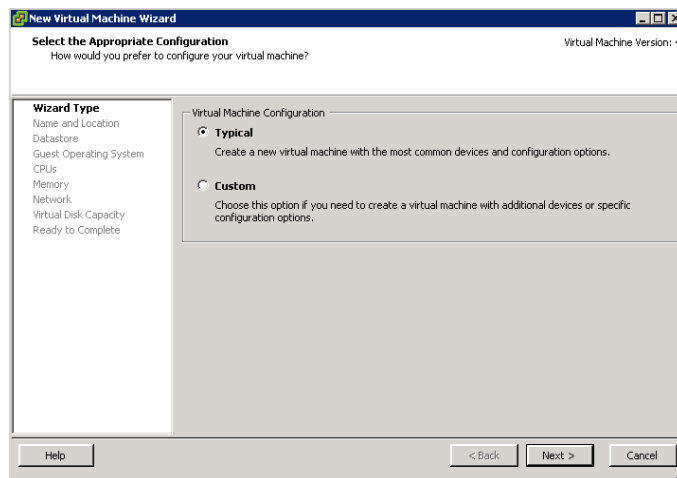


10. ábra.

A virtuális gép létrehozását segítő varázslót (*New Virtual Machine Wizard*, 11. ábra.) a host gép összefoglaló (*Summary*) fülén (10. ábra) található ikonnal indíthatjuk.

A varázsló a következő beállításokon halad végig:

1. Virtuális gép neve
2. Adattár helye
3. Operációs rendszer típusa. Ez a támogatott virtuális hardverelemek és teljesítmény optimalizálása miatt fontos.
4. Processzorok száma
5. Memóriaméret
6. Hálózati interfészek és kapcsolódásaik a virtuális switch-ekre
7. Virtuális diszk tárhely allokáció



11. ábra.

5.3. Virtuális gépek használata

Egy virtualizációs rendszerben a leggyakoribbak a virtuális gépek általános használatával kapcsolatos műveletek:

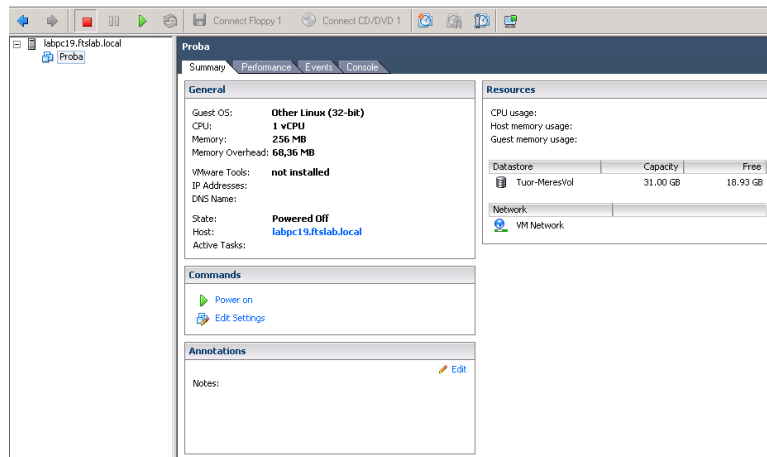
- indítás
- leállítás
- újraindítás
- felfüggesztés (hibernálás, *suspend*)
- konzolhoz kapcsolódás
- konfiguráció módosítás
- állapotmentés (*snapshot*) készítése

Virtuális gépek üzemállapotát a virtuális gép kiválasztásakor a felső sávban megjelenő ikonokkal (play - elindítás, pause - felfüggesztés, stop - leállítás, körbe nyilak - restart) lehet változtatni. (12. ábra.)

