

Egységes metamodell kialakítása privát IaaS cloud rendszerekhez

Budai Péter István, Dr. Goldschmidt Balázs

Budapesti Műszaki és Gazdaságtudományi Egyetem

Irányítástechnika és Informatika Tanszék

Email: budai@iit.bme.hu, balage@iit.bme.hu

2012. március 5.

Absztrakt

A jelenleg elterjedt publikus és privát IaaS cloud megoldások az általuk kínált szolgáltatásokat tekintve ugyan nagyon hasonlóak, azonban belső felépítésükben és konfigurációs lehetőségeik terén jelentős mértékben különböznek egymástól, így az egyik rendszerről a másikra való áttérés nehézségei miatt felléphet az úgynevezett "vendor lock-in" probléma. Ez elkerülhető, ha a cloud alapú alkalmazásokat, illetve a rájuk épülő menedzsment keretrendszereket valamilyen köztes, a tényleges infrastruktúra szolgáltatótól független modell felett állítjuk össze, melyet ezután már csak le kell képezni az éppen aktuális rendszerre.

Kutatásunkban elsősorban a nyílt forráskódú, privát cloud megoldásokat vizsgáljuk. Célunk egy olyan egységes metamodell kialakítása, melynek segítségével az IaaS cloud rendszerek összes lényeges virtuális építőeleme és funkciója modellezhető. Azonosítottuk azokat a funkcionális területeket, melyeket minden szolgáltatásnak meg kell valósítania valamilyen formában, és ezek mentén vizsgáljuk a létező megoldásokat és lehetséges új megközelítéseket, így haladva az egységes és lehetőleg ideális modell kialakítása felé.

1. Bevezetés

A számítási felhő az utóbbi idők egyik legfelkapottabb informatikai kifejezésévé vált, ám valójában nem takar gyökeresen új fogalmat, hanem olyan, már korábban létező technológiákat kapcsol össze és rejt új köntösbe, mint a hardver virtualizáció, grid computing vagy éppen a szolgáltatás orientált architektúra (SOA).

A cloud computing szolgáltatás ma már számos formában létezik (infrastruktúra, platform vagy szoftver virtualizálása publikus vagy privát környezetben), de a különböző megoldások az megvalósított funkciók, logikai modell és a belső felépítés tekintetében is eltérnek egymástól, ami megnehezíti a különböző szolgáltatások egységes kezelését és az egyik szolgáltatóról a másikra

való áttérést. Célunk egy olyan metamodell kialakítása, mely lehetővé teszi a felhő alapú infrastruktúra szolgáltatások modellezését, ezáltal egy olyan köztes reprezentációt nyújtva, amely független a tényleges szolgáltatótól.

A cikk első részében összefoglaljuk a cloud computing szolgáltatások különböző fajtáit, illetve az azok körében előforduló legfontosabb fogalmakat, majd bemutatjuk a kutatásunk főbb irányvonalait és a hasonló kezdeményezésekkel való kapcsolatát.

2. Cloud computing

Ugyan a mai napig nem létezik egységes definíciója, általánosságban úgy fogalmazhatunk, hogy a cloud computing számítási erőforrás,

tárolási kapacitás és különböző szoftverek egyikének vagy mindegyikének szolgáltatásként való megjelenését jelenti.

A szakirodalomban leggyakrabban az alábbi tulajdonságokkal jellemzik a felhő alapú szolgáltatásokat, és egyben ezek azok a kulcs jellemzők, melyek megkülönböztetik a cloud computingot a bevezetésben említett rokon technológiáktól [1]:

- **Önkiszolgáló működés** (self-service): A felhőben található erőforrások felhasználási jellegükből adódóan aktuálisan felmerülő igények gyors kiszolgálására hivatottak. Ennek teljesítéséhez szükséges, hogy a felhasználók önkiszolgáló módon igényel-hessenek erőforrásokat.
- **Felhasználás alapú számlázás** (per-use billing): A felhő alapú szolgáltatások egyik legígéretesebb tulajdonsága, hogy az ügyfelek csak az általuk valóban felhasznált erőforrások (processzoridő, tárkapacitás, hálózati forgalom) után kell, hogy fizessenek. Ez a tulajdonság és az erőforrások dinamikus, önkiszolgáló jelleggel történő igénylésének és a használat utáni felszabadításának lehetősége teszik a cloud alapú megoldásokat rendkívül költséghatékonyá.
- **Skálázhatóság** (elasticity): Egy cloud computing szolgáltatás egyszerre számos felhasználót szolgál ki, akik bármikor igényelhetnek újabb erőforrásokat a meglévők mellé. A felhő szolgáltatásnak képesnek kell lennie kiszolgálnia ezeket a megnövekedett szükségleteket, a kliensek számára észrevehetetlenül biztosítva a mindkét irányba (bővítés és csökkentés) történő skálázhatóságot.
- **Testreszabhatóság** (customization): A felhő által kínált erőforrásokat felhasználók széles köre használja, így kellően általánosnak kell lenniük. A különböző igényekhez való alkalmazkodást az egyes erőforrás típusokhoz mért lehető legnagyobb fokú testre szabhatósággal teszik lehetővé.

