Budapesti Műszaki és Gazdaságtudományi Egyetem
Virtualizációs technológiák és alkalmazásaik (VIMIAV89)

Virtuális gépben futtatott CPU-emulátor teljesítményének vizsgálata

Házi feladat

Tóth Tamás (RRDDD0)
2011. október 31.

1 Bevezető

A beágyazott vagy mobil rendszerek esetén a fejlesztés során a fejlesztő tipikusan valami emulátorban futtatja a készülő alkalmazást, és nem minden egyes esetben valós hardvert használ. A feladat annak megvizsgálása, hogy hogyan alakul a CPU-emulátor teljesítménye és használhatósága, ha ráadásul azt még egy virtuális gépen belül futtatjuk.

1.1 A kitűzött feladat

A vizsgálódás konkrét tárgyát a Windows alatt is elérhető **Bochs** emulátoron belül futtatott **Debian** operációs rendszer teljesítményének vizsgálata képezi. Az emulátoron belül a terhelést a **bc** program futtatása fogja generálni.

2 Ismerkedés a Bochs-szal

A Bochs egy nyílt forráskodú, C++ nyelven írt IA-32 (x86) emulátor. Az eszköz lehetőséget ad az emulált processzor sebességének és utasításkészletének (MMX, SSEx, 3DNow!) beállítására, ezáltal többféle processzor emulálására alkalmas (386, 486, Pentium, PentiumII, PentiumIII, Pentium4, vagy akár 64 bites processzorok). Az Intel x86 processzorcsalád emulációján kívül képes különböző I/O-eszközök, illetve egy saját BIOS emulációjára.

2.1 Előre konfigurált DLX Linux kipróbálása

A Bochs installálásakor a telepítendő komponensek kiválasztásakor megadható, hogy települjön egy előre konfigurált, DLX Linuxot futtató környezet.

Az opció kiválasztásával a telepítés után az alkalmazás mappájában létrejön egy *dlxlinux* nevű könyvtár, mely az alábbi fontosabb állományokat tartalmazza:

Fájlnév	Tartalom
bochsout.txt	A futtatáskor létrejövő log.
bochsrc.bxrc	Az emulált környezet beállításait tartalmazó szöveges állomány.
hd10meg.img	10 MB-os, a telepített operációs rendszert tartalmazó image.
run.bat	"bochs -q -f bochsrc.bxrc"

2.1.1 Beállítások vizsglata

A konfigurációs állomány néhány érdekesebb bejegyzése:

Memóriabeállítások: host által foglalt és a guest által elérhető memória MB-ban, valamint a beállított BIOS és Video BIOS képállományának elérési útja:

```
memory: host=32, guest=32
romimage: file="../BIOS-bochs-latest"
vgaromimage: file="../VGABIOS-lgpl-latest"
```

Bootolás és merevlemez beállításai: bootolási sorrend és merevlemez fizikai paraméterei, valamint lemez képállományának elérési útja.

```
boot: disk
ata0-master: type=disk, mode=flat, translation=auto, path="hd10meg.img",
cylinders=306, heads=4, spt=17, biosdetect=auto, model="Generic 1234"
```

Processzorbeállítások: CPU darabszáma és sebessége IPS-ben, hibakezelési beállításai, utasításkészlete, CPU stepping, valamint az emulátor belső idejének a hoszthoz történő szinkronizálásának módja (ld. később)

```
cpu: count=1, ips=4000000, reset_on_triple_fault=1, ignore_bad_msrs=1
cpuid: cpuid_limit_winnt=0, mmx=1, sse=sse2, xapic=1, sep=1, aes=0, xsave=0,
movbe=0, 1g_pages=0, pcid=0 fsgsbase=0
cpuid: stepping=3, vendor_string="GenuineIntel", brand_string="
Intel(R) Pentium(R) 4 CPU        "
clock: sync=none, time0=local
```

