

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

POROVNÁNÍ VIRTUALIZAČNÍCH NÁSTROJŮ PRO
OPERAČNÍ SYSTÉM MS WINDOWS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN MAREK

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

POROVNÁNÍ VIRTUALIZAČNÍCH NÁSTROJŮ PRO OPERAČNÍM SYSTÉM MS WINDOWS

COMPARISON OF VARIOUS VIRTUALISATION TOOLS FOR MS WINDOWS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN MAREK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN SAMEK Ph.D.

BRNO 2012

Abstrakt

Tato práce se zabývá testováním virtualizačních nástrojů na platformě MS Windows. Jsou zde probrány techniky virtualizace a způsoby testování výkonu jednotlivých prvků počítače. V práci jsou popsány testy, které byly použity pro měření výkonu. V závěru práce jsou interpretovány výsledky z těchto testů a je ukázáno výsledné zhodnocení jednotlivých nástrojů.

Abstract

This thesis deals with comparison of virtualisation tools for MS Windows. It describes techniques of virtualization, forms of measuring computer performance and also benchmarks used to measuring performance. At the end of this thesis are interpreted results of benchmarks and final comparison tested tools.

Klíčová slova

virtualizace, VMware Player, VirtualBox, MS VirtualPC, testování

Keywords

virtualisation, VMware Player, VirtualBox, MS VirtualPC, benchmarking

Citace

Jan Marek: Porovnání virtualizačních nástrojů pro operační systém MS WINDOWS, bakalářská práce, Brno, FIT VUT v Brně, 2012

Porovnání virtualizačních nástrojů pro operační systém MS WINDOWS

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Samka, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jan Marek
7. května 2012

Poděkování

Rád bych poděkoval svému vedoucímu práce Ing. Janu Samkovi, Ph.D. za jeho ochotu, čas a cenné rady, které mi poskytl během psaní práce. Také bych mu chtěl poděkovat za poskytnutí počítače, který jsem ke své práci potřeboval.

© Jan Marek, 2012

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod.....	2
2	Techniky virtualizace.....	3
2.1	Plná virtualizace.....	3
2.2	Softwarová virtualizace	4
2.3	Hardwarová virtualizace	4
2.4	Paravirtualizace.....	6
3	Virtualizační nástroje	7
3.1	Virtual PC.....	7
3.2	VMware Player.....	8
3.3	VirtualBox	9
3.4	Srovnání.....	10
4	Testování výkonnosti PC	12
4.1	Základní pojmy a principy testování	12
4.1.1	Syntetické benchmarky.....	12
4.1.2	Mikrobenchmarky.....	13
4.1.3	Aplikační benchmarky	13
4.1.4	SPEC CPU	13
4.1.5	Whetstone a Dhrystone.....	14
4.2	Testování CPU.....	14
4.3	Testování pevného disku	15
4.4	Testování RAM paměti.....	15
4.5	Testování síťové karty	16
5	Metodika testování.....	17
5.1	Hostitelský počítač.....	17
5.2	Programy pro virtualizaci	17
5.2.1	VMware Player.....	17
5.2.2	VirtualBox	18
5.2.3	Virtual PC.....	18
5.3	Programy pro testování výkonu.....	18
5.3.1	Testování CPU.....	20
5.3.2	Testování pevného disku	20
5.3.3	Testování RAM paměti.....	21
5.3.4	Testování síťové karty	21
6	Výsledky	23
6.1	CPU – Windows	23
6.2	CPU – Linux	25
6.3	HDD – Windows	26
6.4	HDD – Linux	28
6.5	RAM – Windows.....	30
6.6	RAM – Linux.....	32
6.7	LAN – Windows.....	33
6.8	LAN – Linux	35
6.9	Výsledné zhodnocení.....	37
7	Závěr	39
	Příloha A.....	42
	Příloha B.....	49

1 Úvod

Pojem virtualizace pochází z 60. let 20. století [1]. Jde o vytvoření nového, virtuálního počítače pomocí kombinace softwaru a hardwaru. Takovýto počítač se skládá z virtuálních komponent – virtuální paměť, procesor, disk a další periferie. Virtualizace jednotlivých komponent je různě složitá.

Plná virtualizace simuluje hardware, který je nutný pro běh počítače. Tuto vlastnost jako první představila firma *IBM* na svých sálových počítačích vybavených operačním systémem OS/370 [1]. Bylo na něm možné rozdělit jeden fyzický počítač na několik virtuálních a pod každým virtuálním provozovat samostatný operační systém. Tuto virtualizaci bylo možné provádět pomocí hypervizoru, neboli virtuálního monitoru. Jedná se o programovou vrstvu, která přímo komunikuje s hardwarem počítače a zajišťuje virtualizaci všech jeho součástí. Při plné virtualizaci si operační systém není vědom, že je spouštěn ve virtuálním prostředí a není třeba jej nijak upravovat. Dokonce není nutné používat instrukční sadu fyzického procesoru, protože virtuální monitor dokáže emulovat konkrétní procesor. Toto je sice více univerzální, avšak méně výkonné, protože musíme každou instrukci emulovat na daný typ procesoru. Alternativou k plné virtualizaci je paravirtualizace. Nabízí efektivnější řešení virtualizace, ale vyžaduje modifikaci virtualizovaného operačního systému. O obou metodách se dozvíme více v kapitole 2.

V současné době nám virtualizace umožňuje testování nových operačních systémů, aplikací pro jiné operační systémy. Další oblastí využití jsou servery. Pokud potřebuje mít administrátor oddělené různé služby například DNS, webový server, nebo jiné servery, nemusí vlastnit tři fyzické servery. Stačí mu pouze jeden, ve kterém budou spuštěny další, již virtuální servery. Virtualizace umožňuje šetřit finanční i časové zdroje.

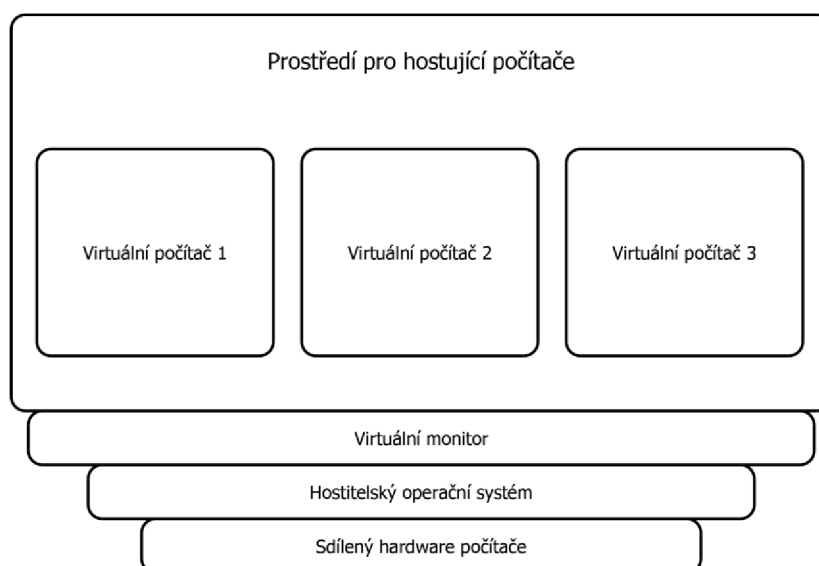
2 Techniky virtualizace

Než se budeme věnovat jednotlivým technikám virtualizace, zavedeme si následující pojmy:

Hostitelský systém: Označuje počítač a operační systém, ve kterém je spuštěn virtualizační software. Pojem *hostitelský operační systém* označuje pouze samotný operační systém hostujícího počítače [4].

Hostující systém: Označuje virtuální počítač a operační systém, který je spuštěn ve virtualizačním nástroji. Pojem *hostující operační systém* označuje operační systém hostujícího systému [4].

Také uvedeme základní schéma virtualizace. Jde o koncept jakým je realizován běh virtuálních počítačů. Základem je sdílený fyzický hardware, nad kterým je hostitelský operační systém, virtuální monitor a samotné virtuální stroje. Schéma je znázorněno na obrázku 2.1. Při popisu jednotlivých technik virtualizace jsem použil informace zejména z článku *Techniky virtualizace počítačů (2)* [2] a z manuálové stránky *Oracle VM VirtualBox* [5].



Obrázek 2.1: Schéma virtualizace.

2.1 Plná virtualizace

Pokud virtualizujeme všechny součásti počítače, hovoříme o plné virtualizaci. Hostující operační systém nemůže poznat, že běží ve virtuálním počítači. Také nepotřebuje žádné úpravy. Tímto způsobem zcela oddělíme fyzickou vrstvu. Samotné programy tak běží nad virtuálním hardware a přístup k fyzickému je vždy zprostředkovan.

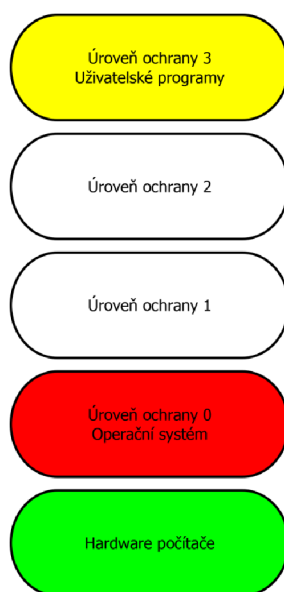
Tento přístup má řadu výhod. Můžeme navrhnout virtuální prostředí, tak aby nám vyhovovalo, výběrem typu procesoru, velikosti paměti, typem a kapacitou disku apod. Také samotné programy jsou nezávislé na konkrétním technickém vybavení. Pokud jej změníme, nebude to mít na virtuální prostředí žádný vliv. Nemusí zde existovat vazba mezi virtuálním prostředím a konkrétním hardwarem, na němž je hostující systém provozován. Toto umožňuje plnou přenositelnost.

Toto řešení, kdy dochází k oddělení programové a fyzické vrstvy, výrazně omezuje výkon virtuálního systému. Virtuální monitor musí odstínit virtuální počítač od jakékoliv možné změny hardware. Toho dosahuje emulováním většiny instrukcí procesoru virtuálního počítače a následným vlastním softwarovým provedením, namísto aby je přímo vykonával fyzický procesor.

Současné virtualizační nástroje používají dva způsoby plné virtualizace. Jedním je softwarová virtualizace a druhým je hardwarová virtualizace, která je možná pouze pokud jí podporuje procesor. Oba způsoby si popíšeme dále.

2.2 Softwarová virtualizace

Provádění virtualizace na architektuře x86 bez podpory hardwarové virtualizace je mimořádně složitý úkol, protože tato architektura nebyla navržena k virtualizaci. Tato architektura byla původně představena v 70. letech a od té doby prošla řadou změn [5]. Současné procesory jsou postavené na architektuře IA-32, které mají implementovány čtyři úrovně ochrany (značí se čísly 0-3). Úroveň 0 má nejvyšší stupeň ochrany a běží v ní operační systém. Naopak v úrovni 3 běží uživatelské programy. Ostatní úrovně se běžně nepoužívají [2]. Jednotlivé úrovně jsou přehledně zobrazeny na obrázku 2.2.



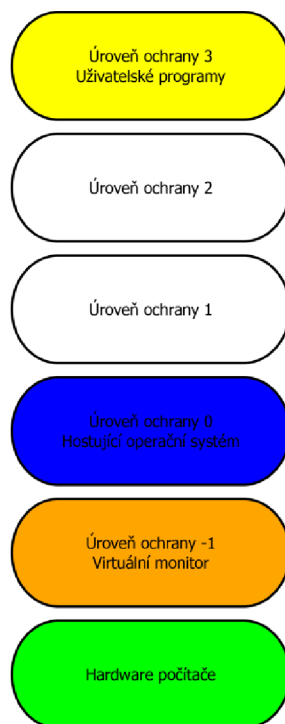
Obrázek 2.2: Úrovně ochrany procesoru.

Budeme rozlišovat dva kontexty – kontext hosta a kontext hostitele. V kontextu hostitele se systém chová, jakoby neobsahoval žádný virtuální monitor a systém funguje v úrovni 0 a 3. V kontextu hosta je aktivní virtuální monitor a hlídá prováděné instrukce. Pokud v hostovaném operačním systému běží aplikace v úrovni 3 není problém provádět jejich instrukce bez jakýchkoliv úprav. Problém nastává při některých instrukcích, které chce provádět samotný operační systém, který běží v úrovni 0. Tyto instrukce jsou kritické a mohou způsobit problémy s přístupem k hardware počítače. Tento problém řeší virtuální monitor pomocí emulace. Kód, který vykonává hostující operační systém je analyzován a převeden do podoby, kdy nemůže modifikovat skutečný stav procesoru a poté je vykonán. Jiný způsob řešení je pomocí paravirtualizace, o které se zmíníme v samostatné kapitole.

2.3 Hardwarová virtualizace

Díky hardwarové podpoře virtualizace je možné některé úkoly virtuálního monitoru převést do režie hardwaru. Vzniká nová úroveň ochrany -1. V této úrovni běží virtuální monitor a operační systém

běží v úrovni 0 a nemusí být modifikován. Umožní nám to jednodušší implementaci virtuálního monitoru a vyšší výkon celého systému. Toto řešení je předvedeno na obrázku 2.3.



Obrázek 2.3: Úrovně ochrany procesoru při hardwarové virtualizaci.

V roce 2006 přestavily firmy *Intel* a *AMD* tuto technologii u svých procesorů. *Intel* ji nazývá *VT-x (Intel Virtualization Technology)* a *AMD* *AMD-V (AMD Virtualization)*. Obě firmy tuto technologii implementují odlišně, ale princip činnosti je ve výsledku stejný [5].

Procesory s *VT-x* pracují ve dvou režimech – *VMX root mode* a *VMX non-root mode* [5]. V režimu *root mode* pracuje procesor jako starší modely bez podpory hardwarové virtualizace. Obsahuje čtyři úrovně ochrany a stejnou instrukční sadu s přidavkem několika speciálních virtualizačních instrukcí. Operační systém běží v tomto režimu, pokud není používána virtualizace. Pokud je virtualizace používána, běží v tomto režimu virtuální monitor. V režimu *non-root mode* pracuje procesor odlišně. Stále používá čtyři úrovně ochrany a stejnou instrukční sadu, ale s novou strukturou *VMCS (Virtual Machine Control Structure)*, která kontroluje operace procesoru a rozhoduje, jak se jednotlivé instrukce mají chovat [5]. V tomto režimu je řada instrukcí zakázána a pokus o jejich provedení způsobí přechod do režimu *VMX root mode*, kde je ošetří virtuální monitor.

Přepínání z *root mode* do *non-root mode* se nazývá *VM entry* a návrat zpátky *VM exit*. *VMCS* obsahuje hostitelskou a hostující oblast, která je uložena, nebo obnovena při přepínání mezi jednotlivými módy. *VMCS* také kontroluje, který host způsobil *VM exit*.

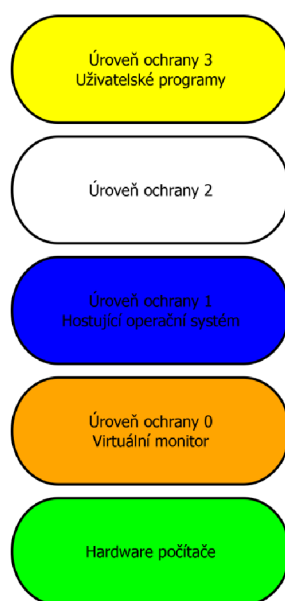
VT-x se vyhýbá několika problémům, které mohou nastat u softwarové virtualizace. Hosté mají svůj vlastní paměťový prostor, který je oddělený od virtuálního monitoru. To způsobuje menší počet kolizí. Navíc díky této technologii běží hostující operační systém v úrovni ochrany 0 a v *VMX non-root mode*, což odstraňuje problémy, kdy musí být běh operačního systému posunut do nižší úrovně ochrany, jak je tomu při paravirtualizaci.

2.4 Paravirtualizace

Paravirtualizace provádí jen částečnou abstrakci na úrovni virtuálního počítače. Nabízí virtuální prostředí podobné skutečnému hardwaru fyzického počítače (například virtuální počítač bude nabízet stejný procesor, avšak s odlišným, menším výkonem). Takováto virtualizace není úplná, některé vlastnosti např. procesoru mohou být omezeny a hostující operační systém může rozpoznat, že běží ve virtuálním prostředí. Díky podobnosti virtuálního a fyzického hardwaru může systém efektivněji fungovat, protože nemusí emulovat všechny komponenty virtuálního počítače.

Je nutné oddělit virtuální monitor a operační systém do jednotlivých úrovní ochrany. Virtuální monitor musí běžet v nejvyšší úrovni ochrany. Na stejné úrovni ochrany tedy nemůže běžet operační systém, protože by mohl ovlivnit stav virtuálního monitoru. Jednou z možností řešení by bylo pozměnit kód operačního systému, tak aby neprováděl žádnou instrukci, které potřebuje nejvyšší úroveň ochrany. Tyto instrukce by zachytil virtuální monitor, který by zkontroloval zda je operace povolena a následně jí provedl tak, aby změnila stav pouze virtuálního počítače. Při tomto řešení by nastal problém při instrukcích čtení z paměti. Jádro operačního systému předpokládá, že má přímý přístup k fyzické části paměti, to ovšem není u virtuálního operačního systému možné. Předem není možné poznat, které instrukce čtení paměti přistupují k privilegovaným údajům, proto musíme všechny tyto instrukce emulovat, což značně snižuje výkon.