A cloud computing szolgáltatók által biztosított erőforrásokat eddig általános módon említettük,

azonban a gyakorlatban ezek absztrakciós szintje szerint különböző kategóriákba sorolhatjuk a felhő alapú szolgáltatásokat:

- **Infrastruktúra** (Infrastructure as a Service, IaaS): A cloud szolgáltató egy virtuális számítógépet, illetve teljes virtualizált hardver környezetet biztosít, processzorral, memóriával, hálózati interfészekkel és -eszközökkel. Ez a legalacsonyabb szintű szolgáltatás, és egyben a legszélesebb körben alkalmazható, hiszen tetszőleges program (vagy operációs rendszer) virtuális számítógépre való feltelepítésével tetszőleges alkalmazás összeállítható. A legelterjedtebb cloud szolgáltatók, mint például az Amazon Web Services[2] vagy a Rackspace[3] is IaaS alapú megoldásokat kínálnak.
- **Platform** (Platform as a Service, PaaS): Ebben az esetben a cloud szolgáltató elrejtja a felhasználó elől a mögöttes hardvertartalmat és egy magasabb absztrakciós szintű platformot és hozzá kapcsolódó middleware szolgáltatásokat kínál. Ez jellemzően egy alkalmazásprogramozási interfészen (API) keresztül érhető el, melynek felhasználásával alkalmazások készíthetők az adott platformra. PaaS szolgáltatást nyújt többek között a Google is a Google App Engine elnevezésű termék formájában, melyen Java, Python és Go programozási nyelveken készült alkalmazások futtathatók.
- **Szoftver** (Software as a Service, SaaS): Számos olyan alkalmazás van, mely jellegéből adódóan kiválóan alkalmas az úgynevezett több bérlős (multi-tenant) működés megvalósítására. A SaaS cloudok esetében a szoftver jelenti magát az igénybe vett erőforrást, melyet az említett több bérlős koncepció szerint használnak a felhasználók, mindegyik úgy látja, mintha csak ő venné igénybe az alkalmazást. Ebben az esetben a testreszabhatóság már jóval szűkebb keretek között értelmezhető, leggyakrabban a megjelenésre vagy egyes funkciók be-, illetve kikapcsolására korlátozódik. Jó példa SaaS szolgáltatásra a Google Mail, Docs és

Calendar alkalmazáscsomagja [5], melyeket vállalati ügyfelek is igénybe vehetnek.

A fenti három csoporthoz hasonló betűszavak léteznek még az adat (DaaS), a kommunikáció (CaaS) és a hardver (HaaS) szolgáltatásként történő megjelenési formáira is. Jelen kutatásunkban a legelterjedtebb, infrastruktúra cloud szolgáltatásokkal foglalkozunk.

Az IaaS szolgáltatások esetében a számlázás a felhasznált virtuális hardver erőforrások (processzoridő, operatív memória, háttértár) alapján történik. A technológia legnagyobb előnye a költséghatékonyságban rejlik. A hardver virtualizációs technológiák segítségével lehetővé válik, hogy egyazon fizikai számítógépen olyan alkalmazások is futtathatók egyszerre, melyek erőforrásigényük tekintetében ugyan nem igényelnének dedikált hardvert, de sajátos szoftverkompatibilitási követelményeik (pl. operációs rendszer támogatás) miatt csak különálló számítógépre lennének telepíthetőek.

Az IaaS cloud szolgáltatásokat tovább csoportosíthatjuk aszerint, hogy azokhoz a felhasználók mely halmaza férhet hozzá.