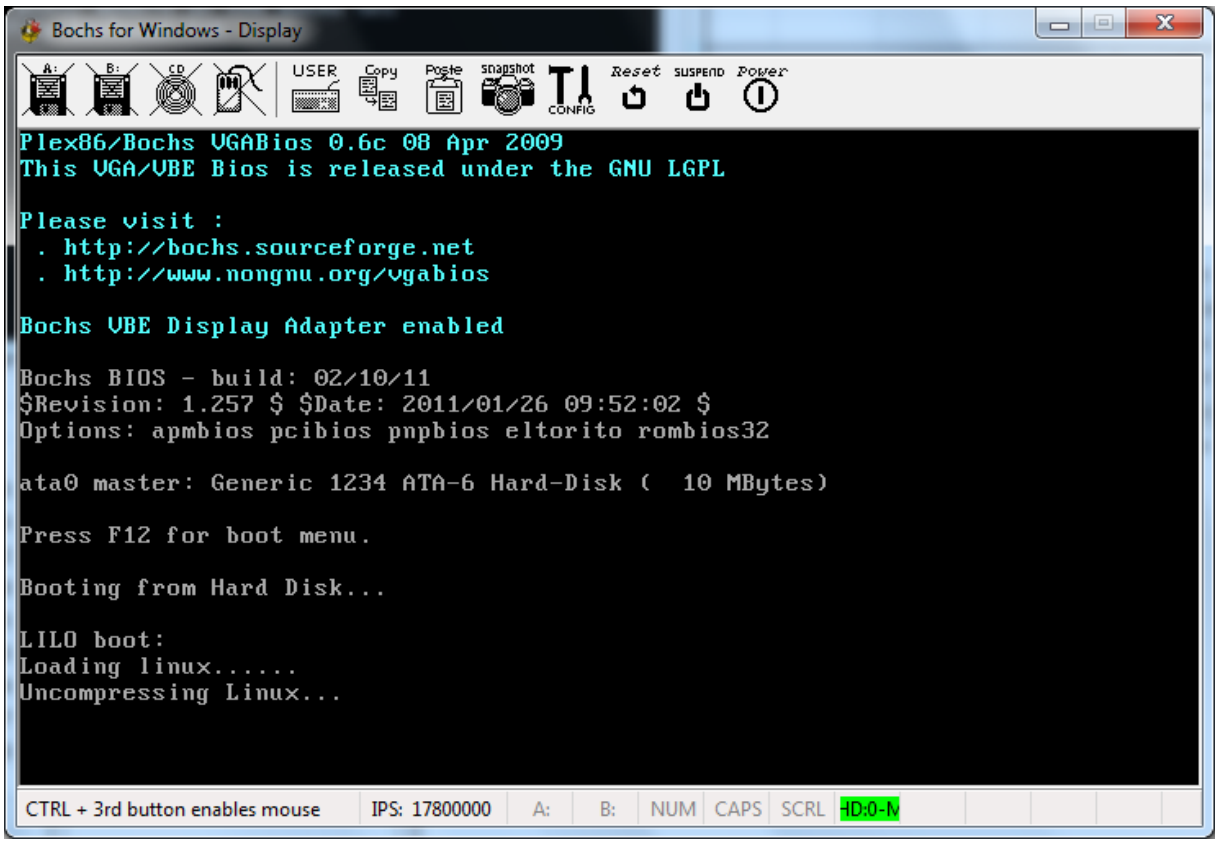
Naplózási beállítások: naplóállomány és különböző súlyosságú események kezelési módja.

```
log: bochsout.txt
panic: action=ask
error: action=report
info: action=report
debug: action=ignore
pass: action=fatal
```

[1]

2.1.2 A rendszer futtatása

A *run.bat* futtatásával elindul az emulátor:

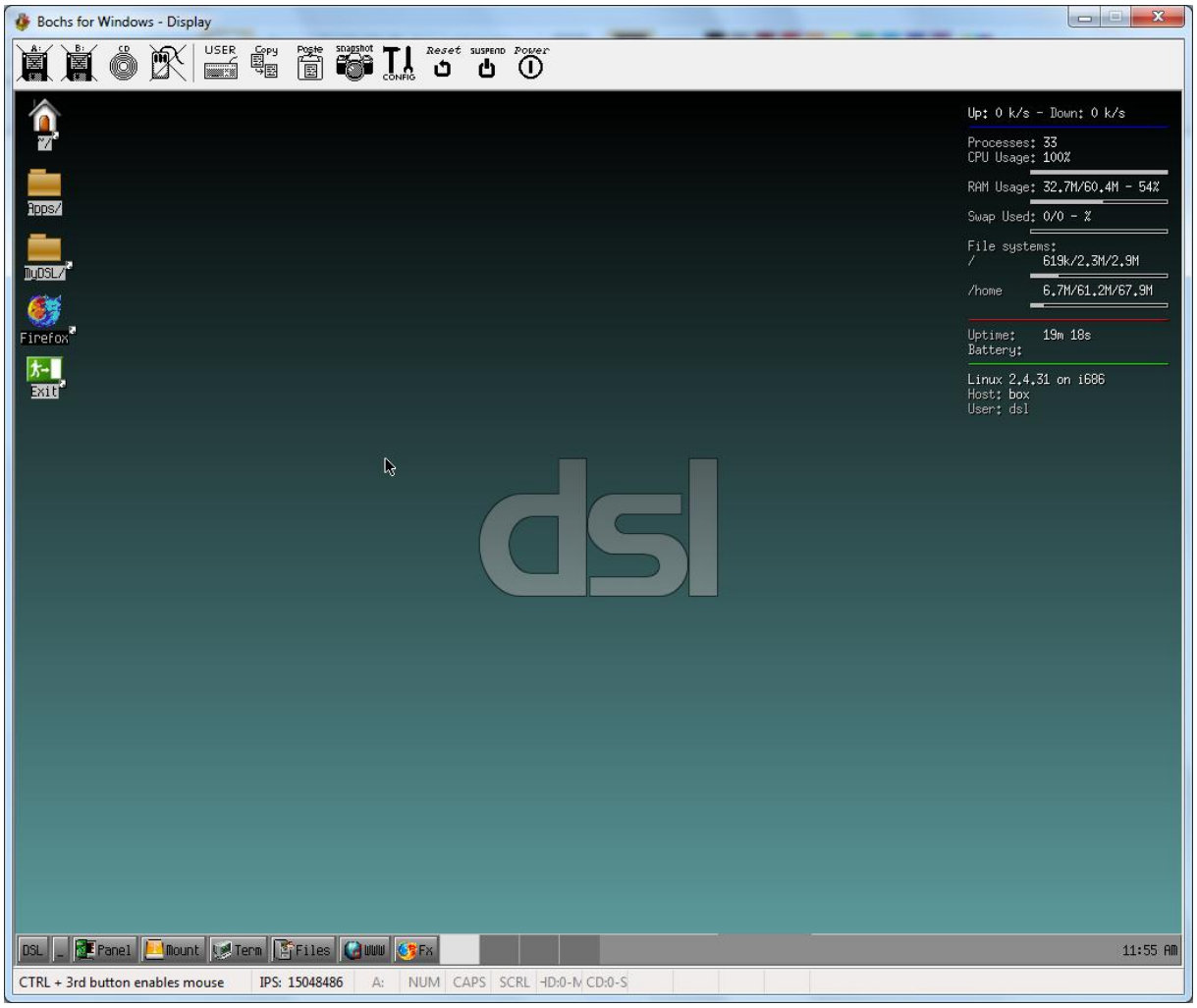


Megjegyzés: mivel a telepítő az emulálandó rendszer állományait a *Program Files* mappa alá telepíti, Windows 7-en az UAC beállításainak függvényében előfordulhat, hogy a szkriptet adminisztrátorként kell futtatni, különben az emulátor nem fér hozzá a merevlemez image-éhez.

2.2 Damn Small Linux futtatása CD-képfájlból

Egy, az előbbinél erősebb környezet kialakításával lehetőség nyílik olyan operációs rendszerek emulálására, melyek grafikus felülettel rendelkeznek. Mivel a Bochs képes CD-olvasó emulálására (ekkor a CD-t egy *.iso* állomány helyettesíti), egy kellően kevés erőforrást igénylő, CD-ről bootolható, GUI-val rendelkező operációs rendszer birtokában ez telepítés nélkül kipróbálható. A DSL (Damn Small Linux) megfelel az iménti kritériumoknak.

A környezet paramétereinek (pl. bootolási sorrend) beállítása és a Bochs futtatása után néhány perc múlva leindul az operációs rendszer



Megjegyzés: a rendszer 45 MIPS-es emulált CPU esetében erősen akadozik (használhatatlan).

3 A mérési környezet

Az alábbi pontok a mérési környezet ismertetését tartalmazzák.