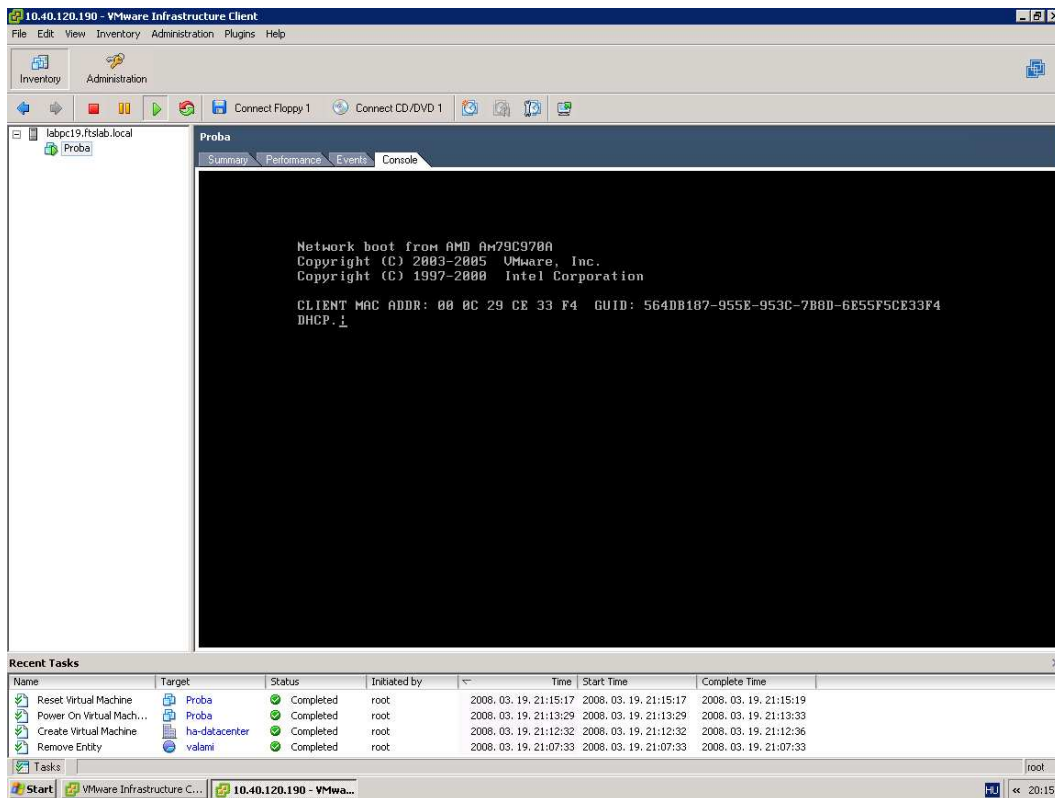
A konzolt a *Console* fülön érhetjük el (13. ábra.), illetve külön ablakba is kirakhatjuk a felső sáv utolsó ikonjával. Az ESX Server egyazon virtuális géphez tetszőlegesen sok egyidejű kapcsolatot kezel. Több egyidejű kapcsolatnál figyelmeztetést kapunk, hogy rajtunk kívül más is kapcsolódott a virtuális gép konzoljához. A VMware termékeknél megszokott módon lehet a billentyűzettel és egérrel irányítani a virtuális gép konzolját. A képernyőképbe kattintással fókuszot kap a virtuális gép, innentől kezdve a billentyűzet és az egér a virtuális gépet irányítja. **Ctrl-Alt** lenyomásával lehet kilépni virtuális gépből. A VMware Tools lehetővé teszi, hogy az egér a virtuális képernyő szélén ki illetve be tudjon lépni virtuális gépbe.

A virtuális gépek *Summary* fülén van lehetőségünk a virtuális hardver beállításait módosítani. (14. ábra) A beállítások közül az eltávolítható eszközök (Floppy, CD/DVD, Hálózat) a gép futása közben átállíthatók, a többi beállítás csak leállított guest mellett módosítható. Ebből a szempontból a felfüggesztett állapot is a bekapcsolt állapothoz hasonlít.

Az eltávolítható eszközök azért is érdemelnek külön említést, mert ezeket módosítjuk a leggyakrabban, például, ha CD-akarunk behelyezni guest virtuális meghajtójába. Az eszközök csatlakoztatott állapotát a *Connected*, illetve *Connect at power on* jelölőnégyzeteivel lehet állítani. Ha bootolni akarunk CD-ről, akkor feltétlenül be kell jelölni a *Connect at power on-t*, mert a BIOS inicializálás gyorsan lefut a virtuális gépben, ennyi idő alatt kézzel nem lehet csatlakoztatni az



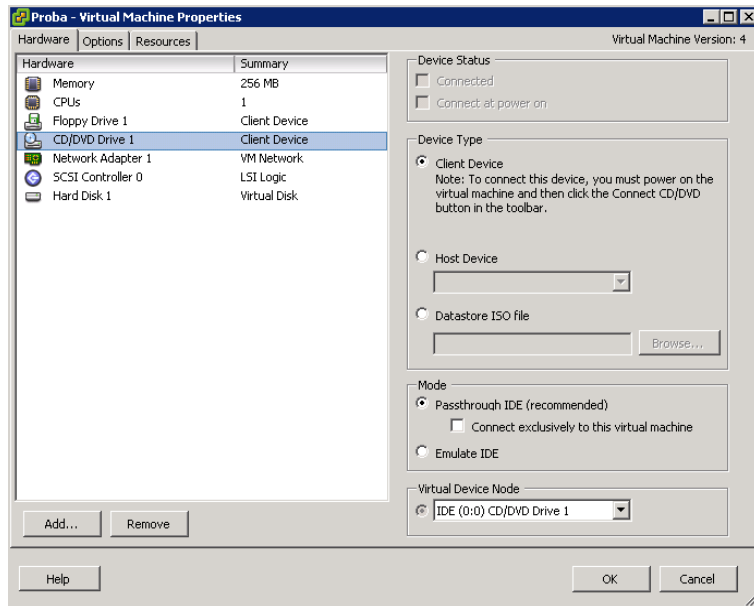
12. ábra.



