



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Irányítástechnika és Informatika Tanszék

Kiszolgálók üzemeltetése nagyvállalati környezetben és kapcsolódó szoftverek fejlesztése

SZAKDOLGOZAT

Készítette
Főglein Simon István

Konzulens
dr. Somogyi Péter

2024. március 24.

Tartalomjegyzék

Kivonat	i
Abstract	ii
1. Bevezetés	1
1.1. Nagyvállalati környezetek ismertetése	1
1.2. Virtualizáció	1
1.2.1. Népszerű virtualizációs technológiák	2
1.2.2. Virtuális gépek használatának néhány előnye	3
1.2.2.1. Erőforrások testreszabása	3
1.2.2.2. Snapshotok	3
1.2.2.3. Migráció	3
1.2.3. Teljes virtualizáció és paravirtualizáció összehasonlítása	4
1.2.4. Konténerizáció	4
1.3. Logikai kötetkezelés	4
1.3.1. Snapshotok, mentések készítése	4
2. Saját munka bemutatása	5
2.1. Kialakítani kívánt környezet meghatározása	5
2.2. Fizikai gép ismertetése	5
2.3. Operációs rendszer	5
2.3.1. SUSE	5
2.3.2. Debian, Ubuntu	5
2.4. Gépmenedzsment: Salt	5
2.4.1. Más hasonló megoldások	5
2.5. Monitoring	5
2.6. Továbbfejlesztési lehetőségek	5
Köszönetnyilvánítás	6
Irodalomjegyzék	7

HALLGATÓI NYILATKOZAT

Alulírott *Főglein Simon István*, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2024. március 24.

Főglein Simon István
hallgató

Kivonat

Jelen dokumentum egy diplomaterv sablon, amely formai keretet ad a BME Villamosmérnöki és Informatikai Karán végző hallgatók által elkészítendő szakdolgozatnak és diplomatervnek. A sablon használata opcionális. Ez a sablon \LaTeX alapú, a *TeXLive* \TeX -implementációval és a PDF- \LaTeX fordítóval működőképes.

Abstract

This document is a L^AT_EX-based skeleton for BSc/MSc theses of students at the Electrical Engineering and Informatics Faculty, Budapest University of Technology and Economics. The usage of this skeleton is optional. It has been tested with the *TeXLive* T_EX implementation, and it requires the PDF-L^AT_EX compiler.

1. fejezet

Bevezetés

Napjainkban az informatika és az internet életünk szerves részévé vált. A számos szolgáltatás folyamatos rendelkezésre állásának biztosítása és a megnövekedett forgalom kiszolgálása jó néhány új technológia kifejlesztését követelte meg.

Dolgozatomban elsősorban egy általános képet szeretnék adni arról, hogy milyen üzemeltetési kihívásokkal kell szembenéznünk, ha egy ilyen szolgáltatás működtetésébe vágjuk a fejszénket. Értekezésemben nem fogok kitérni bizonyos infrastrukturális hátterekre – mint például a kiszolgálók folyamatos energiaellátásának biztosítása –, ezeket adottnak fogom tekinteni, hiszen ezt egy adatközpontban bérelt hely esetén sem magunknak kell biztosítanunk. A következőkben sokkal inkább az informatikai lehetőségek tárgyalására fogom helyezni a hangsúlyt: hogyan tudunk hatékonyan üzemeltetni több kiszolgálót, milyen módon lehet biztosítani a szolgáltatásaink lehető legnagyobb rendelkezésre állását, és hogyan védhetjük meg adatainkat egy esetlegesen félresikerült rendszerfrissítést követően. Mindezt egy kisebb volumenű tesztrendszerben is be fogom mutatni.

1.1. Nagyvállalati környezetek ismertetése

A legtöbb hétköznapi felhasználó számára ismeretlen vagy meglepő lehet, hogy maga az internet és az ezen keresztül elérhető szolgáltatások – gondoljunk például az Ügyfélkapura vagy az internetbank-szolgáltatásokra – nagyon komplex rendszerek nem csak szoftveres, hanem informatikai infrastruktúra szempontjából is. A legtöbb ilyen szolgáltatás egy adatközpontban lévő szerveren fut, ami a beérkező kérésekre ad válaszokat. Ezt a folyamatot úgy is felfoghatjuk, hogy az ilyen szolgáltatások felhasználói lényegében az adott szolgáltató (a fenti példánál maradva a Magyar Állam és az adott bankok) számítógépeivel kommunikálnak.