Řešení, které se používá, spočívá v přesunutí virtuálního operačního systému na nižší úroveň ochrany, konkrétně na úroveň 1. Aplikační programy pracují stále na úrovni 3, tedy operační systém má stále vyšší úroveň ochrany. Toto řešení je ukázáno na obrázku 2.4. Ovšem tímto přesunem ztrácí možnost provádění některých operací, které vyžadují plně privilegovaný přístup. Pokud bude chtít provést takovouto operaci, dojde k přerušení a řízení převezme virtuální monitor. Ten operaci zkontroluje a provede tak, aby správně změnila stav virtuálního počítače. Instrukce, které vyžadují plně privilegovaný přístup poběží mnohem pomaleji, díky nutnosti jejich provádění virtuálním monitorem. Pro paravirtualizaci je nutné modifikovat některé součásti operačního systému. Tuto změnu lze provádět u operačních systémů s otevřeným zdrojovým kódem. Pro operační systémy Windows, u kterých není možná takováto úprava je možnost využít hardwarové virtualizace, která nabízí podobné výhody jako paravirtualizace, tedy možnost přímého vykonávání instrukcí virtuálního počítače fyzickým procesorem.



Obrázek 2.4: Úrovně ochrany procesoru při paravirtualizaci.

3 Virtualizační nástroje

Ve své práci se budu věnovat třem nástrojům pro virtualizaci počítače, které běží pod operačním systémem Windows. Jsou jimi *Virtual PC*, *VMware Player* a *VirtualBox*. Všechny tyto nástroje jsou poskytovány zdarma. V této kapitole si projdeme postupně všechny nástroje a ukážeme jejich možnosti. Na závěr kapitoly ukážeme krátké srovnání všech tří nástrojů a zmíníme se o dalších produktech, které jsou k virtualizaci dostupné.

3.1 Virtual PC

Virtual PC je produkt společnosti *Microsoft*. Je nabízen zdarma ke stažení v centru stahování. Ve Windows 7 je tento nástroj součástí všech edicí operačního systému. Pro starší operační systémy existuje verze *Microsoft Virtual PC 2007*. Tuto verzi lze spustit na operačních systémech:

- Windows Vista Business,
- Windows Vista Enterprise,
- Windows Vista Ultimate,
- Windows XP Professional,
- Windows XP Tablet PC Edition.

Windows XP mode je služba, která umožňuje používat starší verze obchodního softwaru se systémem Windows XP přímo v systému Windows 7. Tato služba je dostupná pro verze Professional, Ultimate a Enterprise operačního systému Windows 7. Zahnuje licenci na užívání systému Windows XP ve virtuálním prostředí programu [9].

Virtual PC lze provozovat na těchto verzích operačního systému:

- Windows 7 Home Basic
- Windows 7 Home Premium
- Windows 7 Enterprise
- Windows 7 Professional
- Windows 7 Ultimate

Oficiálně podporované operační systémy jsou Windows XP, Windows Vista a Windows 7. Ostatní systémy, například Linux, mohou také fungovat, ale nejsou oficiálně podporovány. Minimální systémové požadavky jsou:

- 1GHz 32-bitový nebo 64-bitový procesor
- Doporučeno 2 GB RAM paměti a více
- Doporučeno 15 GB volného místa na pevném disku pro každé virtuální prostředí Windows

Virtual PC podporuje hardwarovou virtualizaci. Pokud procesor nemá podporu hardwarové virtualizace je nutné nainstalovat aktualizaci, která umožní spouštění virtuálního počítače [6]. *Virtual PC* virtualizuje standardní PC a jeho přidružený hardware. Konkrétně emuluje následující hardware:

- Procesor Intel Pentium II (32bit) s čipovou sadou Intel 440 BX
- SVGA VESA standard grafických karet (S3 Trio 32 se 4 MB RAM apod.)
- AMI BIOS
- Sound Blaster 16 ISA PnP
- Síťovou kartu DEC 21041

Dalšími vlastnostmi jsou podpora a přesměrování USB, podpora více vláknového zpracování, přesměrování paměťových karet připojených k hostiteli, integrace s Windows Explorerem. Výhody

tohoto nástroje jsou zejména v jednoduchosti ovládání, integraci v systému Windows 7, specializované a odladěné prostředí pro běh operačních systémů Windows. Mezi nevýhody patří oficiální podpora pouze systémů Windows, nemožnost běhu 64 bitových operačních systémů. *Virtual PC* také podporuje virtualizaci pouze jednoho jádra procesoru, tedy pokud virtualizujeme na více jádrovém procesoru, virtuálnímu počítači bude přiděleno pouze jedno jádro.

3.2 VMware Player

VMware Player je produkt společnosti *VMware*. Jde o multiplatformní nástroj pro virtualizaci operačních systémů. Může běžet pod operačním systémem Windows i Linux. Ve své práci se budu zabývat variantou pro Windows. V současné době je nejnovější verze 4. Je dostupná zdarma ke stažení na stránkách *VMware*. Pro stažení je nutné se nejprve registrovat.

Ve starších verzích programu nebylo možné vytvářet virtuální počítače. Bylo možné spustit virtuální počítač, který byl vytvořený například některým placeným produktem *VMware*. Od verze 3.0 je virtuální počítače možné vytvářet, spouštět a konfigurovat. Minimální systémové požadavky pro spuštění virtuálních PC jsou:

- Procesor 64-bit x86
 - 1,3GHz nebo rychlejší
 - Podpora instrukcí LAHF/SAHF v *long mode*
- Operační paměť
 - Minimálně 1 GB RAM
 - Doporučeno 2 GB a více
- Pevný disk
 - IDE nebo SCSI
 - Nejméně 1 GB místa pro každý hostující operační systém

VMware Player také podporuje více jádrové procesory a procesory s hardwarovou virtualizací. Další komponenty mohou být mechanika CD/DVD, která může využívat fyzickou mechaniku připojenou přes rozhraní IDE nebo SCSI, nebo soubor typu ISO. Pokud chceme pracovat se sítí, je také potřebná síťová karta. Pro samotnou instalaci programu je vyžadováno 250 MB na pevném disku.

Podporované hostitelské operační systémy Windows:

- Windows XP
- Windows 2000
- Windows Vista
- Windows 7
- Windows Server 2003
- Windows Server 2008

Kompletní seznam podporovaných hostitelských systémů je možné nalézt na stránkách společnosti v sekci kompatibility [8].

Podporované hostující operační systémy:

- Windows
- Linux
- UNIX

Kompletní přehled podporovaných systémů je možné nalézt na stránkách společnosti [7]. Jsou podporovány jak 32-bitové systémy, tak 64-bitové. Pro spuštění 64-bitového systému je nutné mít procesor s podporou hardwarové virtualizace. Konkrétní hardware, který *VMware* virtualizuje, záleží na použité *Hardware version*. Nejnovější *VMware Player* používá *Hardware version 8* [10]. Aplikace

podporuje spuštění virtuálních počítačů, které byly vytvořeny v aplikacích *VMware Workstation 4* a novější, *GSX Server 3.x*, *ESX Server 2.5* a novější, *VMware Server* a také v aplikaci *Microsoft Virtual PC*.

Výhody nástroje jsou snadné ovládání, moderní způsoby virtualizace, možnost spuštění i 64 bitových operačních systémů.

3.3 VirtualBox

VirtualBox původně vyvíjela společnost *GmbH*, kterou poté zakoupila společnost *Sun Microsystems* a tu následně zakoupila společnost *Oracle*. Jde o nástroj pro virtualizaci operačních systémů vhodný pro domácí i firemní využití. Lze jej zdarma stáhnout ze stránek společnosti [11]. V době psaní této práce byla nejnovější verze programu 4.1.10. Nástroj lze spustit na operačních systémech Windows, Linux, Solaris, Mac OS X. Ve své práci se budu zabývat verzí určenou pro Windows.

Minimální systémové požadavky pro spuštění programu *VirtualBox*:

- Procesor s architekturou x86
- 512 MB RAM a více
- 30 MB pro instalaci programu na pevný disk

Systémové požadavky jsou závislé na konkrétním hostujícím a hostovaném operačním systému. Například pokud chceme virtualizovat Windows XP na hostitelském počítači, kde běží také Windows XP budeme potřebovat alespoň 1 GB RAM. Místo na pevném disku, které je potřeba pro virtuální počítače, závisí na velikosti pevného disku virtuálního počítače. V aplikaci máme u konkrétního virtuálního počítače možnost zapnout hardwarovou virtualizaci, pokud jí podporuje procesor na hostitelském počítači.

Podporované hostitelské operační systémy Windows:

- Windows XP (32-bit)
- Windows Server 2003 (32-bit)
- Windows Vista (32-bit a 64-bit)
- Windows Server 2008 (32-bit a 64-bit)
- Windows 7 (32-bit a 64-bit)

Seznam všech podporovaných hostitelských operačních systémů je možné nalézt v manuálu k programu [5].

Podporované hostující operační systémy:

- Windows
- Linux
- Solaris
- FreeBSD
- Mac OS X Server

Kompletní seznam všech hostujících operačních systémů a jejich jednotlivých verzí nalezneme v manuálu na stránkách programu [5]. *VirtualBox* podporuje 32 i 64 bitové hostující operační systémy. 64 bitové systémy můžeme spouštět i na 32 bitových hostitelských systémech. K tomu je však zapotřebí mít procesor podporující hardwarovou virtualizaci.

VirtualBox emuluje téměř všechny hardware hostitelského PC. Podle nastavení virtuálního počítače může být k dispozici tento hardware:

- **Vstupní zařízení.** V základním nastavení je emulována PS/2 klávesnice a myš.
- **Grafika.**
- **Paměť.** Jsou emulovány standardní ATA, SATA a SCSI disky.
- **Sítě.** *VirtualBox* umožňuje virtualizovat 6 různých typů síťových karet [5].

- **USB.** Jsou emulovány dva USB ovladače, EHCI a OHCI.
- **Audio**

Ve *VirtualBoxu* můžeme rozlišit dvě skupiny, jak program pracuje se soubory a procesy [3]:

- **Architektura klient-server** – všechny aspekty *VirtualBoxu* a běžících virtuálních počítačů mohou být spravovány pomocí *COM/XPCOM API*.
- **Frontend/backend architektura** – vnitřní implementace je schována ve sdílené knihovně *VBoxVMM*. Ta může být považována za backend. Jen neměnná a lze pro ni poměrně snadno napsat jiný *frontend*. *Frontend* je rozhraní, které určuje vzhled programu. *VirtualBox* nabízí takovýto rozhraní několik. Klasické, založené na Qt knihovnách. *VBoxManage*, nástroj pro příkazovou řádku. Čisté grafické rozhraní založené na *SDL*. *Remote Desktop Protocol server*, dostupný pro konzoli a dovoluje připojení vzdálených počítačů.

Výhody nástroje jsou snadné ovládání, podpora 64 bitových hostujících systémů na 32 bitových hostitelských operačních systémech, široký výběr hostujících i hostitelských operačních systémů

3.4 Srovnání

Všechny nástroje nám nabízí podobné možnosti. Technické řešení jednotlivých programů je odlišné, ale úlohu virtualizovat počítač musí zvládat všechny. Patrné rozdíly jsou v uživatelském prostředí. *VirtualBox* při spuštění nejprve zobrazí okno s virtuálními počítači, a poté co některý virtuální počítač spustíme, se nám otevře nové okno. Takže spuštěné virtuální počítače nejsou závislé na ovládacím okně *VirtualBoxu*. Oproti tomu *VMware Player* po spuštění zobrazí ovládací okno a ve stejném okně také spouští virtuální počítače, pro spuštění dalšího virtuálního počítače musíme otevřít nové okno. Dalším rozdílem je podpora hostitelských a hostovaných operačních systémů. Zatímco *VirtualBox* a *VMware Player* podporují širokou škálu operačních systémů, *Virtual PC* se specializuje na systémy Windows. V dalších kapitolách se budu věnovat porovnání výkonu jednotlivých nástrojů. Bude srovnáván výkon řadou testů na různých hostujících operačních systémech. Hostovaný operační systém bude Windows 7 v 64 bitové verzi.

Nyní si uvedeme stručný přehled jednotlivých nástrojů, který zobrazíme v přehledné tabulce 4.1 převzaté z knihy *Virtuální počítač* [3].

	Virtual PC	VMware Player	VirtualBox
Čeština	Ano	Ne	Ano
Virtualizace Windows	Ano	Ano	Ano
Virtualizace Linux	Negarantované	Ano	Ano
Použití .ISO souborů	Ne	Ano	Ano
Nastavení velikosti RAM	Ano	Ano	Ano
Volba počtu CPU	Ne	Ano	Ano
Možnost nastavení sítě	Ne	Ne	Ano
Vytváření snímků	Ne	Ne	Ano
Možnost volby typu disků	Ne	Ano	Ano

Tabulka 4.1: Srovnání virtualizačních nástrojů.

Všechny tři srovnávané nástroje jsou nabízeny zdarma. Pro operační systém Windows jsou dostupné další nástroje, které jsou však placené. Můžeme uvést například *VMware Workstation*, který nabízí některá vylepšení oproti verzi *Player*, jako propracovaný systém snímkování stavů virtuálních

počítačů nebo jejich klonování. Další program nabízí firma *Parallels*, jde o jejich produkt *Parallels Desktop 4 for Windows & Linux*. Oba tyto nástroje jsou k dispozici zdarma k vyzkoušení po určitou dobu, potom je nutné zakoupit licenci.

4 Testování výkonnosti PC

Porovnávat jednotlivé nástroje můžeme v několika ohledech. Při porovnání uživatelského rozhraní, práce s programem, jeho ovládání a nastavení jsou výsledky často subjektivní. Pokud chceme prakticky srovnávat výkon jednotlivých nástrojů, musíme stanovit konkrétní testy. Ty nám prokážou jasné výsledky na konkrétních číslech, která pak můžeme srovnávat. Pokud správně navrheme metodiku testování, výsledky budou již objektivní. Je důležité dodržet několik základních podmínek. Všechny testy musí být prováděny na stejné hardwarové sestavě a se stejným hostitelským operačním systémem. Rovněž testovací nástroje musí být ve stejné verzi. V jednotlivých virtualizačních nástrojích budu testovat jednotlivé komponenty virtuálního počítače pomocí uvedeného softwaru. Tento software bude vybrán s ohledem na konkrétní operační systém, který bude běžet v hostujícím operačním systému.

Nejprve si uveďme pojem benchmark. Definici převezmeme ze serveru Svět hardware [12]. *Jde o označení programu, který zjišťuje výkon, schopnost či kvalitu hardwaru. Zpravidla se jedná o test s výsledným skóre, díky čemuž se mohou vzájemně porovnávat různé počítačové sestavy.*

Syntetické testy zahrnují jednotlivé benchmarky, které se zaměřují na měření výkonu jednotlivých součástí počítače. Jde o programy, které počítají výkon při jednotlivých operacích. Například u procesoru to může být výkon FPU jednotky, nebo měření propustnosti paměti. Ke konkrétním nástrojům se dostaneme dále v této kapitole.

Praktické testy ukazují výkon v reálně používaných aplikacích. To může být široká škála programů. I zde se vybírají různé aplikace, pomocí kterých testujeme jednotlivé součásti počítače. Některé aplikace mají přímo vestavěné testování. Pro testování procesoru se používají různé nástroje pro kompresi dat, multimediální programy, nebo také programy pro šifrování disku.

V podkapitole 4.1 se budeme věnovat základním pojmům testování a v dalších podkapitolách si uvedeme konkrétní aplikace, které budeme používat při samotném testování.

4.1 Základní pojmy a principy testování

Nejprve si ukážeme základní dělení benchmarků. Půjde zejména o rozdělení na syntetické a aplikační benchmarky. Z těchto dvou tříd benchmarků budeme vybírat konkrétní aplikace, které budeme používat k samotnému testování výkonu. Tyto aplikace budou popsány v kapitole 4.2. Dále v této kapitole nalezneme popis obecných typů benchmarků jako *SPEC CPU*, třídy úloh *Whetstone* a *Dhrystone*. Informace, které jsou uvedeny v této kapitole vycházejí z knihy *Measuring computer performance: A practitioner's guide* [13].

4.1.1 Syntetické benchmarky

Tento typ benchmarků se skládá z programů, které mají za úkol měřit samotný výkon počítače, tedy neprovádí žádnou užitečnou práci. Provádí se v nich instrukce, které jsou vybrány, aby se shodovaly s instrukcemi, které se provádí v některých typech aplikačních programů. Myšlenka takto založených testů je, že pokud se budou provádět stejné instrukce jako v reálných aplikačních programech, výkon vyzorovaný z těchto syntetických programů bude stejný jako při provádění reálných aplikací. Tyto programy se ovšem nechovají jako reálné aplikace. Problém je, že tyto programy nemodelují přesně výkon, což je způsobeno vzájemným provázáním instrukcí, které jsou ve specifickém pořadí. Odkazy na paměťové vzory reálných aplikací je těžké duplikovat v syntetických programech. Tyto vzory určují paměťové oblasti, které ovlivňují výkon celého paměťového podsystemu. Díky hardwarovým a kompilačním optimalizacím je čas potřebný k vykonání syntetických a reálných aplikací velmi

odlišný. Složitost reálných aplikací nedovoluje takovéto aplikace používat jako benchmarkové aplikace. Výhodou syntetických benchmarků je jejich abstrakce od detailů reálných aplikací. Zkušenosti však ukázaly, že právě tyto detaily jsou důležitým faktorem měření výkonů počítačů.

4.1.2 Mikrobenchmarky

Mikrobenchmarky jsou malé programy, určené k testování výkonu konkrétních částí počítače. Například to může být program určený k testování vstupně/výstupního paměťového podsystemu u procesoru. Nebo konkrétní jednotky procesoru, jako například jednotky pracující s plovoucí desetinnou čárkou. Tyto programy jsou typicky využívány k určení maximálního výkonu takovéto části systému. Mohou být využívány ke zjištění, zda jsou všechny části uvnitř systému vyvážené. Programování takovýchto aplikací vyžaduje od programátora hlubší znalost jednotlivých komponent uvnitř systému.

