JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH
Ekonomická fakulta
Katedra matematiky a informatiky

Studijní program: B6209 Systémové inženýrství a informatika
Studijní obor: Ekonomická informatika

Porovnání technologií virtualizačních systémů

Bakalářská práce

Vedoucí bakalářské práce: Mgr. Radim Remeš
Autor: Ing. Lukáš Sládek

2012

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH
Fakulta ekonomická
Akademický rok: **2010/2011**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Ing. Lukáš SLÁDEK**
Osobní číslo:	**E10021**
Studijní program:	**B6209 Systémové inženýrství a informatika**
Studijní obor:	**Ekonomická informatika**
Název tématu:	**Porovnání technologií virtualizačních systémů**
Zadávající katedra:	**Katedra aplikované matematiky a informatiky**

Zásady pro vypracování:

V současné době se z důvodu větší efektivity a flexibility v síťovém prostředí firem a institucí stále častěji využívá principu virtualizace, která umožňuje využít fyzický hardware jednoho počítače pro více systémů.
Vypracujte přehled proprietárních i opensource softwarových prostředků pro virtualizaci hardware.
Porovnejte jejich funkce a vlastnosti, uveďte využití těchto softwarových nástrojů. Vhodnou metodou porovnejte jednotlivá řešení.

Metodický postup:

1. Studium odborné literatury.
2. Obecný popis virtualizačních metod.
3. Teoretický popis konkrétních dostupných produktů pro virtualizaci.
4. Porovnání a analýza vybraných produktů, zhodnocení jejich použitelnosti pro nasazení v reálném prostředí.
5. Závěr.

Rozsah grafických prací:

Rozsah pracovní zprávy: **40 - 50 stran**

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

1. **Ruest, D., Ruest, N. Virtualizace: podrobný průvodce. Brno : Computer Press, 2010.**
2. **Tulloch, M., et al. Understanding Microsoft Virtualization Solutions: From the Desktop to the Datacenter. Redmond, WA : Microsoft Press, 2010.**
3. **Introduction to VMware vSphere. Palo Alto, CA : VMware, Inc., 2010.**
4. **Oracle VM VirtualBox: User Manual. Redwood Shores, CA : Oracle Corporation, 2011.**
5. **Xen: Users' Manual Xen v3.3. University of Cambridge, UK : Citrix Systems, Inc., 2008.**

Vedoucí bakalářské práce: **Mgr. Radim Remeš**
Katedra aplikované matematiky a informatiky

Konzultant bakalářské práce: **Ing. Ladislav Beránek, CSc.**
Katedra aplikované matematiky a informatiky

Datum zadání bakalářské práce: **14. února 2011**

Termín odevzdání bakalářské práce: **13. dubna 2012**

doc. Ing. Ladislav Rolínek, Ph.D.
děkan

JIHOČESKÁ UNIVERZITA
V ČESKÝCH BUDĚJOVICÍCH
EKONOMICKÁ FAKULTA
Studentská 13 (26)
370 05 České Budějovice

L.S.

prof. RNDr. Pavel Tlustý, CSc.
vedoucí katedry

V Českých Budějovicích dne 25. března 2011

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47 zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích 31.8.2012 Ing. Lukáš Sládek

Poděkování

Děkuji Mgr. Radimu Remešovi za cenné rady a vedení práce a všem blízkým za podporu při psaní této práce.

Abstrakt

Bakalářská práce se zabývá popisem technologií virtualizačních systémů běžně používaných v IT praxi. Rozebírá základní pojmy, definuje pojem virtualizace. Větší část práce je věnována rozboru principů, na kterých virtualizační systému pracují. Setkáváme se zde s termíny hardwarová virtualizace, softwarová virtualizace, paravirtualizace, hypervisor a dalšími. Všechny tyto pojmy jsou vysvětleny. Následuje popis několika dostupných produktů, které jsou v běžné praxi používány k virtualizaci, včetně jejich architektur a popisu jednotlivých součástí virtualizačních systémů. Podrobněji jsou popsány tři zástupci hardwarové virtualizace – VMware ESXi, Microsoft Hyper-V a Xen Hypervisor a dva zástupci virtualizace softwarové – Oracle VM VirtualBox a VMware Workstation. Těchto pět produktů je v poslední části práce podrobně otestováno a zhodnoceno. Systémy jsou podle celkového hodnocení přibližně stejně výkoné, avšak v dílčích oblastech můžeme nalézt rozdíly.

Klíčová slova

virtualizace, hypervisor, porovnání výkonu, paravirtualizace

Abstract

This bachelor thesis deals with the description of virtualization technology systems commonly used in IT practice. It discusses basic concepts, defines the term virtualization. The larger part is devoted to analysis of the principles on which the virtualization works. We can find here the concept of hardware virtualization, software virtualization, paravirtualization, hypervisor and other. All of these concepts are explained. The following is a description of products that are in common practice used to virtualization, including the description of their architecture and components of virtualization systems. Three representatives of hardware virtualization are described in detail – VMware ESXi, Microsoft Hyper-V, Xen Hypervisor and two representatives of the software virtualization – Oracle VM VirtualBox and VMware Workstation as well. These five products are tested and evaluated in the last part of the thesis. Based on the overall benchmarking are systems approximately the same performance, but in particular tests we can find differences.

Key words

virtualization, hypervisor, benchmarking, paravirtualization

Obsah

Kapitola 1

Úvod

Virtualizace je fenoménem poslední doby. S tímto pojmem se můžeme setkat napříč celým oborem informatiky a v několika úrovních. My se však budeme zabývat pouze virtualizací serverů, počítačů a operačních systémů. Tato virtualizace totiž přináší prostředky a nástroje, jak jeden hardware zpřístupnit pro více virtuálních, neboli hostovaných, počítačů. Ty pak dokáží dokonale využít kapacity fyzického stroje, což je výhodné, neboť alokovat výkon celého stroje pro jedinou aplikaci je neekonomické. Jeden server či počítač nahradí několik jiných, dokáže tak šetřit náklady na IT infrastrukturu.

Virtualizace také přináší nové možnosti využití hardware – pro vývojáře otevírá cesty ke snadnému vytvoření testovacích prostředí pro své projekty, pro správce IT možnost, jak konsolidovat spravovaný hardware a software.

Pomocí virtualizace můžeme vytvářet celé infrastruktury v podnicích a právě díky virtualizaci je daleko snadnější takovou infrastrukturu udržovat a spravovat. Virtualizace tak přispívá k efektivnímu využití prostředků na IT služby. Navíc, s pomocí virtualizace lze snadněji čelit výpadkům a zvýšit tak dostupnost aplikačního prostřední a spolehlivost celého systému.

Nespornou výhodou virtualizace je možnost vytváření různých testovacích a školících prostředí, které je velmi snadné v případě potřeby vrátit do výchozího stavu.

Cílem této práce je představit oblast virtualizace serverů a desktopů, zhodnotit principy fungování softwaru pro virtualizaci a výkonnostně porovnat systémy mezi sebou.

Tato práce se v druhé kapitole zaměřuje na vysvětlení základních pojmů a na popis jednotlivých přístupů k virtualizaci včetně jejich charakteristiky. Třetí kapitola představuje několik vybraných virtualizačních systémů, představuje jejich strukturu a principy, na kterých fungují. Poslední, čtvrtá kapitola se pak snaží odpovědět na otázku, který z těchto virtualizačních systémů je nejvýkonnější pomocí několika měření v testovacích aplikacích. Hlavním cílem práce je právě toto zhodnocení.

Kapitola 2

Virtualizační metody

Nejdříve se podíváme, co vůbec znamená pojem virtualizace a vysvětlíme si další důležité pojmy. Následovat bude popis jednotlivých přístupů k virtualizaci a shrnutí důvodů proč vůbec virtualizaci používat.

2.1 Definice pojmu virtualizace

„Jako virtualizace se v prostředí počítačů označují postupy a techniky, které umožňují k dostupným zdrojům přistupovat jiným způsobem, než jakým fyzicky existují, jsou propojeny atd."[4]. Znamená to tedy, že virtualizované[1] prostředí se lépe přizpůsobuje potřebám uživatelů a snáze se používá. Před uživateli lze zakrývat pro ně nepodstatné detaily (např. rozmístění hardwarových prostředků). Virtualizovat lze na různých úrovních – od celého počítače (tzv. virtuální desktop[2] i server), po jeho jednotlivé hardwarové komponenty (např. virtuální procesory, virtuální paměť atd.), případně pouze softwarové prostředí (virtualizace operačního systému). Všechno závisí pouze na tom, jaké prostředky máme k dispozici na reálném počítači, tedy výkon a vlastnosti procesoru, operační paměť, velikost pevného disku atd.

Virtualizace je v poslední době široce diskutovaná oblast IT technologie. To, co dnes umožňuje proveditelnost virtualizace desktopu nebo serveru, je vysoký výkon hardwaru. Díky němu je možné mezi fyzický hardware a operační systém spolu s běžícími aplikacemi vložit další softwarové vrstvy. Tyto vrstvy umožňují onu virtualizaci prostředků, umožňují více virtuálním počítačů fungovat a využívat prostředků jednoho fyzického stroje. Tato idea se rozšířila nejen na servery, ale i na osobní počítače.

Díky možnosti spouštět více virtuálních počítačů na jednom fyzickém počítači, je tato technologie velice zajímavá pro každého, kdo potřebuje vytvářet, instalovat a znovu použít počítače, na nichž běží v podstatě libovolný operační systém. Právě

[1] Též virtuální.

[2] Pojmem desktop je myšlen osobní počítač. Dále v textu budu používat oba výrazy s identickým významem.

v tom tkví největší hodnota virtualizace.

Při virtualizaci se budeme setkávat s několika základními termíny, které bych zde rád s pomocí obrázku 2.1 nastínil. Některé budou více popsány v dalším textu.

Kruh je oblast ochrany, lepé řečeno zabezpečení. Architektura Intel x86 definuje 4 úrovně ochrany zvané kruhy (angl. rings), které jsou číslovány od 0 (nejvíce privilegovaná vrstva) do 3 (nejméně privilegovaná vrstva), viz obrázek 2.1. Tyto kruhy jsou využívány operačními systémy k ochraně kritické systémové paměti před programovými chybami vzniklými v méně privilegovaných kruzích uživatelských aplikací. Kruh 0 je specifický v tom, že dovoluje přístup k procesoru a dalším zdrojům. Kernely OS běží v kruhu 0, hypervisor v kruhu 0 nebo kruhu 1 a běžné aplikace běží v kruhu 3. Kruh 2 se prakticky nepoužívá.

Hypervisor je vrstva, která je nahrána ještě dříve než samotné jádro systému. Je to prostředník mezi hardwarem a softwarem, resp. hostovaným operačním systémem. Umí zpřístupnit hardware více operačním systémům najednou. Při spuštění (bootu) stroje je hypervisor nahrán do kruhu 0 [2].

Domény Základní doména Dom0 se též nazývá hostitel (angl. host) a lze z něj spouštět další domény, zavírat je, migrovat nebo jim nastavovat parametry. Další domény, které z hostitele spustíte, se označují jako domény U (DomU) a jádra jejich OS běží taktéž v kruhu 1, stejně jako jádro hostitele.

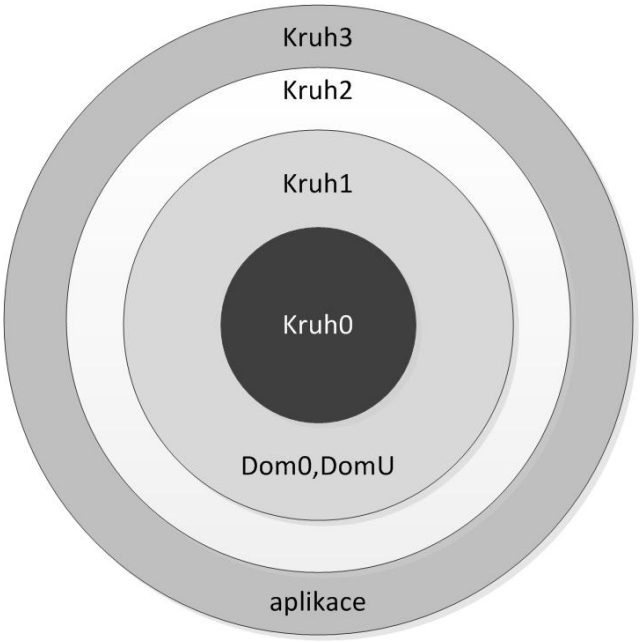
Host (angl. guest) je každý další virtuální stroj nebo systém, který se vytvoří pomocí virtualizačního systému. Je důležité si uvědomit, že české slovo host a anglické slovo host označují dvě odlišné věci [17].

Hostitel (angl. host) je systém, který hostí virtuální stroje.