- **Publikus** (public): Ebben a körbe tartozik az eddigi összes, példaként megemlített cloud szolgáltató. A publikus cloud szolgáltatások jellemzően bármely felhasználó számára elérhetőek a megfelelő díjazás ellenében, a hozzáférés mind az önkiszolgáló felülethez, mind pedig az igényelt virtuális erőforrásokhoz a világhálón keresztül történik. Az igényelhető processzoridő, memória- és tárhelykapacitás tekintetében ezek a rendszerek gyakorlati szempontból szinte végtelennek tekinthetők.
- **Privát** (private): Ezen típusú cloud rendszerek fizikai építőelemei felett egy intézmény vagy vállalat önálló hatáskörben rendelkezik (lehetnek saját tulajdonú vagy bérelt eszközök), és a saját telephelyén (hálózati tűzfalán, virtuális magánhálózatán) belül alakítja ki a cloud infrastruktúrát. Ennek a kialakításnak leggyakrabban adatvédelmi okai lehetnek, de az infrastruktúra cloud architektúrák előnyös tulajdonságai a privát IaaS rendszereknél is

megmutatkoznak. A rendelkezésre álló virtuális erőforrások rugalmas átrendezhetősége a fizikai erőforrások nagyobb fokú kihasználását segítik elő, így az egész infrastruktúrát tekintve alacsonyabb hardverköltséggel ugyanolyan teljesítményt érhetünk el.

- **Hibrid** (hybrid): Ezek a rendszerek egy privát és egy vagy több publikus szolgáltatás összekapcsolásával keletkeznek. A biztonságkritikus adatok és alkalmazások a privát számítási felhőben kapnak helyet, a kevésbé érzékeny vagy időnyjellegű, nagy kapacitást igénylő feladatokra pedig a publikus cloud erőforrásait veszik igénybe.

Ezzel a megoldással tovább csökkenthetők a költségek, hisz az intézményi hardver infrastruktúrát elég az átlagos kapacitásra méretezni, nem lesznek kihasználatlan erőforrások, és jelentősen csökkenthető a publikus cloud szolgáltató felé irányuló adatforgalom is.

Kutatásunkban főként a privát alkalmazásra készült, és azon belül is elsősorban a könnyen hozzáférhető, nyílt forráskódú IaaS cloud keretrendszereket vizsgáljuk. Egy ilyen keretrendszer feladata a fizikai (szerverek, hálózati kapcsolatok) és virtuális (virtuális számítógép és hálózat) erőforrások menedzselése, kiosztása és összerendelése. Vizsgálataink az alábbi IaaS rendszerekre terjednek ki:

- **Eucalyptus**: A rendszert a University of California berkelein belül kezdték fejleszteni 2008-ban és az Amazon Elastic Compute Cloud (EC2) menedzsment interfészével kompatibilis szolgáltatás kialakítása volt a cél [6]. Egy Eucalyptus konfiguráció háromféle komponensből áll. A Node Controller (NC) feladata egy fizikai kiszolgálón található virtuális számítógépek kezelésével kapcsolatos műveletek elvégzése, a Cluster Controller (CC) fizikai számítógépek egy csoportjának kezeléséért és a virtuális hálózat kialakításáért felel. A Cloud Controller (CLC) tartja nyilván az elérhető fizikai erőforrásokat és biztosítja a menedzsment interfészt a felhasználók felé [7].

- **OpenStack:** Az OpenStack projekt az amerikai űrkutatási hivatal (NASA) és a Rackspace vállalat együttműködésében készül 2010 óta és három, többé-kevésbé különálló komponensből áll, melyek a számítási teljesítmény (Compute/Nova), a tárolási kapacitás (Storage/Swift) és a virtuális gép sablonok (Imaging/Glance) szolgáltatásáért felel. A Nova komponens önmagában is moduláris felépítésű, többek között egy Amazon EC2-kompatibilis menedzsment interfész kiszolgálót (nova-api), a virtuális számítógép példányok kezeléséért felelős elemet (nova-compute), és a különböző modulokat összekapcsoló üzenetsor kiszolgálót tartalmaz [8].
- **CloudStack:** A Cloud.com által fejlesztett Java alapú rendszer 2011 óta a virtualizációs megoldások szakértőjének számító Citrix Systems támogatását is élvezi [9]. Egy CloudStack konfiguráció egy, a fizikai és virtuális erőforrásokat nyilvántartó és elosztó, valamint a felhasználói hozzáférést biztosító Management Server-ből és a fizikai kiszolgálókon futó, a lokális virtuális számítógépeket vezérlő Host Agent szoftverkomponensből áll. A lényegében egyszintű felépítésből az adminisztrátorok a fizikai komponensek négy szintű hierarchikus struktúrát állíthatnak össze a felhasználói felületen.
- **OpenNebula:** Az Európai Unió Reservoir elnevezésű kutatási projektjének keretein belül kezdték meg az OpenNebula rendszer fejlesztését. A többi privát IaaS megvalósításnál alacsonyabb szintű hozzáférést és ezáltal aprólékosabb vezérlési lehetőséget kínál a fizikai infrastruktúrához. A keretrendszer moduláris felépítésének köszönhetően lehetőséget ad publikus cloud szolgáltatásokhoz való kapcsolódásra, azaz hibrid felhő kialakítására is [10].