4.1.3 Aplikační benchmarky

Jelikož syntetické benchmarky mají své omezení, byly vybrány různé typy aplikačních programů, aby splňovaly účely benchmarků. Toto jsou již reálné aplikace, které také produkují užitečné výsledky. Toto je hlavní rozdíl od syntetických benchmarků. Jsou používány programy často zdůrazňující konkrétní třídu aplikací. To jsou například aplikace z oblasti vědy, strojírenství, aplikace pro paralelní počítačové systémy a podobně.

Aplikační benchmarky mohou přesněji vystihnout využití systému než jakékoliv jiné typy benchmarkových aplikací. Aby vyhodnocení výkonu pomocí souboru aplikací tohoto typu netrvalo nadměrně dlouho, jsou těmto programům předávána uměle vytvořená, malá vstupní data. Toto může omezit chování paměťového modelu a také vstupně/výstupní požadavky uživatelských aplikačních programů. I přes tohle omezení jsou to však nejlepší nástroje pro měření výkonnosti [13].

Při výběru aplikačních benchmarků je třeba dát pozor na takzvané ‘toy’ benchmarky. Tyto programy sice počítají reálné úlohy, ale jsou příliš malé, aby byly považovány za samostatné aplikace, spouštěné uživateli systému. Tyto typy benchmarků zahrnují například *quicksort*, *problém N dam*, nebo *Hamoiské věže*. Tyto programy byly populární v 80. letech minulého století pro svou možnost být vykonávány na simulovaných procesorech. V současné době již díky rychlejším simulátorům nejsou tyto programy užitečné.

4.1.4 SPEC CPU

System Performance Evaluation Cooperative (SPEC) bylo založeno v roce 1988 skupinou firem (*Apollo/Hawlett-Packard, Digital Equipment Corporation, MIPS, a Sun Microsystems*) aby poskytly standardizovanou sadu aplikací a metodologii k měření výkonu. První sada benchmarků, *SPEC89*, se skládala ze čtyř programů napsaných v jazyce C a šesti v jazyce Fortran.

Výhodou *SPEC* benchmarků je, že často vydávají nové verze, které poskytují lepší výsledky vzhledem k neustále se měnícímu výkonu počítačů. Jelikož různé oblasti aplikací potřebují různé benchmarky, *SPEC* se rozšířila o *Open System Group (OSG)*, *High-Performance Group (HPG)* a *Graphics Performance Characterization Group (GPC)*.

SPEC OSG je přímým pokračovatelem původní *SPEC* komunity. Tato skupina se zaměřila na typy benchmarků pro pracovní stanice a servery, běžící zejména pod otevřeným operačním systémem, jako je UNIX. Zaměřili se na benchmarky určené pro procesor, paměťový systém a komponenty kompilátoru. Tuto třídu představoval původní soubor benchmarků *SPEC89*, který obsahoval deset programů s operacemi v pohyblivé řádové čarce *CFP95* a osm programů s operacemi

v pevné řádové čárce *CINT95*. Výsledkem těchto testů je hodnota *SPECratio*. Jde o referenční čas, získaný z těchto benchmarků podělený celkovým časem běhu všech benchmarků během testu.

4.1.5 Whetstone a Dhrystone

Whetstone a *Dhrystone* jsou dva typy syntetických benchmarků, které byly využívány k měření výkonnosti. V dnešní době se již nepoužívají, ale jsou zde zmíněny, jelikož jsou z historického hlediska zajímavé. Operace v benchmarkích *Whetstone* byly stanoveny na základě studie chování programů napsaných v jazyce Algol v 70. letech minulého století. Jejich hlavním úkolem bylo měřit výkon v operacích, kde se využívá pohyblivá řádová čárka. Naopak úkol benchmarků *Dhrystone* byl v měření výkonnosti pomocí operací, které používají pevnou řádovou čárku. K tomu byly využívány operace jako kopírování a porovnávání řetězců.

4.2 Testování CPU

Při testování výkonnosti procesoru budeme používat celou škálu benchmarků. Budou zde zařazeny jak syntetické, tak i aplikační benchmarky. Některé aplikace dokážou využívat více jader procesoru, ovšem *Virtual PC* nedokáže virtualizovat více jader, proto i v ostatních virtualizačních nástrojích omezíme počet jader na jedno, abychom zachovali stejné testovací podmínky pro všechny nástroje. Syntetické testy budou měřit výkon paměti procesoru a výkon ve výpočetních operacích. V praktických testech budeme měřit výkon v jednotlivých reálných aplikacích. Tyto aplikace zde budou dále představeny. Při porovnávání a testování procesorů jsou dále často zahrnovány kritéria jako spotřeba a teplota. Tyto kritéria však nebudeme zohledňovat, protože ve virtuálním prostředí nemají smysl. Při výběru některých aplikací používaných v testu jsem se inspiroval metodikou testování procesorů, kterou používá server *PC tuning* [14].

SuperPi

První syntetický benchmark zastupuje aplikace *SuperPi*. Jde o benchmark, který využívá pouze jedno jádro procesoru. Ukazuje reálný výkon FPU jednotky, práci s cache paměťmi a reflektuje také propustnost paměti RAM. Aplikace počítá Ludolfovo číslo na zadaný počet desetinných míst. Výsledkem je pak čas, který aplikace potřebovala k této operaci. Aplikace je dostupná pro operační systém Windows i Linux.

TrueCrypt

Pro zjištění výkonu procesoru při šifrování obsahu disků bude použita aplikace *TrueCrypt*. Zde mají výhodu procesory s AES instrukcemi. V našem případě zde bude výhodou, pokud daný virtualizační nástroj podporuje hardwarovou akceleraci těchto instrukcí. Aplikace dokáže využít všechna jádra procesoru a je dostupná pro operační systémy Windows i Linux.

LINPACK benchmark

Třetí aplikací bude *Linpack* benchmark. Jde o benchmark, který měří výkon počítače pomocí operací v plovoucí desetinné čárce. Jde o syntetický benchmark, který řeší různé lineární rovnice. V našem testu použijeme verzi benchmarku optimalizovaného firmou *Intel* pro jejich procesory.

4.3 Testování pevného disku

Při testování pevných disků budou využívány jak syntetické benchmarky, tak i praktické testy. Bude se testovat rychlost sekvenčního čtení a zápisu. V praktických testech budeme měřit dobu potřebnou ke startu operačního systému, nebo dobu potřebnou pro zkopírování určitého množství dat. U pevných disků se v testech objevují dále kritéria jako cena za 1 GB dat, teplota, hlučnost, spotřeba. Tyto kritéria nebudeme brát v úvahu, jelikož ve virtuálním prostředí nemají smysl. V testu je zařazen jeden syntetický benchmark, který měří rychlost sekvenčního čtení a zápisu. Další dva testy jsou praktické, abychom přesněji změřili výkon disku při běžných operacích používaných uživatelem.

Bonnie++

Tento benchmark se skládá ze sady testů, které testují výkon pevného disku a souborového systému. Benchmark se skládá ze dvou částí. V první části jsou prováděny operace čtení, vytváření a mazání souborů. Druhá část je zaměřena na samotnou práci s pevným diskem a ukazuje propustnost. Tento benchmark je dostupný pro operační systém Linux.

IOzone

Jde o syntetický benchmark, který měří výkon pomocí sady operací nad soubory. My využijeme testy s operacemi zápisu a čtení souboru. Abychom změřili reálnou rychlost práce s diskem zvolíme zápis i čtení souboru většího, než je dostupná RAM ve virtuálním systému. Výsledky programu jsou udávány v KB/s. Program je dostupný pro operační systém Windows i Linux.

Start operačního systému

Z praktických testů bude porovnávána doba potřebná ke spuštění testovaných operačních systémů Windows a Linux. Bude měřen start od spuštění operačního systému v konkrétním virtualizačním nástroji až po plné naběhnutí systému.

Kopírování souborů

Posledním praktickým testem bude doba potřebná ke zkopírování určitého množství dat, kdy bude použito kopírování přímo z operačního systému, tedy nebudou použity žádné programy pro správu souborů. Kopírování provedeme za pomoci příkazové řádky. Budeme kopírovat data mezi dvěma fyzickými disky.

4.4 Testování RAM paměti

Při testování RAM paměti je hlavním faktorem jejich propustnost. Tu budeme měřit pomocí různých vektorových operací. Půjde o syntetické testy, které zjistí maximální výkon z pohledu tohoto kritéria. Další částí budou aplikační testy. Při těchto testech budeme vybírat podobné programy jako při testování procesoru, jelikož tyto součásti jsou provázané a při aplikačních benchmarkcích záleží na obou z nich.

STREAM benchmark

Stream je syntetický benchmark, který měří propustnost paměti. Výkon je měřen pomocí čtyř vektorových operací. Tyto operace pracují s bloky dat, které jsou větší než je velikost cache paměti. Výsledky jsou udávány v MB/s. Aplikace je dostupná pro operační systémy Windows i Linux.

SuperPi

Další program pro měření výkonnosti bude již zmíněný *SuperPi*. Ve výsledcích tohoto programu se také projeví vlastnosti RAM paměti, takže výsledky také můžeme započítat do celkového hodnocení výkonnosti paměti.

WinRar

Z aplikačních benchmarků využijeme program *WinRar*. I v tomto programu hraje výkonnost paměti RAM důležitou roli na jeho výsledky. Tento program zpracovává velké množství dat, které musí projít přes paměť RAM, tedy je důležitá její rychlá odezva. Program umí pracovat i s více jádry procesoru. Aplikace je dostupná pro operační systémy Windows i Linux.

4.5 Testování síťové karty

Testování síťové karty budeme provádět na základě přenosu dat přes síťovou kartu. Data budeme přenášet mezi dvěma fyzickými počítači. Na jednom počítači bude nainstalován samotný operační systém a na druhém budeme data vysílat i přijímat ve virtualizovaném počítači. Pro přenos použijeme protokol SMB. Další testy budou provedeny pomocí syntetických benchmarků, ty vytvoří na jednom fyzickém počítači server a na druhém se jako klient na tento server připojí a testují přenosovou rychlost.

LANBench

K testování použijeme aplikaci *LANBench*. Jde o aplikaci nabízenou zdarma. Lze nastavit cílovou adresu a port, na kterém bude probíhat komunikace. Dále se v aplikaci nastaví doba, po kterou budou přenášena data. Jde o pakety určité velikosti, kterou si můžeme nastavit. Mód přenosu nám určí, zda se budou pakety vysílat, přijímat nebo obojí. Můžeme také nastavit počet spojení, která budou souběžně otevřena. Program je určen pro operační systém Windows.

nttcp

Program *nttcp* testuje propustnost sítě pomocí zasílání TCP nebo UDP paketů. Program je nutné spustit na dvou počítačích připojených do stejné sítě. Programy mezi sebou zasílají pakety a měří propustnost. Velikost těchto paketů, celkovou zasílanou velikost je možné měnit pomocí parametrů programu. Program pracuje pod operačním systémem Linux.

Iperf

Jde o syntetický benchmark, který měří propustnost počítačové sítě. Program se spustí na dvou fyzických počítačích, jeden bude představovat server a druhý klienta. Poté si mezi sebou vyměňují pakety, které mohou být podle parametru TCP nebo UDP. Výsledná propustnost je udávána v Mb/s.

Přenos dat

Praktickým testem bude přenos určitého množství dat pomocí sdílení v síti. Pro tento účel zvolíme souborového manažera a budeme měřit čas potřebný k přenosu předem daného množství souborů. Tyto data budeme přenášet mezi dvěma fyzickými počítači propojenými v síti.

5 Metodika testování

Je důležité stanovit vhodnou metodiku testování, abychom zajistili objektivní výsledky. V této kapitole si popíšeme, za jakých podmínek budeme jednotlivé nástroje testovat. Uvedeme si přesně, které verze programů budeme používat, které budou hostované i hostující operační systémy a také přesné nastavení jednotlivých programů k testování. Je také důležité stanovit interpretaci výsledků. Zde půjde zejména o to, jakým způsobem budeme pracovat s naměřenými výsledky. V následujících podkapitolách si popíšeme tři základní body, těmi jsou hostitelský počítač spolu s hostitelským a hostujícími operačními systémy, samotné programy pro virtualizaci a jednotlivé programy pro testování.

5.1 Hostitelský počítač

Všechny nástroje budou testovány na jednom hostitelském počítači. Půjde o počítač s těmito parametry:

- Procesor: Intel Core i5-2500 3,4GHz (procesor podporuje hardwarovou virtualizaci)
- RAM paměť: 2x 4 GB DDR3-1333
- Pevný disk 1: ST3500410AS ATA Device (500 GB, 7200 RPM, SATA-II)
- Pevný disk 2: ST3500418AS ATA Device (500 GB, 7200 RPM, SATA-II)
- Síťová karta: Realtek PCIe GBE Family Controller

Všechny testované nástroje budeme testovat na hostitelském počítači pod operačním systémem Windows. Konkrétně půjde o Microsoft Windows 7 Professional se Service Packem 1. Tento operační systém je nabízen k nekomerčnímu použití zdarma pro studenty FIT v rámci licence MSDN AA [15]. Půjde o čistou instalaci s doplněním všech dostupných aktualizací ke dni, kdy bude test probíhat. Dále zde budou nainstalovány všechny tři nástroje pro virtualizaci, které budou popsány v kapitole 5.2.

Pro účely testování budeme ve virtuálních počítačích provozovat dva operační systémy. První bude Windows a druhý Linux. Konkrétně pak Microsoft Windows XP Professional se Service Packem 3, který je taktéž dostupný v rámci licence MSDN AA. V případě Linuxu se bude jednat o Ubuntu 11.10 „Oneiric Ocelot“, který je dostupný zdarma ke stažení na stránkách Ubuntu [16]. Na obou hostujících systémech budou nainstalovány všechny nástroje pro vlastní testování.

Hostitelský operační systém bude v 64-bitové verzi a všechny hostující operační systémy v 32-bitové verzi.

5.2 Programy pro virtualizaci

Všechny tři programy pro virtualizaci jsme si již podrobně popsaly. V této sekci bude uvedeno, které verze těchto programů budeme používat a jejich nastavení. Toto nastavení bude stejné pro oba testované hostující operační systémy.

5.2.1 VMware Player

VMware Player budeme používat ve verzi 4.0.2, která byla uvedena 24. ledna 2012. Bude nainstalována 64-bitová verze programu. Program bude nainstalován se základním nastavením instalátoru. Do každého hostovaného počítače budou doinstalovány *VMware Tools*.

Nový virtuální počítač bude vytvořen s těmito parametry:

- Velikost paměti RAM: 3340 MB
- Typ souboru pevného disku: VMDK (Virtual Machine Disk)
- Velikost pevného disku 1: 40 GB
- Velikost pevného disku 2: 10 GB
- Počet procesorů: 1
- Síťová karta: v režimu NAT

5.2.2 VirtualBox

VirtualBox budeme používat ve verzi 4.1.10, která byla uvedena 13. března 2012. Program bude nainstalován se základním nastavením instalátoru. Nový virtuální počítač bude vytvořen s těmito parametry:

- Velikost paměti RAM: 3340 MB
- Typ souboru pevného disku: VDI (VirtualBox Disk Image)
- Velikost pevného disku 1: 40 GB
- Velikost pevného disku 2: 10 GB
- Počet procesorů: 1
- Síťová karta: v režimu NAT

5.2.3 Virtual PC

Virtual PC bude nainstalován ve verzi, která je součástí Windows 7. Nový virtuální počítač bude vytvořen s těmito parametry:

- Velikost paměti RAM: 3340 MB
- Typ souboru pevného disku: VHD (Virtual Hard Disk)
- Velikost pevného disku 1: 40 GB
- Velikost pevného disku 2: 10 GB
- Počet procesorů: 1
- Síťová karta: v režimu NAT

5.3 Programy pro testování výkonu

V této podkapitole si postupně popíšeme jednotlivé programy, které budeme používat v testech jednotlivých komponent. U každého programu si popíšeme jeho nastavení a za jakých podmínek bude spuštěn. Také bude uvedeno v jakých jednotkách budou výstupní hodnoty.

Než si popíšeme jednotlivé konkrétní programy, uvedeme jakým způsobem budeme pracovat s výsledky. Testy budou rozděleny do čtyř kategorií – procesor, RAM paměť, pevný disk a síťová karta. V každé kategorii budou zvoleny 3 testy. Každý test bude ve své kategorii spuštěn třikrát a výsledkem bude průměr naměřených hodnot. Nejlepšímu výsledku testu z dané kategorie přiřadíme hodnocení 100%. Ostatní výsledky budou ohodnoceny vzhledem k tomuto výsledku, tedy v příslušném poměru přepočítány na procenta. Ve výsledku se tyto dílčí výsledky z každého testu sečtou a vydělí počtem provedených testů, tak získáme výsledné hodnocení v dané kategorii. Tímto postupem zjistíme, který virtualizační program nabízí nejlepší výkon v dané kategorii. Nakonec ukážeme celkový výsledek, který bude vypočítán z konečných výsledků testů v daných kategoriích. Nyní uvedeme příklad, jak budeme výpočet provádět. Vezmeme si tabulky výsledků pro CPU na systému Windows, konkrétně test *LINPACK* a test *SuperPi*.

MS virtual PC	
pokus	výkon (Gflops)
1.	8,5660
2.	8,2906
3.	8,5737
průměr	8,4768

Tabulka 5.1: LINPACK.

Virtual Box	
pokus	výkon (Gflops)
1.	11,4701
2.	11,5076
3.	11,8464
průměr	11,6080

Tabulka 5.2: LINPACK.

Vmware Player	
pokus	výkon (Gflops)
1.	12,7473
2.	13,0127
3.	12,3574
průměr	12,7058

Tabulka 5.3: LINPACK.