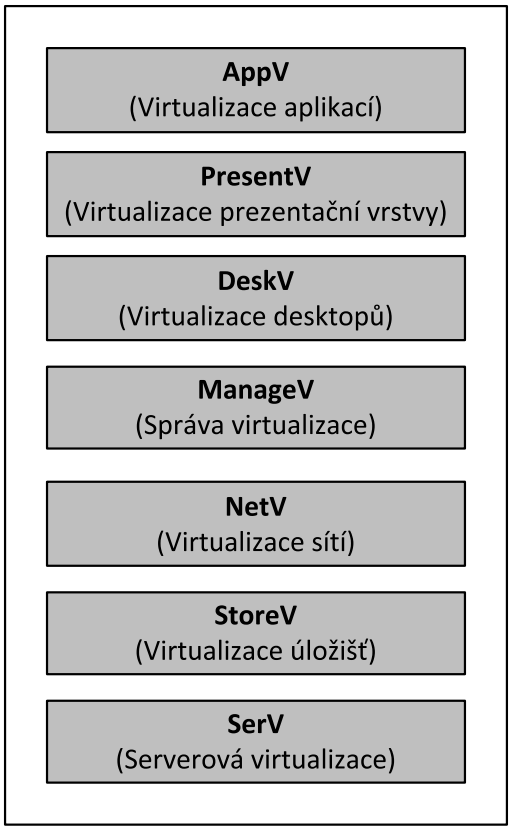
2.2 Model virtualizace

Postupným vývojem virtualizace se dospělo k několika úrovním, kterými se samotné virtualizační nástroje zabývají. Můžeme tedy definovat různé typy virtualizace. Tyto typy ukazuje obrázek 2.2 a dále si je podrobně představíme. Označení vrstev virtualizace včetně popisu bylo převzato z [1]:

- Serverová virtualizace (SerV) – „rozděluje fyzickou instanci operačního systému na virtuální instanci nebo virtuální počítač"[1]. Umožňuje tak virtualizovat libovolný operační systém a to jak 32 bitové, tak i 64 bitové platformy



Obrázek 2.1: Kruhy a domény [17]



Obrázek 2.2: Model virtualizace [1]

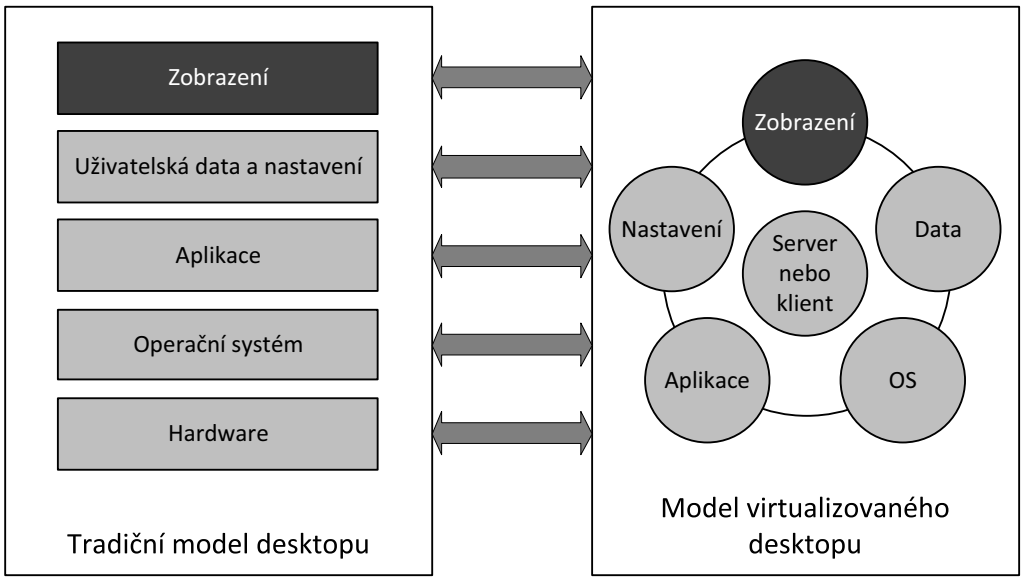
(Windows, Linux, některé formy systému UNIX). Při serverové virtualizaci se hardwarový server stává hostitelem více serverů virtualizovaných. Existují dva pohledy na virtualizaci:

  - Softwarová virtualizace (SoftV) – spouští virtualizovaný operační systém nad softwarovou virtualizační platformou na existujícím operačním systému.
  - Hardwarová virtualizace (HardV) – spouští virtualizovaný operační systém nad softwarovou platformou přímo nad hardwarem bez existujícího operačního systému. Prostředek použitý ke spouštění hardwarové virtualizace se nazývá *hypervisor*. Tento prostředek má za úkol nabídnou prostředky počítače virtualizovaným operačním systémům.

- Virtualizace úložišť (StoreV) – tento prostředek se využívá ke sloučení fyzického úložiště z více zařízení tak, aby se jevilo v systému jako jeden soubor úložišť. Takovýto soubor může nabývat více podob: přímo připojené úložiště (DAS – Direct-attached storage), síťové úložiště (NAS – Network-attached storage) nebo sítě SAN[3] (storage area network) a lze je připojit různými síťovými protokoly: Fibre Channel, Internet SCSI (iSCSI), Fibre Channel over Ethernet[4] nebo prostřednictví souborového systému NFS (Network File System). Virtualizace úložišť umožňuje využití dynamických logických jednotek. Taková jednotka spotřebovává pouze tolik diskového prostoru, kolik aktuálně potřebuje. Je-li vytvořena logická jednotka o velikosti 120 GB, ale využito z ní je pouze 30 GB, na skutečném úložišti spotřebuje právě 30 GB.
- Virtualizace sítí (NetV) – prostředek umožňuje řídit dostupnou šířku přenosového pásma jejím rozdělením na nezávislé kanály, které lze pak přidělit jednotlivým zdrojům. Nejjednodušší formou virtualizace sítí je virtuální lokální síť VLAN. Ta umožňuje logické oddělení fyzické sítě pomocí aktivních síťových prvků.
- Správa virtualizace (ManageV) – správa virtualizačních technologií, rozdělená alespoň do dvou rovin – fondy zdrojů (hardwarové prostředky) a virtuální služby (virtuální počítače nebo servery pro zákazníky). Smyslem je důsledné bezpečností oddělení obou rovin, aby nemohly být vzájemně ovlivněny.

[3]Síť SAN je dedikovaná (oddělená od LAN, WAN atd.) datová síť, která slouží pro připojení externích zařízení k serverům.

[4]FCoE využívá zapouzdření rámců protokolu Fibre Channel do rámců sítí Ethernet. To umožňuje protokolu využívat sítí Ethernet o rychlost 10 Gbps (nebo rychlejších) při zachování využití protokolu Fibre Channel.



Obrázek 2.3: Virtualizace boří vazby mezi vrstvami [2]

- Virtualizace desktopů (DeskV) – umožňuje nasazení virtuálních počítačů spolu s desktopovými operačními systémy. Podrobněji se této úrovni virtualizace věnuje následující kapitola 2.2.1.
- Virtualizace aplikační vrstvy (PresentV) – dříve označována jako terminálové služby. Dnešní využití směřuje do oblasti správy virtualizace, kde se využívají protokoly této služby.
- Virtualizace aplikací (AppV) – využívá podobných principů jako softwarově založená serverová virtualizace, kdy virtualizační aplikace odděluje hostitelský operační systém od spouštěné aplikace.

2.2.1 Přístupy k virtualizaci – desktop

U tradičního modelu desktopu běží nad hardwarem operační systém, ve kterém jsou spouštěny jednotlivé aplikace. Tyto aplikace používají uživatelská data a nastavení (uložená lokálně) a výstup je skrze zobrazovací rozhraní prezentováno uživateli na obrazovce. Dle [2] je pomocí virtualizace toto tradiční uspořádání narušeno a to na všech nebo jen některých vrstvách (viz Obrázek 2.3).

V praxi existuje několik způsobů, jak k virtualizaci přistupovat. Neexistuje metoda virtualizace univerzálně použitelná pro všechny implementace. I když jsou metody rozdílné, spojuje je odlišnost od tradičního modelu desktopu – jak již bylo naznačeno v kapitole 2.1.

Pokud se podíváme na obrázek 2.3, kde zobrazení poběží stále na klientské stanici, můžeme se rozhodovat, zda operační systém (OS) a aplikace poběží na stanici

či na serveru a kde budou uložena uživatelská data a nastavení. Podle toho, která část nebo části jsou virtualizovány, můžeme rozeznat následující způsoby virtualizace dle [2]:

Virtualizace dat a nastavení Jedná se o nejjednodušší variantu virtualizace – na serveru jsou uložena data a nastavení pro jednotlivé uživatele. Uživatel tedy může přistupovat ke svým datům skrze standardizované desktopové prostředí, kde každý počítač pracuje s podobnou sadou aplikací (lokálně nainstalovaných). Tyto data a nastavení aplikací se přenesou na stanici a do aplikací. Uživateli je tak dána možnost pracovat z libovolné stanice. Výhodou je snadná a levná implementace.

Virtualizace aplikací Obvykle jsou aplikace instalovány na každý desktop (klienta) v prostředí operačního systému a vyžadují sdílení lokálních zdrojů, jako jsou runtime knihovny, registry apod. Při virtualizaci aplikací jsou jednotlivé aplikace přímo svázány s prostředky, které potřebují pro svůj běh, a nejsou instalovány na desktop, což zabraňuje potenciálním konfliktům s ostatními aplikacemi a minimalizují se tak zásahy do operačního systému. Hlavním přínosem je uživatelská flexibilita – aplikace není svázána s konkrétním počítačem, ale s konkrétním uživatelem. To umožňuje snížení nákladů za licence.

Relační virtualizace Tato virtualizace je odvozena od architektury tenkých klientů[5]. V tomto modelu se jednotliví klienti připojují k *jedné* instanci operačního systému běžící na jednom serveru. V podstatě všechny aplikace a celý operační server běží pouze na tomto jediném serveru a pouze zobrazení je přenášeno k uživatelům na (tenkého) klienta. Přínosem tohoto způsobu virtualizace je centralizace prostředků (včetně úložišť) na jediném serveru. Z toho sice vyplývá větší náročnost na spolehlivost těchto prostředků, ale je výrazně usnadněna správa a údržba.

Virtuální desktop Jedná se o rozšíření myšlenky tenkého klienta. Na rozdíl od předchozí varianty, má každý uživatel k dispozici *samostatnou* instanci operačního systému běžící na serveru jako virtuální stroj. Tento přístup dokáže pokrýt různorodé potřeby uživatelů. Jelikož se jedná o rozšíření předchozí varianty, jsou i stejné výhody, avšak zde při vyšších nárocích na prostředky a úložiště serveru.

Single-desktop virtuální stroj V tomto modelu běží virtualizační software přímo na desktopu. Tento model nevyžaduje síťové připojení, protože virtualizační

[5]Tenký klient obsahuje v podstatě pouze síťové rozhraní a grafickou kartu.

Typ virtualizace	Flexibilita	Řízení	Bezpečnost	Dostupnost	Offline přístup	Výkon
Virtualizace dat a nastavení	○	○	✕	✓	✓	✓
Virtualizace aplikací	✓	○	○	○	✓	✓
Relační virtualizace	○	✓	○	✓	✕	○
Virtuální desktop	○	✓	○	✓	✕	○
Blade-based virtuální desktop	○	○	○	○	✕	✓
Single-desktop virtuální stroj	✓	○	○	○	✓	○

✓– plná podpora, ○ – částečná podpora, ✕ – nepodporováno

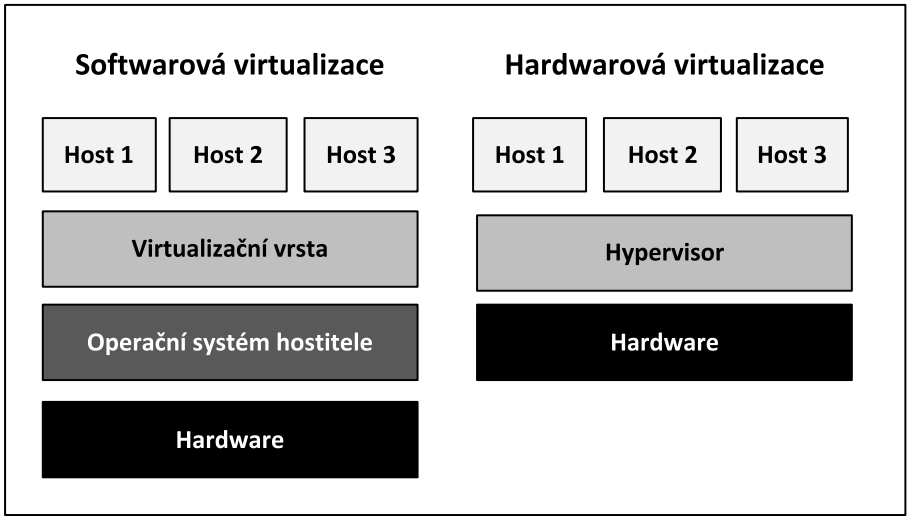
Tabulka 2.1: Výhody jednotlivých typů virtualizací [2]

software běží na stejném zařízení jako operační systém desktopu.Tento druh virtualizace je velmi flexibilní pro jednotlivého uživatele. Umožňuje vývojářům vytvořit testovací prostředí na pracovní stanici, nebo oddělit pracovní a soukromé prostředky. Dále zde existuje možnost centrální distribuce přednastaveného virtuálního počítače, např. pro běh starších aplikací. Tento princip využívá např. Windows XP mode obsažený ve Windows 7 (viz kapitola 3.2.3). Jedná se v podstatě o softwarovou virtualizaci serverů popsanou v kapitole 2.2.2.

Přehled výhod a vlastností jednotlivých přístupů ukazuje tabulka 2.1.

2.2.2 Přístupy k virtualizaci – server

V kapitole 2.2 bylo uvedeno, že existují dva modely serverové virtualizace – a to hardwarový (HardV) a softwarový model (SoftV). Druhý jmenovaný, *softwarový model*, je méně robustní a méně výkonné řešení, neboť spoléhá na jednodušší nástroje pro virtualizaci. Tento přístup vyžaduje základní operační systém hostitele, který samozřejmě spotřebovává zdroje serveru a proto ovlivňuje provoz virtuálních počítačů běžících nad ním. Právě proto je toto řešení méně výkoné a spíše vhodné pro fázi vývoje nebo testování, jak je uvedeno v [1]. Jednotlivé vrstvy modelu znázorňuje obrázek 2.4. To, že mezi hardwarem a virtuálními stroji běží hostitelský operační systém sebou nese řadu nepříjemností, např. proces aktualizace tohoto operačního systému, často spojený s jeho restartováním, což vede i k restartování virtuálních počítačů. Roli hypervisoru zde tedy plní virtualizační vrstva. Virtualizační vrstva zachytává všechny V/V operace hostovaných systémů a předává je operačnímu systému k vykonání a zpět předává výsledky operací.



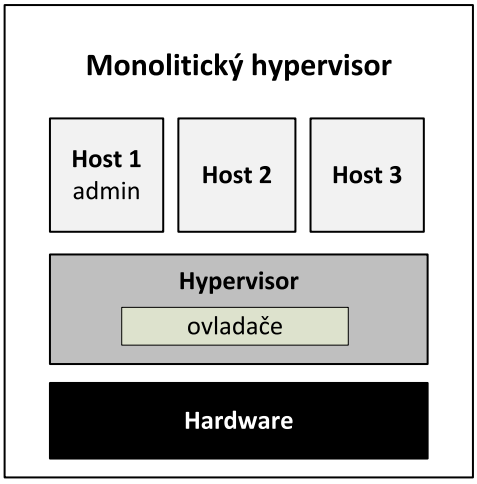
Obrázek 2.4: Modely serverové virtualizace [1]

Tento způsob virtualizace je též nazýván *plná virtualizace* [3]. Tento přístup je také často využíván při virtualizaci desktopů, používají ho komerční nástroje, jako je VirtualBox, VMware Workstation nebo Microsoft Virtual PC.

Tyto nedostatky odstraňuje druhý model – *hardwarová virtualizace*, též Hardware Virtual Machine (HVM). V tomto případě je kód hypervisoru spuštěn přímo na hardwaru a nepotřebuje tak ke svému běhu hostitelský operační systém. Hardwarová virtualizace nevyžaduje upravovat hostovaný operační systém, jako je to u jiných modelů. Tato vrstva tak jednoduše zpřístupní hardware hostitelského serveru pro jednotlivé virtuální stroje. Tyto virtuální stroje běží nad vrstvou hypervisoru. Hypervisor může být uložen přímo ve firmware serveru nebo např. na flash paměti. Obvykle se jedná o velmi malé objemy dat. Další výhodou je, že tento kód vrstvy hypervisoru není potřeba tak často „záplatovat" a aktualizovat jako operační systém a proto je minimalizováno ovlivnění virtuálních počítačů v důsledku aktualizací. Jelikož se jedná o hardwarovou virtualizaci, je nutná podpora ze strany hardwaru, zejména procesoru a základní desky. Můžeme tak hardwarovou virtualizaci označit jako hardware-assisted virtualizaci. Bez této podpory ze strany hardwaru není možné virtuální stroje spustit. U společnosti Intel se jedná o technologii Intel VT-x, u AMD pak o technologii AMD-V. Nástroje pro virtualizaci serverů, které využívají tento model, jsou např. MS Virtual Server, VMware Server. Podrobněji jsou tyto nástroje popsány v kapitole 3.