3.1 Fizikai és virtuális gép

A fizikai és az azon belül, **VMware Player** segítségével futtatott virtuális gép fontosabb paraméterei a következők:

	Fizikai gép	Virtuális gép
Processzor	AMD Athlon II X2 240e (2 mag, 2.8 GHz magórajel, 2 MB L2 cache)	
Elérhető memória	3,5 GB	1 GB
Operációs rendszer	Windows 7 Ultimate SP1	Windows XP Professional SP3

3.2 Az emulált környezet

Az emulációhoz használt eszköz: Bochs 2.4.6.

Az emulált környezet egy Debian 3.0 operációs rendszer, mely előre telepített disk image formájában letölthető a Bochs weboldaláról a hozzá tartozó beállításokkal együtt [2].

Az emulálandó környezet fontosabb (kezdeti) beállításai:

```
memory: host=32, guest=32
cpu: count=1, ips=15000000, stb…
clock: sync=slowdown, time0=local
```

4 A mérés végrehajtása

4.1 A mérés lehetőségei

A méréssel azt kell megvizsgálni, hogy az adott fizikai, illetve virtualizált környezetben hogyan alakul az emulátor teljesítménye azonos beállítások mellett. A teljesítmény mérésére több lehetőség is adódik:

- az emulátor mennyi idő alatt hajt végre egy számításigényes műveletet
- az emulátor átlagosan hány utasítást hajt végre másodpercenként egy számításigényes feladat futtatása közben

Mindkét módszer alkalmazásakor adódnak nehézségek, ezt az alábbi két pont tartalmazza.

4.1.1 Teljesítmény mérése a végrehajtási idő segítségével

A legtöbb operációs rendszer időmérésének az alapja a CPU időzítője által bizonyos időközönként előidézett megszakítás (tick). Emulált processzor esetében az így mért idő alapesetben nem korrelál a futtatókörnyezet idejével, ugyanis az emulált CPU sebessége (Bochs esetében ennek mértékegysége IPS - instructions per second) a futtatókörnyezet CPU-igénye és a különböző utasítások emulációjának eltérő overheadje miatt ingadozik.

A Bochs az emulált környezet hoszthoz szinkronizálására négy üzemmódot biztosít [1]:

```
clock: sync=[none|slowdown|realtime|both]
```

- **none**: az időmérés alapja a beállított IPS paraméter
- **slowdown**: az emulátor igyekszik az emulált processzor átlagos sebességét a beállított IPS paraméter környékén tartani, az időmérés alapja itt is a beállított IPS érték.
- **realtime**: az időmérés alapja a hoszt ideje
- **both**: az emulátor igyekszik az emulált processzor átlagos sebességét a beállított IPS paraméter környékén tartani, az időmérés alapja a hoszt ideje

Ezek alapján a különböző beállítások mellett az alábbi jellemzőket lehet a rendszeren belül mérni:

- **none**: mennyi idő alatt hajtana végre egy adott feladatot egy olyan CPU, mely konstans a megadott IPS értéken üzemel
- **slowdown**: mint a *none* esetében, ugyanakkor (mivel ebben a módban az emulátor igyekszik a CPU átlagos sebességét a megadott IPS érték környékén tartani), egy bizonyos IPS értékig (amíg a hoszt képes maradéktalanul kiszolgálni az emulátor processzorigényét) a mért eredmény nagyjából egyezni fog a valós végrehajtási idővel.
- **realtime**: minden esetben nagyjából a valós végrehajtási időt méri
- **both**: minden esetben nagyjából a valós végrehajtási időt méri, ráadásul a CPU sebessége - amíg ez kivitelezhető - a beállított IPS érték körül ingadozik.

4.1.2 Teljesítmény mérése az egységnyi idő alatt átlagosan végrehajtott utasítások száma segítségével

Az előző pont alapján a teljesítmény pontosabban mérhető az aktuális IPS értékek mérésével. Erre adottak az alábbi lehetőségek:

- saját instrumentációs könyvtár létrehozása Bochs-hoz [3]
- IPS értékek naplózása

Megjegyzés: a naplózás kapcsán a felhasználói dokumentáció az alábbiakat írja:

"You can recompile Bochs with --enable-show-ips option enabled, to find your workstation's capability. Measured IPS value will then be logged into your log file or in the status bar (if supported by the gui)."

A GUI-n valóban megjelenik az aktuális IPS érték, azonban a naplóban még a Debug üzenetek között sem sikerült az aktuális IPS értékre vonatkozó bejegyzést találnom.