Z tabulek můžeme vyčíst, že nejlepší průměrný výsledek dosáhl VMware Player (čím více Gflops, tím lépe), proto tento výsledek prohlásíme za 100%. Výsledky zbylých dvou nástrojů vypočítáme podle vztahu 5.1:

$$\text{aktuálně_počítaný_výsledek} = \frac{\text{hodnota_aktuálně_počítaného}}{\text{hodnota_nejlepšího_výsledku}} * 100 \quad (5.1)$$

Na příkladu jsme si ukázali přepočítání výsledků na procenta, kdy byla použita přímá úměra, tedy čím vyšší číslo ve výsledku, tím více procent. Nyní si uvedeme druhý příklad pro nepřímou úměru, kdy nižší číslo ve výsledku bude znamenat více procent (tento příklad platí u testů, ve kterých jsou výsledky uváděny v jednotkách času).

MS virtual PC	
pokus	čas (s)
1.	11,467
2.	11,386
3.	11,216
průměr	11,356

Tabulka 5.4 SuperPi.

Virtual Box	
pokus	čas (s)
1.	11,922
2.	11,563
3.	11,453
průměr	11,646

Tabulka 5.5: SuperPi.

Vmware Player	
pokus	čas (s)
1.	11,111
2.	11,080
3.	11,079
průměr	11,090

Tabulka 5.6: SuperPi.

Z tabulek je patrné, že nejlepší průměrný výsledek dosáhl VMwarePlayer (čím kratší čas, tím lépe). Tento výsledek tedy prohlásíme za 100% a výsledky zbylých dvou nástrojů vypočítáme podle vztahu 5.2:

$$\text{aktuálně_počítaný_výsledek} = \frac{\text{hodnota_nejlepšího_výsledku}}{\text{hodnota_aktuálně_počítaného}} * 100 \quad (5.2)$$

Na tomto příkladu lze vidět přepočítání výsledků na procenta za použití nepřímé úměry, tedy kratší čas znamená lepší výsledek.

Na posledním příkladu si uvedeme výpočet celkového výsledku v dané kategorii.

MS virtual PC	
pokus	výsledek
SuperPi	97,7%
TrueCrypt - Aes	19,9%
TrueCrypt – Serpent-Twofish AES	82,4%
LINPACK	66,7%
průměr	66,7%

Tabulka: 5.7: Celkový výsledek.

Virtual Box	
pokus	výsledek
SuperPi	95,2%
TrueCrypt - AES	19,4%
TrueCrypt – Serpent-Twofish AES	80,0%
LINPACK	91,4%
průměr	71,5%

Tabulka 5.5: Celkový výsledek.

Vmware Player	
pokus	výsledek
SuperPi	100,0%
TrueCrypt - AES	100,0%
TrueCrypt – Serpent-Twofish AES	100,0%
LINPACK	100,0%
průměr	100,0%

Tabulka 5.6: Celkový výsledek.

Položka průměr, která je zobrazena v grafech u kapitoly 6 jako celkový výsledek je vypočítána jako aritmetický průměr všech dílčích výsledků naměřených v jednotlivých testech, jak je vidět z výše uvedených tabulek.

5.3.1 Testování CPU

Prvním testem pro CPU bude aplikace *SuperPi*. Tato aplikace budou použita na systému Windows ve verzi, která měří dobu s přesností na tisíce sekund, *Super Pi / mod1.5 XS*. Pro operační systém Linux bude použita verze 2.0. Jediným parametrem aplikace je počet desetinných míst, na které počítá číslo Pi. Tento parametr bude nastaven na 1 milion. Výsledek je uveden v sekundách.

Druhým testem bude aplikace *TrueCrypt*, u které využijeme benchmark, který obsahuje. Aplikaci použijeme ve verzi 7.1a u obou operačních systémů. V tomto benchmarku nastavíme velikost bufferu na hodnotu 100 MB. Benchmark poté spustí sadu algoritmů. Do výsledného hodnocení budeme počítat výsledky prvního a posledního algoritmu. Tedy výsledky algoritmu *AES* a *AES-Twofish-Serpent*. Tyto výsledky jsou udávány v MB/s.

Poslední testem bude aplikace *LINPACK* benchmark. Aplikaci použijeme pro oba operační systémy ve verzi 10.3.9. Spouštět ji budeme pomocí skriptu *runme_xeon32*, jelikož budeme používat 32-bitovou verzi. V souboru *lininput_xeon32* nastavíme vstupní parametry programu takto: počet rovnic 8000, velikost pole 8008, počet opakování 2 a přiřazenou hodnotu 4. Výsledkem bude hodnota udávající výkon v jednotkách Gflops.

5.3.2 Testování pevného disku

Prvním testem bude benchmark *IOzone*. Pro operační systém Windows použijeme verzi 3.381 a pro Linux verzi 3.373. Aplikaci spustíme v režimu auto mode a spustíme testy pro čtení a zápis. Jako další parametr nastavíme velikost souboru, který se bude zapisovat a poté číst. Ten nastavíme větší než je použitá RAM paměť v systému, nastavíme jej tedy na 5 GB. Posledním parametrem bude velikost záznamu, tu nastavíme na 64 KB. Výsledná rychlost čtení a zápisu bude udávána v KB/s.

Druhý test bude proveden pomocí programu *Bonnie++*. Tímto programem budeme testovat operační systém Linux. Program bude použit ve verzi 1.03e. Výsledky budeme používat z bloků testů pro sekvenční vstup a výstup. Tyto testy pracují s daty po bytech a po blocích dat. Výsledkem je hodnota pro rychlost zápisu a čtení udávána v jednotkách KB/s.

Třetím testem, společným pro oba operační systémy bude doba spouštění operačního systému. V obou operačních systémech bude nastaveno spuštění příkazové řádky po startu systému. Výsledný čas

budeme měřit stopkami od spuštění operačního systému v daném virtualizačním nástroji až po spuštění příkazové řádky v daném hostujícím systému.

Posledním testem, který bude opět společný pro oba operační systémy, bude měření doby potřebné pro zkopírování složky s obsahem 404 fotografií ve formátu JPEG o celkové velikosti 992 MB. Kopírování bude provedeno přes standardní rozhraní operačního systému a bude se kopírovat mezi dvěma fyzickými disky. Jeden virtuální disk, na kterém je nainstalován hostující operační systém bude umístěn na fyzickém disku 1 a druhý virtuální disk, na který se budou daná data kopírovat bude umístěn na fyzickém disku 2. Doba potřebná pro kopírování bude měřena stopkami.

5.3.3 Testování RAM paměti

Prvním testem pro RAM paměti bude aplikace *STREAM benchmark*. Tato aplikace budou použita na systému Windows i Linux ve verzi 5.8. Výsledky jsou udávány ve čtyřech veličinách *Copy*, *Scale*, *Add*, *Triad*. Jednotkou jsou MB/s.

Dalším testem bude aplikace *SuperPi*. Výsledky uvedené ve sekundách budou stejné jako při testování procesoru.

Posledním testem bude aplikace *WinRar*. Aplikaci použijeme pro oba operační systémy ve verzi 4.11 32-bit. S aplikací budeme pracovat pomocí příkazového řádku. Pomocí této aplikace budeme komprimovat zdrojové kódy webového prohlížeče *Firefox*. Konkrétně půjde o verzi 12.0b3. Velikost všech souborů je 378 MB a jejich celkový počet 55 928. V tomto testu budeme sledovat čas potřebný pro zkomprimování všech dat, ten budeme měřit stopkami. Aplikaci budeme spouštět s parametrem určujícím míru komprese dat, ten nastavíme na nejmenší kompresi. Výsledek bude uveden v sekundách.

5.3.4 Testování síťové karty

Pro testování síťové karty si nejprve vytvoříme síť mezi dvěma fyzickými počítači. Oba počítače budou používat gigabitovou síťovou kartu. Vzdálený počítač bude mít adresu 192.168.16.1 a hostitelský počítač s virtuálními počítači bude mít adresu 192.168.16.2. Virtuální počítače budou mít síťovou kartu nastavenou v režimu NAT a adresu přidělenou z DHCP serveru, který zajišťuje daný virtualizační nástroj.

První test provedeme pomocí programu *LANBench*. Použijeme verzi 1.1.0 určenou pro Windows. Na obou PC spustíme program s tímto nastavením:

```
Server address: 192.168.16.1
Server port: 8998
Test duration: 60 secs
Packet size: 8196 KB
Connections: 1
Transfer Mode: Send And Receive
```

Na vzdáleném počítači spustíme server a na hostujícím počítači spustíme samotný test. Výsledkem jsou dvě hodnoty, jedna pro vysílaná data a druhá pro přijímaná. Jednotkou jsou Kb/s.

Druhý test provedeme pomocí programu *nttcp*. Jde o program pro Linux. Na vzdáleném počítači spustíme program v režimu serveru s parametrem `-i`, tedy:

```
nttcp -i
```

Na hostujícím počítači spustíme program s těmito parametry:

```
nttcp -t/-r -T 192.168.16.1
```

Výsledek je uveden v Mb/s.

Třetí testem bude aplikace *Iperf*. Program použijeme ve verzi 2.0.5 pro operační systém Linux a ve verzi 1.7.0 pro operační systém Windows. Tento test bude proveden na obou operačních systémech. Na vzdáleném počítači spustíme program v režimu server, tedy s parametrem `-s`. Na hostujícím počítači pak s parametrem `-c` a IP adresou serveru.

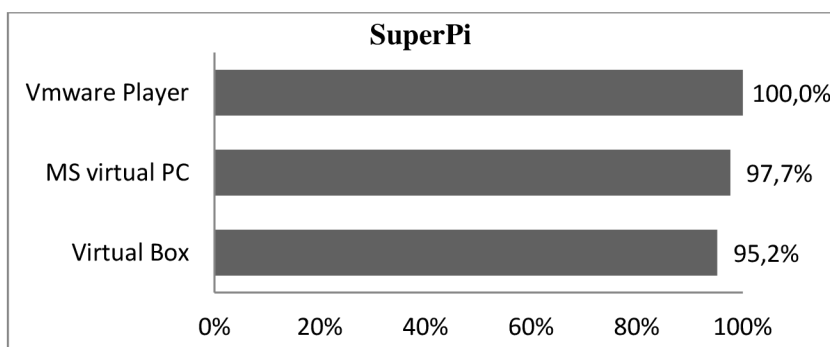
Poslední test bude proveden na obou operačních systémech. Půjde o přenos souborů pomocí standardního grafického rozhraní mezi oběma fyzickými počítači. Budeme přenášet komprimovaný archiv o velikosti 992 MB. Na operačním systému Windows budeme používat standardní sdílení souborů pomocí protokolu SMB a na operačním systému Linux budeme používat sdílení souborů pomocí Samby. Na vzdáleném počítači bude v systému Windows sdílen adresář s archivem a na hostujícím počítači budeme tento archiv kopírovat ze sítě na lokální disk. Výsledkem bude doba potřebná k přenosu dat.

6 Výsledky

V této kapitole si představíme všechny výsledky z naměřených testů. Výsledky budou postupně uváděny pro dané kategorie a na konci každé kategorie bude uveden celkový výsledek za danou kategorii. Výsledky budou uvedeny v podobě grafu, kde budou průměrné hodnoty převedeny na procenta. Všechny výsledky testů budeme uvádět odděleně pro oba testované operační systémy. Kompletní výsledky všech měření v každém testu si můžeme prohlédnout v [příloze A](#), kde jsou uvedeny v přehledných tabulkách.

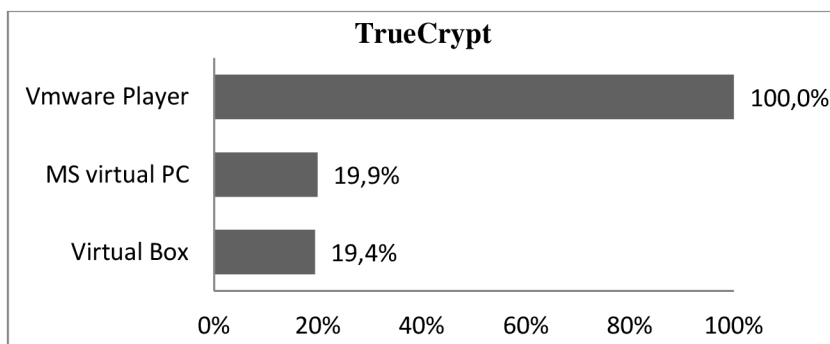
6.1 CPU – Windows

V testech procesoru pro operační systém Windows jsme testovali pomocí tří nástrojů, které spadají do kategorie syntetických testů. U programu *TrueCrypt* jsou uvedeny dva grafy, každý pro jiný algoritmus, který byl použit při testování.

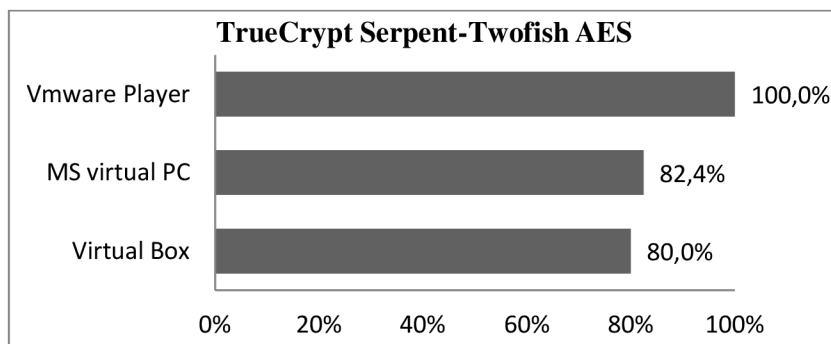


Graf 6.1: SuperPi.

V aplikaci *SuperPi* byly výsledky mezi jednotlivými nástroji velmi nepatrné. Mezi nejrychlejším *VMwarem* a nejpomalejším *VirtualBoxem* byl rozdíl 0,556 sekund.

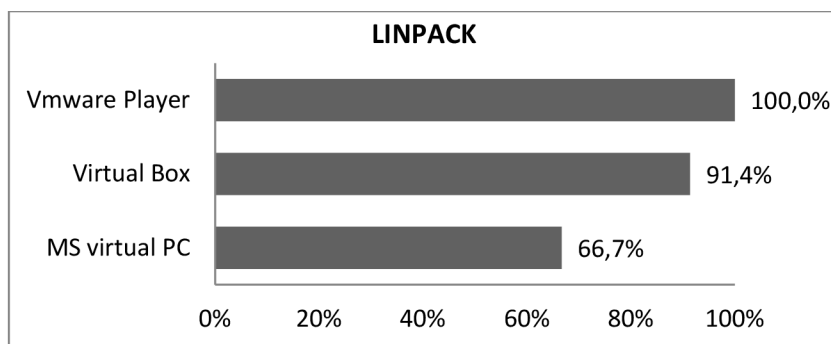


Graf 6.2: TrueCrypt AES.



Graf 6.3: TrueCrypt Serpent-Twofish AES.

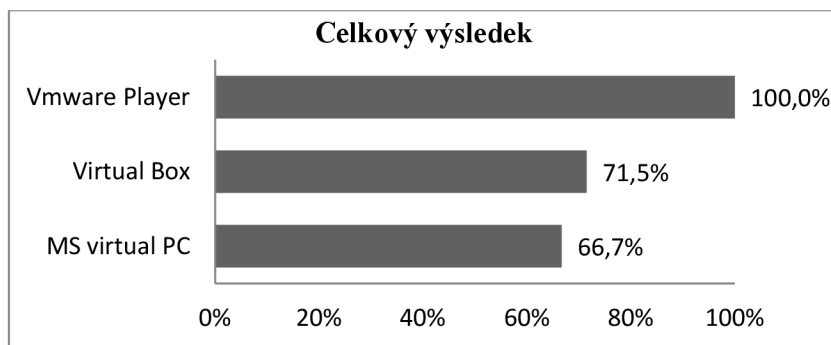
U aplikace *TrueCrypt*, konkrétně u algoritmu AES byly výsledky velmi rozdílné. *VMware* byl více než pětikrát lepší než ostatní nástroje. Tento velký rozdíl je způsoben hardwarovou podporou AES instrukcí, kterou *VMware* nabízí. U algoritmu Serpent-Twofish nebyly rozdíly tak patrné.



Graf 6.4: LINPACK.

V posledním benchmarku byl nejlepší opět *VMware*, kdy oproti nejpomalejšímu *VirtualPC* naměřil o 4 Gflops více, tedy celkem 12,7 Gflops.

Výsledné hodnocení pro kategorii procesoru pod operačním systémem Windows:

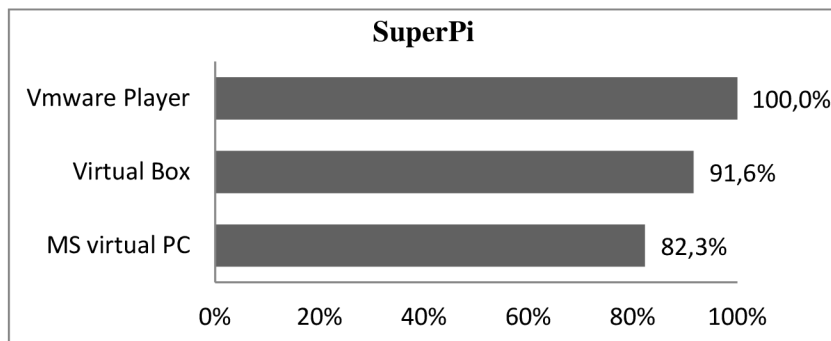


Graf 6.5: Celkový výsledek.

V testu procesoru byl *VMware* ve všech testech nejlepší, proto má i v celkovém hodnocení 100%. Nejvýraznější rozdíl byl v testu *TrueCrypt*. *VirtualPC* byl nejslabší v testu *LINPACK*. Rozdíl byl natolik velký, že i v celkovém hodnocení skončil na posledním místě.