Jistým mezičlánkem mezi softwarovou a hardwarovou virtualizací je *paravirtualizace*. Paravirtualizace je využívána tam, kde není zajištěna podpora virtualizace ze strany procesoru. Je to technika, kdy hypervisor poskytuje API[6] a hostovaný

[6]API (Application Programming Interface) označuje rozhraní pro programování aplikací. Jde o sbírku procedur, funkcí či tříd nějaké knihovny, jiného programu nebo jádra operačního systému, které může využívat jiný program.



Obrázek 2.5: Monolitický hypervisor [21]

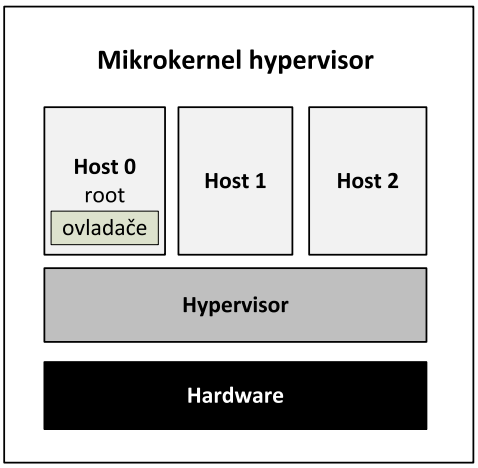
operační systém jej následně využívá. To přináší nutnost modifikovat jádro hosta i hostitele umístěného v doméně Dom0 i DomU, jinak není operační systém schopen toto API využít. Této metody využívá open-source projekt Xen, VMware ji řeší pomocí ovladačů. Jádro hosta je možné modifikovat pouze u otevřených operačních systémů.

Následují kapitoly popisují dva různé návrhy hypervisoru – monolitickou a mikrokernel architekturu hypervisoru.

2.2.2.1 Monolitický hypervisor

Návrh monolitického hypervisoru vyžaduje použití ovladačů pro hardware kompatibilních s daným hypervisorem. Ovladače jsou umístěny přímo ve vrstvě hypervisoru a jsou jím i spravovány. Strukturu ukazuje obrázek 2.5.

Integrovaný návrh přináší nesporné výhody, avšak i některé nedostatky. Například, monolitický hypervisor nepotřebuje ovládací (administrátorský) operační systém, protože všechny hostované systémy komunikují přímo se základní fyzickou vrstvou hardwaru hostitelského počítače za využití ovladačů integrovaných do hypervisoru. Toto je nesporná výhoda návrhu. Na druhou stranu, skutečnost, že ovladače musí být speciálně vyvíjeny pro daný hypervisor vytváří významné překážky, neboť existuje na trhu mnoho typů základních desek, řadičů úložišť, síťových adaptérů a dalšího hardwaru. Výsledkem je, že distributoři monolitických hypervisorů jsou nuceni úzce spolupracovat s výrobci hardware a zajistit tak pro svůj hypervisor dostatečnou podporu ovladačů hardware. To také znamená, že jsou vývojáři hypervisorů závislí na výrobcích hardware, kteří vydávají ovladače podporující daný hypervisor. Z výše uvedeného vyplývá, že počet zařízení, které mohou být využity s tímto typem hypervisoru, je více omezený, než kdyby dané hostované operační systémy běžely přímo na fyzickém počítači.



Obrázek 2.6: Mikrokernel hypervisor [21]

Je důležité zmínit jeden podstatný fakt tohoto návrhu – ignoruje jednu z nejdůležitějších bezpečnostních zásad: ochranu do hloubky. S ochranou do hloubky je definováno několik vrstev jako prevence útoků. Podrobněji jsou vrstvy popsány v kapitole 2.1. V tomto modelu není žádná ochrana do hloubky, protože všechny prvky běží v nejvíce privilegované vrstvě systému [21]. Tohoto monolitického systému využívá např. VMware ESXi Server.

2.2.2.2 Mikrokernel hypervisor

Mikrokernel hypervisor nevyžaduje speciální ovladače určené pro daný hypervisor. To je způsobeno faktem, že tento hypervisor obsahuje operační systém v root (kořenovém) oddílu. Tento root oddíl poskytuje spustitelné prostředí potřebné pro ovladače zařízení nižší fyzické vrstvy hostitelského systému. Pojem oddíl ve virtualizační terminologii představuje jednotku izolovanou od hypervisoru, pro kterou je alokována určitá paměť a určitý počet procesorů. Jsou dva typy oddílů:

- Root nebo také rodičovský oddíl, je řídící oddíl, ve kterém běží virtualizační software. Je to také oddíl, kterému je umožněn přístup k hardware, a který určuje přidělené zdroje ostatním (podřízeným) oddílům.
- Hostovaný oddíl nebo také oddíl potomka je jakýkoliv oddíl vytvořený pomocí root oddílu. Hostované operační systému a jeho aplikace běží v tomto oddílu.

V modelu mikrokernel hypervisoru je potřeba nainstalovat ovladače pro fyzický hardware pouze v operačním systému běžícím v root oddílu. Není třeba instalovat ovladače pro fyzický hardware v jednotlivých hostovaných systémech v podřízených oddílech. Je to dáno tím, že pokud požaduje tento hostovaný systém přístup

k fyzickému hardware systému, provede požadavek pomocí komunikace s operačním systémem běžícím v root oddílu. Jinými slovy, hostovaný operační systém nemá přímý přístup k hardwaru počítače, pouze komunikuje skrze hostitelský operační systém v root oddílu. Strukturu modelu ukazuje obrázek 2.6.

Mikrokernel hypervisor má několik výhod oproti monolitickému. Za prvé, jelikož mikrokernel hypervisor nepotřebuje speciální ovladače, vystačí si s existujícími ovladači od výrobců hardware. Za druhé, protože ovladače nejsou součástí hypervisoru, je hypervisor méně zatížený, což znamená, že je menší a spolehlivější. Třetí a pravděpodobně nejdůležitější výhodou je, že jsou minimalizovány mezery, kterými může dojít k útoku, neboť cizí kód (v tomto případě ovladač, který je programován třetí stranou a nikoliv producentem hypervisoru) není implementován v hypervisoru. Riziko infikace hypervisoru škodlivým kódem je tak sníženo na minimum a nehrozí ztráta kontroly nad všemi virtuálními stroji běžícími v hostitelském systému.

Jedinou nevýhodou mikrokernel návrhu je potřeba speciálního (root) oddílu. To přidává měřitelnou, ale obvykle minimální, zátěž z důvodu potřebné komunikace mezi root oddílem a podřízeným oddílem, která je potřebná pro přístup k systémovému hardwaru [21].

2.3 Důvody pro využití virtualizace

1. *Vytvoření dynamického datového centra*: Virtualizace dovoluje alokovat zdroje přesně tam, kde a kdy jsou potřeba z celkového objemu dostupných zdrojů. To vše může probíhat automaticky, bez interakce správce systému. Výsledkem je, že budeme mít systém s celkově menším objemem zdrojů, které jsou však dynamicky a efektivně rozdělovány mezi aplikace.

2. *Snížení spotřeby energie a náročnosti chlazení*: Efektivní distribucí zdrojů můžeme docílit nižšího instalovaného výkonu hardwaru, a tím i značného snížení spotřeby elektrické energie a snížení nároků na chlazení systémů. Další výhodou je menší prostorová náročnost. Ukažme si to na příkladu. Jeden systém s osmijádrovým procesorem dokáže obsloužit (v závislosti na náročnosti hostovaných aplikací) až 80 virtuálních serverů/počítačů při spotřebě cca 1600 W při plné zátěži. To je v přepočtu 20 W na jeden virtuální stroj. Toto číslo je velmi vzdáleno spotřebě 200 W reálného (fyzického) stroje s dvoujádrovým procesorem při minimálním zatížení. Stejných poměrů bychom se dopočítali i u nároků na chlazení. Jeden takový server pro hostování virtuálních strojů přitom zabere 4U pozice v racku[7] oproti 80 1U dvoujádrových systémům [3].

[7]Rack je standardizovaný systém umožňující přehlednou montáž a propojování různých elek-

3. *Lepší bezpečnost*: Virtualizace umožňuje oddělit citlivá firemní data od dat uživatelů, stejně jako oddělit data uživatele od dat ostatních uživatelů i v případě, že sdílejí stejný fyzický počítač.

4. *Spouštění Legacy software*[8] *na novém hardwaru*: Nevhodnou, náročnou a nákladnou cestou, jak využívat legacy programy, je udržování starého hardwaru. Mnohem spolehlivější a levnější je využít služeb virtualizace a s její pomocí virtualizovat legacy hardware, který umožní běh potřebného operačního systému a následně i legacy softwaru.

5. *Snadné testování a vývoj*: Softwaroví vývojáři mohou při testování softwaru vytvořit virtuální prostředí pro běh různorodých operačních systémů nebo dokonce pro běh různých verzí operačního systému. Podobně může být ve firemním prostředí při přechodu na novější softwarovou výbavu část infrastruktury virtualizována a testováno její chování před upgradem celého systému.

6. *Běh rozličných operačních systémů na stejném hardware*: Mnoho společností využívá více operačních systémů, virtualizace umožňuje běh těchto operačních systému na stejném hardware.

7. *Zlepšení škálovatelnosti*: Virtualizace umožňuje snadné rozšíření stávající infrastruktury podniku.

8. *Zvýšení využití hardwaru*: Zabýváme se pouze celkovým požadovaným výkonem, nikoliv požadavky jednotlivých serverů nebo aplikací. Je nežádoucí, aby byl stroj využit pouze na zlomek svého výkonu. Virtualizace dokáže rozdělit zdroje tak, že jsou plně využity.

9. *Snadný upgrade*: Upgrade softwaru nebo operačního systému v podniku může být nelehký úkol. Virtualizace umožní rozdělení tohoto úkolu na dílčí úlohy, které jsou snadněji implementovány a ověřovány v praxi.

10. *Řízení výpadků*: Při řízených výpadcích (odstávkách) systémů je možné ve virtualizovaném prostředí přesunout zdroje pro pokrytí těchto výpadků. S migračními nástroji lze dokonce virtuální stroje přesouvat spolu s uživatelskými přístupy dynamicky mezi reálnými systémy. Podobně lze řešit i neplánované výpadky.

---

trických a elektronických zařízení.

[8]Legacy systém je starý počítačový systém, technologie nebo program, který je stále používán (typicky protože vyhovuje potřebám uživatele), i když jsou dostupné novější a efektivnější technologie, systémy.

Virtualizace s sebou přináší i výhody ekonomické – dokáže šetřit náklady. Většina dnes provozovaných serverů je jen velmi málo vytížena, přibližně do 10%. Každý z těchto serverů potřebuje při svém běhu mnoho energie a prostoru. Při správné konfiguraci lze na stejném fyzickém serveru spustit více než 10 virtuálních počítačů, serverů nebo desktopů – žádný z těchto strojů nepotřebuje žádně další elektrické napájení nebo místo a přitom nabízí stejné služby jako fyzické počítače.

Každý virtuální počítač není ničím jiným než množinou souborů někde na disku. Když se vezme fyzická instance serveru a převede se na instanci virtuální [9], tedy převedení fyzického počítače na množinu souborů ve složce. Jakmile bude v tomto stavu, lze jej přesunout z jednoho serveru na jiný server, vypnout, restartovat, přejít do režimu spánku a v podstatě s ním dělat vše, co šlo dříve, a to bez jakéhokoli významného snížení výkonu [1].

Tato kapitola nám shrnula základní přístupy k virtualizací serverů a desktopů. Následovat bude představení prostředků, které k virtualizaci slouží.

[9] P2V konverzi – Physical-to-Virtual Migration za pomocí nejrůznějších nástrojů.

Kapitola 3

Popis a analýza dostupných produktů

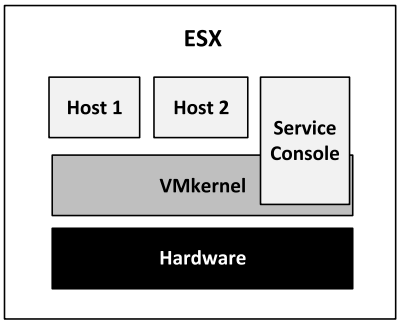
Následující kapitola představuje výběr virtualizačních systémů dostupných na trhu. U jednotlivých produktů je popsána jejich struktura, architektura, komponenty a principy, na kterých systém a komponenty pracují. Kapitola je rozdělena na dvě hlavní části – část popisující hypervisory a část popisující desktopové virtualizační systémy.

3.1 Hypervisory

Hypervisory představují prostředky hardwarové virtualizace představené v kapitole 2.2.2. Jejich společným znakem je instalace na holý hardware.

3.1.1 VMware vSphere

Systém VMware vSphere, který je pokračováním virtualizačních produktů VMware předchozích generací, je robustnější, více škálovatelný a spolehlivější serverový virtualizační produkt [13]. Součástí systému je řada nástrojů, které umožňují řízení



Obrázek 3.1: Architektura ESX [13]

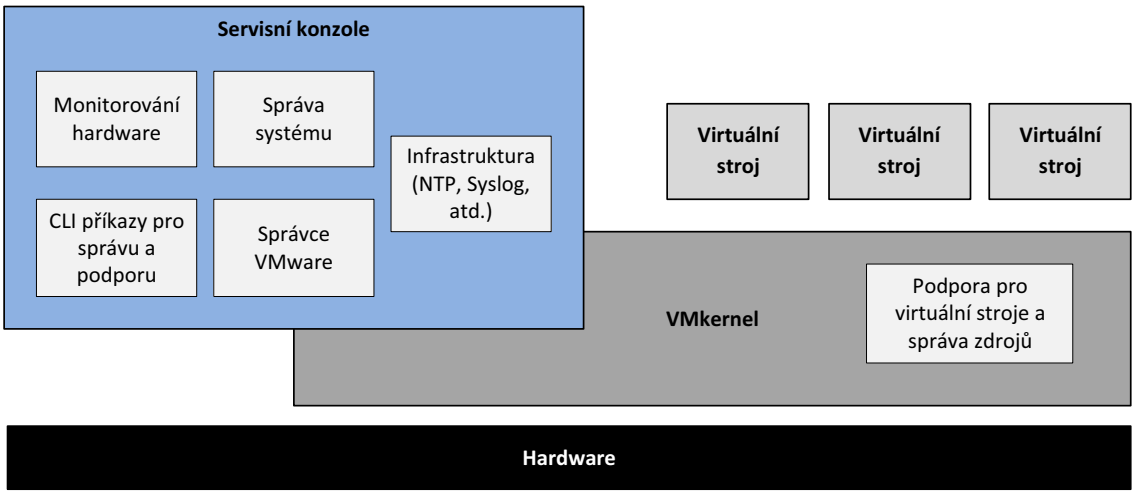
dynamických procesů, zvyšují dostupnost, odolnost proti chybám, zajišťují správu distribuovaných prostředků a zálohování. Správci tak mají k dispozici veškeré nástroje, které jsou nutné k provozu podnikového prostředí od několika až po tisíce serverů. Jedná se o produkty a nástroje:

- VMware ESX a ESXi,
- VMware Virtual Symmetric Multi-Processing,
- VMware vCenter Server,
- VMware vCenter Update Manager,
- VMware vSphere client,
- VMware VMotion a Storage VMotion,
- VMware Distributed Resource Scheduler,
- VMware High Availability,
- VMware Fault Tolerance,
- VMware Consolidated Backup,
- VMware vShield Zones,
- VMware vCenter Orchestrator.