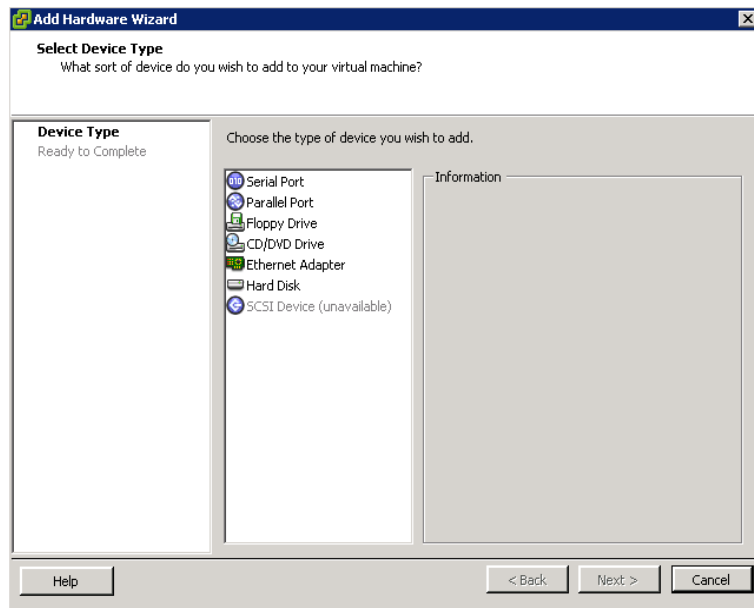
13. ábra.

eszközt. A virtuális CD lemez lehet valódi CD a klienst futtató gép CD meghajtója (*Client Device*, az ESX szerver meghajtója *Host Device* vagy ISO image fájl (*Datastore ISO*). A leggyakrabban az utóbbit használjuk, a telepítőkészleteket valamilyen adattárra összegyűjtjük így telepítéskor csak ki kell választani a megfelelőt. A VMware Tools telepítő CD image-ei is így érhetőek el, a *vmimages* könyvtár alatt.

Kikapcsolt állapotban lehetőségünk van minden hardverelem módosítására, elvételére és hozzáadására (15. ábra.). Vannak olyan (ritkábban használt) virtuális hardvereszközök, amiket az új gépet létrehozó varázsló nem kínál fel, pl.: soros illetve párhuzamos portok.



14. ábra.

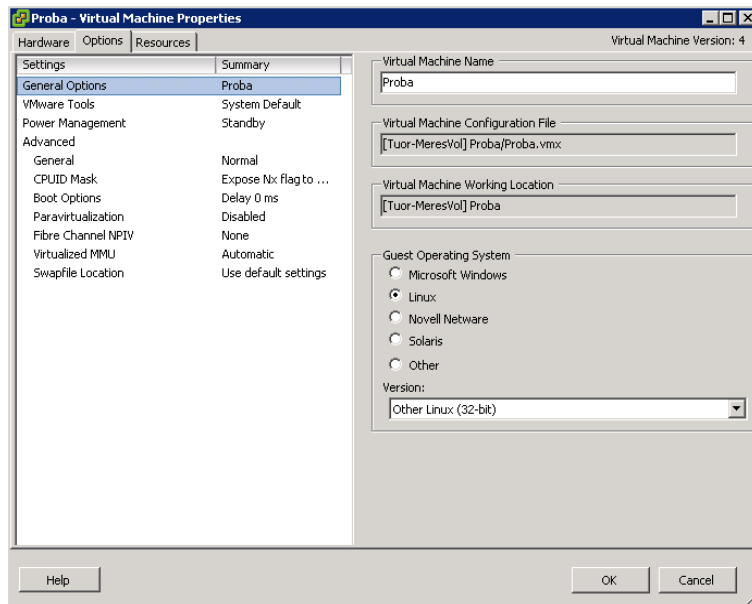


15. ábra.

A virtuális gép tulajdonságait szerkesztő ablakban az *Options* fülben lehet speciális beállításokat módosítani, ezek teljesítmény optimalizálási lehetőségekre és kompatibilitásra vannak hatással (16. ábra.)

A *Resources* fülön találhatóak az erőforrás-kiosztási beállítások, amikkel a későbbiekben részletesen is foglalkozunk.

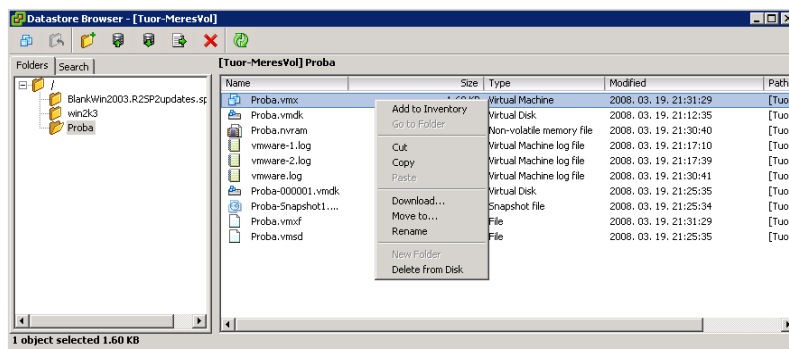
Külön figyelmet érdemel a teljes gépre kiterjedő állapotmentési lehetőség, ami fizikai gépen általában nem megoldható, vagy csak kézi konfigurálással valósítható meg az operációs rendszer szintjén. Fájlrendszerekről snapshot készíthető logikai kötetkezelő (LVM) rendszerekkel, ám memóriatartalomra ez nem terjed ki. Az VMware snapshot rendszere futó gépről is képes pillá-



16. ábra.

natképet készíteni, így nemcsak a merevlemez tartalma, hanem a memória pillanatnyi állapota is mentésre kerül. További fontos tulajdonság, hogy egy gépről több, akár egymásból leszármazó, vagy közös forrásból kiinduló párhuzamos snapshot is kezelhető, és igény szerint bármelyik mentett állapot gyorsan visszatölthető. A *snapshot manager* ikonjai a felső sávon a jobb szélhez közel helyezkednek el.