4.2 A mérés módja

A fentiek alapján az időmérés jóval egyszerűbb megoldás, ráadásul *"clock: sync=both"* módban a végrehajtási idő kellő pontossággal mérhető. Mivel ilyenkor az emulált CPU átlagos sebessége a rendszer sebességkorlátjának eléréséig a beállított IPS érték szerint alakul, értelmesen és reprodukálható módon vizsgálható a teljesítmény alakulása a beállított IPS érték függvényében. Ezáltal meghatározható a rendszer sebességkorlátja is.

Az így kapott érték összevethető az ugyanazon IPS érték mellett *"clock: sync=none"* vagy *"clock: sync=slowdown"* módban kapott értékkel, mely annak az ideális rendszernek a teljesítményét reprezentálja, melynek átbocsátóképessége konstans a megadott IPS.

4.2.1 Terhelés generálása

A mérés során az alábbi bash parancs generálja a terhelést:

```
echo "scale=1000; a(1)*4" | bc -l
```

A parancs a "*scale*" értékéül adott számtól függő pontossággal számolja "*a(1)*4*", vagyis π értékét (a az arctan függvényt jelenti) [4].

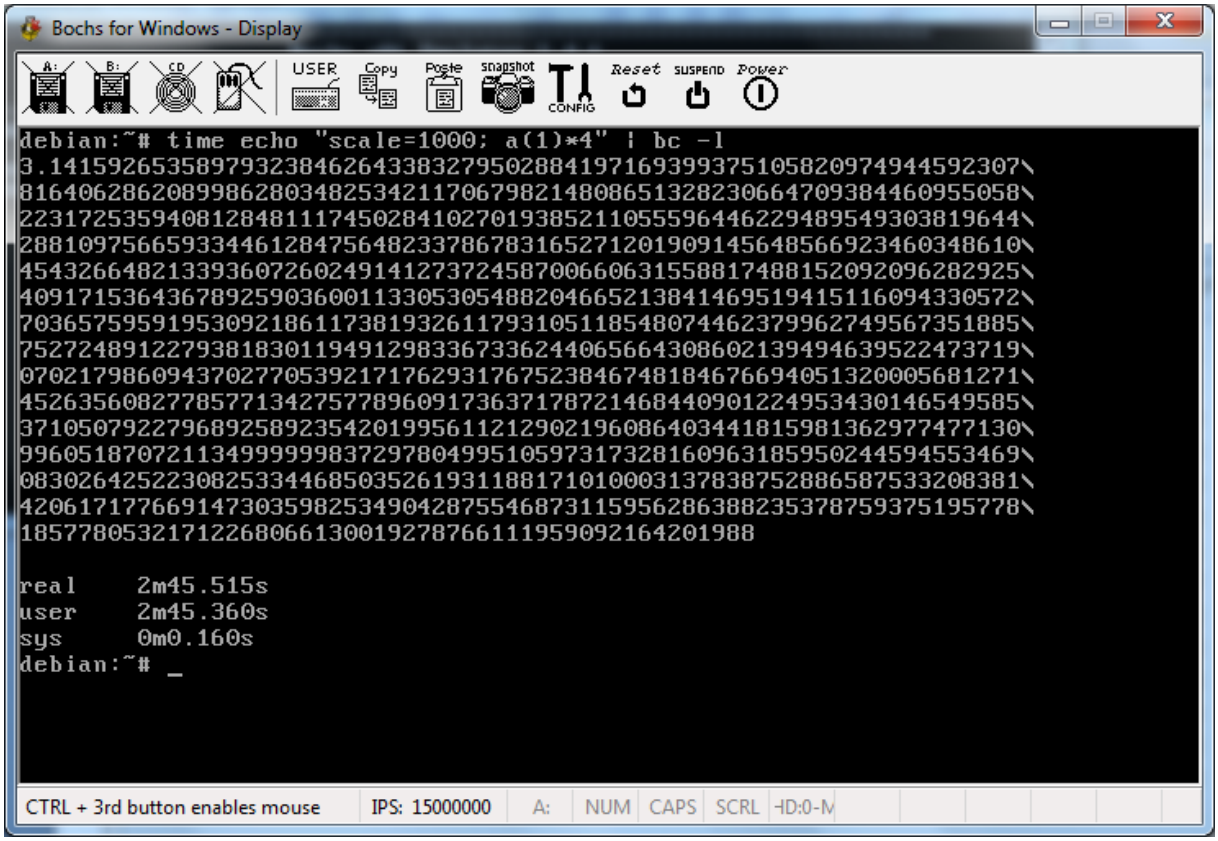
4.2.2 Mérendő és megfigyelendő jellemzők