Jedná se o produktovou řadu určenou pro hardwarovou virtualizaci serveru. Jádrem řady je hypervisor, tvořící virtualizační vrstvu pro ostatní produkty vSphere, VMware ESX a VMware ESXi. Oba hypervisory obsahují stejné nástroje, podporují stejné funkce a oba se instalují přímo na hardware. Základem ESX jsou dvě vzájemně se doplňující komponenty Servisní konzole a VMkernel. Komponenta Servisní konzole je založená na operačním systému Linux a poskytuje služby, které jsou dostupné v běžných operačních systémech (např. firewall, agenty SNMP[1] nebo webový server) a které jsou zároveň potřebné jako podpora virtualizace. Druhá komponenta, VMkernel, je skutečným základem virtualizačního procesu. „Prostřednictvím plánování CPU, správy paměti a virtuálního přepínání zpracování dat zajišťuje přístup virtuálních počítačů k základnímu fyzickému hardwaru."[13] Servisní konzole zajišťuje přístup k VMkernel (viz Obrázek 3.1).

Oproti tomu, ESXi je hypervisor pracující bez komponenty Servisní konzole a vystačí si tak s velmi malým paměťovým prostorem a přesto nabízí plnohodnotný

[1]Simple Network Management Protocol – nástroj umožňující průběžný sběr nejrůznějších dat pro potřeby správy sítě a jejich následné vyhodnocování.



Obrázek 3.2: Architektura VMware ESX [11]

funkční rozsah [13]. Veškeré ovladače pro hardware a V/V zásobník jsou implementovány přímo ve vrstvě hypervisoru VMkernel.

Hypervisory byly krátce představeni, nyní je popíši podrobněji. O ostatních komponentách je možno získat více informací v [11, 13].

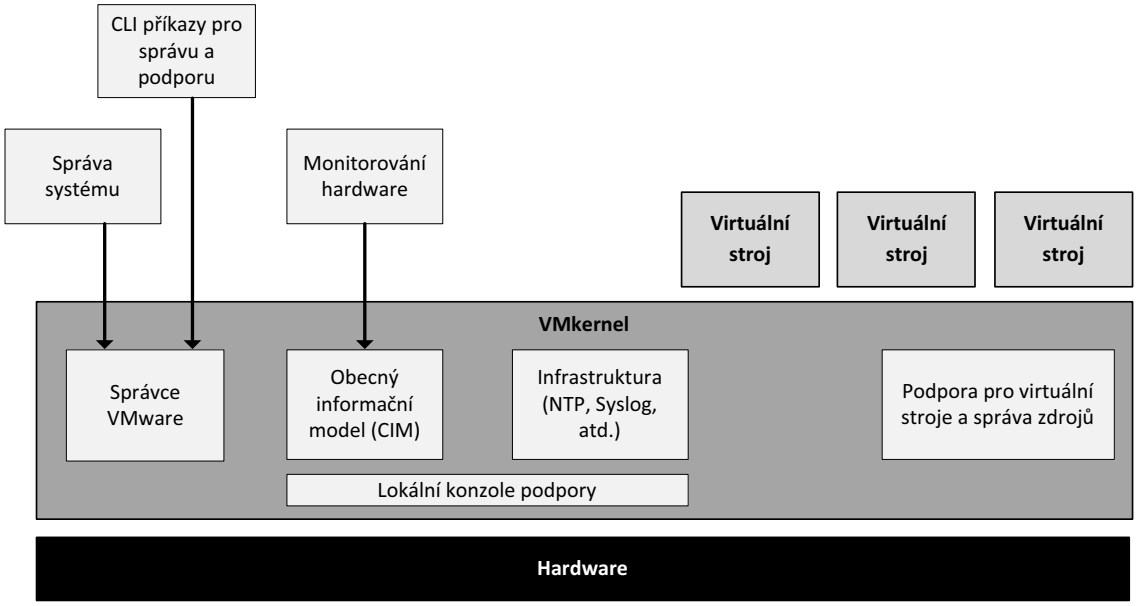
3.1.1.1 Porovnání hypervisorů ESXi a ESX

Nejdříve bych rád porovnal oba typy hypervisorů, které jsou ve vSphere obsaženy. Poté podrobněji popíši ESXi.

Architektura VMware ESX V této původní ESX architektuře je virtualizační jádro (označeno jako VMkernel) rozšířeno o oddíl správy označovaný jako konzole operačního systému (COS nebo servisní konzole). Primárním úkolem servisní konzole je poskytovat rozhraní pro správu. V servisní konzoli jsou umístěny nejrůznější agenti[2] VMware správy dohromady s dalšími agenty (logování, služby času atd.). Tato architektura umožňuje nasazení ostatních agentů od třetích firem a umožňuje tak rozšíření funkcionality o další služby jako je monitorování hardware či správu systému. Mimoto umožňuje servisní konzole přihlášení administrátorům a spouštění konfiguračních a diagnostických příkazů a skriptů. Architektura je naznačena na obrázku 3.2.

Architektura VMware ESXi U architektury ESXi byla servisní konzole (COS) odebrána a všichni VMware agenti jsou spouštěni přímo ve VMkernelu. Služby infrastruktury jsou poskytovány nativně pomocí modulů obsažených ve VMkernelu. Stejně tak mohou být spuštěny ve VMkernelu moduly autorizovaných třetích stran,

[2]Agent je samostatná entita umístěná do určitého prostředí, která může vykonávat nějakou činnost.



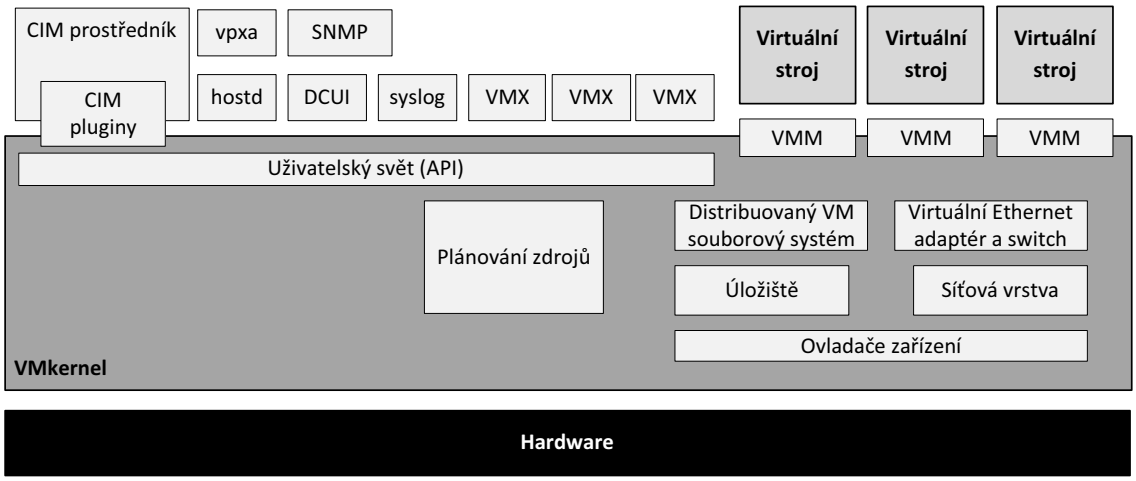
Obrázek 3.3: Architektura VMware ESXi [11]

jako jsou ovladače hardwaru nebo komponenty monitorování hardwaru. Pouze moduly, které byly digitálně podepsány společností VMware, mohou být použity v systému a tvoří tak pevně uzavřenou architekturu. Ochrana před spuštěním libovolného kódu na ESXi hostiteli výrazně zvyšuje bezpečnost celého systému. Tuto architekturu znázorňuje obrázek 3.3.

3.1.1.2 VMware ESXi architektura

„VMware ESXi přináší nový základ pro tvorbu infrastruktury. Tato inovativní architektura pracuje nezávisle na jakémkoliv operačním systému, nabízí vylepšenou bezpečnost, zvyšuje spolehlivost a zjednodušuje správu."[14] Tato kompaktní architektura je navržena pro integraci přímo do optimalizovaného hardwaru pro virtualizaci, umožňuje rychlou instalaci, konfiguraci a nasazení. Funkčně je ESXi ekvivalentní hypervisoru ESX, poskytující stejnou úroveň výkonu a škálovatelnosti. A to i přes to, že byla odebrána servisní konzole a zredukována potřebná paměť pro instalaci až na hranici 32 MB [14]. Funkce servisní konzoly je nahrazena novým vzdáleným rozhraním příkazového řádku (CLI) se zachováním standardů správy systému. ESXi podporuje všechny nástroje z produktové řady vSphere [14].

Komponenty ESXi ESXi je tvořeno základním operačním systémem VMkernel a procesy, které běží nad ním. VMkernel poskytuje prostředky pro běh všech procesů v systému, včetně aplikací pro správu, agentů a stejně tak hostovaných virtuálních strojů. Ovládá veškerý hardware serveru, spravuje zdroje pro všechny aplikace. Hlavní procesy, které běží nad VMkernelem jsou podle [14]:



Obrázek 3.4: Detailní architektura VMware ESXi [14]

- Přímé uživatelské rozhraní (Direct Console User Interface – DCUI) – nízkoúrovňové rozhraní pro správu a konfiguraci dostupné skrze konzole serveru, primárně určené pro počáteční základní nastavení.
- Monitor virtuálního stroje (Virtual Machine Monitor – VMM) – je proces, který poskytuje pro každý virtuální stroj exekuční prostředí, stejně jako pomocný proces označovaný jako VMX. Každý spuštěný virtuální stroj má k dispozici vlastní VMM a VMX proces.
- Nejrůznější agenti používaní pro VMware správce infrastruktury vyšší úrovně pomocí vzdálených aplikací.
- Obecný informační model (Common Information Model – CIM) – jedná se o rozhraní, které umožňuje správu systému na úrovni hardware ze vzdálených aplikací pomocí standardních API.

Obrázek 3.4 podrobněji ukazuje architekturu ESXi.

VMkernel VMkernel je POSIX[3] operační systém vyvinutý společností VMware a poskytující řadu funkcí, které můžeme najít i v ostatních operačních systémech, jako je řízení procesů, souborový systém, signály a řízení vláken. Je speciálně navržen tak, aby podporoval současný běh více virtuálních strojů a poskytoval základní funkcionality jako je:

- plánování zdrojů

[3]POSIX (zkratka z Portable Operating System Interface) je v informatice označení standardu používaného hlavně unixovými operačními systémy. Jeho úkolem je vytvořit jednotné rozhraní, které má zajistit přenositelnost programů (aplikací) mezi různými hardwarovými platformami. Definuje rozhraní nejen pro programátory (tzv. API), ale i pro uživatele (v podobě nástrojů pro příkazový řádek).

- V/V zásobník
- ovladače zařízení

Souborový systém VMkernel používá jednoduchého souborového systému k uložení konfiguračních souborů, log souborů a aplikovaných opravných balíčků. Struktura souborového systému je navržena shodně se servisní konzolí u ESX. Tento souborový systém je nezávislý na VMware VMFS souborovém systému použitého pro ukládání virtuálních strojů. VMware VMFS úložiště může být vytvořeno na místním disku hostitelského systému nebo na sdíleném úložišti. Pokud je využito pouze externí úložiště, pak ESXi nevyžaduje použití lokálního disku. Využitím bezdiskové instalace je zvýšena spolehlivost a sníženy nároky na napájení a chlazení. Vzdálená správa umožňuje správu souborů jak na vnitřní paměti, tak i na VMFS úložišti. Jelikož vnitřní paměť je obvykle závislá na napětí, umožňuje ESXi uchování logů na nakonfigurovaném syslog serveru i při výpadku napájení.

Uživatelé a skupiny Uživatelé i skupiny mohou být v ESXi systému definovány lokálně. Poskytují způsob, jak odlišit uživatele v přístupu k systému pomocí klienta virtuální infrastruktury (VIC) nebo vzdáleného rozhraní příkazového řádku. Skupiny mohou být využity ke kombinování více uživatelů stejně jako v jiných operačních systémech. V systému je několik předdefinovaných uživatelů a skupin. Administrátorská práva mohou být nastaveny pro individuální uživatele či skupiny uživatelů.

Uživatelský svět Pojem uživatelský svět (angl. user world) označuje procesy běžící ve VMkernel operačním systému. Prostředí, ve kterém uživatelský svět běží, je omezeno ve srovnání s tím, co lze nalézt v systému kompatibilním s POSIX (např. v Linuxu):

- sada dostupných signálů je limitována,
- systémové API je podskupinou POSIXu,
- velmi omezený `/proc` souborový systém[4],
- jednotný swap soubor je dostupný pro všechny procesy v uživatelském světě. Pokud existuje lokální disk, je swap soubor automaticky vytvořen v malém VFAT oddílu. Dále je možné uživatelsky definovat umístěni swap souboru na připojeném VMFS úložišti.

[4] /procfs (nebo také /proc souborový systém) je speciální souborový systém používaný v UNIX operačních systémech k prezentaci informací o procesech a dalších systémových informací v hierarchické formě podobně jako souborová struktura.

Stručně řečeno, uživatelský svět není určen jako univerzální prostředí pro spuštění libovolné aplikace, ale nabízí dostatek prostoru pro procesy, které je nutné spouštět přímo v hypervisoru.