Ezek a szervergépek több lényeges különbséggel is bírnak a személyi számítógépekkel szemben. Egyik legfontosabb tulajdonságuk, hogy hibatűrőek bizonyos hardverhibákat illetően: szinte minden főbb komponensből legalább kettő áll rendelkezésre, így ha az egyik meg is hibásodik, akkor a hiba elhárításáig a beépített redundancia miatt a gép képes tovább funkcionálni, általában a felhasználók felé észrevétlenül, míg a gép üzemeltetői figyelmeztetést kapnak a hiba típusáról és a kapcsolódó tennivalókról.

1.2. Virtualizáció

A fent említett megnövekedett forgalom kiszolgálását hatékonyan lehet kezelni úgy, hogy olyan fizikai számítógépet helyezünk üzembe, mely több, egymástól független operációs rendszer futtatására is alkalmas. Ilyenkor ezeket a fizikai gépen futó rendszereket virtuális gépeknek (virtual machine, VM) nevezzük. Egy virtuális gép elkülönített erőforrásokat

kap a fizikai géptől, hozzáférhet például bizonyos mennyiségű processzormaghoz, memóriához, illetve külön háttértár-partíciói is lehetnek. A virtualizált hardverek és operációs rendszerek a legtöbb esetben a külvilág felé nem különböztethetők meg a fizikai számítógépektől, és ezzel a megoldással jelentősen csökkenthető a rendszerek és a hozzájuk szükséges informatikai infrastruktúra üzemeltetésének költsége.

A virtualizáció nagy ereje abban rejlik, hogy bizonyos hardverek virtualizációjával egységyi teljesítményt olcsóbban kaphatunk meg, mintha külön fizikai gépeket helyeznénk üzembe, illetve nagyobb rugalmasságot kapunk a kezelésükben, üzemeltetésükben. Képzeljük el, hogy megveszünk egy számítógépet, amin szeretnénk futtatni egy számunkra fontos alkalmazást, mondjuk a honlapunkat. Ilyenkor az ezen a gépen futó operációs rendszer teljes mértékben megszabhatja, hogy milyen erőforrásokból mennyit használ. Ha egy másik szolgáltatást – például levelezőszervert – szeretnénk emellett futtatni, akkor limitáltabbak a lehetőségeink, hiszen a korábban telepített webszerver már foglal bizonyos erőforrásokat, illetve a program függőségeit és konfigurációs fájljait is telepítettük már, ami esetleg negatívan hat a levelezőszerverünk működésére. Ha mindezt virtualizált környezetben tesszük meg, akkor a topológia megváltozik: a két alkalmazás teljesen elkülönítetten, egymás zavarása nélkül, különböző virtuális gépekben futhatnak, míg magán a fizikai gépen egy úgynevezett hypervisor látja el az erőforrások ütemezésének és kiosztásának (pl. processzoridő, memória) feladatát.

1.2.1. Népszerű virtualizációs technológiák

Mivel a virtualizáció napjainkban nagyon elterjedt technológia, számos olyan megoldás született, mely egyszerűsíti a virtuális gépek üzemeltetését. Ezek közül nagy ismertségnek örvend az Oracle VirtualBox és a VMware Player, azonban ezek a megoldások nem skálázódnak annyira jól, mint a továbbiakban tárgyalt társaik, melyek sokkal megfelelőbbek nagyvállalati szerverkörnyezetben való alkalmazásra. Ezek a megoldások lehetőséget biztosítanak a virtuális gépek távoli elérésre, kezelésére, egyszerűbb telepítésükre, valamint szükség esetén elosztott működésükre.

Ilyen nagyvállalati környezetben is kedvelt megoldás például a VMware ESXi, amely egy igen modern hypervisor számos kényelmi funkcióval ellátva (lehetőség van például a rendszer webes felületről való kezelésére és virtuális gépek sablonból való gyors (nagyságrendileg 5-10 perc) telepítésére). Egy másik kedvelt megoldás az ESXi-vel ellentétben teljesen ingyenesen, GPLv2-es licenc alatt elérhető XEN hypervisor, mely ugyan kevesebb kényelmi funkciót tartalmaz, de szintén népszerűségnek örvend széleskörű támogatása, kedvező teljesítménye és szabad szoftver voltából eredő ingyenessége miatt. A XEN a 2014 márciusában kiadott 4.4-es verzió óta stabilan működik együtt a libvirt virtualizációs API-val, amely nagyban megkönnyíti a hypervisorral való kommunikációt a virtuális gépek konfigurálása során. [1] A XEN-hez hasonlóan szabad szoftver licenccel érhető el a Kernel-based Virtual Machine (KVM) is, mely a XEN-nél modernebb megoldásnak tekinthető, és manapság széles körben használják a Linux kernelbe való integráltságának és stabilitásának köszönhetően. Bár maga a KVM nem tartalmaz ilyet, de számos interfész elérhető az ezen keresztül futtatott virtuális gépek kezelésére (például virt-manager), valamint akadnak olyan megoldások is, melyek a KVM-re alapozva nyújtanak szélesebb körű virtualizációs megoldást, ilyen lehet¹ például a Proxmox és a Cockpit.