A virtuális gépek fájljait az adattár böngészővel (*Datastore Browser*) kezelhetjük. Az összefoglaló fülön egy adattárat kijelölve, majd jobb kattintással a kontextus menüjéből indítható az adattár böngésző. (17. ábra) A felső sorban található ikonok lehetővé teszik, hogy könyvtárakat hozzunk létre, töröljünk fájlokat vagy könyvtárakat, áthelyezzünk, illetve fel- és letöltsünk fájlokat a kliens gépről az adattárba. A virtuális gépeket leíró *.vmx* fájlokra jobb gombbal kattintva, majd a *Add to inventory...* menüpontot választva lehet az erőforrásába beregisztrálni olyan virtuális gépet ami nincs oda felvéve (pl. mert nem az adott szerveren hoztuk létre). Az erőforrásából természetesen el is távolíthatóak a virtuális gépek, akár a fájlok törlésével, akár meghagyásával (ez esetben később újra hozzáadható lesz az erőforrásához). A Workstationnel létrehozott diszkeknél figyelni kell arra, hogy a virtuális diszkek egy fájlban, fixen lefoglalt formátumban (*-flat*) legyenek, mert az ESX szerver csak ezt támogatja, ne pedig 2GB-os, dinamikusan lefoglalt egységekben (*2GB sparse*).

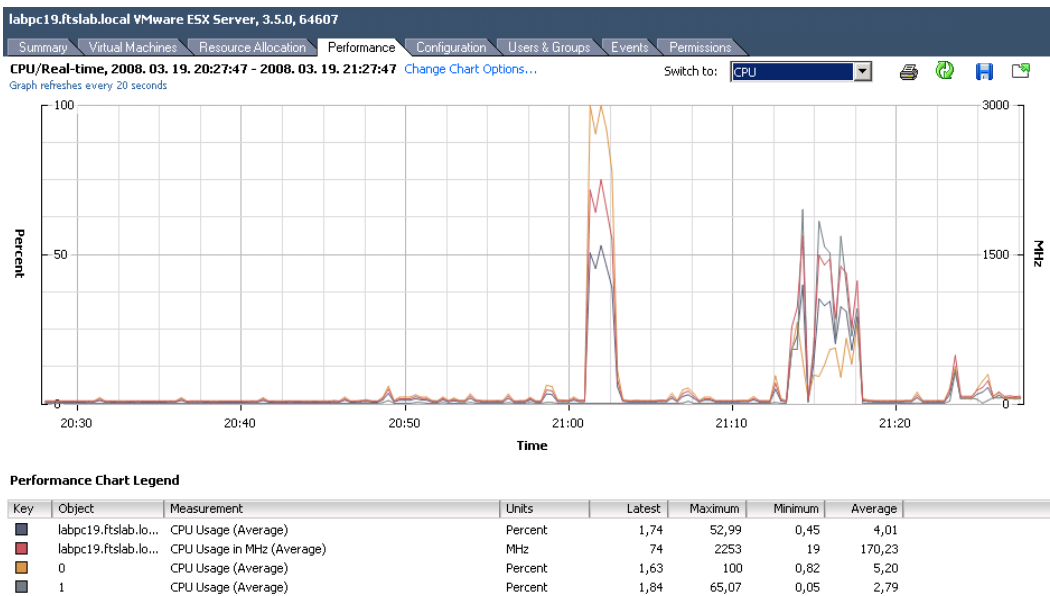


17. ábra.

5.4. Teljesítménymérés és erőforrás-gazdálkodás

A virtualizált rendszerek alaptulajdonsága, hogy közös erőforrásokon osztozó virtualizált környezeteket kezelnek. A virtuális gépek teljesítményét alapvetően meghatározza, hogy a virtualizációs futtatókörnyezet hogyan gazdálkodik a közös erőforrásokkal. Az ESX Server számos megfigyelési és beállítási lehetőséget kínál az erőforrások kiosztásának befolyásolására.