Přímé uživatelské rozhraní DCUI DCUI je lokální uživatelské rozhraní, které je zobrazeno pouze v konzoli ESXi systému. Nabízí rozhraní s menu podobné BIOSu a umožňuje interakci se systémem. Jeho hlavním úkolem je provedení počáteční konfigurace a případné řešení problémů. Úkoly DCUI jsou následující:

- nastavení administrátorského hesla,
- nastavení síťového adaptéru, pokud nebylo provedeno pomocí DHCP,
- provedení jednoduchého síťového testu,
- prohlížení logů,
- restartování agentů,
- obnovení výchozího nastavení.

Záměrem je, aby uživatel prováděl pouze minimum konfiguračních úkonů pomocí DCUI, tedy pouze tu úvodní, případně při řešení problémů. Následná konfigurace i administrace, se provádí pomocí vzdálené správy, za pomoci nástrojů, jako je například VI klient, VirtualCenter nebo vzdáleného rozhraní příkazového řádku. Přístup přes všechny tyto kanály je chráněn administrátorským (root) heslem.

Další procesy uživatelského světa Agenti používaní pro implementaci některých administrátorských funkcích byli přeneseni ze servisní konzole do uživatelského světa:

- `hostd` – poskytuje programové rozhraní k VMkernel a je využíváno přímo VI klientem, stejně jako VI API. Jedná se o proces, který ověřuje uživatele a udržuje záznamy, kteří uživatelé nebo skupiny mají jaká práva. Dále dovoluje vytvářet a spravovat lokální uživatele.
- `vpxa` – je to proces, který se využívá pro připojení k VirtualCenteru.
- `syslog` – je-li nastaveno vzdálené logování, agent odesílá všechny logy na vzdálený cíl (syslog server).

Kromě toho, ESXi obsahuje procesy, které umožňují NTP časovou synchronizaci a SNMP monitoring.

Obecný informační model (CIM) Obecný informační model (Common Information Model – CIM) je otevřený standard, který definuje, jak mohou být počítačové zdroje reprezentovány a spravovány. To umožňuje frameworku[5] monitorování hardware na základě standardů a bez použití agentů. Tento framework se skládá z CIM správce objektů, často nazývaného CIM prostředník (CIM broker) a ze sady CIM poskytovatelů. CIM poskytovatelé jsou využíváni jako mechanismy k poskytnutí přístupu k ovladačům zařízení, a tedy hardwarem pod nimi. Výrobci hardware mohou naprogramovat poskytovatele, kteří umožní monitorování a správu svých dílčích zařízení. VMware vydává poskytovatele, kteří slouží k monitorování hardwaru serveru, ESX/ESXi úložišť a dalších specifických virtualizačních zdrojů. Tito poskytovatelé běží uvnitř ESXi systému, a proto jsou extrémně nenároční a zaměření na specifický administrativní úkol. Poskytovatelé jsou umístěni přímo v systémovém obraze a není je možná upravovat za chodu. CIM prostředník sbírá data od všech CIM poskytovatelů a prezentuje je dále mimo uživatelský svět pomocí standardního API.

VI API VMware Virtual Infrastructure API poskytuje mocné rozhraní pro vývoj aplikací k integraci s VMware infrastrukturou. VI API umožňuje komunikaci s webovou službou VirtualCenteru a tím spravovat a ovládat ESX/ESXi hypervisor. Toto rozhraní také využívají pro správu VI klient a vzdálený příkazový řádek i již zmiňovaný VirtualCenter.

Spolu s CIM standardem poskytuje VI API komplexní způsob, jak spravovat ESXi systém, a to jak vzdáleně, tak i centrálně. Výhodou těchto vzdálených nástrojů pro správu je, že není nutno aktualizovat a upravovat agenty správy po každé úpravě nebo aktualizaci systému.

Komplexnější popis nejen těchto komponent je možné nalézt v [14].

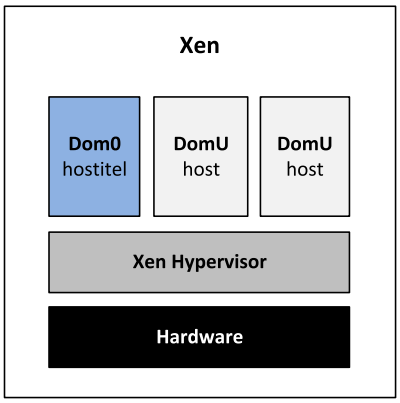
3.1.2 Xen Hypervisor

Xen Hypervisor je hypervisor poskytující rozhraní pro virtualizaci hardwaru a běh více operačních systémů na jednom počítači současně. První verze Xenu vznikla v laboratořích na univerzitě v Cambridge. Od roku 2010 je Xen komunitním projektem publikovaným pod licencí GPL v2. Xen se vyvíjí pro platformy x86, x86-64, Itanium, PowerPC a architekturu ARM [16]. Umožňuje běh rozličných operačních systému jako je Linux, NetBSD, FreeBSD, Solaris, Windows, a dalších běžných operačních systémů nad vrstvou hypervisoru jako hostovaný operační systém.

Systém se spuštěním Xen Hypervisorem obsahuje následující komponenty:

- samotný Xen Hypervisor

[5]Framework je programové rozhraní.



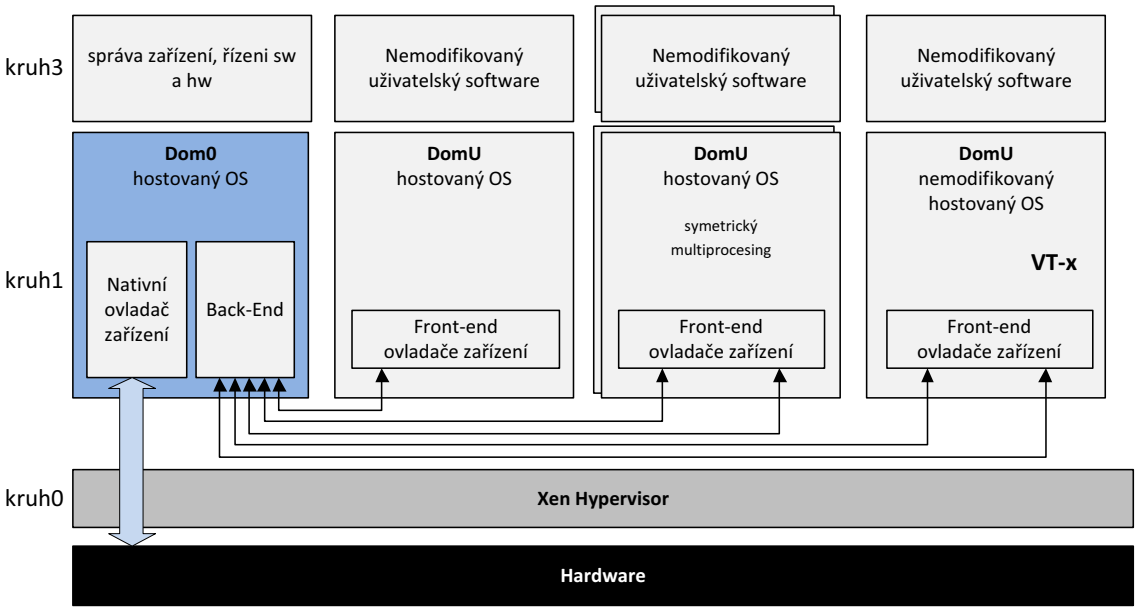
Obrázek 3.5: Struktura Xen [16]

- jednu privilegovanou doménu Dom0 – privilegovaný host běžící nad hypervisorem mající přímý přístup k hardware a možnost spravovat hypervisor
- jednu nebo více domén DomU – doména určená pro běh hostovaného operačního systému bez přímého přístupu k hardwaru; hypervisor zprostředkovává přístup k hardware pomocí virtuálních CPU, disků apod.

Xen hypervisor běží přímo na hardwaru a vytváří rozhraní pro všechny požadavky na hardware (CPU, V/V, disk atd.) pro hostovaný operační systém. Oddělením hostovaných OS od hardware umožňuje Xen hypervisor spouštění více operačních systémů bezpečně a nezávisle.

Doména 0 je označována jako Dom0. Je spouštěna hypervisorem během počáteční inicializace systému a může obsahoval libovolný operační systém s výjimkou Windows. Dom0 má unikátní práva (privilegia) přístupu k vrstvě Xen hypervisoru, která nejsou přenositelná na jakoukoliv jinou doménu. Tato práva umožňují spravovat ostatní domény, jejich parametry, spouštět je nebo je ukončovat, stejně tak umožňuje spravovat samotný hypervisor nebo celý počítačový systém.

Doména hosta označována jako DomU nebo také neprivilegovaná doména. Je spouštěna a ovládána doménou Dom0 a pracuje nezávisle na systému. Jako host může být spouštěn modifikovaný operační systém s využitím paravirtualizace (viz 2.2.2) nebo nemodifikovaný operační systém vyžadující podporu virtualizace ze strany hardware (viz kapitola 2.2.2) označovanou jako Hardware Virtual Machine (HVM). Typickým operačním systémem, který vyžaduje HVM je MS Windows [16].



Obrázek 3.6: Architektura Xen hypervisoru [20]

Bezpečnost a spolehlivost Kritickým aspektem tvorby hypervisoru je zajištění toho, že řešení bude bezpečné, obzvláště je-li toto řešení nasazené v podniku či prostředí cloudu. Xen zajišťuje vysokou úroveň bezpečnosti různými metodami:

- Izolace hostů – každý host v doméně DomU je izolován od ostatních domén DomU a není možné žádným způsobem získat přístup k paměti nebo síťovému připojení jiných DomU. Hosté komunikují pouze s doménou Dom0 (viz obrázek 3.6).
- Privilegovaný přístup – pouze doména Dom0 nebo jeden host určený k ovládání mají možnost komunikovat s hardwarem skrze hypervisor.
- Izolace operačních systémů – oddělením hypervisoru od operačního systému znamená, že hypervisor nemůže být využit k útoku na operační systém (Xen nemůže útočit na hostitelský operační systém, jelikož zde žádný k útoku není).
- Malá kódová základna – Xen hypervisor má malý kódový otisk (angl. footprint), což omezuje oblast útoku.

Celková struktura systému je znázorněna na obrázku 3.6. Doména Dom0 je vytvořena během bootování a má povoleno využívat řídící rozhraní. Tato inicializační doména je zodpovědná za hostování softwaru pro správu v aplikační vrstvě. Kontrolní rozhraní poskytuje možnost vytvářet i ukončovat ostatní domény a kontrolovat jejich přiřazené zdroje, alokovanou fyzickou paměť a přístup k fyzickým diskům stroje a síťovým rozhraním. Zároveň je možné vytvářet nebo mazat virtuální síťová rozhraní a virtuální bloky zařízení (je to zařízení, pomocí něhož host přistupuje k fyzickému

disku). Tato virtuální V/V zařízení mají přiřazeny informace o přístupových právech, která určují, které domény jsou přístupné a s jakými restrikcemi [18].

Domény hostovaných virtuálních strojů, tedy DomU komunikují přes rozhraní front-end ovladačů (vyvíjených pro daný systém) s privilegovanou doménou Dom0 a jejím back-end rozhraním. Dom0 tyto požadavky předává přes nativní ovladače pro instalovaný hardware a vrstvu hypervisoru. Je-li hostován otevřený operační systém, tedy systém jehož kód je možné upravit, je možno využít paravirtualizace (viz kapitola 2.2.2). Pokud tomu tak není, je nutné využít podpory HVM (viz kapitola 2.2.2) ze strany výrobce hardwaru. Taková doména DomU pak komunikuje stejným způsobem, jako doména DomU s upravitelným kódem hostovaného operačního systému, (viz schéma na obrázku 3.6), avšak je požadována hardwarová podpora virtualizace.

3.1.2.1 Oracle VM

Na Xen Hypervisoru je založen Oracle VM. Jedná se o virtualizační systém pro virtualizaci serverů a podporu snadnějšího nasazení podnikových aplikací, zjednodušení životního cyklu aplikací a správy. Obsahuje plnou podporu přenosu aplikací z fyzických na virtuální servery.

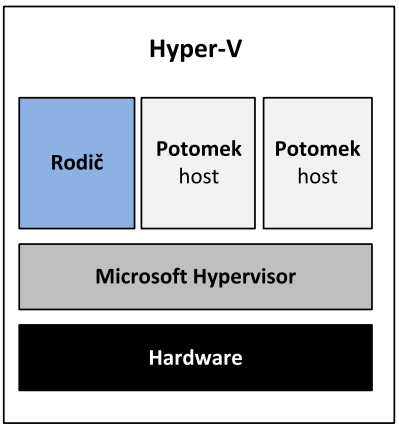
Oracle VM je složen z Oracle VM Server for x86 a Oracle VM Manageru. Oracle VM server for x86 dokáže pracovat až s 160 fyzickými CPU a 2 TB paměti. Pro každý virtuální stroj lze přiřadit až 128 virtuálních CPU a 1 TB paměti. Podpora je zaručena pro všechny x86 operační systémy všech výrobců na trhu. Samozřejmostí je podpora širokého spektra hardwaru, což usnadňuje integraci na stávající infrastrukturu.

Oracle VM Manager poskytuje jednoduché centralizované administrátorské prostředí pro konfiguraci a provoz serverů, sítí a úložišť na bázi webového prostředí. Prostředí tak přístupné odkudkoliv. Pomocí správce lze vytvářet pravidla stejně jako kopírovat, sdílet, nastavovat, bootovat či migrovat virtuální stroje [8].

3.1.2.2 Citrix XenServer

Citrix XenServer je kompletní podniková serverová virtualizační platforma postavená na Xen hypervisoru. XenServer je navrhnut pro efektivní správu Windows a Linux virtuálních serverů, podporuje cloud, obsahuje všechny potřebné nástroje pro správu virtuální infrastruktury včetně dynamické.

Volná edice XenServer přináší 64-bitový hypervisor, centralizovanou správu, nástroj pro migraci a konverzní nástroje pro vytvoření virtuální platformy a maximalizaci výkonu hostovaných systémů. Placené verze pak přinášejí podporu pro organizace libovolné velikosti, integrační a automatizované procesy správy a řešení pro virtuální datové centrum.



Obrázek 3.7: Zjednodušená architektura Hyper-V [21]

3.1.3 Microsoft Hyper-V

Hyper-V je hypervisor z dílny společnosti Microsoft. Zjednodušeně můžeme jeho architekturu představit na obrázku 3.7. Jak je z obrázku 3.7 vidět, serverový systém Hyper-V se skládá z Microsoft Hypervisoru běžícího na holém hardwaru, z čehož vyplývá, že se jedná o hardwarovou (HardV) virtualizaci. Nad vrstvou hypervisoru běží jeden rodičovský[6] (root) oddíl a jeden nebo více oddílů potomků (hostů). Jedná se tedy o mikrokernel architekturu (viz kapitola 2.2.2.2). Hyper-V s touto architekturou tak využívá ochranu do hloubky – hypervisor je minimalistický prvek a největší část funkcionality tak přináší běžící operační systém v rodičovském oddíle.