¹A konfigurációtól függően akár többfajta virtualizációs környezet is beállítható, de a KVM az egyik legjobban támogatott.

1.2.2. Virtuális gépek használatának néhány előnye

A virtualizáció számos előnnyel bír a szervertinfrastruktúra karbantartása, könnyű kezelhetősége szempontjából, ebből néhány fontosabbat szeretnék kiemelni.

1.2.2.1. Erőforrások testreszabása

Amikor több tíz vagy több száz szerver üzemeltetéséről van szó, akkor hatványozottan számításba kell vennünk az egyes gépekre jutó költségeket. Virtuális gépek esetén ez azért kedvezőbb egy fizikai gépnél, mert ugyan a nagyvállalati környezetbe szánt szervergépek jelentősen drágábbak a személyes felhasználásra tervezett társaiknál, de akár több tíz virtuális gép egyidejű futtatását is lehetővé teszik. Ezáltal az egy fizikai gépre eső, asztali gépeknél megszokott áramfogyasztáshoz képest jóval nagyobb energiafelvétel sokkal kedvezőbb arányt mutat, ha számításba vesszük a futtatott virtuális kiszolgálók számát is.

Mindezek mellett a nagyvállalati felhasználáshoz tervezett számítógépek jóval hibátűrőbbek, hiszen a főbb komponensek redundánsan lettek kialakítva: ezekből az ilyen szerverekben legalább kettő van, és a rendszer automatikusan képes detektálni a hardveres hibákat, és ezek figyelembe vételével tovább működni. További előny lehet még hardveres hibák esetén, hogy ezek a számítógépek széles körben támogatják az úgynevezett *hot swappingot*, amely azt jelenti, hogy bizonyos hardverelemek (általában például háttértárak és memóriamodulok) a számítógép bekapcsolt állapotában, annak működése közben is szolgáltatás nélkül cserélhetőek.

Előnyös lehet továbbá, hogy a virtuális gépek erőforrásai szabadon módosíthatók, így akár két újraindítás között is változtathatjuk a rendelkezésre álló memória mennyiségét vagy épp a processzormagok számát. Sőt, egyes hypervisorok és operációs rendszerek ezen erőforrások futásidejű megváltoztatását is támogatják bizonyos korlátozások mellett, így gyakorlatilag a fontosabb virtualizált erőforrások is hot swappelhetőnek tekinthetők.

1.2.2.2. Snapshotok

Egy másik kedvező lehetőség virtuális gépek használata esetén az, hogy úgynevezett snapshotokat készíthetünk róluk. Ezek a snapshotok a gépet egy adott pillanatbeli állapotban reprezentálják, és később ezeket az állapotokat visszaállíthatjuk, ha szükségünk lesz rá. Egyes megoldások a memóriakép mentését is támogatják, így akár egy futó gép is könnyen visszaállítható. A snapshotok készítése hasznos lehet például rendszerfrissítések esetén, így ha valamiféle hiba lép fel a frissítés során, vagy egy adott szoftver nem megfelelően működik azt követően, akkor a frissítés előtt készített snapshotra visszaállva újra teljes értékűen üzemelhet a szerver, amíg a frissítés során fellépő hibát elhárítjuk.

1.2.2.3. Migráció

Részben az előző ponthoz kapcsolódik a virtuális gépek migrációja. Ez a funkció azt jelenti, hogy egy adott fizikai gépről, mely virtuális gépeket futtat (virtual host), készíthetünk egy snapshotot, amit áthelyezhetünk egy másik virtual hostra, és a virtuális gép ezen futhat tovább egyéb újrakonfigurálás nélkül. Lehetőség van azonban a háttértárak tartalmát elhagyva is átmozgatni egy VM-et egy másik hosztra. Ehhez bevett szokás leírófájlok használata, mely egy virtuális gép konfigurációját tartalmazza. A leírófájlt egy másik hosztgépre áthelyezve ott újra elindíthatjuk a definiált virtuális gépet. Ilyenkor szükség lehet a VM háttértárainak inicializálására, de ettől eltekintve a konfiguráció szabadon hordozható virtual host-ok között. Ilyen migrációra egyes megoldások fejlettebb támogatást is adnak, így akár valós időben, az aktuális terheltség figyelembe vétele mellett automatikusan is áthelyezhetőek virtuális gépek a megadott fizikai hosztok között.