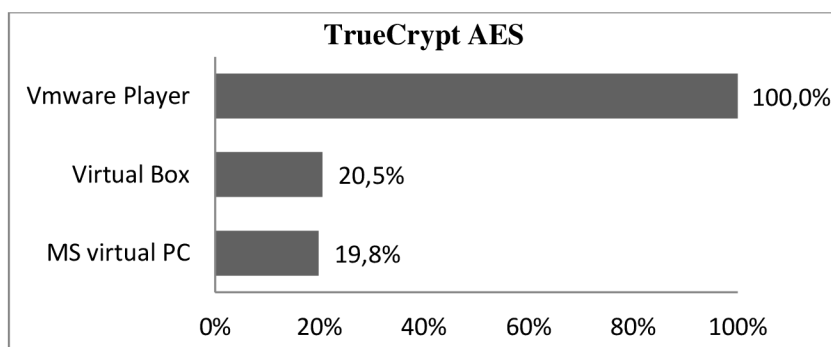
6.2 CPU – Linux

U operačního systému Linux byly výsledky velmi podobné. Testy byly provedeny se stejnými nástroji jako u operačního systému Windows.

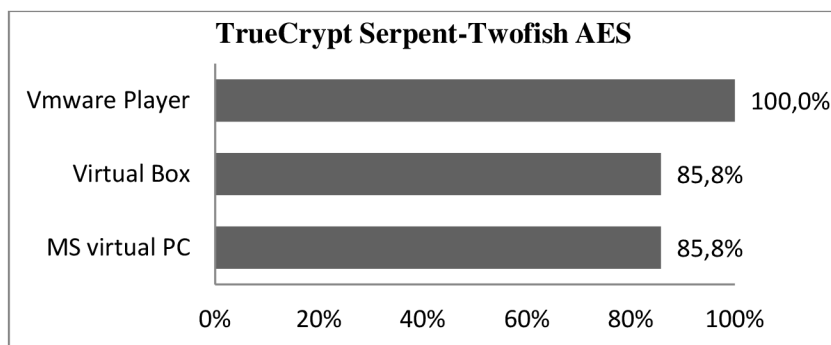


Graf 6.6: SuperPi.

V testu programu *SuperPi* byly ve výsledcích již větší rozdíly než na Windows. Rozdíl mezi nejrychlejším *VMwarem* a nejpomalejším *VirtualPC* byl více než 2 sekundy.

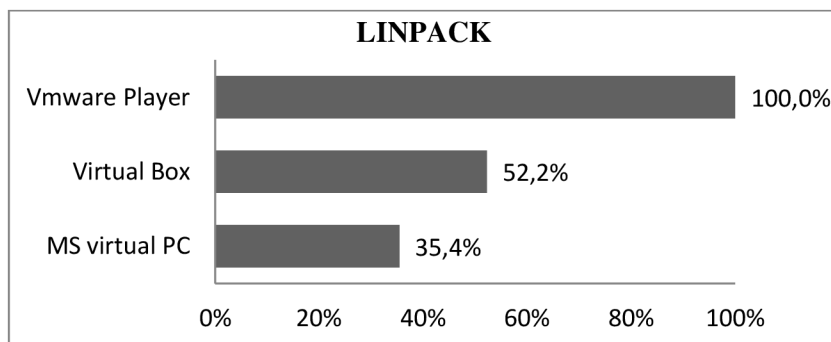


Graf 6.7: TrueCrypt AES.



Graf 6.8: TrueCrypt Serpent-Twofish AES.

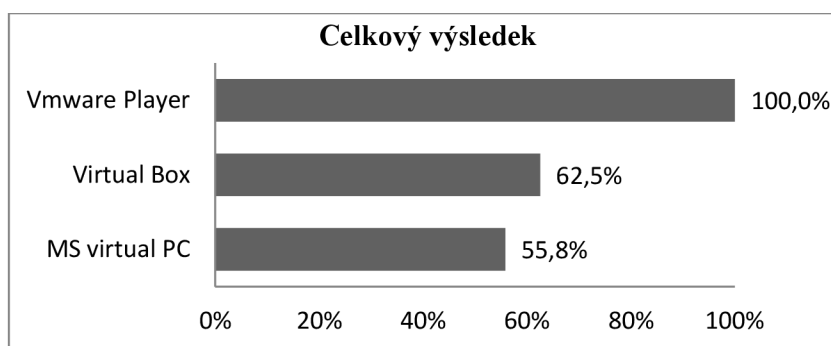
V testu *TrueCrypt* byly výsledky podobné jako u operačního systému Windows, tedy u algoritmu AES byl *VMware* až pětikrát lepší než ostatní nástroje. Opět je to způsobeno jeho hardwarovou podporou AES instrukcí.



Graf 6.9: LINPACK.

V posledním testu pod operačním systémem Linux se ukázal *VMware* jako nejvýkonnější. Byl téměř třikrát lepší než *VirtualPC* a téměř dvakrát lepší než *VirtualBox*.

Výsledné hodnocení pro kategorii procesoru pod operačním systémem Linux:

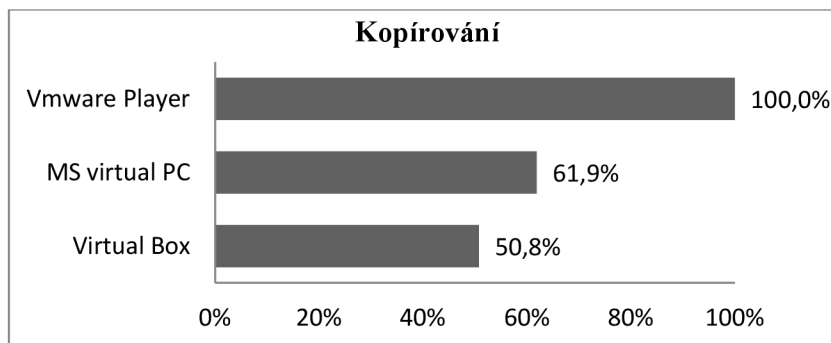


Graf 6.10: Celkový výsledek.

I pod operačním systémem byl *VMware* nejlepší ve všech testech, proto i v celkovém hodnocení má 100%. I zde se projevila hardwarové podpora AES instrukcí v testu *TrueCrypt*. Naopak *VirtualPC*, který oficiálně nepodporuje operační systém Linux, byl ve všech testech nejhorší.

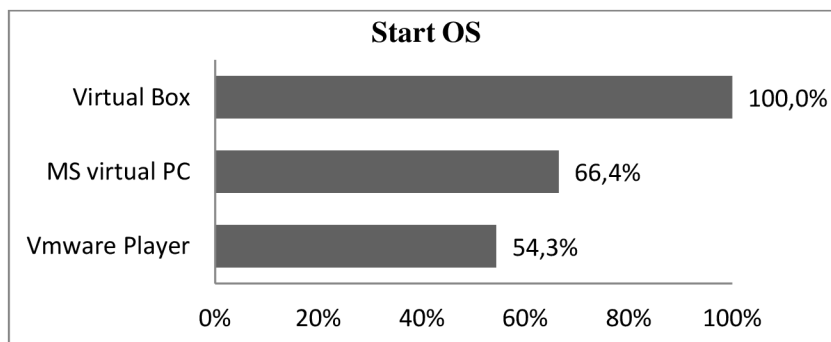
6.3 HDD – Windows

V testech pro pevný disk nejsou výsledky tak jednoznačné jako u procesoru. V testu byly zařazeny dva praktické testy a jeden syntetický.



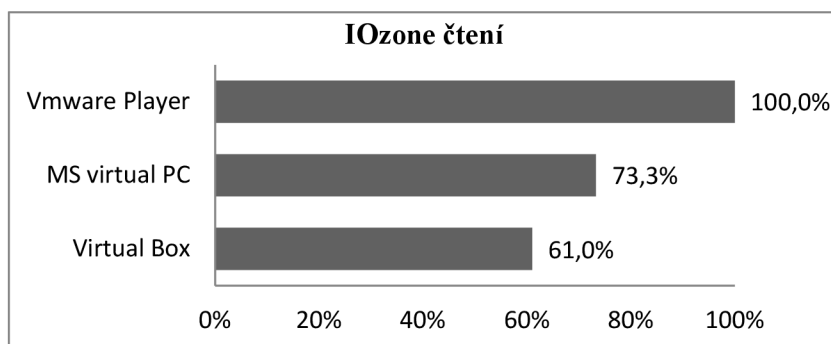
Graf 6.11. Kopírování.

V praktickém testu kopírování velkého počtu malých souborů byl *VMware* nejrychlejší. Téměř dvakrát oproti *VirtualBoxu*.

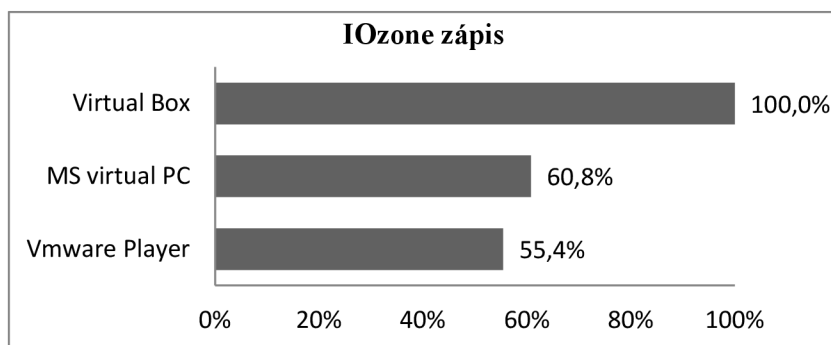


Graf 6.12: Start OS.

Naopak v testu startu operačního systému byl *VMware* nejpomalejší. Může to být způsobeno zaváděním jeho *VMware Tools*. Rozdíl mezi nejrychlejším *VirtualBoxem* a *VMwarem* je téměř 16 sekund.



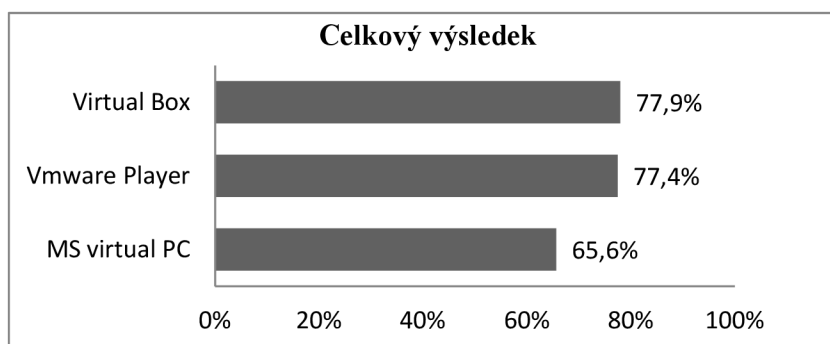
Graf 6.13: IOzone čtení.



Graf 6.14: IOzone zápis.

V syntetickém testu byly výsledky velmi rozdílné. Zatímco v testu čtení byl nejlepší *VMware* a nejhorší *VirtualBox*, v testu zápisu tomu bylo právě naopak.

Výsledné hodnocení pro kategorii pevného disku pod operačním systémem Windows:

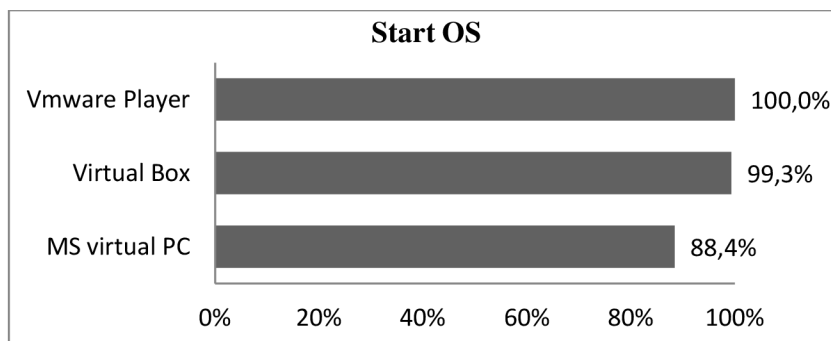


Graf 6.15: Celkový výsledek.

Ve výsledcích testů pevného disku jsou rozdíly mezi *VMwarem* a *VirtualBoxem* nepatrné. V praktickém testu kopírování byl nejlepší *VMware* a *VirtualBox* byl téměř dvakrát horší, naopak v testu startu operačního systému byly výsledky mezi těmito dvěma nástroji naopak. *VirtualPC* dopadl nejhůře v obou praktických testech, proto i ve výsledném hodnocení je nejhorší.

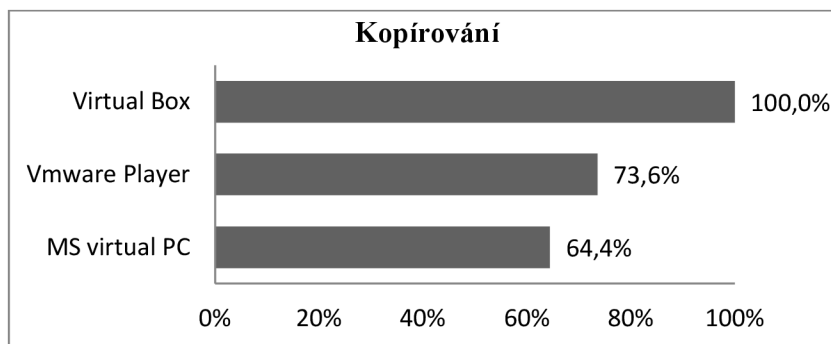
6.4 HDD – Linux

I v případě testů pro Linux jsou výsledky rozdílné. V syntetickém benchmarku a praktickém testu kopírování se jako nejlepší ukázal *VMware*, naopak v testu startu operačního systému stejně tak jako v případě systému Windows byl nejpomalejší.



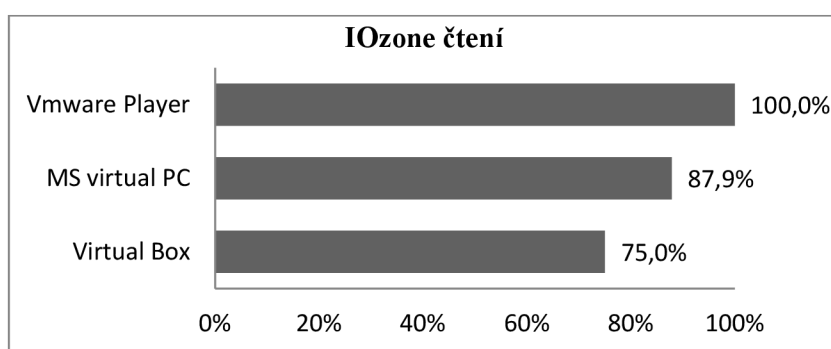
Graf 6.16: Start OS.

U operačního systému Linux nebyly výsledky kopírování tak rozdílné jako v případě Windows. I zde byl však nejrychlejší *VMware*.

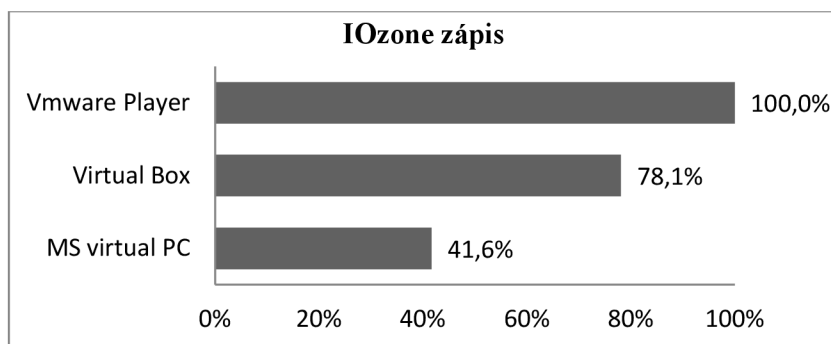


Graf 6.17: kopírování.

V testu startu operačního systému byl nejlepší *VirtualBox*, stejně tak jako u operačního systému Windows. Naopak *VMware* v případě startu Linuxu byl lepší než *VirtualPC*.



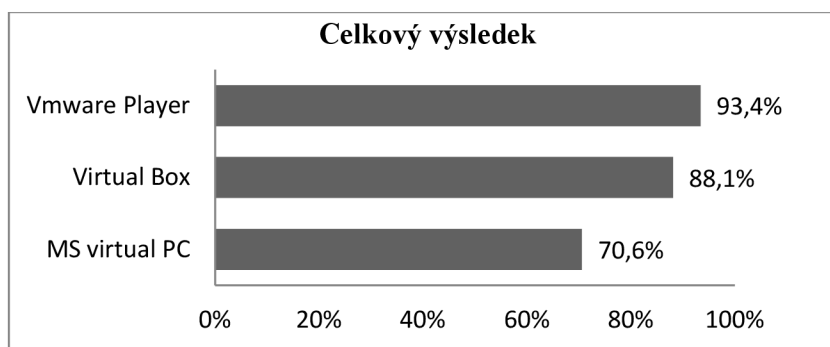
Graf 6.18: IOzone čtení.



Graf 6.19: IOzone zápis.

V syntetickém testu na operačním systému Linux byl v obou případech lepší *VMware*. *VirtualPC* byl lepší než *VirtualBox* v případě testu čtení, u testu zápisu byl naopak horší.

Výsledné hodnocení pro kategorii pevného disku pod operačním systémem Linux:

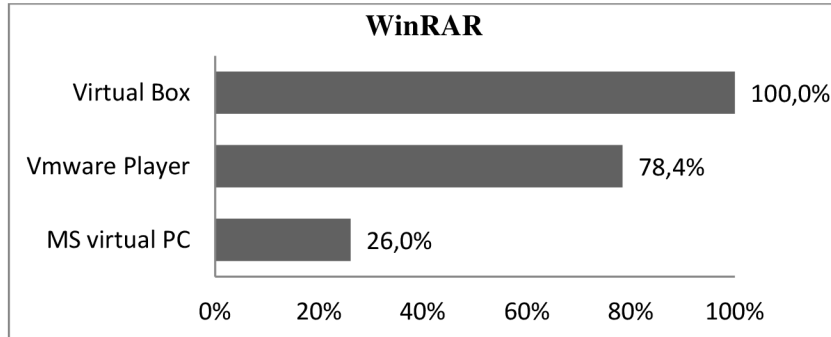


Graf 6.20: Celkový výsledek.