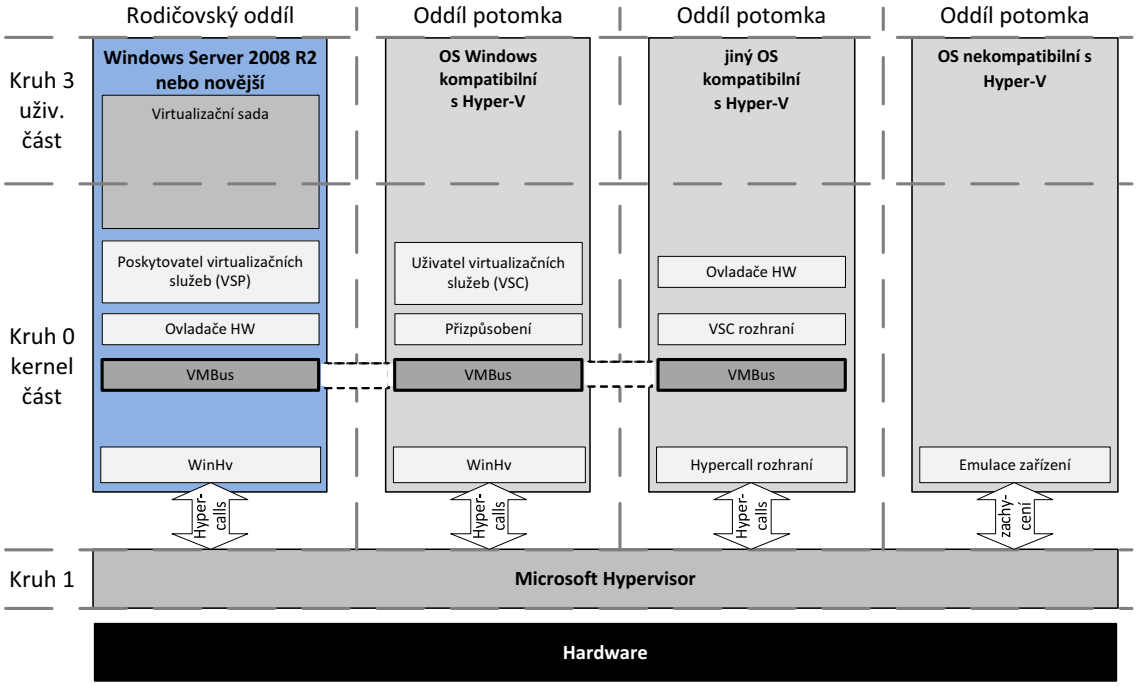
V implementaci hardwarové virtualizace a mikrokernel jádra společnosti Microsoft běží v rodičovském oddíle Windows Server 2008 nebo novější ve Standard, Enterprise nebo Datacenter edici. Oddíly komunikují s hypervisorem pomocí hypercall rozhraní. Hypercall je API, které umožňuje hostovanému operačními systému využít optimalizací, které nabízí hypervisor [21]. Celá struktura je na obrázku 3.8. V dalším textu si představíme jednotlivé oddíly a jejich součásti.

3.1.3.1 Rodičovský oddíl

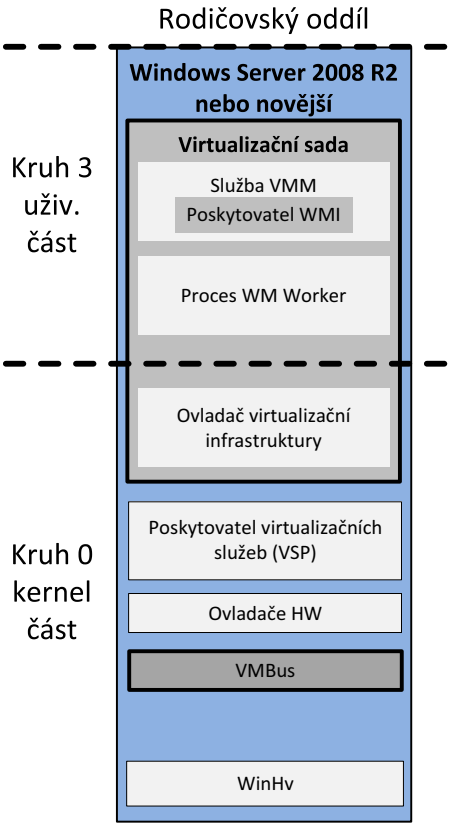
Rodičovský oddíl obsahuje řadu speciálních komponent, které nejsou obsaženy v oddílu potomka. Obrázek 3.9 podrobněji ukazuje rozličné komponenty rodičovského oddílu včetně vyznačení, v kterém kruhu se nacházejí. Uživatelské procesy jsou spouštěny v kruhu 3, kernel procesy pak v kruhu 0. Rodičovský oddíl je první oddíl, který je v systému vytvořen po startu hypervisoru, slouží pro běh instance operačního systému Windows Server 2008 R2 nebo novějšího a umožňuje tak hypervisoru Hyper-V plnit roli serveru. Rodičovský oddíl má v Hyper-V následující úkoly:

- je použit pro vytváření a správu dalších oddílů (potomků) v systému a obsahuje

[6]Pojmy *rodič* a *potomek* vycházejí z terminologie firmy Microsoft převzaté z [21].



Obrázek 3.8: Architektura Hyper-V [21]



Obrázek 3.9: Komponenty rodičovského oddílu [21]

WMI poskytovatele (viz dále), který poskytuje rozhraní pro vzdálenou správu,

- spravuje a přiřazuje hardware kromě procesorového času a alokované paměti, které jsou využívány hypervisorem,
- hardwarové zdroje rodičovského oddílu jsou sdíleny nebo alokovány pro využití jedním nebo více oddíly potomků,
- ovládá správu napájení, operace plug & play, zaznamenává chyby hardware (pokud nastaly).

Virtualizační sada Virtualizační komponenty umístěné v rodičovském oddíle souhrnně nazýváme virtualizační sada. Virtualizační sada je spuštěna v tomto oddíle a má přímý přístup k hardwaru hostitelského systému skrze mechanismy, které popíši později. V Hyper-V implementaci HardV modelu má virtualizační sada tyto komponenty:

- VMM – Virtual Machine Management Service – služba správy virtuálního systému,
- VM worker – Virtual Machine Worker Process,
- virtuální zařízení,
- ovladač virtualizační infrastruktury,
- knihovna rozhraní Windows Hypervisor.

Ostatní komponenty obsažené v rodičovském oddíle jsou následující:

- VSP – Virtualization Service Providers – poskytovatel virtualizačních služeb,
- VMBus – Virtual Machine Bus.

Následující odstavce detailně popisují jednotlivé výše uvedené komponenty.

Služba správy virtuálního systému Služba správy virtuálního systému (služba VMM nebo také VMMS) zodpovídá za správu stavu všech virtuální strojů v oddílech potomků. To znamená správu zastavených nebo offline virtuálních strojů, řízení vytváření snapshotů[7] a řízení přidávání nebo odebírání zařízení. Jakmile je virtuální stroj v oddíle potomka spuštěn, VMMS vytvoří nový Virtual Machine worker (VM worker) proces, který je využit k vykonávání správy pro tento spuštěný virtuální stroj.

[7] V počítačových systémech je snapshot (čes. snímek) stav systému v daném časovém okamžiku [26].

VMMS také řídí, které operace mohou být vykonány na virtuálním stroji v daném stavu. Například pokud je mazán snapshot virtuálního stroje, VMMS brání v jeho užití (více v [21] s. 81). Konkrétně, VMMS spravuje následující stavy virtuálního stroje:

- starting – spouštění,
- active – aktivní,
- not active – neaktivní,
- taking snapshot – snímání snapshotu,
- applying snapshot – aplikace snapshotu,
- deleting snapshot – mazání snapshotu,
- merging disk – spojení disk.

Operace za běhu virtuálního stroje (tzv. online operace) – jako je Pauza, Uložení, Vypnutí – nejsou spravovány pomocí VMMS. Místo ní jsou spravovány procesem VM Worker. VMMS je implementována v kruhu 3 i v kruhu 0 jako systémová služba a má, mimo jiné, návaznost na WMI (Windows Management Instrumentation) služby. Podrobnosti lze zjistit v [21], s. 30.

Proces Virtual Machine Worker Proces WMW je uživatelský proces spuštěný v instanci operačního systému v rodičovském oddílu. Poskytuje služby správy virtuálních strojů běžících v oddílech potomků. VMMS vytváří oddělené WM worker procesy pro každý spuštěný virtuální stroj, aby je navzájem oddělil od sebe. To má tu výhodu, že selže-li VM worker proces, je postižen pouze daný virtuální stroj, nikoliv ostatní nebo dokonce všechny virtuální stroje. VM worker proces spravuje následují aspekty přiřazeného virtuálního stroje:

- vytváření, konfiguraci a běh virtuálního stroje,
- pozastavení a pokračování běhu virtuálního stroje,
- uložení a obnovu virtuálního stroje,
- vytváření snapshotů virtuálního stroje.

Vedle toho, proces VM worker obsahuje virtuální základní desku (Virtual Motherboard – VMB). VMB zpřístupňuje hostovaným systémům jako oddělená zařízení paměť, IRQ a mapování paměti a V/V. VMB je také zodpovědná za správu virtuálních zařízení, která jsou popsána v následujícím textu.

Virtuální zařízení Virtuální zařízení (Virtual Devices – VDevs) jsou softwarové moduly, které poskytují konfiguraci zařízení pro oddíly potomků. VMB obsahuje základní sadu virtuálních zařízení včetně PCI sběrnice a chipsetu, stejná, která byla na základních deskách s Intel 440BX chipsetem. Tento typ byl zvolen kvůli široké kompatibilitě. Existují dva typy virtuálních zařízení:

- Core VDevs – tato virtuální zařízení modelují existující hardware a jsou dostupná pro každý virtuální stroj. Typicky jsou využívána v situacích, kde je kladen důraz na kompatibilitu a proto může již existující software, jako je BIOS nebo ovladače, fungovat bez úprav. Core virtuální zařízení dělíme na následující:
  - Emulované zařízení – tato virtuální zařízení emulují specifický hardware, jako např. VESA grafická karta. Většina Core virtuálních zařízení je právě emulovaná, jako například BIOS, DMA, APIC, ISA sběrnice, PCI sběrnice, správa napájení, RTC, sériový řadič, Speaker, PS/2 klávesnice a myš, síťová karta, řadič disket, řadič pevných disků, VGA grafická karta.
  - Syntetická zařízení – jsou to virtuální zařízení, která nemodelují konkrétní hardware. Syntetická zařízení jsou dostupná pouze hostovaným operačním systémům, které podporují integrační služby. Jedná se např. o virtuální grafickou kartu.
- Plug-in VDevs – tato virtuální zařízení nemodelují existují hardware, ale jsou využita k vytvoření, konfiguraci a správě poskytovatelů virtualizačních služeb (VSP, viz dále) běžících v rodičovském oddílu. Je to část, která ovládá hardware. Plug-iny virtuálních zařízení umožňují přímou komunikaci mezi rodičovským oddílem a potomky pomocí VMBus.

Ovladač virtualizační infrastruktury Ovladač běží v kernel části virtualizační sady. Poskytuje služby správy oddílu, virtuálních procesorů a paměti pro všechny oddíly potomků. Dále umožňuje uživatelským komponentám virtualizační sady komunikovat s hypervisorem.

Knihovna rozhraní Windows Hypervisor Jedná se o dynamickou knihovnu (WinHv) v kernel části, která se nahrává spolu s Windows Server instancí běžící v rodičovském oddíle a spolu s hostovaným operačním systémem v jakémkoliv oddíle potomka (pokud je hostovaný operační systém Hyper-V kompatibilní). Umožňuje ovladačům operačního systému volat hypervisor standardním voláním systému Windows.

Poskytovatel virtualizačních služeb Poskytovatelé virtualizačních služeb (VSP) jsou hostováni v rodičovském oddílu. Umožňují distribuci služeb zařízení pro potomky pomocí V/V nástrojů skrze klienty virtualizačních služeb (VSC) běžících v jednotlivých oddílech potomků. VSP v tomto vztahu představuje server a VSC klienta. Veškerá komunikace mezi nimi probíhá pomocí VMBus.

VMBus Sběrnice VMBus (Virtual Machine Bus) je logická, kanálová sběrnice, umožňující komunikaci mezi rodičovským oddílem a potomky. Úkolem VMBus je poskytnout vysokorychlostní, vysoce optimalizovaný komunikační nástroj mezi virtualizovanými oddíly. Předčí tak jiné, pomalé, vysoko režijní a emulované způsoby komunikace.[8]

Hostovaný operační systém, který nepodporuje Integrační služby, je tedy Hyper-V nekompatibilní a musí využít emulace. To znamená, že hypervisor musí intervenovat zachycením volání fyzického hardwaru z tohoto hostovaného OS a přesměrovat je na emulované zařízení, která běží v rámci VM worker procesu v rodičovském oddíle. Emulace vyžaduje daleko více režijního provozu než komunikace pomocí VMBus.

VSP běžící v rodičovském oddílu komunikující skrze VMBus obsluhuje požadavky na zařízení. Ty jsou mu předávány od VSC běžícího v oddílu potomka opět skrze VMBus. Tím je dána vysoká rychlost obsloužení volání fyzického hardware. Tato komunikace je pro hostovaný operační systém zcela transparentní [21].