3. Kutatási irányok

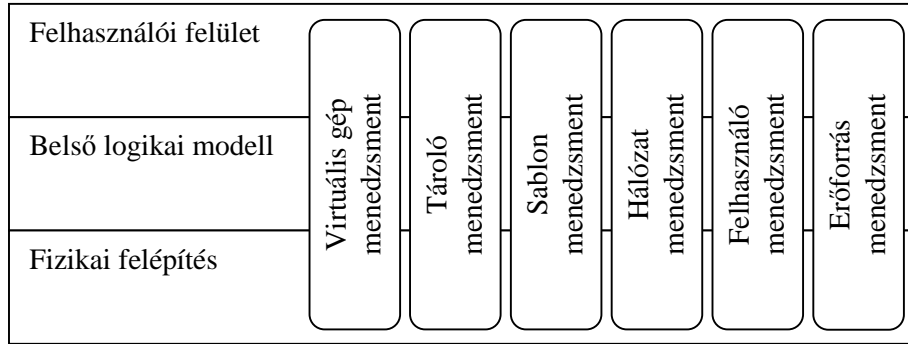
Az előző szakaszban említett IaaS keretrendszerek közös vonása, hogy a hardver

virtualizáció megvalósításához már létező megoldásokat (KVM [11], Xen [12], VMware [13]) használnak fel, azaz önmagukban is már egyfajta absztrakciót végeznek. Hasonló módon, mindegyik IaaS megoldásnak egy sajátos logikai modellje van a rendszerben található menedzselt fizikai és virtuális erőforrásokról. A szolgáltatások felhasználói elől ezek a belső reprezentációk rejtve maradnak, ők a rendszer egyszerűsített képét, a saját virtuális erőforrásaikat, illetve az azokon végezhető műveleteket látják, jellemzően valamilyen alkalmazásprogramozási interfészen keresztül.

Az egyes IaaS megoldások természetesen más és más megközelítést alkalmaznak a menedzsment szoftver és a mögöttes modell tekintetében, ez azonban hátrányként jelentkezik, ha több különböző cloud szolgáltatóval kell együttműködnünk. Még egymással kompatibilis API-t (pl. Amazon EC2) biztosító rendszereknél sem garantált, hogy minden funkciót megvalósítja az összes implementáció. Ha át kívánunk térni az egyik szolgáltatóról a másikra, akkor felléphet az úgynevezett vendor lock-in jelenség, amikor is olyan funkcionalitásra támaszkodunk, melyet a cél rendszer nem támogat.

A kutatásunk célja egy olyan metamodell megalkotása, melynek segítségével leírhatók az IaaS szolgáltatásokban fellelhető fizikai és virtuális komponensek, valamint az ezeket felhasználó komplex felhő alapú alkalmazások. Az általános modell elfedi a különböző szolgáltatók közti különbségeket és lehetőséget nyújt az erőforrás menedzsment operációk automatikus, az adott rendszerre adaptált változatának előállítását és végrehajtását. Amellett, hogy egy ilyen metamodell segíthet egy egységes fogalomrendszer kialakításában, jól alkalmazható az IaaS szolgáltatásokra épülő cloud menedzsment szoftverekben is, ahol elsődleges cél az összetett alkalmazások és változatos szolgáltatók támogatása.