1.2.3. Teljes virtualizáció és paravirtualizáció összehasonlítása

1.2.4. Konténerizáció

1.3. Logikai kötetkezelés

1.3.1. Snapshotok, mentések készítése

2. fejezet

Saját munka bemutatása

2.1. Kialakítani kívánt környezet meghatározása

Dolgozatomban egy kisebb léptékű, de a fontosabb elvek ismertetését kellő mértékben lehetővé tevő tesztkörnyezetet fogok kialakítani és részletesen bemutatni. A tesztkörnyezetben egy fizikai gépen (virtual host, dom0) fogok virtuális gépeket kialakítani a KVM hypervisor segítségével. Ezen környezet célja, hogy betekintést engedjen a nagyvállalati környezetek kialakításának fontosabb lépéseibe. Ahogy arról a bevezetés során szó volt, ilyen környezetekben nem ritka, hogy több száz vagy több ezer gépet kell kezelnünk. Nyilvánvaló, hogy ilyen nagyságrendű rendszert nem lehet kizárólag manuális megoldásokkal kezelni, nem várhatjuk el a rendszergazdától és az üzemeltetést végző mérnököktől, hogy egy-egy frissítést minden gépen külön-külön telepítsenek: ehhez sok esetben nem is adottak a lehetőségek, hiszen a legtöbb ilyen számítógép egy adatközpontban fut, ahova a biztonsági előírások miatt nem tudna minden munkatárs bejutni, továbbá körülményes lenne az egyes számítógépekhez külön-külön perifériákat csatlakoztatni, amire például a nagyobb operációsrendszer-verzióváltások során szükség lehetne az úgynevezett *offline upgrade*-ek miatt – ilyen helyzetekben a távoli elérés (pl. SSH, VNC) sem jelent megoldást.

Mindezen okok miatt a dolgozatomban egy infrastruktúramenedzsmint eszközt fogok használni, amivel növelhető az egyes kiszolgálók karbantartásának hatékonysága mind fizikai, mind pedig virtuális gépek esetén.

2.2. Fizikai gép ismertetése

2.3. Operációs rendszer

Értekezésemben nagy szerepe lesz a választott operációs rendszereknek, hiszen ezek fognak a virtualizációs rendszer alapjául szolgálni, valamint képesnek kell lennünk a gépek távoli menedzsmentjére is, így mindenképpen olyan megoldásra van szükség, amely jól támogatott a választott infrastruktúramenedzsmint eszköz által. Fontos szempont volt továbbá, hogy a tesztkörnyezet a lehetőségekhez mérten jól reprezentálja a nagyvállalati környezetben használatos rendszereket, így sok olyan OS-verzió kikerült a lehetőségek közül, amelyek ugyan népszerűek például asztali megoldásként, de egyes nagyvállalati szoftverek (legyen az adatbázismotor, vagy bizonyos eszközvezérlők, driverek) hivatalosan nem támogatottak rajtuk. Emiatt az operációs rendszerek kiválasztása során körültekintően jártam el, több Linux-disztribúció is szóba került, az ezekről született konklúziót itt foglalom össze néhány mondatban.

2.3.1. Debian, Ubuntu

2.3.2. SUSE

2.4. Gépmenedzsment: Salt

2.4.1. Más hasonló megoldások

2.5. Monitoring

2.6. Továbbfejlesztési lehetőségek

Köszönetnyilvánítás

Ez nem kötelező, akár törölhető is. Ha a szerző szükségét érzi, itt lehet köszönetet nyilvánítani azoknak, akik hozzájárultak munkájukkal ahhoz, hogy a hallgató a szakdolgozatban vagy diplomamunkában leírt feladatokat sikeresen elvégezze. A konzulensnek való köszönetnyilvánítás sem kötelező, a konzulensnek hivatalosan is dolga, hogy a hallgatót konzultálja.

Irodalomjegyzék

- [1] XEN Project: Xen project 4.4 release notes (2024. március 20.). https://wiki.xenproject.org/wiki/Xen_Project_4.4_Release_Notes.