3.1.3.2 Oddíl potomka

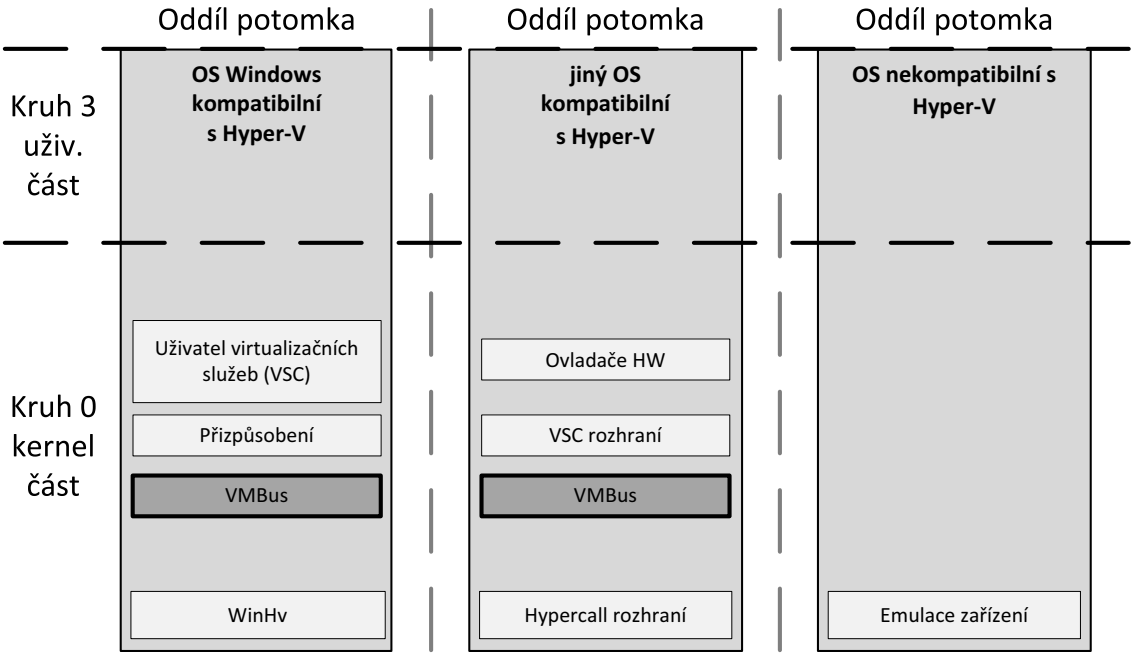
Jak ukazuje obrázek 3.10, v Hyper-V jsou implementovány tři typy oddílů potomků:

- oddíl potomka hostující operační systém Windows kompatibilní s Hyper-V,
- oddíl potomka hostující operační systém kompatibilní s Hyper-V,
- oddíl potomka hostující operační systém nekompatibilní s Hyper-V, ať už Windows nebo jiný OS.

Oddíl potomka hostující operační systém Windows kompatibilní s Hyper-V Jak zobrazuje obrázek 3.10, oddíl s OS Windows podporující Hyper-V obsahuje v kernel části následující virtualizační komponenty:

- Klient virtualizačních služeb (VSC) je syntetické zařízení umístěné v oddíle potomka a využívající hardwarových zdrojů poskytovaných VSP[9] v rodičovském oddíle pomocí komunikace přes VMBus[10]. VSC je automaticky zpřístupněn po

[8] [21] s. 32
[9] viz kapitola 3.1.3.1
[10] viz kapitola 3.1.3.1



Obrázek 3.10: Oddíly potomků Hyper-V [21]

instalaci integračních služeb v oddíle potomka a umožní tak používání syntetických zařízení. Bez těchto integračních služeb může oddíl potomka využívat pouze emulovaná zařízení, jak ukazuje obrázek 3.10 v pravé části.

- Přizpůsobení – tento termín odkazuje na modifikace provedené v kódu operačního systému, aby byl kompatibilní s Hyper-V. To znamená, že umí detekovat svůj běh v prostředí hypervisoru nebo obecněji ve virtualizovaném prostředí a dokáže tak pracovat efektivněji. Hyper-V umožňuje efektivnější komunikace s úložištěm, síťovými prostředky, grafickým a vstupním podsystémem.

Oddíl potomka hostující operační systém kompatibilní s Hyper-V Tento oddíl je znázorněn na obrázku 3.10 uprostřed. Je to takový oddíl potomka, který hostuje jiný operační systém než Windows a využívá VSC třetích stran ke komunikaci přes VMBus s VSP v rodičovském oddílu a odbavuje tak požadavky na přístup k hardwaru. Tyto VSC jsou instalovány v rámci instalace integračních služeb. Integrační služby také poskytují komponenty, které umožní oddílu potomka komunikovat s ostatními oddíly a hypervisorem. Dále poskytují oddílu potomka následující funkce:

- Heartbeat – využívá se k ověření, že oddíl potomka reaguje na požadavky z rodičovského oddílu.
- Key\Value Pair Exchange – uchovává páry klíčů vyměněných mezi rodičovským oddílem a potomkem (využití pro nástroje správy).

- Synchronizace času – synchronizace času mezi rodičovským oddílem a oddílem potomka.
- Vypnutí – Umožňuje potomkovi odpovědět na požadavek vypnutí z rodičovského oddílu.
- Služba stínové kopie svazku – spolupracuje se stejnou službou v rodičovském oddílu a usnadňuje konzistentní zálohu dat.

Hyper-V obsahuje integrační služby pro většinu moderních Windows operačních systémů počínaje Windows XP SP3 a to jak ve 32 bitové, tak i 64 bitové verzi. Microsoft dokonce vyvinul integrační komponenty pro Linux. Jsou šířeny pod GPL v2 licencí.

Oddíl potomka hostující operační systém nekompatibilní s Hyper-V Tento oddíl ukazuje pravá část obrázku 3.10. V oddílu potomka běží operační systém (ať už Microsoft Windows nebo jakýkoliv jiný), který není kompatibilní s Hyper-V, jinými slovy, není pro něj možno nainstalovat integrační služby. V důsledku to znamená, že hostovaný operační systém musí místo syntetických zařízení využívat zařízení emulovaná, což má za následek snížení výkonu [21].

3.1.4 KVM

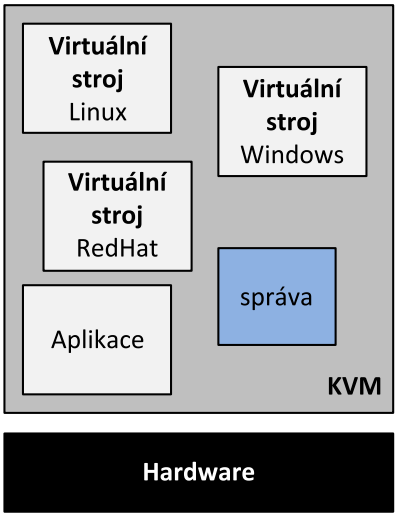
KVM (zkratka pro Kernel-based Virtual Machine) je kompletní virtualizační open-source řešení pro Linux a x86 hardware obsahující podporu virtualizace (technologie Intel VT-x nebo AMD-V). Obsahuje kernel modul, který poskytuje samotnou virtualizační strukturu, dále specifický modul pro daný typ procesoru (resp. podporu virtualizace). Poslední součástí je QEMU[11] – emulátor strojů. Ten umožňuje běh programu určeného pro jinou platformu (např. ARM) na jiné platformě (např. x86). Použitím těchto prostředků umožňuje KVM spouštět virtualizované nemodifikované operační systémy Linux i Windows. Každý virtuální stroj má přidělené vlastní prostředky [22].

Struktura KVM hypervisoru je na obrázku 3.11. KVM hostí každý virtuální stroj jako jakýkoliv jiný Linux proces, proto každý hostovaný virtuální stroj může využívat všechny dostupné funkce linuxového jádra, jako je přístup k hardwaru a úložišti, bezpečnostní prvky a aplikace [23].

3.2 Virtualizační systémy – desktop

Jedná se o prostředky softwarové virtualizace (kapitola 2.2.2) či virtualizace desktopů (kapitola 2.2.1). Často tyto nástroje slouží vývojářům pro testování jejich apli-

[11] QEMU je univerzální a open-source emulátor procesorů [24].



Obrázek 3.11: Struktura KVM [23]

kaci na dalších operačních systémech.

3.2.1 Oracle VM VirtualBox

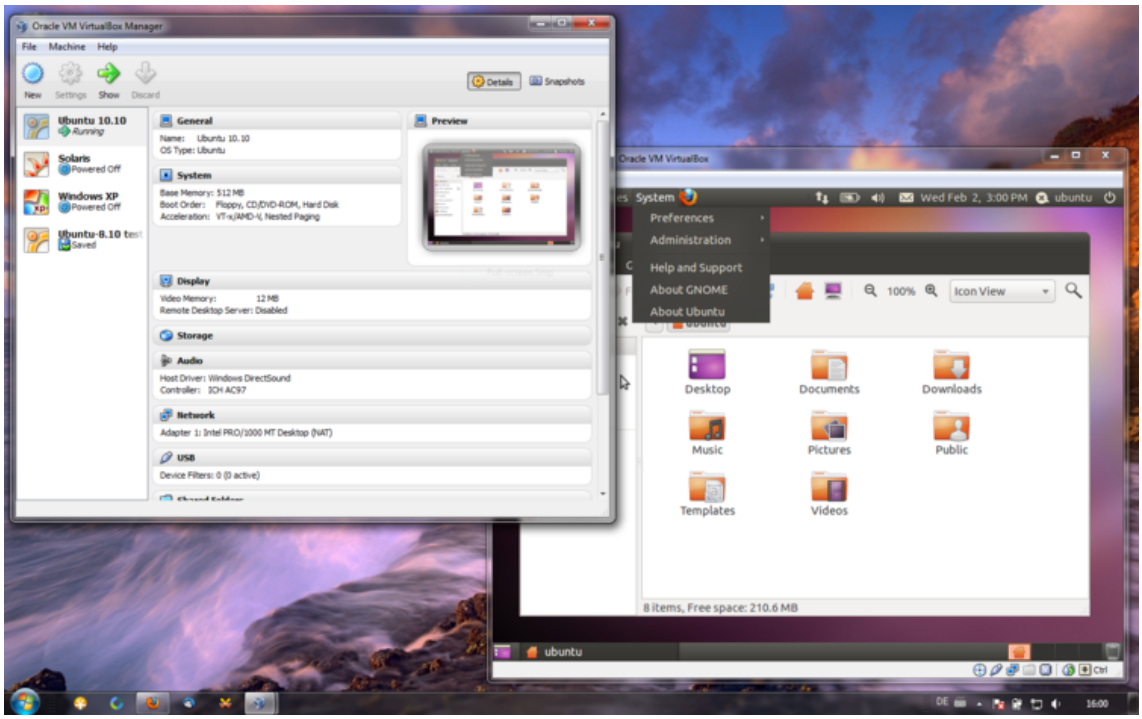
Oracle VM VirtualBox lze charakterizovat jako bezplatný, částečně otevřený, mnoha funkcemi vybavený nástroj pro virtualizaci, i když to tak na první pohled nevypadá. Díky těmto vlastnostem se jedná o velmi rozšířený a oblíbený nástroj. Vyvíjí ho společnost Oracle.

VirtualBox je multiplatformní virtualizační aplikací. To znamená, že ji lze instalovat na systém s procesorem Intel nebo AMD a nezáleží, jestli na tomto systému běží operační systém Windows, Mac OS, Linux nebo Solaris. VirtualBox umožňuje takovýto systém rozšířit o podporu běhu více operačních systémů současně a to nejen čtveřice výše uvedených. Lze tedy spustit například Windows a Linux na systému Mac OS, spustit Windows Server 2008 na Linux serveru, spustit Linux na Windows atd. Jediným omezením v počtu běžících systémů je velikost diskové paměti a operační paměti.

VirtualBox nalézá uplatnění kdekoliv, od malých embedded systémů nebo desktopů až po datová centra a cloud prostředí [5]. Obrázek 3.2.1 ukazuje běh operačního systému Ubuntu Linux na Windows 7.

VirtualBox vytváří pro potřeby tvorby virtuálního počítače virtuální procesor, generickou grafickou kartu, virtuální síťovou kartu a další potřebný „hardware". Co se týče výkonnosti a rychlosti patří VirtualBox ke špičce a směle se může poměřovat s placeným VMware [5]. Práce s ním je jednoduchá a intuitivní, dokáže využívat periferní zařízení přes USB, virtuální BIOS umožňuje zavádění systémů z USB médií.

Díky své částečné otevřenosti lze programovat vlastní zásuvné moduly, na kterých se podílí i široká komunita uživatelů. Samozřejmostí je lokalizace do češtiny.



Obrázek 3.12: VirtualBox - Ubuntu a Windows 7 [7]

Hardwarová náročnost produktu je nízká, což je známkou dobré optimalizace kódu. Přesto výkon a funkce nezaostávají – pohyb myši a zadávání příkazů z klávesnice je velice rychlé. Vynikající je i kompatibilita mezi hostitelským systémem a tím virtuálním – lze lehce sdílet soubory mezi aplikacemi, obsah schránky i vybrané složky i bez funkčního síťového sdílení [5].

3.2.1.1 Princip VirtualBoxu

VirtualBox spouští v hostitelském operačním systému proces pro každého virtuální hosta. Všechen uživatelský kód spouštěný na hostovaném systému, je spouštěn v kruhu 3, stejně jako by byl spouštěn v hostitelském systému. Výsledkem je stejný výkon aplikace jak ve virtuálním, tak nativním (hostitelském) systému [9].

Z důvodu ochrany hostitele před chybami vzniklými v hostovaném virtuálním prostředí, není možné spouštět kernel hostovaného systému v kruhu 0. Místo toho běží v kruhu 1, není-li v systému hardwarová podpora virtualizace, nebo ve VT-x kruhu 0, je-li v systému hardwarová podpora virtualizace. To představuje překážku, jelikož host spouští v kruhu 1 instrukce, které je možné spouštět pouze v kruhu 0. Tyto instrukce se tak mohou chovat odlišně nebo nesprávně. K zajištění správnosti provádění operací hostovaného jádra systému, skenuje VirtualBox Monitor virtuálního stroje (VMM – Virtual Machine Monitor) kód v kruhu 1 a buď nahradí problémovou cestu kódu přímým voláním hypervisoru nebo kód spustí na bezpečném emulátoru.

V některých situacích nedokáže VMM přesně určit, co přesměrovaný hostovaný kód z kruhu 1 dělá. V těchto případech použije VirtualBox QEMU emulátor k dosažení stejného cíle. Příkladem je spouštění BIOSu, kdy real-mode operace během bootování hosta způsobí zákaz volání přerušení, nebo pokud je známo, že instrukce způsobí výjimku.

Protože tato emulace je ve srovnání s přímým spuštěním hostovaného kódu pomalejší, VMM obsahuje unikátní skener kódu pro každého podporovaného hosta. Jak bylo zmíněno dříve, tento skener nahradí cesty kódu přímým voláním hypervisoru a zajistí tak správné a efektivnější vykonání operace. Navíc pokaždé, kdy host způsobí chybu, VMM analyzuje příčinu chyby a určí, jestli může být problémová cesta v budoucnu nahrazena lepší metodou. V důsledku toho přístupu je VirtualBox inteligentnější než typický emulátor. Dokáže spouštět plně virtualizované (viz [3], s. 22) hosty téměř stejně rychle, jako hosty s hardwarovou podporou virtualizace Intel VT-x nebo AMD-V.

Některé operační systémy umožňují běh ovladačů v kruhu 1, což může způsobit konflikty s přesměrovaným kódem kernelu hosta. Takové operační systémy pak vyžadují hardwarovou podporu virtualizace [9].