A szakasz elején elmondottakból következik, hogy a különböző IaaS keretrendszereket különböző absztrakciós szinteken lehet vizsgálni, ahol az egyik szinten szereplő elem esetleg



1. ábra: A modellezés különböző szintjeinek és a cloud szolgáltatások funkcionális területeinek kapcsolata

megfeleltethető egy szomszédos szint valamely elemének vagy elemeinek. Vizsgálataink során három különböző modellezési szintet különböztettünk meg:

- **Fizikai felépítés:** Ez a szint a cloud infrastruktúra tényleges fizikai elemeinek (szerverek, hálózati kapcsolatok) felépítését takarja. Szintén ide sorolható a menedzsment keretrendszer szoftveres komponenseinek (vezérlők, ágensek) struktúrája és elhelyezkedése is. Habár ez a réteg jórészt rejtve marad, áttételesen azonban befolyásolhatja a felsőbb szintek működését (pl. a fizikai hálózat topológiája).
- **Belső logikai modell:** Ez a közttes modell írja le a cloud infrastruktúra lehetséges fizikai és virtuális elemeit, az azok között fennálló logikai kapcsolatokat, minden olyan információt tárol, mely a rendszer működtetéséhez szükséges. Ez a belső modell nagyban meghatározza, hogy milyen szolgáltatásokkal rendelkezik az adott cloud keretrendszer, hisz csak azok a funkciók megvalósíthatóak, melyeket le lehet képezni a modell elemeire.
- **Felhasználói interfész:** Az a csatorna, melyet a cloud szolgáltatás felhasználója a virtuális erőforrások igényléséhez és kezeléséhez igénybe vehet, jellemzően egy webes felület vagy webszolgáltatás API formájában. Elérhetővé teszi a mögöttes infrastruktúra szolgáltatásait, elfedve annak a felhasználóra nézve nem releváns elemeit.

Ha egy cloud szolgáltatás több felhasználói interfésszel is rendelkezik, gyakori jelenség, hogy az egyik egy natív felület, míg a másik kényelmesebben kezelhető, vagy éppen más szolgáltatóval kompatibilis, de esetenként korlátozottabb funkcionalitással bír.

Annak érdekében, hogy az egyes szintek metamodelljei kezelhető méretűek maradhassanak, érdemesnek láttunk valamilyen vízszintes irányú tagolást is bevezetni. Ehhez felmértük azokat a tágabb területeket, melyek logikailag összetartozó funkciók egy csoportját jelentik, és melyeket többé-kevésbé minden IaaS keretrendszernek támogatnia kell, hogy teljes értékű szolgáltatást nyújthasson. Ezek alapján az alábbi funkcionális területeket azonosítottuk:

- **Virtuális gép menedzsment:** A virtuális gépek életciklusának kezelése és az ütemezéssel (erőforrás hozzárendeléssel) kapcsolatos feladatok.
- **Tároló menedzsment:** A virtuális gépekhez kapcsolható tartós tároló egységek (pl. virtuális partíciók, blokkos tárolók) műveletei és az életciklusuk kezelése.
- **Sablon menedzsment:** A virtuális gépek létrehozásához használt merevlemez-képfájlok és sablonok kezelése.
- **Hálózat menedzsment:** Az elszigetelt virtuális hálózatok kiépítése, tűzfalak és egyéb hálózati eszközök konfigurációja.
- **Felhasználó menedzsment:** A cloud szolgáltatás felhasználóinak kezelése, az azonosítással és hozzáférés kezeléssel kapcsolatos feladatok.

- **Erőforrás menedzsment:** A rendszerben található erőforrások nyilvántartása és az erőforrás-használat nyomonkövetése, számlázási feladatok.

Ezek a funkcionális területek nem különülnek el egymástól teljesen és mindegyik absztrakciós szinten megjelennek, azonban az egyes szinteken más és más feladatokat rónak a keretrendszerre.

A metamodell vízszintes és függőleges irányú felosztását az 1. ábra mutatja be. A kétirányú kapcsolatokon keresztül lehetővé válik majd a kisebb részmodellek összekapcsolása egy átfogó metamodellé.

4. Kapcsolódó munkák

Az aktívan használt cloud computing szolgáltatások létrejötte megelőzte a szabványos megoldások kidolgozását. Azóta több kezdeményezés is indult, mely a miénkhez hasonló célt tűzött ki maga elé, azaz egységes interfész kialakítását, mely elfedi a felhő szolgáltatások közti különbségeket.

Az Open Cloud Manifesto [14] egy önkéntes alapon szerveződő közösségi kezdeményezés, amely a technológiai megoldásoktól független irányelveket határoz meg annak érdekében, hogy a cloud szolgáltatások átjárhatóak legyenek. A kezdeményezéshez csatlakozott vállalatok szorosabb együttműködés mellett kötelezik el magukat a cél elérése érdekében.