Az alábbi jellemzők mérendőek, illetve figyelendőek meg a különböző beállított IPS értékek mellett:

- végrehajtás ideje: a *time* parancs, illetve stopperóra segítségével. Time esetében elsősorban a *real* érték, vagyis a valós végrehajtási idő érdekes.

```
time echo "scale=1000; a(1)*4" | bc -l
```

Megjegyzés: a feladat Ural2-n nagyjából 7,5 s alatt fut le. A futás eredménye 15 MIPS esetében:



- CPU-használat a Task Managerben (átlag becslése megfigyelés alapján): fizikai gép esetében a *bochs.exe* folyamaté, a virtuális gép esetében a guest *bochs.exe* folyamatáé, illetve a host *vmware-vmx.exe* folyamatáé

4.3 Mért adatok és kiértékelésük

Megjegyzés: *both*, illetve *realtime* szinkronizáció esetében az emulált CPU-n futó operációs rendszer a beállított IPS érték nagyságától függő valószínűséggel időnként az alábbi hibaüzenetet küldi:

```
Probable hardware bug: clock timer configuration lost - probably a VIA686a.
Probable hardware bug: restoring chip configuration.
```

Ugyanezen beállítások mellett fordul elő, hogy a *time* parancs *real* értéke alacsonyabb, mint a *user* érték (ld. táblázatok).

Mivel a hibák ellenére a *both* szinkronizáció bizonyult a legjobb módszernek a mérések elvégzésére, ráadásul a stopperes mérés eredménye nagyjából korrelál a *time* parancs által mért értékekkel, a fenti anomáliákat figyelmen kívül hagytam. A kiértékelés alapjául a *real* értéket választottam, ugyanis az esetek többségében ez az adat tér el legkevésbé a stopperes mérés eredményétől.

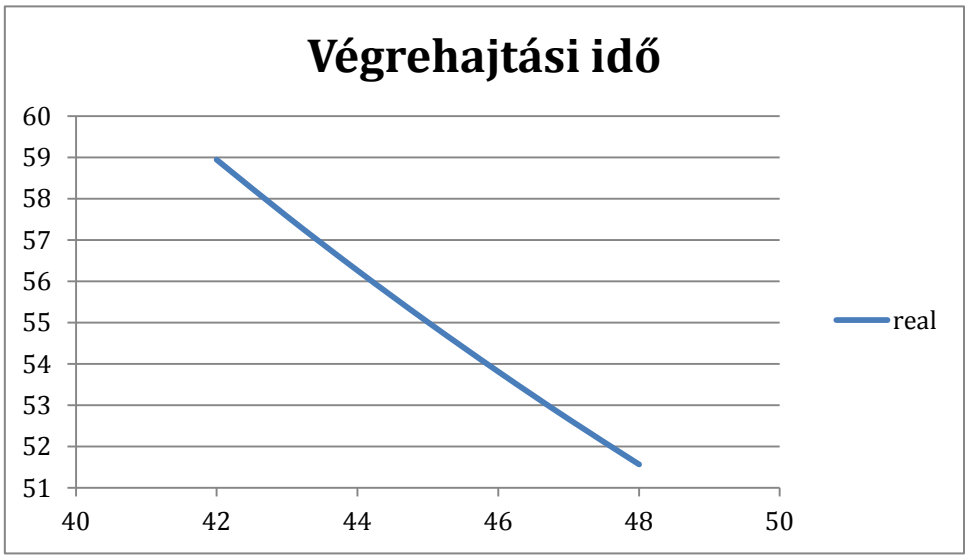
4.3.1 Ideális rendszer

Fizikai gép, *slowdown* szinkronizáció

MIPS	real (s)	user (s)	kernel (s)	stopper (s)
42	58,94	58,92	0,02	58,36

43	57,567	57,55	0,02	58,65
44	56,259	56,23	0,02	57,28
45	55,008	54,98	0,03	54,07
46	53,81	53,78	0,03	54,26
47	52,661	52,63	0,02	53,56
48	51,565	51,54	0,03	53,27

A valós végrehajtási idő a MIPS-beállítás függvényében:



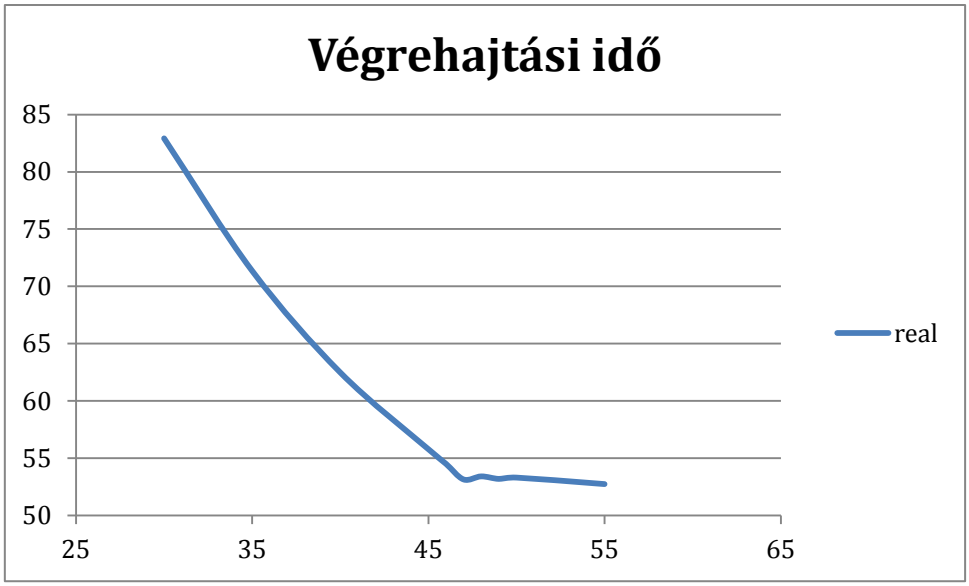
Látható, hogy a CPU sebességének növelésével a végrehajtási idő arányosan csökken.

4.3.2 Fizikai gépen futó rendszer

Fizikai gép, *both* szinkronizáció

MIPS	real (s)	user (s)	kernel (s)	stopper (s)	bochs.exe
30	82,91	82,87	0,03	83,29	26%
35	71,349	71,25	0,09	71,74	31%
40	62,481	62,44	0,05	63,02	38%
45	55,79	55,68	0,09	56,2	48%
46	54,514	54,56	0,07	55	49%
47	53,15	53,07	0,07	53,41	50%
48	53,416	53,31	0,08	53,68	50%
49	53,198	53,14	0,06	53,46	50%
50	53,313	53,94	0,06	53,6	50%
55	52,744	52,66	0,09	53,13	50%

A valós végrehajtási idő a MIPS-beállítás függvényében:



Jól látható az ábrán, hogy 46 MIPS környékén található az a pont, ahol a rendszer eléri a teljesítménykorlátját. Ez nagyjából ugyanaz az érték, ahol a *bochs.exe* folyamat eléri az 50%-os CPU-használatot (egy magot teljesen kihasznál). A töréspontig a rendszer az ideális rendszerhez hasonló módon skálázódik, ez igazolja, hogy az emuláció sebessége átlagban valóban nagyjából a beállított érték szerint alakul.

A kapott eredmény egybevág a Bochs felhasználói dokumentációjában szereplő szemléltető adatokkal:

Bochs	Speed	Machine/Compiler	Typical IPS
2.3.7	3.2Ghz	Intel Core 2 Q9770 with WinXP/g++ 3.4	50 to 55 MIPS
2.3.7	2.6Ghz	Intel Core 2 Duo with WinXP/g++ 3.4	38 to 43 MIPS
2.2.6	2.6Ghz	Intel Core 2 Duo with WinXP/g++ 3.4	21 to 25 MIPS
2.2.6	2.1Ghz	Athlon XP with Linux 2.6/g++ 3.4	12 to 15 MIPS
2.0.1	1.6Ghz	Intel P4 with Win2000/g++ 3.3	5 to 7 MIPS