Pro operační systém Linux nebyly výsledky tak rozdílné v jednotlivých testech. Zde byl *VMware* nejlepší téměř ve všech testech, v testu startu operačního systému byl stejně tak jako na systému Windows nejhůřší. Naopak *VirtualPC* byl téměř ve všech testech nejhůřší, proto i v celkovém skóre je na posledním místě.

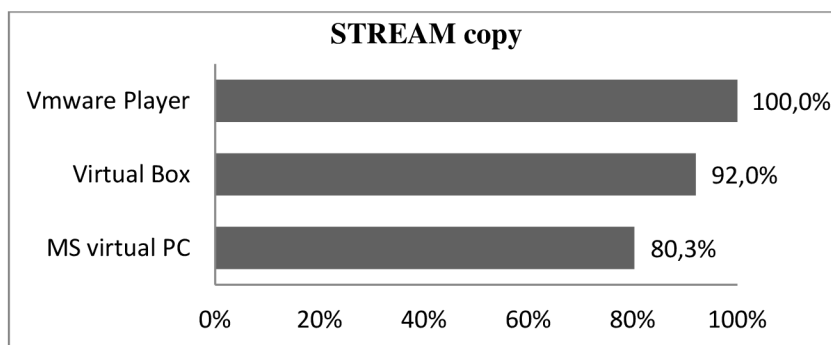
6.5 RAM – Windows

Při testování RAM jsme používali stejný test, jako u procesoru. Jde o program *SuperPi*. Graf výsledků zde nebude uveden, protože si je můžeme prohlédnout v části CPU. Do celkového výsledku však bude zahrnut.

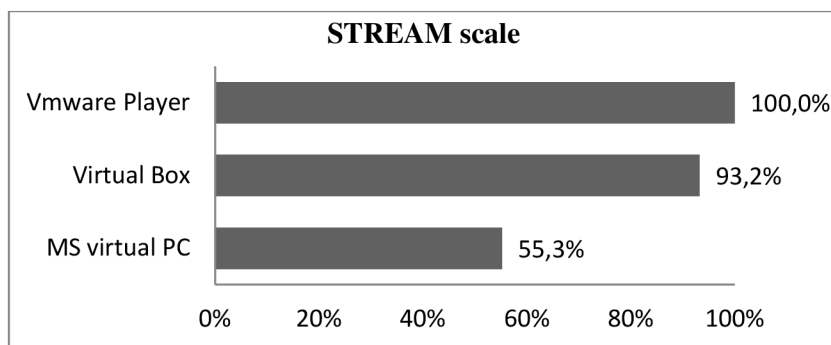


Graf 6.21: WinRAR.

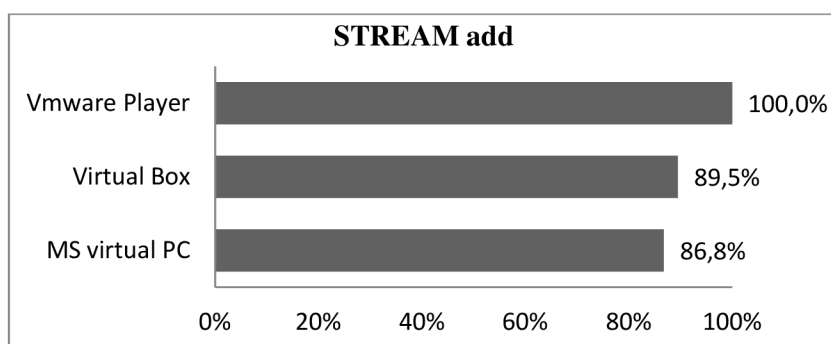
V praktickém testu komprimace zdrojových kódů byl nejlepší *VirtualBox*. Naopak nejhůřše dopadl program *VirtualPC*, který byl téměř pětkrát horší než *VirtualBox*.



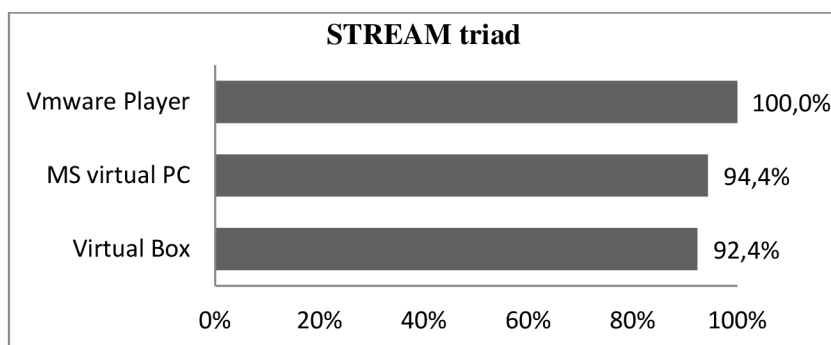
Graf 6.22: STREAM copy.



Graf 6.23: STREAM scale.



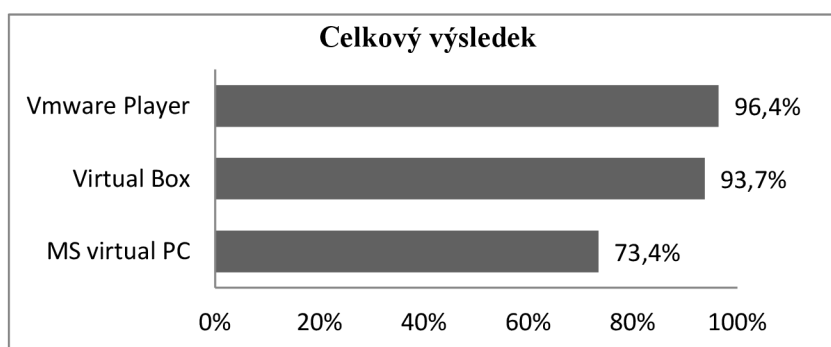
Graf 6.24: STREAM add.



Graf 6.25: STREAM triad.

V syntetickém testu pomocí programu STREAM byly výsledky téměř ve všech operacích vyrovnané. Ve všech testech byl však nejlepší program *VMware*.

Výsledné hodnocení pro kategorii RAM paměti pod operačním systémem Windows:

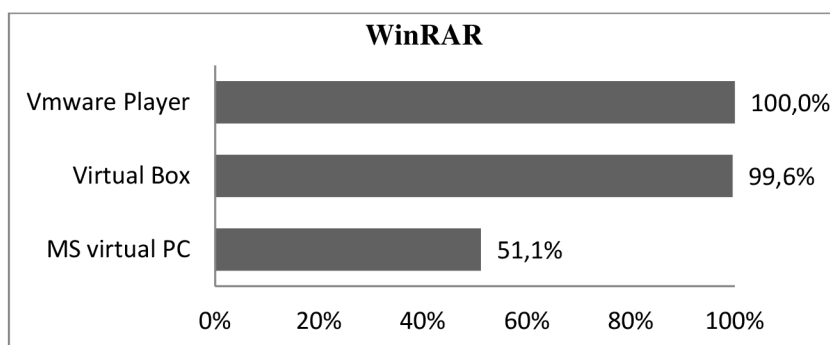


Graf 6.26: Celkový výsledek.

V testech paměti RAM jsou výsledky *VMwaru* i *VirtualBoxu* velmi vyrovnané. V syntetickém testu propustnosti paměti byl nejlepší *VMware* ve všech čtyřech operacích. Naopak nejpomalejší byl *VirtualPC*, který byl nejhorší ve třech operacích. Největší propad výkonu u programu *VirtualPC* byl při operaci *scale*, kdy byl téměř dvakrát horší než nejrychlejší *VMware*. *VirtualPC* nejvíce zaostal za ostatními nástroji v testu komprimace pomocí programu *WinRAR*. I v celkovém skóre se umístil jako poslední.

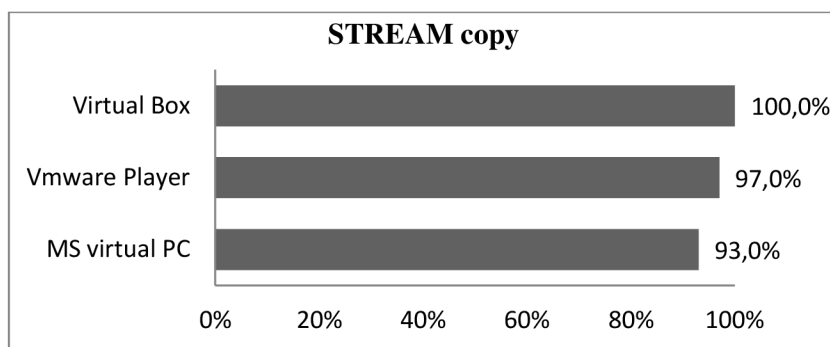
6.6 RAM – Linux

Stejně jako u testů pro Windows i u testů pro Linux zde nebude uveden výsledný graf pro test *SuperPi*. Do výsledného hodnocení však bude zahrnut.

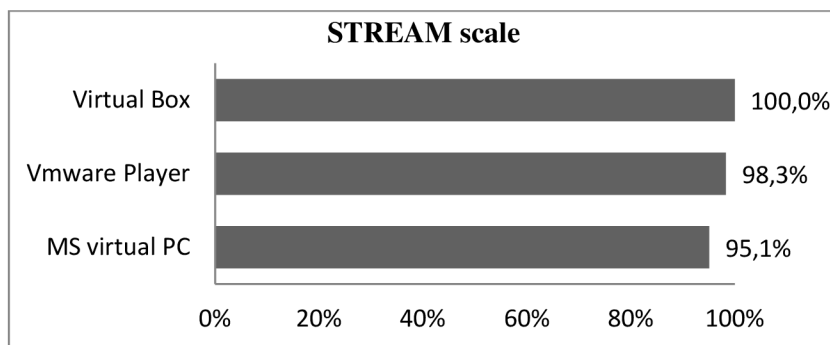


Graf 6.27: WinRAR.

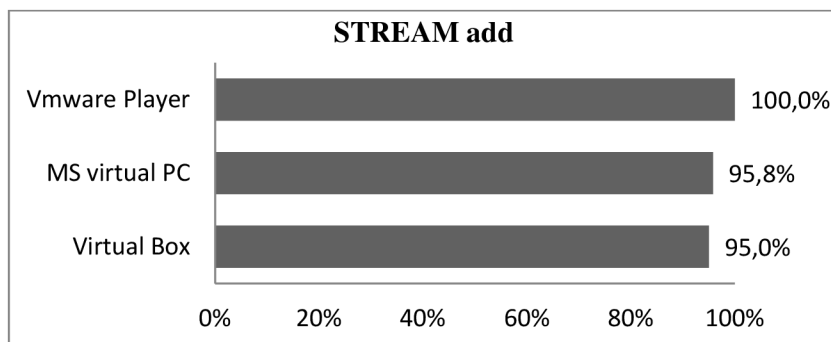
U operačního systému Linux byl v testu komprimace nejrychlejší *VMware*, který byl téměř dvakrát lepší než nejpomalejší *VirtualPC*.



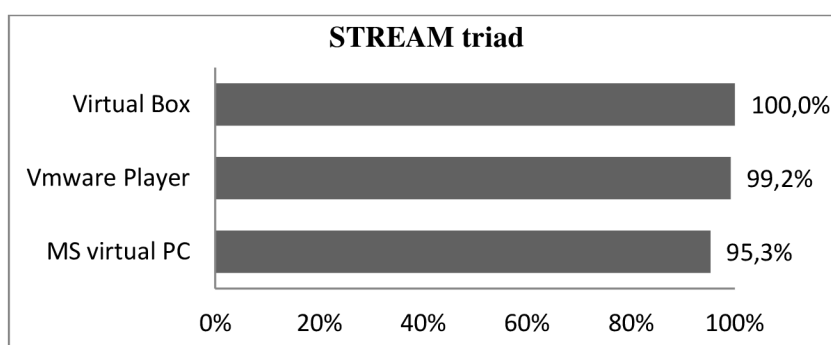
Graf 6.28: STREAM copy.



Graf 6.29: STREAM scale.



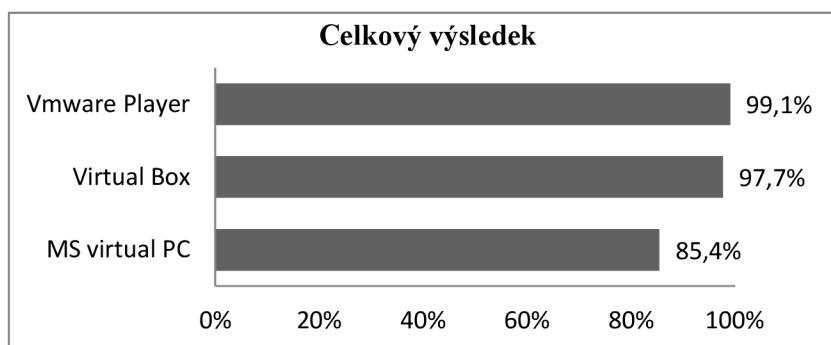
Graf 6.30: STREAM add.



Graf 6.31: STREAM triad.

V syntetickém testu programu *STREAM* byly u operačního systému Linux výsledky ve všech operacích velmi vyrovnané s minimálními rozdíly. Nejlepší ve třech ze čtyřech testovaných operací byl však program *VirtualBox*.

Výsledné hodnocení pro kategorii RAM paměti pod operačním systémem Linux

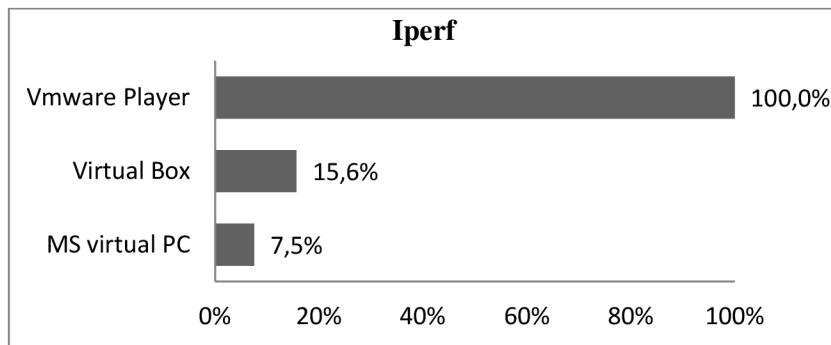


Graf 6.32: Celkový výsledek.

Na operačním systému Linux jsou výsledky velmi podobné. *VMware* i *VirtualBox* dosáhly podobných výsledků v celkovém hodnocení a *VirtualPC* skončil opět na posledním místě.

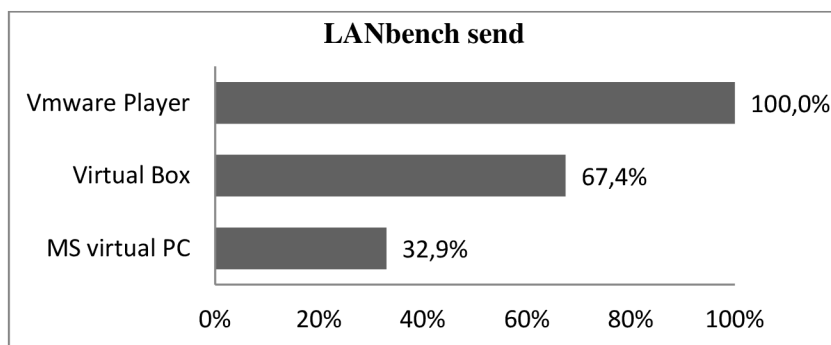
6.7 LAN – Windows

V testech síťové karty byly provedeny dva syntetické a jeden praktický test. V praktickém testu byla přenášena data o velikosti 992 MB. Šlo o jeden soubor.

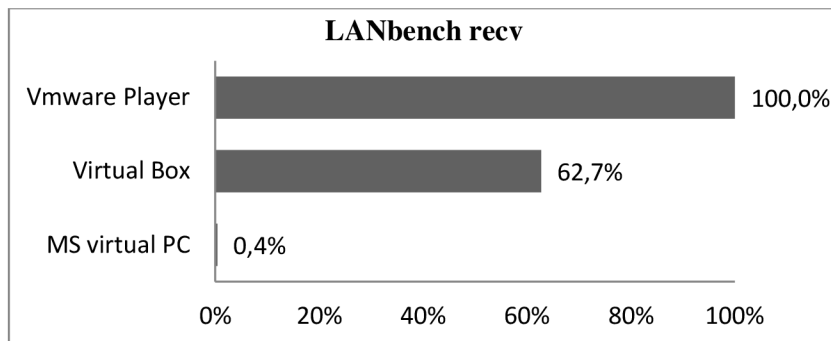


Graf 6.33: Iperf.

V prvním syntetickém testu se jako nejrychlejší ukázal *VMware*, který byl několikanásobně rychlejší, než ostatní testované nástroje.

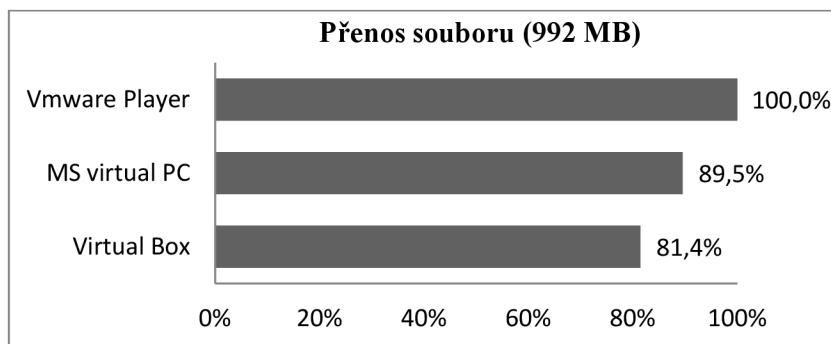


Graf 6.34: LANbench send.



Graf 6.35: LANbench recv.