Sokkal gyakorlatiasabb megközelítést jelent a Dasein Cloud API elnevezésű projekt, melynek célja egy Java nyelvű alkalmazásprogramozási interfész kialakítása [15]. Mint a neve is mutatja, a különböző cloud szolgáltatások legfelső rétege, a programozható felhasználói felület fölé nyújt egy absztrakciós réteget. A Dasein API által biztosított funkcionalitás nem bővíthető, viszont egy adott felhő szolgáltatóhoz készült implementációnak nem feltétlenül kell támogatnia minden specifikált műveletet.

5. Összefoglalás

Kutatásunkban egy egységes metamodell megalkotását tűztük ki célul magunk elé, mellyel a privát IaaS cloud szolgáltatások modellezhetők.

Ehhez megalkottunk egy funkcionális és absztrakciós szintek szerinti kétdimenziós felosztást, ami alapján kisebb, könnyebben kezelhető részmodellek alakíthatók ki, melyek később összeilleszthetők a teljes metamodell létrehozásához.

A kutatásunk következő szakaszában a már meglévő modellek bővítését és további IaaS szolgáltatók vizsgálatát tervezzük elvégezni, a teljes metamodell elkészülte után pedig egy referencia implementáció elkészítésével bizonyítjuk annak gyakorlati alkalmazásában rejlő előnyöket.

6. Köszönetnyilvánítás

A jelen cikk alapjául szolgáló munka szakmai tartalma kapcsolódik a "Új tehetséggondozó programok és kutatások a Műegyetem tudományos műhelyeiben" c. projekt szakmai célkitűzéseinek megvalósításához. A projekt megvalósítását a TÁMOP - 4.2.2.B-10/1-2010-0009 program támogatja.

7. Hivatkozások

- [1] R. Buyya, J. Broberg, A. Goscinski: *Cloud Computing: Principles and Paradigms*, Wiley, 2011, ISBN 978-0-470-88799-8
- [2] Amazon Web Services: *Products and Services*, <http://aws.amazon.com/products/>, letöltve: 2012. 02. 29.
- [3] Rackspace Hosting: *Cloud Products and Cloud Services*, http://www.rackspace.com/cloud/cloud_hosting_products/, letöltve: 2012. 02. 29.
- [4] Google: *What is Google App Engine?*, http://code.google.com/appengine/docs/what_isgoogleappengine.html, letöltve: 2012. 02. 29.
- [5] *Google Apps for Business*, <http://www.google.com/apps/intl/en/business/smb.html>, letöltve: 2012. 02. 29.
- [6] D. Nurmi et al., *The Eucalyptus Open-Source Cloud-Computing System*, Cloud Computing and Applications 2008 (CCA 08), 2008

- [7] Eucalyptus Systems, Inc.: *Eucalyptus Open-Source Cloud Computing Infrastructure - An Overview*, 2009
- [8] CSS Corp: *OpenStack Compute Starter Guide*, 2011, <http://docs.openstack.org/diablo/openstack-compute/starter/openstack-starter-guide-diablo.pdf>, letöltve: 2011. 03. 02.
- [9] CloudStack: *CloudStack - Open Source Cloud Computing Project*, <http://cloudstack.org/about-cloudstack.html>, letöltve: 2012. 03. 01.
- [10] B. Sotomayor, R. S. Montero, I. M. Llorente, I. Foster: *Virtual Infrastructure Management in Private and Hybrid Clouds*, IEEE Internet Computing, vol. 13, no. 5, pp. 14-22, Sep./Oct. 2009.
- [11] *Kernel Based Virtual Machine weboldal*, http://www.linux-kvm.org/page/Main_Page, letöltve: 2012. 03. 04.
- [12] *Xen hypervisor weboldal*, <http://xen.org/>, letöltve: 2012. 03. 04.
- [13] *VMware weboldal*, <http://www.vmware.com/worldwide.html>, letöltve: 2012. 03. 04.
- [14] *Open Cloud Manifesto*, 2009, <http://www.opencloudmanifesto.org/OpenCloudManifesto.pdf>
- [15] enStratus Networks LLC: *Dasein Cloud API weboldal*, <http://dasein-cloud.sourceforge.net/>, letöltve: 2012. 03. 02.