Lehetőség van létrehozni csoportosító elemeket (*Resource Group*), melyekbe virtuális gépeket helyezhetünk. A csoportoknak lehet meghatározott erőforrás korlátja (*Limit*) vagy garantált minimális részesedése (*Reservation*). Korlátot vagy fenntartott minimumot az egyes virtuális gépek szintjén is megadhatunk az előzőekben említett *Virtual Machine Properties* ablak *Resources* füle alatt. Így komplex hierarchikus erőforrás kiosztási fát definiálhatunk, bár leggyakrabban ezt a lehetőséget arra használjuk, hogy egyes virtuális gépek túlzott erőforrás-foglalást korlátozzuk (pl. egy virtuális gép folyamatosan 100%-ra terhelt CPU-ja ne lassítsa le a többi virtuális gépet). Erőforrások szerinti (CPU, memória, I/O terhelés) prioritásokat is adhatunk a virtuális gépeknek. Ez akkor kap szerepet, ha valamely erőforrásból többre van igény, mint amennyi rendelkezésre áll, ilyenkor a prioritások függvényében von el erőforrásokat a guest gépektől.



18. ábra.

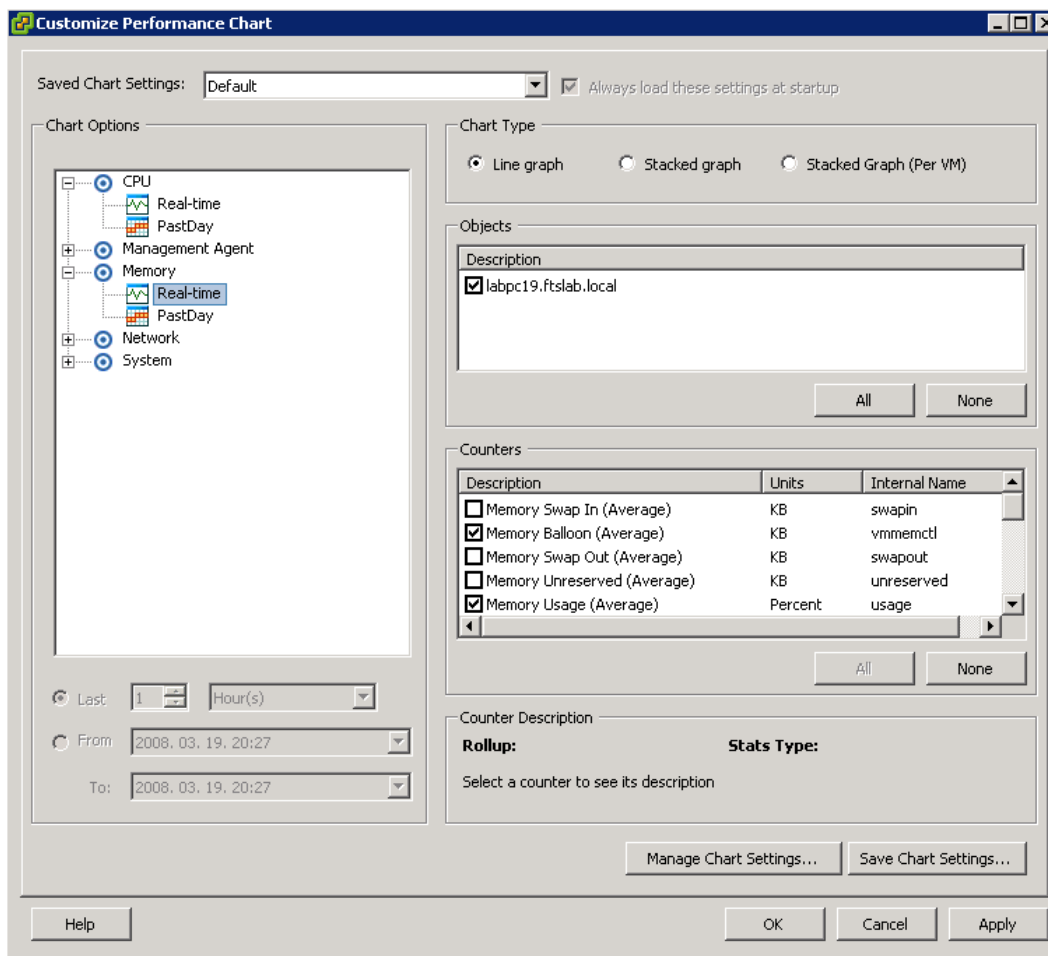
Az erőforrás-foglalást illetve terhelést az ESX Server kicsivel több, mint egy napra visszamenőleg rögzíti minden virtuális gépre. A Virtual Center ezt kiterjeszti saját adatbázissal, ami akár 1 évre visszamenő historikus adatokat is tárolhat. A visszamenőleges adatok grafikonok formájában megtekinthetők a *Performance* fül alatt (18. ábra).

A grafikonokon megjeleníthető értékek és az időbeli felbontás a *Change Chart Options...* alatt választható ki (19. ábra).

A virtuális gépen belüli óra és a host gép valós idejű órája sok esetben nem szinkronizált (óraszinkron szolgáltatást a VMware Tools biztosít), jelentős elcsúszás lehet közöttük, ami ráadásul nem konstans, függ a terheléstől. Ezáltal a virtuális gépen belüli teljesítményméréseknél nem szabad a belső órára hagyatkozni.