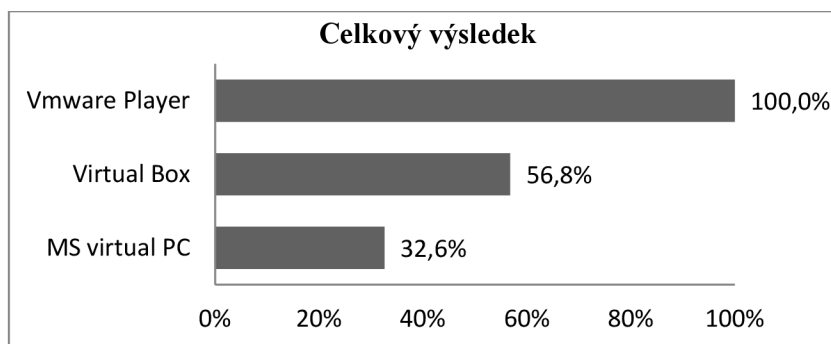
V druhém syntetickém testu se opět ukázal jako nejrychlejší program *VMware*. Naměřené výsledky nejsou tak rozdílné jako při testu pomocí programu *Iperf*, ale pořadí výsledků je stejné.



Graf 6.36: Přenos souboru.

V praktickém testu přenosu souboru jsou výsledky nejvíce vyrovnané. I zde je ovšem nejrychlejší *VMware*, naopak nejpomalejší byl v testu program *VirtualBox*.

Výsledné hodnocení pro kategorii síťové karty pod operačním systémem Windows:

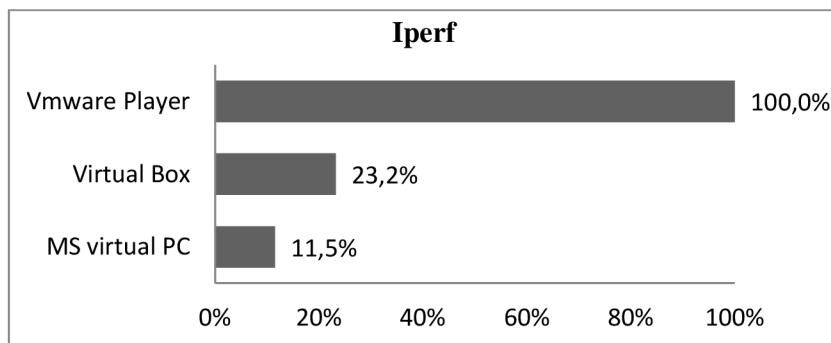


Graf 6.37: Celkový výsledek.

Při testování síťové karty se opět jako nejlepší ve všech testech ukázal *VMware*. Naopak jako nejhorší se téměř ve všech testech ukázal *VirtualPC*. V syntetických testech byly výsledky velmi rozdílné, kdy *VMware* byl několikanásobně lepší. V praktickém testu přenosu souboru byly výsledky již více vyrovnané.

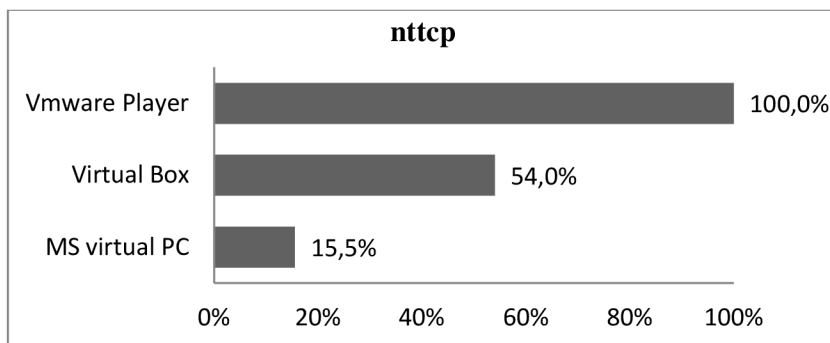
6.8 LAN – Linux

I na Linuxu jsme testovali pomocí dvou syntetických testů a jednoho praktického.



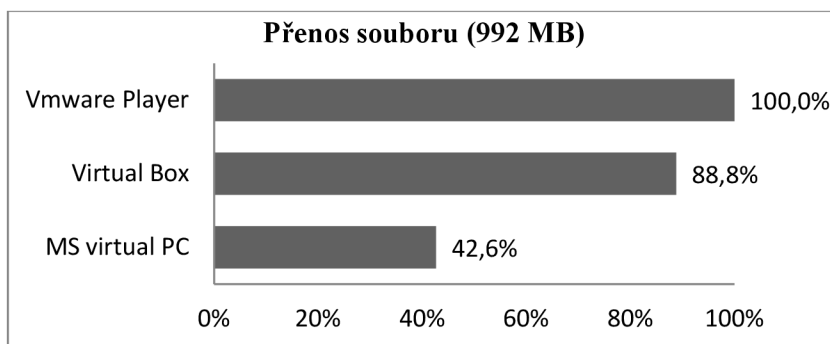
Graf 6.38: Iperf.

Test pomocí programu *Iperf* dopadl velmi podobně jako v případě Windows. I zde se ukázal *VMware* jako několikanásobně rychlejší.



Graf 6.39: nttcp.

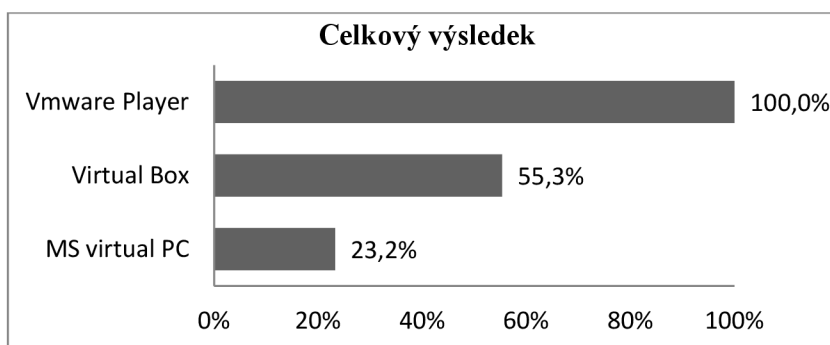
Podobné výsledky jako u programu *Iperf* jsme naměřili i pomocí programu *nttcp*. Nejrychlejší byl opět *VMware* a nejpomalejší *VirtualPC*.



Graf 6.40: Přenos souboru.

I na Linuxu jsou výsledky praktického testu více vyrovnané než v případě syntetických testů. Nejrychlejší byl opět *VMware* a nejpomalejší *VirtualPC*.

Výsledné hodnocení pro kategorii síťové karty pod operačním systémem Linux:



Graf 6.41: Celkový výsledek.

Pod operačním systémem Linux byl stejně tak jako v případě Windows nejlepší nástroj *VMware*, který byl ve všech testech nejrychlejší. Nejhůře si vedl *VirtualPC*.

6.9 Výsledné zhodnocení

Všechny virtualizační nástroje jsme testovali ve čtyřech kategoriích testů. Byly to testy zaměřené na procesor, RAM paměť, pevný disk a síťovou kartu. Pro každou kategorii byly zvoleny tři nástroje, kde byly zastoupeny aplikační i syntetické benchmarky. Nástroje byly vybírány s ohledem na jejich použitelnost pro oba operační systémy. V kategoriích procesor, paměť RAM a pevný disk byly použity stejné nástroje pro oba operační systémy, lišily se pouze v použitých verzích. V kategorii síťové karty byly použity dva stejné nástroje a třetí test proběhl pomocí různé aplikace pro Windows a Linux. Abychom eliminovali výchyly jednotlivých měření, které mohly během testování nastat, každý test byl spuštěn třikrát a výsledky zprůměrovány. U některých benchmarků bylo měřeno více různých veličin. V celkovém výsledku každé kategorie jsou započítány všechny výsledky. V některých kategoriích se celkový výsledek skládá z průměru tří naměřených hodnot v jednotlivých testech, v jiných kategoriích to může být více.

V kategorii CPU byl pro oba operační systémy nejkvonnější nástroj VMware. V celkovém výsledku byl téměř o 30% lepší než konkurence pod operačním systémem Windows a téměř o 40% lepší než zbývající nástroje pod Linuxem. Tento náskok si získal zejména díky testu pomocí programu *TrueCrypt*, kde se pracuje s instrukcemi AES. VMware nabízí hardwarovou akceleraci těchto instrukcí. Naopak nejhorší výsledek zaznamenal program *VirtualPC* od Microsoftu. Na *VirtualBox*, který skončil na druhém místě, ztrácel u obou operačních systémů téměř 7%. Z těchto výsledků je tedy patrné, že pokud požadujeme vysoký výpočetní výkon po procesoru, nejlepších výsledků dosáhneme v programu VMware. Z příkladů programů, které vyžadují vysoký výkon procesoru, můžeme uvést program *TrueCrypt*, který se používá pro šifrování disků.

V kategorii pevného disku byly pro operační systém Windows celkové výsledky velmi podobné u VMware a VirtualBoxu, kdy byl VirtualBox lepší o 0,5%. Při kopírování souboru mezi dvěma fyzickými disky byl v případě Windows nejlepší VMware, kdy jeho náskok byl téměř 40% před druhým VirtualPC. Lze tedy říci, že při kopírování dat mezi více fyzickými disky pod Windows je nejlepší nástroj VMware. V syntetickém testu pod Windows se v případě čtení z disku ukázal nejlepší VMware, naopak v případě zápisu na disk nejlepší VirtualBox. Zde má VirtualBox výhodu. Pod operačním systémem Linux byly výsledky kopírování souboru mezi dvěma fyzickými disky velmi podobné. Zde byl VMware jen o 0,7% lepší než VirtualBox, tedy v případě Linuxu a kopírování mezi více fyzickými disky nemá VMware tolik navrch jako u Windows. V syntetických testech byl však VMware nejlepší jak v případě čtení, tak i v případě zápisu na pevný disk. Tedy pro Linux je rychlejší VMware. Největší rozdíly u obou systémů byly v případě testu startu operačního systému. Zde byl v obou případech nejrychlejší VirtualBox. U systému Windows byl nejpomalejší VMware. To může být způsobeno například zaváděním nástroje *VMware Tools* do hostujícího operačního systému pro jeho lepší integraci s hostitelským operačním systémem. V případě Linuxu načíst operační systém nejdéle trvalo programu *VirtualPC*.

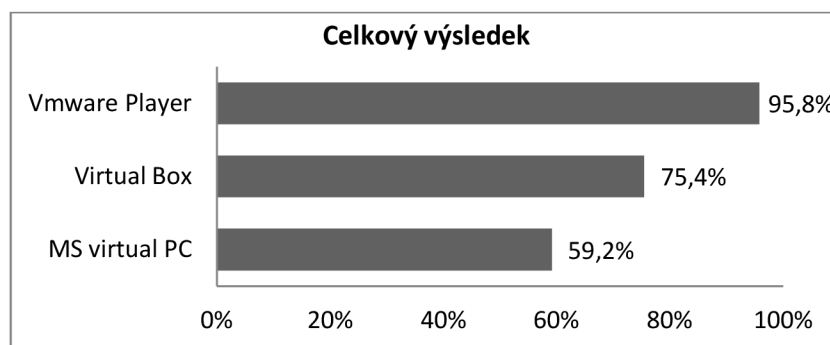
V kategorii paměti RAM byl pro oba systémy lepší VMware. Jeho rozdíl oproti druhému VirtualBoxu není však velký. V celkovém výsledku se oba nástroje liší maximálně o necelá 4%. Naopak VirtualPC je oproti druhému VirtualBoxu až o 20% horší. U obou systémů při syntetickém benchmarku pomocí programu *STREAM benchmark* byly výsledky velmi podobné a těsné. Jen při testu pomocí operace *scale* na Windows byl VirtualPC výrazněji pomalejší než konkurence. Největších rozdílů v této kategorii jsme naměřili u praktického benchmarku pomocí programu *WinRAR*, kdy jsme komprimovali data. Pod Windows byl nejrychlejší VirtualBox, který byl o více než 21% rychlejší než druhý VMware. U operačního systému Linux na tom byly VMware i VirtualBox, kdy VMware byl o 0,4% lepší. Naopak VirtualPC v tomto praktickém benchmarku výrazně zaostal. Lze tedy říci, že při práci s programy, které vyžadují vysokou propustnost RAM je

nejvhodnější *VMware*. Druhý *VirtualBox* ovšem nikterak nezaostává. *VirtualPC* je v této kategorii nejpomalejší.

V poslední kategorii síťové karty byl u obou operačních systémů nejrychlejší *VMware*. V syntetických testech dokonce několikanásobně, než ostatní programy. Naopak *VirtualPC* byl v syntetických testech nejhorší. Ani v jednom testu u obou systémů se propustnost nedostala přes hranici 100Mb/s. Více vyrovnané výsledky přinesl praktický test, kdy jsme kopírovali jeden velký soubor. V případě Windows jsme kopírovali pomocí klasického sdílení souborů a u Linuxu pomocí doinstalované *Samby*. U Windows byly výsledky vyrovnané, kdy na nejrychlejší *VMware* ztrácel *VirtualPC* necelých 11% a *VirtualBox* necelých 19%. V případě Linuxu byl nejhorší *VirtualPC*, který na nejlepší *VMware* ztratil více než 47%. Pro práci se síťovou kartou je nejlepší program *VMware*. Na posledním místě u obou operačních systémů skončil *VirtualPC*. Musíme však poznamenat, že v případě přenosu většího souboru přes síť má u operačního systému Windows navrch. V případě Linuxu je na tom však podstatně hůře.

Z výsledného zhodnocení je vidět, že ve většině testech u obou operačních systémů byl nejlepší program *VMware*, který nabízel nejvyšší výkon v mnoha operacích. Naopak *VirtualPC* byl nejhorší v celkovém hodnocení u všech kategorií pro oba operační systémy. Pod operačním systémem Linux byl ve všech testech kromě dvou nejhorší *VirtualPC*. Toto může být způsobeno neoficiální podporou tohoto systému v programu. Naopak pro operační systém Windows byl v několika testech lepší než *VirtualBox*.

Na závěr si ukážeme graf, kde si zobrazíme celkový výsledek pro všechny měřené kategorie i operační systémy, abychom tak mohli ukázat, který nástroj si celkově vedl nejlépe. Hodnoty budou vypočítány podobně jako u jednotlivých dílčích výsledků z každé kategorie, tedy vezmeme aritmetický průměr všech hodnot z tabulek celkových výsledků.



Graf 6.42: Celkový výsledek.

Z tohoto celkového grafu můžeme říci, že nejlepšími výsledky dosáhl *VMware*, jako druhý nejvýkonnější se ukázal *VirtualBox* a nejhůře si vedl *VirtualPC*.

Jelikož *VirtualPC* neumí pracovat s více jádry procesoru, omezili jsme i ostatní dva programy a všem virtuálním počítačům jsme nastavili pouze jeden procesor. Tak měli všechny programy stejné podmínky. Některé námi zvolené aplikace pro testování umí však využívat více jader procesoru. Pokud bychom tedy *VirtualBoxu* a *VMwaru* nastavili, aby virtuální počítače využívali všechny jádra hostitelského procesoru, rozdíl mezi výsledky těchto dvou programů a programem *VirtualPC* by mohly být výraznější. Tohle je velkou slabinou programu *VirtualPC*. V současné době stále více programů umí využívat více jader procesoru a to je výpočetní výhodou, se kterou v programu *VirtualPC* nemůžeme počítat.

7 Závěr

V práci jsme si vysvětlili jednotlivé techniky virtualizace a podívali se, jak pracují a přistupují k hardwaru počítače. V další části jsme si představili jednotlivé virtualizační nástroje, které jsme v práci testovali. Podrobně jsme si uvedli jednotlivé vlastnosti, zejména systémové požadavky pro spuštění a podporované operační systémy. V závěru kapitoly jsme provedli porovnání těchto nástrojů na základě popsaných vlastností. Než jsme mohli přistoupit na samotné testování bylo nutné se zabývat samotnou problematikou testování výkonnosti. Uvedli jsme teoretický úvod do testování PC a probrali jednotlivé kategorie testování. V závěru kapitoly jsme si představili jednotlivé nástroje, které budou použity pro samotné testování. Důležitou součástí testování bylo stanovit konkrétní metodiku. Tedy detailně popsat podmínky, za kterých bude testování probíhat. Tento problém jsme si rozdělili to třech částí, kterými byly hostitelský počítač a jeho konfigurace, samotné programy pro virtualizaci a jejich nastavení, programy pro testování výkonu a jejich konfigurace.

Výsledkem práce jsou grafy, které ukazují výkon v jednotlivých kategoriích. Díky výsledkům tohoto testování jsme mohli porovnat výkonnost jednotlivých nástrojů vzájemně mezi sebou. Dozvěděli jsme se tak, které nástroje a pro které operační systémy nabízejí nejlepší výkon. Testování bylo zaměřeno na měření výkonu ve čtyřech kategoriích. Práci by bylo možné rozšířit o testování dalších aspektů virtualizačních nástrojů, jakými jsou například přehlednost a komfort práce s programem. Tento aspekt by bylo nutné testovat formou vyhodnocení uživatelských reakcí na práci s danými programy. Dále by bylo možné se zaměřit například na podporu daných programů ze strany vývojářů a kvalitu jejich dokumentace. Testování výkonnosti by bylo možné rozšířit o testy z dalších kategorií, jakými jsou například grafický výkon. Měření by se také mohlo zpřesnit, pokud bychom provedli testování daných nástrojů na rozdílně výkonných hostitelských počítačích.

V práci jsme provedli měření výkonnosti z konkrétního námi zvoleného pohledu. Naměřené výsledky jsou objektivní, tedy můžeme nástroje mezi sebou z tohoto hlediska porovnávat. Pokud bychom do testování zahrnuli i jiné, již zmíněné aspekty, mohli bychom dosáhnout odlišných výsledků.

