



M Ű E G Y E T E M 1 7 8 2

# **Szolgáltatásbiztonságra tervezés laboratórium (VIMIM236)**

Szolgáltatásbiztonsági benchmarkok

Mérési útmutató

Készítette: Kocsis Imre

2010.11.25.

Budapesti Műszaki és Gazdaságtudományi Egyetem  
Méréstechnika és Információs Rendszerek Tanszék

A mérés során két típusú virtuális géppel fogunk dolgozni; egy Windows Server 2003 és egy CentOS 5-el. A mérést géppárokon fogjuk végezni, oly módon, hogy az egyik gépen mérési célpontként futtatunk egy Windows és egy Linux VM-et, a másikon pedig mérőállomásként a Windows VM egy példányát. (A Windows VM mindkét szerepre fell lett készítve.) A gépek „bridged” VMWare hálózati módba vannak állítva, ily módon a fizikai laborgépeket is kiszolgáló DHCP szolgáltatástól kapják IP címüket. Ennek megfelelően az IP cím ütközések elkerülése érdekében kiemelten fontos, hogy a virtuális gépek első indításakor a „másoltuk vagy mozgattuk a VM”-et kérdésre úgy válaszoljunk, hogy

### Másoltuk („I copied it”)

Ennek hatására új MAC cím generálódik a virtuális hálózati kártyához, ami pedig VM-példányonként különböző IP címekhez juttat minket.

A virtuális gépekre belépéshez a root/LaborImage, illetve az Administrator/LaborImage felhasználói név/jelszó párosokat használjuk.

Mérő gépnek a géppárok CoolerMaster, mért gépnek pedig a Lenovo tagját javasoljuk. A mérő gépen javasoljuk továbbá a VM CPU és memória-erőforrásainak bővítését.

## 1 Ismerkedés a mérési környezettel

Mindkét virtuális gépen telepítésre került egy Geronimo J2EE kiszolgáló, melyre egy klasszikus benchmark alkalmazást, az Apache Geronimo daytrader-t telepítettünk (korábban IBM Trade6).

I./1. A Geronimo egyik gépen sem lett telepítve szolgáltatásként. Keressük meg a Geronimo indítási mappáját (C:\geronimo\bin, illetve /home/meres/geronimo/bin) és konzolból/terminálból próbáljuk ki a kiszolgáló elindítását és leállítását! Javasolt a geronimo.bat, illetve geronimo.sh parancsállományok használata. Ezeket paraméterek nélkül futtatva segítséget kapunk használatukról.

A Linux VM-en a kiszolgáló sikeres elindításához a JAVA\_HOME környezeti változónak be kell állítva lennie. Ezt legegyszerűbben a Geronimo indítására használt terminálban tehetjük meg a következő paranccsal:

```
export JAVA_HOME=/usr/lib/jvm/java-6-openjdk
```

Az így eszközölt beállítás persze csak az aktuális bash folyamatra és annak gyermekeire lesz érvényes. Az export parancsot a szokásos módon a /etc/profile-ban is elhelyezheti.

Legalábbis a Geronimo „tisztá” leállításához szükségünk lesz az alapértelmezett adminisztrátori felhasználó/jelszó párosra: system / manager. Az alkalmazás-kiszolgáló nyitóoldala a http://<IP cím>:8080 URL-en található sikeres indítás után.

I./2. A mérő gépről böngészőből próbáljuk ki és fedezzük fel a mért gépen futó VM-eken elindított alkalmazást! Elérése: http://<IP cím>:8080/daytrader. Az alkalmazás egy részvénykereskedési rendszert emulál. Olvassuk el a FAQ-t is!

Benchmarkolás szempontjából az alkalmazás különlegességei:

- Szintetikus, tisztán benchmark jellegű hívásokhoz is biztosít URL-eket. Próbáljuk ezeket is ki!
- Rendelkezik URL-ként „hívható” konfigurációs eszközökkel (mint pl. adatbázis újrainicializálása; lásd „Configuration” fül). Ezeket mérjük fel; alkalmazásuk javasolt a mérési kampány(ok) során.
- A http://<IP cím>:8080/daytrader/scenario szervlet segítségével kimondottan egyszerűen (is) „meghajtható”. (Lásd FAQ, „What is the TradeScenario servlet?”)

I./3. A mérés során a mért virtuális gépekbe hibákat fogunk injektálni. VMWare környezetben ennek a legegyszerűbb módja a vmrun parancs használata a mért gazdahozton. Ismerkedjünk meg a vmrun paranccsal (lásd [www.vmware.com/pdf/vix160\\_vmrun\\_command.pdf](http://www.vmware.com/pdf/vix160_vmrun_command.pdf), illetve az argumentumok nélkül futtatott parancs által kiírt dokumentációt) és próbáljunk meg a Windows vendég-gépen egy calc.exe-t elindítani, a Linuxon pedig egy új állományt touch-olni (lásd man touch)!

A vmrun paraméterezése nem feltétlenül triviális; segítségként néhány tipikus felhasználási eset – Linux gazdagépről, Linux és Windows VM-eken végezve. (A használt kapcsolókat javasolt a megfelelő dokumentációk segítségével értelmezni.)