6. Központi menedzsment a VirtualCenter segítségével

A VirtualCenter egy Windows2003-ra telepíthető programcsomag, mely több ESX Server kezelését képes összevonni egy felületre. Ezenkívül számos funkcióval ki is terjeszti őket:



19. ábra.

- Virtuális gépekhez összetett hozzáférési jogosultságkezelés
- Virtuális gépek automatikus klónozása és üzembeállítása (deploy)
- Virtuális gépek futás közbeni mozgatása hostok között
- Hibatűrő fürtök építése
- Terheléelosztó fürtök építése
- Hostok és virtuális gépek monitorozása, riasztás
- Hostok és virtuális gépek központositott szoftverfrissítése
- Biztonsági mentés (backup) kezelése

Amint látható a VirtualCenter célja, hogy egy nagyvállalati infrastruktúramenedzsmment legtöbb részfeladatát önmaga képes legyen ellátni, így nincs olyan nagyobb terület, ami feltétlenül külső szállító termékét igényelné. Azonban hatóköre célzottan csak az ESX Servereken futó virtuális gépekre terjed ki. Továbbá plugin architektúrájú, így bővíthető új feladatok ellátására (a szoftverfrissítést is külön plugin valósítja meg), nyílt szoftverfejlesztői felületet biztosít⁴, ezáltal

⁴A VMware Virtual Infrastructure SDK ezen címen érhető el: <http://www.vmware.com/support/developer/vc-sdk/>

lehetőség nyílik más menedzsment alkalmazásokkal történő problémamentes integrációjára, illetve menedzsment feladatok automatizálására számos programnyelv felett.

6.1. A VirtualCenter erőforrásfája

A VirtualCenter is az ESX Servernél megismert *inventory*-t használja, ám itt ez kiegészül a következő elemfajtákkal:

- Datacenter - nagy szervezőegység, amin belül kövösek a virtuális hálózatok nevei és az datastore-ok
- Cluster - Datacenteren belül ESX hostok és virtuális gépek egy szűkebb csoportja, mely hibatűrő és/vagy terheléelosztó fürtöt valósít meg

Az ESX Servereket vagy Datacenterben vagy Clusterben helyezhetjük el. A Datacenterben elhelyezett hostok a korábban már megismert módon viselkednek, alattuk jelennek meg az erőforrás csoportok és virtuális gépek. A Clusterben azonban nincsenek a virtuális gépek hostok alá rendelve, dinamikusan üzem közben változtathatják a helyüket a hostok között. A Cluster tulajdonságait a létrehozásakor (Az Inventory-ban a Datacenteren jobb gomb *Add Cluster...*) állíthatjuk be:

- VMware HA - hibatűrő fürtözési funkció
- VMware DRS - terheléelosztó funkció

Mikor egy hostot be akarunk vonni a VirtualCenter menedzselése alá (Datacenteren vagy Clusteren jobb gomb *Add Host...*) meg kell adnunk a gép adminisztrátori (root) belépési jelszavát.

6.2. Virtuális gép sablonok kezelése

A virtualizáció egy nagy előnye, hogy dinamikusan, az igények szerint, hardver megmozgatása nélkül lehet egy-egy feladatra új gépet létrehozni. A virtuális gépek életciklusának (létrehozása, használata, megszüntetése vagy „nyugalmazása”) támogatásában egy fontos szempont, hogy új gépek létrehozásakor ne kelljen minden alkalommal kézzel operációs rendszert telepíteni és konfigurálni, hanem egy már félkész környezet testreszabásával gyorsan lehessen új gépet üzembe állítani. Nagyvállalati környezetben gyakran vannak előírások az operációs rendszerek beállításaira, így ezek betartása is nagyban egyszerűsíthető, ha kész előretelepített *sablonokból* (template) hozzuk létre a virtuális gépeket.

Ennek egyszerű megoldása a virtuális gépek klónozása, ám ez nem képes kezelni azt a problémát, hogy részben a virtuális hardver, részben az operációs rendszer telepítés közben egyéni azonosítókat kap, aminek egy hálózaton belüli egyedisége fontos a helyes működés szempontjából. Ilyenek például a hálózati vezérlő MAC címe, az OS IP címe és hosztneve, esetleg valamilyen UUID-je, Windows-ok alatt biztonsági egyedi azonosító, SID.

A VirtualCenter képes sablonból automatikusan létrehozni virtuális gép példányokat úgy, hogy közben tesztre is szabja azt. Ennek előfeltétele, hogy a sablonnak szánt virtuális gépbe feltelepítsük a VMware Tools-t, Windows-ok testreszabásához a VirtualCenter szerverre telepítsük a Microsoft Sysprep Tools-t, valamint, hogy az elkészült virtuális gépet template-nek jelöljük a *Summary* fülön a *Convert to template* opcióval. Innentől kezdve ez a gép nem lesz elindítható, az inventory-ban csak a *Virtual Machines and Templates* nézetben lesz látható.

Viszont lehetőségünk nyílik ebből a sablonból új virtuális gép példányosítására a *Deploy Virtual Machine from This Template* opcióval.