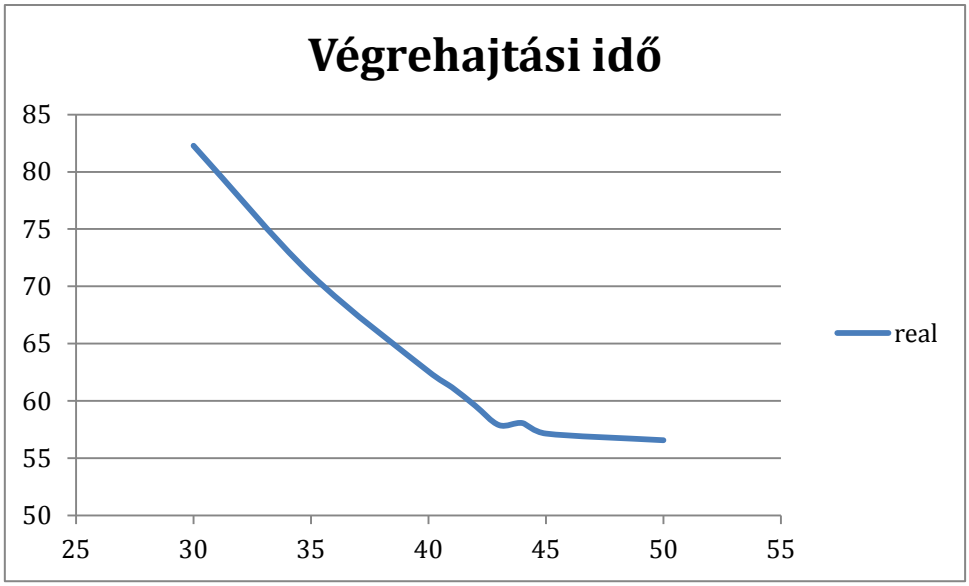
Bár az IPS nem ad tiszta képet egy processzorarchitektúra valós teljesítményéről, pusztán ezt a jellemzőt alapul véve a 46 MIPS-es emulációs teljesítmény nagyjából az 1990-es évben gyártani kezdett Motorola 68040-es processzort idézi [5].

4.3.3 Virtuális gépen futó rendszer

Virtuális gép, *both* szinkronizáció

MIPS	real (s)	user (s)	kernel(s)	stopper (s)	bochs.exe	vmware-vmx.exe
30	82,291	73,06	0,03	83,38	36%	37%
35	71,04	70,63	0,33	71,03	41%	42%
40	62,581	62,19	0,42	62,92	46%	47%
41	61,192	60,66	0,48	61,28	47%	48%
42	59,571	59,09	0,45	59,71	48%	49%
43	57,897	58,36	0,03	58,7	50%	51%
44	58,072	57,48	0,34	58,11	50%	51%

A valós végrehajtási idő a MIPS-beállítás függvényében:



A virtuális gépben futtatott emulátor teljesítménye a fizikai gépen emulált CPU-éhoz hasonlóan alakul, azzal a kivétellel, hogy itt a töréspont korábban, nagyjából 43 MIPS környékén található.

Ez nagyjából 6,5%-os teljesítményvesztést jelent.

5 Következtetések

A mérés a várható eredményt hozta. A virtualizált környezetben futtatott emulátor teljesítménye hasonlóan alakult a fizikai környezetben futtatott példányéhoz képest, ahogy bármely más alkalmazás esetében alakult volna. A virtualizáció CPU-overheadje és a host gép processzorhasználata miatt a virtuális gépen belül futó alkalmazás egy valamelyest gyengébb processzort lát, ez azonban egészen addig nincs kihatással az emulátor teljesítményére, amíg processzorigénye el nem éri a virtuális CPU teljesítménykorlátját.

Ezek alapján az, hogy érdemes-e virtuális gépben futtatni emulátort, attól függ, hogy igényli-e az emulált környezet a fizikai CPU nagyobb teljesítményét, vagy elégséges számára annak 90-95 százaléka.

Hivatkozások

[1] Bochs felhasználói dokumentáció:

http://bochs.sourceforge.net/cgi-bin/topper.pl?name=New+Bochs+Documentation&url=http://bochs.sourceforge.net/doc/docbook/user/index.html

[2] Bochs és disk image-ek letöltése:

http://sourceforge.net/projects/bochs/files/

[3] Bochs instrumentálása:

http://bochs.sourceforge.net/cgi-bin/lxr/source/instrument/instrumentation.txt

[4] GNU bc dokumentációja:

http://www.gnu.org/software/bc/manual/html_mono/bc.html

[5] IPS:

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

[6] Érdekesség: virtuális gépek időzítésének nehézségei:

http://www.vmware.com/files/pdf/Timekeeping-In-VirtualMachines.pdf