Literatura

- [1] Matyska, Luděk. *Virtualizace výpočetního prostředí. Zpravodaj ÚVT MU. ISSN 1212-0901, 2006, roč. XVII, č. 2, s. 9-11.*
- [2] Matyska, Luděk. *Techniky virtualizace počítačů (2). Zpravodaj ÚVT MU. ISSN 1212-0901, 2007, roč. XVII, č. 3, s. 9-12.*
- [3] ŠIŠKA, Michal. *Virtuální počítač : Praktická řešení pro domácí uživatele. Vydání první. Brno : Computer Press, a.s., 2011. 256 s. ISBN 978-80-251-3334-7.*
- [4] WARD, Brian. *VMWARE : Provozujeme více operačních systémů na jednom počítači. Vydání první. Brno : Computer Press, a.s., 2004. 268 s. ISBN 80-251-0129-0.*
- [5] Chapter 10. Technical background. *Oracle VM VirtualBox®* [online]. [cit. 2012-05-05]. Dostupné z: <http://www.virtualbox.org/manual/ch10.html>
- [6] Error message about hardware-assisted virtualization for Windows XP Mode in Windows Virtual PC on a computer that is running Windows 7. *Microsoft Help and Support* [online]. [cit. 2012-01-06]. Dostupné z: <http://support.microsoft.com/kb/977206>
- [7] VMware Guest Operating System Installation Guide. *VMware Virtualization Software for Desktops, Servers & Virtual Machines for Public and Private Cloud Solutions* [online]. [cit. 2012-01-06]. Dostupné z: <http://partnerweb.vmware.com/GOSIG/home.html>
- [8] VMware Compatibility Guide: System Search. *VMware Virtualization Software for Desktops, Servers & Virtual Machines for Public and Private Cloud Solutions* [online]. [cit. 2012-01-06]. Dostupné z: <http://www.vmware.com/resources/compatibility/search.php>
- [9] Windows XP Mode. *Microsoft Windows* [online]. [cit. 2012-01-06]. Dostupné z: <http://windows.microsoft.com/cs-CZ/windows7/products/features/windows-xp-mode>
- [10] VMware KB: Virtual machine hardware versions. *VMware Virtualization Software for Desktops, Servers & Virtual Machines for Public and Private Cloud Solutions* [online]. 2011-09-17 [cit. 2012-01-06]. Dostupné z: http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1003746
- [11] VirtualBox. *VirtualBox* [online]. [cit. 2012-01-06]. Dostupné z: <https://www.virtualbox.org/>
- [12] Svět hardware: Benchmark. *Svět hardware* [online]. [cit. 2012-01-06]. Dostupné z: <http://www.svethardware.cz/glos.jsp?doc=7CC6E9CF57E6E53DC1257349003E5BC8>
- [13] LILJA, David J. *Measuring computer performance: A practitioner's guide.* Cambridge: Cambridge University Press, 2004. ISBN 0-521-03627-2.

- [14] Máme novou metodiku. *PCTuning* [online]. 2011-05-23 [cit. 2012-01-06]. Dostupné z:
<http://pctuning.tyden.cz/hardware/procesory-pameti/21085-mame-novou-metodiku-test-prvnich-7-procesoru?start=1>
- [15] MSDN AA - MerlinWiki. *Wiki@Merlin - MerlinWiki* [online]. [cit. 2012-05-06]. Dostupné z:
http://merlin.fit.vutbr.cz/wiki/index.php/MSDN_AA
- [16] Co je Ubuntu. *Co je Ubuntu* | *www.ubuntu.cz* [online]. [cit. 2012-05-06]. Dostupné z:
<http://www.ubuntu.cz/>

Příloha A

Tabulky naměřených výsledků

CPU – Windows

MS virtual PC	
pokus	čas (s)
1.	11,467
2.	11,386
3.	11,216
průměr	11,356

Tabulka A.1: SuperPi.

Virtual Box	
pokus	čas (s)
1.	11,922
2.	11,563
3.	11,453
průměr	11,646

Tabulka A.2: SuperPi.

Vmware Player	
pokus	čas (s)
1.	11,111
2.	11,080
3.	11,079
průměr	11,090

Tabulka A.3: SuperPi.

MS virtual PC				
algoritmus	průměr (MB/s) #pokus1	průměr (MB/s) #pokus2	průměr (MB/s) #pokus3	průměr
AES	137,0	139,0	136,0	137,3
Serpent-Twofish AES	30,7	30,6	30,3	30,5

Tabulka A.3: TrueCrypt.

Virtual Box				
algoritmus	průměr (MB/s) #pokus1	průměr (MB/s) #pokus2	průměr (MB/s) #pokus3	průměr
AES	134,0	134,0	134,0	134,0
Serpent-Twofish AES	29,5	29,6	29,6	29,6

Tabulka A.4: TrueCrypt.

Vmware Player				
algoritmus	průměr (MB/s) #pokus1	průměr (MB/s) #pokus2	průměr (MB/s) #pokus3	průměr
AES	689,0	683,0	702,0	691,3
Serpent-Twofish AES	37,4	36,9	36,7	37,0

Tabulka A.5: TrueCrypt.

MS virtual PC	
pokus	výkon (Gflops)
1.	8,5660
2.	8,2906
3.	8,5737
průměr	8,4768

Tabulka A.6: LINPACK.

Virtual Box	
pokus	výkon (Gflops)
1.	11,4701
2.	11,5076
3.	11,8464
průměr	11,6080

Tabulka A.7: LINPACK.

Vmware Player	
pokus	výkon (Gflops)
1.	12,7473
2.	13,0127
3.	12,3574
průměr	12,7058

Tabulka A.8: LINPACK.

CPU – Linux

MS virtual PC	
pokus	čas (s)
1.	11,645
2.	11,569
3.	11,649
průměr	11,621

Tabulka A.9: SuperPi.

Virtual Box	
pokus	čas (s)
1.	10,709
2.	10,269
3.	10,341
průměr	10,440

Tabulka A.10: SuperPi.

Vmware Player	
pokus	čas (s)
1.	9,677
2.	9,417
3.	9,609
průměr	9,568

Tabulka A.11: SuperPi.

MS virtual PC				
algoritmus	průměr (MB/s) #pokus1	průměr (MB/s) #pokus2	průměr (MB/s) #pokus3	průměr
AES	121,0	121,0	122,0	121,3
Serpent-Twofish AES	21,0	22,0	22,0	21,7

Tabulka A.12: TrueCrypt.

Virtual Box				
algoritmus	průměr (MB/s) #pokus1	průměr (MB/s) #pokus2	průměr (MB/s) #pokus3	průměr
AES	126,0	124,0	126,0	125,3
Serpent-Twofish AES	22,0	22,0	21,0	21,7

Tabulka A.12: TrueCrypt.

Vmware Player				
algoritmus	průměr (MB/s) #pokus1	průměr (MB/s) #pokus2	průměr (MB/s) #pokus3	průměr
AES	609,0	621,0	604,0	611,3
Serpent-Twofish AES	25,0	26,0	25,0	25,3

Tabulka A.13: TrueCrypt.

MS virtual PC	
pokus	výkon (Gflops)
1.	8,0930
2.	7,8423
3.	8,0076
průměr	7,9810

Tabulka A.14: LINPACK.

Virtual Box	
pokus	výkon (Gflops)
1.	11,6183
2.	11,8627
3.	11,8477
průměr	11,7762

Tabulka A.15: LINPACK.

Vmware Player	
pokus	výkon (Gflops)
1.	21,1793
2.	23,1591
3.	23,3412
průměr	22,5599

Tabulka A.16: LINPACK.

HDD – Windows

MS virtual PC	
pokus	čas (s)
1.	22,683
2.	23,015
3.	21,527
průměr	22,408

Tabulka A.17: Kopírování.

Virtual Box	
pokus	čas (s)
1.	27,680
2.	27,002
3.	27,315
průměr	27,332

Tabulka A.18: Kopírování.

Vmware Player	
pokus	čas (s)
1.	13,674
2.	13,877
3.	14,061
průměr	13,871

Tabulka A.19: Kopírování.

MS virtual PC	
pokus	čas (s)
1.	28,731
2.	27,866
3.	28,507
průměr	28,368

Tabulka A.20: Start OS.

Virtual Box	
pokus	čas (s)
1.	18,839
2.	18,950
3.	18,751
průměr	18,847

Tabulka A.21: Start OS.

Vmware Player	
pokus	čas (s)
1.	35,790
2.	34,197
3.	34,080
průměr	34,689

Tabulka A.22: Start OS.

MS virtual PC				
	sekvenční (KB/s) #1.pokus	sekvenční (KB/s) #2.pokus	sekvenční (KB/s) #3.pokus	sekvenční (KB/s) #průměr
čtení	78326	98268	97016	91203
zápis	38348	38599	40167	39038

Tabulka A.23: IOzone.

Virtual Box				
	sekvenční (KB/s) #1.pokus	sekvenční (KB/s) #2.pokus	sekvenční (KB/s) #3.pokus	sekvenční (KB/s) #průměr
čtení	86111	67623	73830	75855
zápis	64107	68741	59906	64251

Tabulka A.24: IOzone.

Vmware Player				
	sekvenční (KB/s) #1.pokus	sekvenční (KB/s) #2.pokus	sekvenční (KB/s) #3.pokus	sekvenční (KB/s) #průměr
čtení	124768	123426	125132	124442
zápis	36103	34408	36309	35607

Tabulka A.25: IOzone.

HDD – Linux

MS virtual PC	
pokus	čas (s)
1.	9,831
2.	9,276
3.	10,883
průměr	9,997

Tabulka A.26: Start OS.

Virtual Box	
pokus	čas (s)
1.	8,904
2.	8,731
3.	9,076
průměr	8,904

Tabulka A.27: Start OS.

Vmware Player	
pokus	čas (s)
1.	8,639
2.	8,998
3.	8,884
průměr	8,840

Tabulka A.28: Start OS.

MS virtual PC	
pokus	čas (s)
1.	31,982
2.	32,632
3.	33,055
průměr	32,556

Tabulka A.29: Kopírování.

Virtual Box	
pokus	čas (s)
1.	21,853
2.	20,616
3.	20,425
průměr	20,965

Tabulka A.30: Kopírování.

Vmware Player	
pokus	čas (s)
1.	28,628
2.	28,598
3.	28,181
průměr	28,469

Tabulka A.31: Kopírování.

MS virtual PC				
	sekvenční (KB/s) #1.pokus	sekvenční (KB/s) #2.pokus	sekvenční (KB/s) #3.pokus	sekvenční (KB/s) #průměr
čtení	121453	122298	122557	122103
zápis	47247	48353	47415	47672

Tabulka A.32: IOzone.

Virtual Box				
	sekvenční (KB/s) #1.pokus	sekvenční (KB/s) #2.pokus	sekvenční (KB/s) #3.pokus	sekvenční (KB/s) #průměr
čtení	103130	105356	104048	104178
zápis	86585	87912	94050	89516

Tabulka A.33: IOzone.

Vmware Player				
	sekvenční (KB/s) #1.pokus	sekvenční (KB/s) #2.pokus	sekvenční (KB/s) #3.pokus	sekvenční (KB/s) #průměr
čtení	142434	142454	131823	138904
zápis	115814	117022	110961	114599

Tabulka A.34: IOzone.

RAM – Windows

MS virtual PC	
pokus	čas (s)
1.	236,422
2.	259,320
3.	277,332
průměr	257,691

Tabulka A.35: WinRAR.

Virtual Box	
pokus	čas (s)
1.	68,123
2.	66,262
3.	66,257
průměr	66,881

Tabulka A.36: WinRAR.

Vmware Player	
pokus	čas (s)
1.	84,951
2.	79,100
3.	91,838
průměr	85,296

Tabulka A.37: WinRAR.

MS virtual PC				
funkce	#1.pokus (MB/s)	#2.pokus (MB/s)	#3.pokus (MB/s)	průměr (MB/s)
Copy	19131,08	15257,67	11980,49	15456,41
Scale	19050,90	6055,22	6975,55	10693,89
Add	19306,28	15433,51	17219,70	17319,83
Triad	19408,78	17730,03	17975,24	18371,35

Tabulka A.38: STREAM.

Virtual Box				
funkce	#1.pokus (MB/s)	#2.pokus (MB/s)	#3.pokus (MB/s)	průměr (MB/s)
Copy	17789,95	17541,88	17812,45	17714,76
Scale	17828,17	18341,70	17977,20	18049,02
Add	17734,62	18051,69	17809,02	17865,11
Triad	17972,16	17915,17	18058,93	17982,08

Tabulka A.39: STREAM.

Vmware Player				
funkce	#1.pokus (MB/s)	#2.pokus (MB/s)	#3.pokus (MB/s)	průměr (MB/s)
Copy	19379,05	19373,85	19015,08	19256,00
Scale	19442,63	19453,99	19173,42	19356,68
Add	19496,37	21398,22	18984,46	19959,68
Triad	19553,12	19547,24	19315,66	19472,01

Tabulka A.40: STREAM.

RAM – Linux

MS virtual PC	
pokus	čas (s)
1.	48,201
2.	46,279
3.	50,020
průměr	48,167

Tabulka A.41: WinRAR.

Virtual Box	
pokus	čas (s)
1.	24,670
2.	24,692
3.	24,804
průměr	24,722

Tabulka A.42: WinRAR.

Vmware Player	
pokus	čas (s)
1.	24,776
2.	24,553
3.	24,553
průměr	24,627

Tabulka A.43: WinRAR.

MS virtual PC				
funkce	#1.pokus (MB/s)	#2.pokus (MB/s)	#3.pokus (MB/s)	průměr (MB/s)
Copy	12432,57	11922,74	12475,80	12277,04
Scale	11878,35	11922,12	11712,67	11837,71
Add	13314,60	13374,01	13389,20	13359,27
Triad	13399,99	13303,84	13574,68	13426,17

Tabulka A.44: STREAM.

Virtual Box				
funkce	#1.pokus (MB/s)	#2.pokus (MB/s)	#3.pokus (MB/s)	průměr (MB/s)
Copy	13278,22	13675,48	12643,07	13198,92
Scale	12810,77	12672,76	11878,14	12453,89
Add	13628,78	12604,92	13506,06	13246,59
Triad	14345,99	14475,45	13449,57	14090,34

Tabulka A.45: STREAM.

Vmware Player				
funkce	#1.pokus (MB/s)	#2.pokus (MB/s)	#3.pokus (MB/s)	průměr (MB/s)
Copy	12139,78	12945,51	13311,12	12798,80
Scale	11900,54	11812,59	13024,43	12245,85
Add	13617,11	13632,86	14563,25	13937,74
Triad	13726,42	13589,83	14634,23	13983,49

Tabulka A.46: STREAM.

LAN – Windows

MS virtual PC	
pokus	Propustnost (Mb/s)
1.	12,4
2.	30,8
3.	32,4
průměr	25,2

Tabulka A.47: Iperf.

Virtual Box	
pokus	Propustnost (Mb/s)
1.	30,4
2.	96,7
3.	31,2
průměr	52,8

Tabulka A.48: Iperf.

Vmware Player	
pokus	Propustnost (Mb/s)
1.	260,0
2.	375,0
3.	379,0
průměr	338,0

Tabulka A.49: Iperf.

MS virtual PC		
pokus	send (Mb/s)	recv (Mb/se)
1.	86,76	1,17
2.	88,62	1,01
3.	88,28	0,93
průměr	87,89	1,04

Tabulka A.50: LANbench.

Virtual Box		
pokus	send (Mb/s)	recv (Mb/s)
1.	193,25	129,54
2.	190,11	157,29
3.	179,78	147,80
průměr	187,71	144,87

Tabulka A.51: LANbench.

Vmware Player		
pokus	send (Mb/s)	recv (Mb/s)
1.	272,02	235,67
2.	273,78	245,19
3.	254,49	212,58
průměr	266,77	231,15

Tabulka A.52: LANbench.

MS virtual PC	
pokus	čas (s)
1.	76,098
2.	72,510
3.	78,965
průměr	75,858

Tabulka A.53: Přenos souborů.

Virtual Box	
pokus	čas (s)
1.	89,806
2.	86,990
3.	73,427
průměr	83,408

Tabulka A.54: Přenos souborů.

Vmware Player	
pokus	čas (s)
1.	69,297
2.	66,107
3.	68,387
průměr	67,930

Tabulka A.55: Přenos souborů.

LAN – Linux

MS virtual PC	
pokus	Propustnost (Mb/s)
1.	67,6
2.	66,4
3.	69,5
průměr	67,8

Tabulka A.56: Iperf.

Virtual Box	
pokus	Propustnost (Mb/s)
1.	137,0
2.	103,0
3.	169,0
průměr	136,3

Tabulka A.57: Iperf.

Vmware Player	
pokus	Propustnost (Mb/s)
1.	591,0
2.	586,0
3.	586,0
průměr	587,7

Tabulka A.58: Iperf.

MS virtual PC	
pokus	Propustnost (Mb/s)
1.	62,742
2.	66,260
3.	63,119
průměr	64,041

Tabulka A.59: nttcp.

Virtual Box	
pokus	Propustnost (Mb/s)
1.	219,792
2.	227,696
3.	223,302
průměr	223,597

Tabulka A.60: nttcp.

Vmware Player	
pokus	Propustnost (Mb/s)
1.	411,736
2.	420,126
3.	409,375
průměr	413,746

Tabulka A.61: nttcp.

MS virtual PC	
pokus	čas (s)
1.	83,206
2.	77,471
3.	80,258
průměr	80,312

Tabulka A.62: Přenos souborů.

Virtual Box	
pokus	čas (s)
1.	36,853
2.	37,399
3.	41,304
průměr	38,519

Tabulka A.63: Přenos souborů.

Vmware Player	
pokus	čas (s)
1.	35,691
2.	33,781
3.	33,124
průměr	34,199

Tabulka A.64: Přenos souborů.

Příloha B

Obsah CD

- Text bakalářské práce