A példányosítás során az alap kérdéseken (az új virtuális gép neve az inventory-ban, melyik Datacenterbe, Clusterre vagy Hostra kerüljön) túl a *Customize using customization wizard* opció megnyitja a lehetőséget a példány testreszabása előtt. A testreszabásnál lehetőség van a hosztnev illetve a hálózati interfészek beállításait (IP cím, stb.) módosítani. Az első indításkor a VMware Tools automatikusan újrageneráltatja a gép egyedi azonosítóit (Linux alatt sajnos ez a funkció nem

teljes, pl. nem készül új SSH privát kulcs vagy SSL tanúsítvány a gép számára) és módosítja a hálózati beállításokat, így a gép a hálózaton már a példányosításnál megadott címen lesz elérhető.

Természetesen a sablon gépen további módosítások elvégzéséhez vissza is alakíthatjuk a sablont virtuális géppé, így az újra elindítható lesz.

6.3. Működés közbeni áthelyezés

Virtuális gépek leállítás nélkül is áthelyezhetők a különböző hostok között, ezt a lehetőséget nevezzük *live migrációnak*. A VMware termékeiben ezt VMotion néven találhatjuk meg. A működés közbeni áthelyezés legfontosabb és egyben technikailag legnehezebben megoldható részfeladata memóriatartalom átmozgatása. Ennek leállítás nélküli vagy csak minimális kimaradással járó megvalósíthatóságát az teszi lehetővé, hogy a virtuális gépek nem használják folyamatosan az összes memórialapjukat. Az áthelyezés több fázisra bomlik:

1. először az aktívan nem használt memórialapok kerülnek hálózaton keresztül átmozgatásra, közben másolat készül minden olyan lapról, amely mégis módosult (copy-on-write) az eredeti hoston, miután már megtörtént a másolása a cél hostra.
2. Ezután a virtuális gép futását átmenetileg felfüggeszti a forrás host, és a - remélhetőleg kevés - közvetlen közelmúltban aktív memórialap másolása kezdődik meg.
3. Ha ezek átkerültek a cél hostra, akkor megtörténik a vezérlésátadás, és már a cél hoston folytatódik a virtuális gép futása.
4. Végül a még megmaradt, vagy időközben módosult, de később már nem érintett, tehát passzív lapok másolása történik meg.

Látható, hogy a virtuális gép mozgatása egy összetett folyamat, aminek célja minimalizálni a virtuális gép felfüggesztett állapotban töltött idejét. Ezt az időt jelentősen befolyásolja a hálózati kapcsolat áteresztőképessége és a virtuális gépben futó alkalmazások memóriahasználata is. Nagy memóriaterületet aktívan használó alkalmazásnál előfordulhat, hogy a memória többségét felfüggesztett állapotban kell mozgatni a folyamatos módosítások miatt, ilyenkor a kiesés ideje hosszú lehet. Fontos megjegyezni, hogy a guest virtuális háttértár tartalma nem kerül mozgatásra. Datastore-ok közötti mozgatásra a Storage VMotion vagy SVMotion technológia szolgál, amit most részletesen nem tárgyalunk.

A VMotion használatához az érintett hostokon egy-egy *vmkernel* hálózati interfészre van szükség, amin a VMotion engedélyezve van. A VMotion engedélyezés jelölőnégyzete a hálózati interfész tulajdonságait (IP cím, stb.) beállító ablakban található a *General* fül alatt. Továbbá fontos előfeltétel, hogy mindkét host közvetlenül hozzáférhessen a virtuális gép diszk tartalmát tároló datastore-hoz, ez praktikusán annyit jelent, hogy a tárhelyet iSCSI vagy NFS megosztásra kell helyezni és minden érintett host számára elérhetővé tenni. A VMFS3 fájlrendszer lehetővé teszi, hogy ugyanazt a blokkos eszközt egyszerre több ESX Server is használhassa, vagyis a konkurens hozzáférések kölcsönös kizárását biztosítja.

Az áthelyezést a mozgatni kívánt virtuális gép *Summary* fülén a *Migrate to Another Host* vagy a jobb kattintásra megjelenő kontextus menüjéből *Migrate...* menüponttal lehet kezdeményezni. Az előfeltételek ellenőrzését a VirtualCenter automatikusan elvégzi és tájékoztat arról, ha a mozgatás valamilyen okból nem kivitelezhető.

6.4. Hibatűrő fürt kialakítása

A VMware HA megoldás lehetőséget biztosít arra, hogy ha egy host meghibásodik vagy elveszik a hálózati kapcsolata, akkor a rajta futó virtuális gépek ne kerüljenek teljesen elérhetetlen állapotba. Ehhez az ESX Servereken futó külön ágenszt (*ha-agent*) telepít és használ. A megoldás előnye, hogy teljesen elosztott módon működik, a VirtualCenter Server csak a konfigurálást végzi el, a kiesések detektálása és kompenzálása már teljes egészében az ágensek feladata.