Program indítása Windows vendégen:

```
vmrun -gu Administrator -gp LaborImage runProgramInGuest /media/vmware-images/oktatas/07-DepyBenchmarking/dbench_windows/WS2003_R2_Ent_Eng_Blank.vmx „C:\Windows\system32\notepad.exe”
```

Cygwin Bash indítása Windows vendégen „login shellként”, majd azzal program futtatása. (A stress terhelésgeneráló eszköz Windows-on csak Cygwin környezetben érhető el.)

```
vmrun -gu Administrator -gp LaborImage runProgramInGuest /media/vmware-images/oktatas/07-DepyBenchmarking/dbench_windows/WS2003_R2_Ent_Eng_Blank.vmx „C:\cygwin\bin\bash.exe” -l -c „/usr/local/bin/stress -c 2 -t 10000”
```

Nem grafikus program futtatása Linux vendégen:

```
vmrun -gu meres -gp LaborImage runProgramInGuest /media/vmware-images/oktatas/07-DepyBenchmarking/dbench_ubuntu/Ubuntu-10.04.vmx /usr/bin/stress -c 2 -t 10
```

Adott hálózati interfész lekapcsolása Ubuntu 10.04 vendégen:

```
vmrun -gu meres -gp LaborImage runScriptInGuest /media/vmware-images/oktatas/07-DepyBenchmarking/dbench_ubuntu/Ubuntu-10.04.vmx /bin/bash „echo LaborImage | sudo -S ifconfig eth1 down”
```

Adott hálózati interfész lekapcsolása Windows vendégen. (Figyelem: a hálózati kapcsolatok átnevezhetők!)

```
vmrun -gu Administrator -gp LaborImage runScriptInGuest /media/vmware-images/oktatas/07-DepyBenchmarking/dbench_windows/WS2003_R2_Ent_Eng_Blank.vmx „cmd.exe /C \"C:\WINDOWS\system32\netsh.exe\" interface set interface name=local admin=disable”
```

I./4. A mérés során a következő hibaokokat fogjuk injektálni (ezzel operátori, illetve környezeti hibákat emulálva): hálózati interfész kiesése, Geronimo folyamat leállása, illetve a „memory hog”, „I/O hog”, „CPU-hog” hármából két szabadon választott.

- Geronimo kiesése: emulálható a shutdown és startup szkriptekkel.
- Hálózati interfész kiesése: próbáljuk ki az ifdown/ifup eth0, illetve a netsh interface set interface name="thename" admin=disable/enable parancsokat! (Praktikusan vmrun segítségével.)
- Az erőforrás-hibáknál választási lehetőséget biztosítunk a készen hozott alkalmazás és a „stress” nevű eszköz között. A stress által önmagáról adott dokumentáció a méréshez elengedő; a Windows Server hoszton cygwin alkalmazásként került telepítve a C:\cygwin\usr\local\bin\könyvtárba.

## 2 Alapteljesítmény („baseline performance”) mérése

II./1. Készítsünk egy-egy snapshotot a virtuális gépek kiinduló állapotairól, amikhez majd rendre vissza tudunk térni!

II./2. Az alapteljesítmény méréséhez a mérő hoszton található Windows Serveren készítsünk egy terhelési profilt! A terhelést méretezzük úgy kísérleti úton, hogy az mindkét mért virtuális gépen elfogadható teljesítményt adjon, de ne legyen triviálisan alacsony! A terhelésgenerálásra és teljesítménymérésre használja a Windows VM-ben telepített JMeter-t! (A JMeter használata egy megelőző mérésnek már tárgya volt.)

II./3. A mérésen az átlagos áteresztőképesség, az átlagos válaszütem és az átlagos hibaráta metrikákat szeretnénk benchmark metrikaként alkalmazni. Ezek méréséhez és rögzítéséhez egészítse ki megfelelő Listener-ekkel a terhelési profilt!

II./4. Nem túl nagy, de nem is nulla felfutási időt alkalmazva mérje meg az alapteljesítményt egy öt perces méréssel! (N.B. az itt elkészített profilt a hibainjektálások során módosítás nélkül fogjuk alkalmazni.)

### **3 Teljesítménymérések hibák jelenlétében**

III./1. A mérési segédlet és a kiválasztott/előírt hibák alapján definiálja az injektálási réseket! Javasolt, hogy a hálózati interfész és a szervlet-kiszolgáló esetén ~40-50 másodpercenként legyen hibainjektálás (melyet ~5 másodperccel követ a jelen esetben automatikus helyreállítás), az erőforrás-foglaló megoldások esetén pedig ~30 másodpercenként úgy, hogy a hiba max. 10 másodpercig legyen aktív.

III./2. A hibák időzített injektálásához javasolt szkriptelt megoldást alkalmazni a mérő gazdagépen. Tetszőleges szkriptnyelvet (vagy akár programozási nyelvet) alkalmazhat.

III./3. Hajtsa végre a hibainjektálási kampányt és rögzítse az eredményeket! Jelen mérés során a hibainjektálást és a terhelés elindítását nem szinkronizáljuk; a hibainjektálást indítsuk el „kézzel” a felfutási szakasz alatt! (Automatizált megoldásokat szívesen látunk, de nem követelmény.)

III./4. Hasonlítsa össze a két mért környezetet szolgáltatásbiztonság szempontjából a mérési eredmények alapján!