A kiesések detektálása ún. „szívverés” (*heartbeat*) jelek hálózaton keresztüli periodikus küldésével történik (alapértelmezetten 30 sec periódussal), a jelzések elmaradása a ágens számára azt jelzi, hogy a távoli hoston, vagy az azt bekötő hálózatban meghibásodás történt. A véletlen jelvesztésből adódó téves hiba-jelzések elkerülésére konfigurálható (*Advanced Options...* alatt), hogy hány egymást követő kimaradást értelmezzen meghibásodásnak (alapértelmezetten 3).

Meghibásodás esetén a kompenzációs művelet nem más, mint a kieső hoston futó virtuális gépek újraindítása egy másik, még ép hoston. Ez azt jelenti, hogy az aktuális memóriatartalom elveszik. A gyakorlatban számos esetben az is elegendő, ha a kieső virtuális gépek újraindulás után lesznek újra elérhetőek. Ilyen esetek például, ha a virtuális gép állapotmentes szolgáltatást nyújt (pl. statikus web) vagy háttértáron naplózott tranzakciókezelést valósít meg, így képes a félbeszakadt műveletek folytatására vagy hatásuk visszagörgetésére, tehát kiesés után konzisztens állapotba visszatérésre. Alkalmazás szintű hibátűrő fürtök esetén a kieső virtuális gép feladatát azonnal képes átvenni a fürt egy másik tagja, majd újraindulás után vissza tud térni az eredetileg kiesett géphez. Automatikus guest újraindítás nélkül az alkalmazás szintű fürt redundanciája is elfogyna a kieső gépek miatt, tehát a virtuális gépek szintjén működő hibátűrő fürtözés és az alkalmazás szintjén működő fürtözés egymás kiegészítő technológiái.

A VMware HA beállításait a *Cluster Edit Settings...* menüpontjában határozhatjuk meg. A VMware HA-t engedélyezhetjük vagy teljesen le is tilthatjuk az adott clusterben, vagy akár virtuális gépenként külön-külön. Továbbá meghatározhatjuk, hogy engedjen-e elindítani új virtuális gépeket, ha nincs meg a kellő redundancia a clusterben, illetve, hogy pontosan hány működőképes host ESX számít elegendő redundanciának. Megadható a virtuális gépek újraindítási prioritása, ezt a mérés során célszerű magasra állítani a várakozási idő csökkentése érdekében. Végezetül megadható, hogy egy *izolált* állapotba került, de önmagában működőképes host mit csináljon a rajta még futó virtuális gépekkel, amik várhatóan más hoston újraindulnak a hálózati kapcsolat megszakadásakor. A kapcsolat helyreállásakor konfliktusba kerülő eredeti és újonnan elindított példányai ugyanannak a virtuális gépnek kivédhetőek, ha az izolált host leállítja a gépeit.

6.5. Dinamikus terheléelosztás

A VMware DRS (Distributed Resource Scheduling) alapvetően megváltoztatja a virtuális gépek és hostok viszonyát, mivel a DRS modelljében a virtuális gépek nincsenek szorosan hosthoz rendelve, hanem az egész clusterben közösek. A hostok erőforrásai is lényegében egy nagy erőforráshalmazként jelennek meg. Természetesen a virtuális gépek ettől még mindig egy meghatározott hoston futnak, nem oszthatóak szét több hostra, de a hozzárendelést nem manuálisan, hanem teljesen automatizáltan a rendszer végzi. A pillanatnyi hozzárendelés megtekinthető a *Cluster Virtual Machines* fülén, illetve a *Maps* nézetben is.

Ha a hostok között nagy terhelésbeli eltérések alakulnak ki (ne feledjük, a virtuális gépek erőforrás-használata üzem közben folyamatosan változhat), akkor a DRS automatikusan (vagy beállítás szerint csak ajánlást téve manuális műveletre) elindítja egy-egy virtuális gép működés közbeni áthelyezését, ezáltal jobban kiegyenlítve a memória és a CPU használatot a hostok között.

Mivel kis eltérések mindig lesznek, illetve a virtuális gépek terhelése a rajtuk futó alkalmazástól függően rövid idő alatt ingadozhat, ezért nem célravezető folyamatosan minden változásra egy erőforrás-használati szempontból költséges live migrációval válaszolni. Ezért a DRS is tartalmaz beállítási lehetőségeket (szintén a *Cluster Edit Settings...* alatt található). Mindenekelőtt meghatározható, hogy automatizáltan vagy manuálisan akarjuk a műveletet kezdeményezni, továbbá az is, hogy mennyire legyen „agresszív”, azaz mennyire gyorsan reagáljon a változásokra.

Fontos beállítási lehetőség, hogy szabályokat adhatunk meg az egyes virtuális gépekre, például, hogy egyes gépeket feltétlenül tartson egy hoston (pl. a gyors, fizikai gépen belüli virtuális hálózati kapcsolat miatt), vagy ellenkezőleg mindig tartsa őket külön hostokon (ennek hibátűrő fürtöknél van jelentősége, ha egy gépre kerül minden tagja, akkor egy host hiba az egész fürtöt egyben teheti működésképtelenné). Ilyen szabályokat a *Cluster Settings*-ben a *VMware DRS/Rules* alatt adhatunk meg.