3.2.1.2 Architektura VirtualBoxu

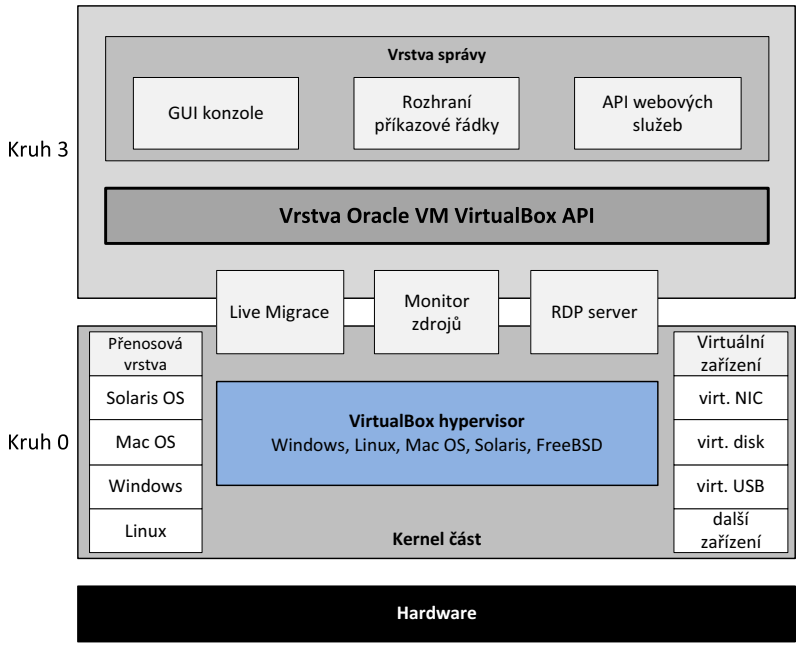
VirtualBox se skládá z jednotlivých vrstev, a to z kernel modulů určených pro běh virtuálních strojů, z API pro správu hostů a sady uživatelských programů a služeb. Jádrem je hypervisor implementovaný jako služba v kruhu 0. Obrázek 3.13 ukazuje vztah všech těchto komponent. Kernel služba se skládá z ovladače `vboxsrv`, který je zodpovědný za alokování fyzické paměti pro hosta, použití modulů hypervisoru zodpovědných za uložení a obnovení procesu hosta, rozhodnutí potřeby obsloužit VT-x nebo AMD-V události a předání řízení hostovanému OS.

Hypervisor nijak nezasahuje do plánování procesů v operačním systému hosta. Tyto úkoly jsou zcela v režii hosta v průběhu jeho chodu. Celý host je spuštěn jako jeden proces v systému hostitele. Je-li to potřeba, může administrátor využít nástroje řízení zdrojů hostitele (třídy řízení procesů, CPU caps[12]) k přidělení výkonu určitému hostovanému virtuálnímu stroji.

Dodatečné ovladače zařízení umožňují hostovi přístup k ostatním zdrojům hostitele, zejména k diskům, síťovým rozhraním, audio a USB zařízením. Pro podporu běhu hosta je použito, kromě modulů jádra, několik dalších procesů. Všechny tyto procesy jsou spuštěny automaticky, jakmile je to potřeba [9]:

- `VBoxSVC` je proces služeb VirtualBoxu. Uchovává stav všech virtuálních strojů, které běží na hostiteli. Spouští se automaticky s prvním startem hosta.

[12]Nástroj k řízení a přidělování výpočetní kapacity CPU jednotlivým procesům



Obrázek 3.13: Architektura VirtualBoxu [9]

- `vboxzoneacess` je speciální démon[13]pro systém Solaris, který umožňuje, aby zařízení VirtualBoxu byla přístupná z Oracle Solaris Container.
- `VBoxXPCOMIPCD` je XPCOM proces využitý pro hosty neobsahující operační systém Windows pro komunikaci mezi hostem a ovládací aplikací. Pro hosty s Windows je použita nativní služba COM.
- `VirtualBox` je proces, který běží pro každý spuštěný virtuální stroj. Pokud jsou v hostiteli uplatněny limity na zdroje, pak jsou právě na tento proces aplikovány.

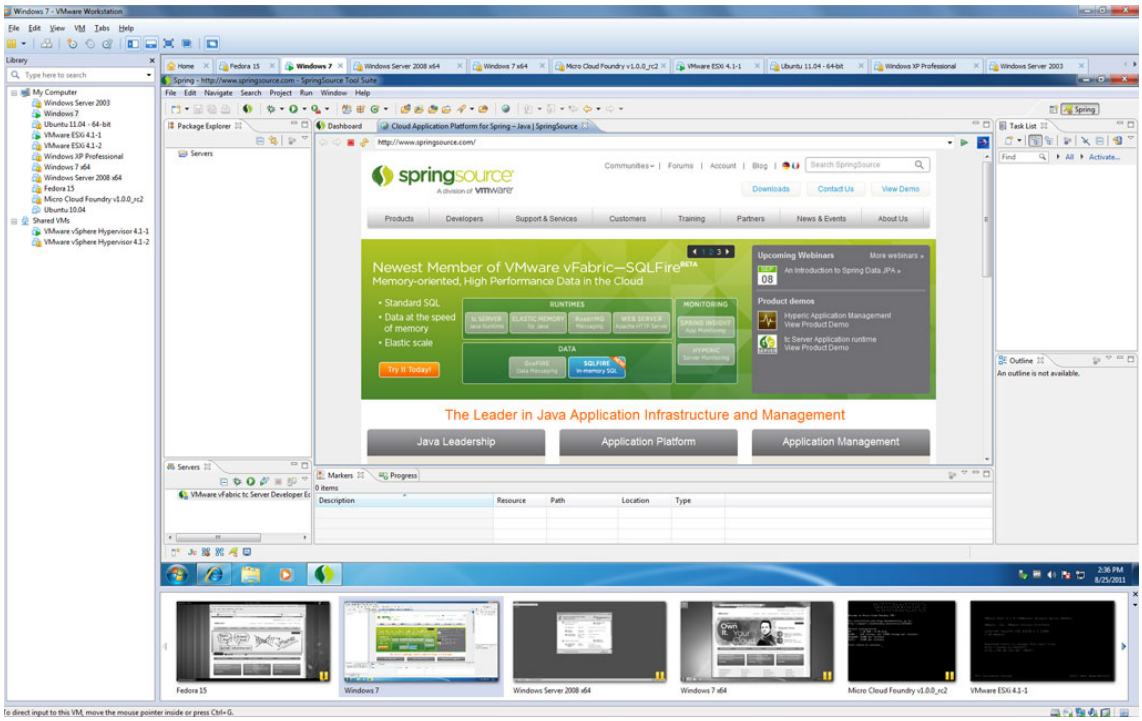
3.2.2 VMware

Společnost VMware nabízí v oblasti desktopové virtualizace několik nástrojů. Co se týče hostitelských operačních systémů, existují verze pro Windows, Linux a Mac OS.

3.2.2.1 VMware Workstation

VMware Workstation vydala společnosti VMware jako svůj první produkt v roce 1999. V současnosti se jedná o jeden z nejznámějších placených programů pro tvorbu virtuálních systémů [5].

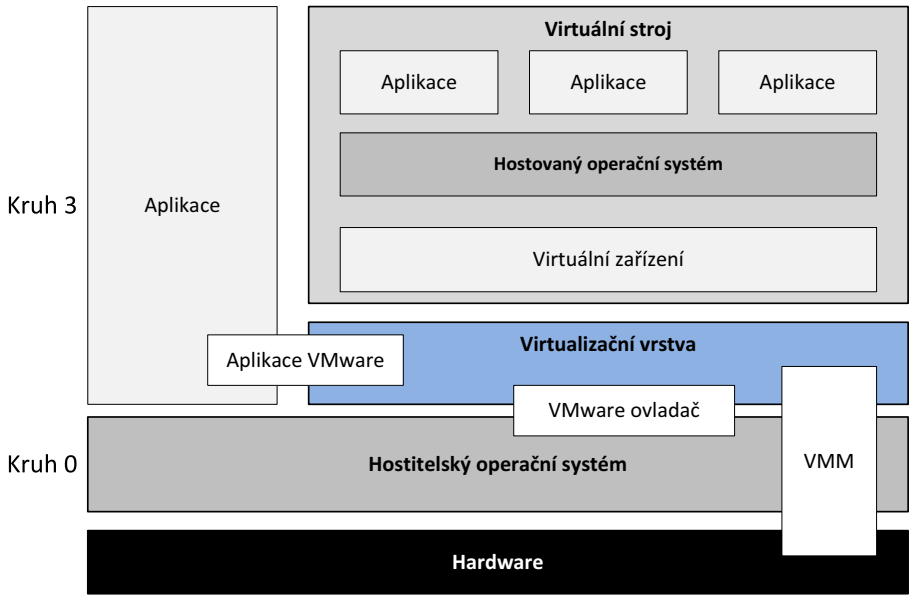
[13]Démon je označení programu, který je spuštěn dlouhodobě a není v přímém kontaktu s uživatelem (na rozdíl od běžných aplikací).



Obrázek 3.14: VMware Workstation 8 [11]

VMware Workstation je určen pro Windows a Linux coby hostitelský operační systém, v jeho prostředí lze však spustit libovolný operační systém. Tím, čím vyniká VMware nad konkurencí, je sada nástrojů VMware Tools. Ty umožňují instalaci ovladačů pro virtuální stroje, čímž lze dosáhnout optimalizace výkonu počítače. Virtuální systém je tak téměř stejně rychlý, jako stroj nativní. Nabízí plnou podporu síťové infrastruktury, uživatel má možnost nastavení sdílení síťového rozhraní pro virtuální stroj. Velice dobře je odladěna spolupráce virtuálního a hostitelského systému – z virtuálního lze snadno přetáhnout okno aplikace a dále s ním pracovat v hostitelském systému. Stejně snadno lze přetahovat soubory a složky. Obrázek 3.14 ukazuje aktuální verzi Workstation 8 s běžícími hostovanými operačními systémy [5].

Architektura VMware Workstation Workstation se skládá ze tří hlavních komponent, a to ovladače VMX, monitoru virtuálního stroje (VMM) a aplikace VMware. VMX ovladač a VMM monitor jsou spuštěny v kruhu 0. VMware aplikace pak běží v kruhu 3 a chová se jako jakákoliv jiná aplikace. Během instalace je v rámci operačního systému hostitele nainstalován ovladač VMX. Ten tak získá privilegovanou úroveň, kterou potřebuje komponenta VMM. Aplikace skrze VMX ovladač nahraje VMM do operační paměti s právy kruhu 0. V tuto chvíli ví operační systém hostitele pouze o aplikaci VMware a VMX ovladači. VMM komunikuje přímo s procesorem nebo skrze VMX ovladač a aplikaci s operačním systémem hosta. Virtuální stroj pak využívá střídavý přístup VMM a hostitelského operačního systému k procesoru, který vyžaduje udržovat stav obou prostředí, což s sebou přináší urči-



Obrázek 3.15: Architektura VMware Workstation

tou daň v podobě úbytku výkonu. Architektura VMware Workstation je na obrázku 3.15. Workstation aplikace obsahuje průvodce instalací virtuálního stroje, umožňuje vytvářet a editovat virtuální počítače.

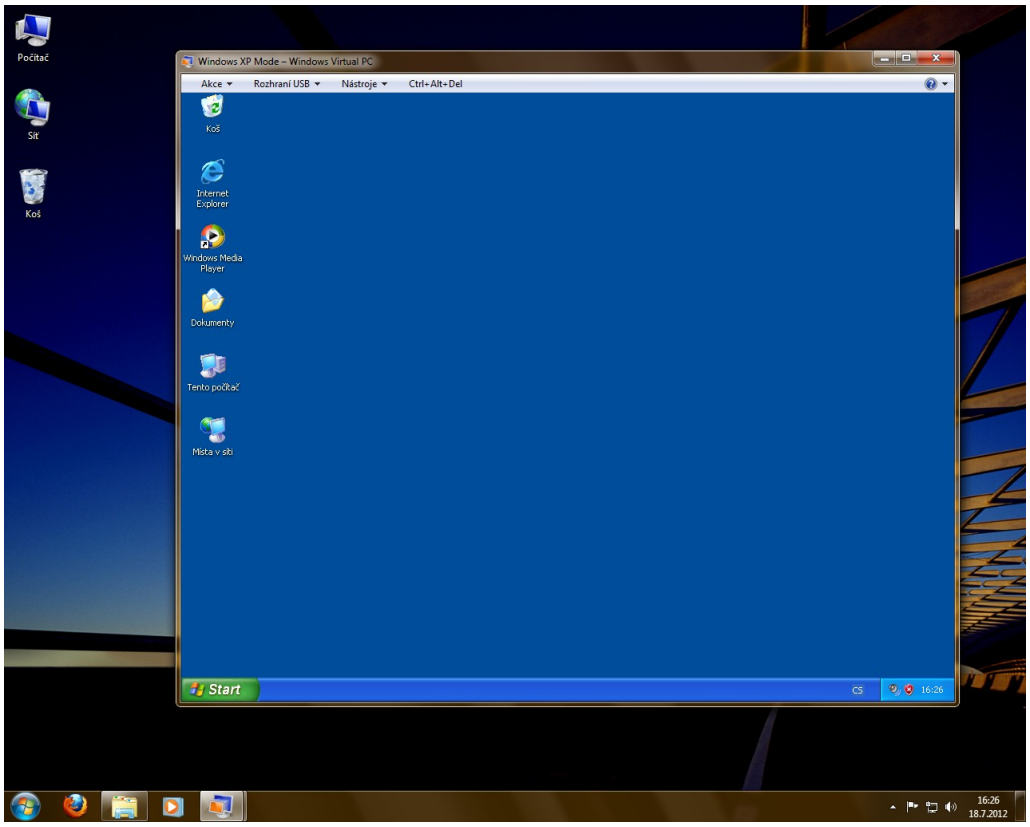
Pokud jsou v hostovaném operačním systému spuštěny čistě výpočetní programy, běží v samostatném režimu skrze VMM přímo na CPU. Workstation dokáže přesměrovat V/V operace a provést je hostitelským systémem místo systémem virtuálním. V/V instrukce jsou instrukce privilegované, VMM zachycuje všechny tyto požadavky a předává je do hostiteli. Jakmile aplikace VMware tento požadavek zpracuje, provede za VMM V/V požadavek (čtení z disku, přístup k síti) standardním voláním operačního systému hostitele.

3.2.2.2 VMware Player

VMware Player je VMware Workstation ochuzený a některé funkce a pro nekomerční použití je dostupný zdarma. Přestože má Player méně funkcí než Workstation, nabízí pro běžného uživatele dostatek možností pro vytváření a spouštění virtuálních strojů [6].

3.2.2.3 VMware Fusion

VMware Fusion, aktuálně ve verzi 4, je systém určený pro počítače Apple s operačním systémem Mac OS X. Dokáže, stejně jako Workstation, přetahovat okna do hostitelského systému a stejně jako Workstation, je i Fusion placený. Zajímavostí je, že dokáže zpřístupnit hardware použitelný pouze ve Windows i pro operační systém Mac [5].



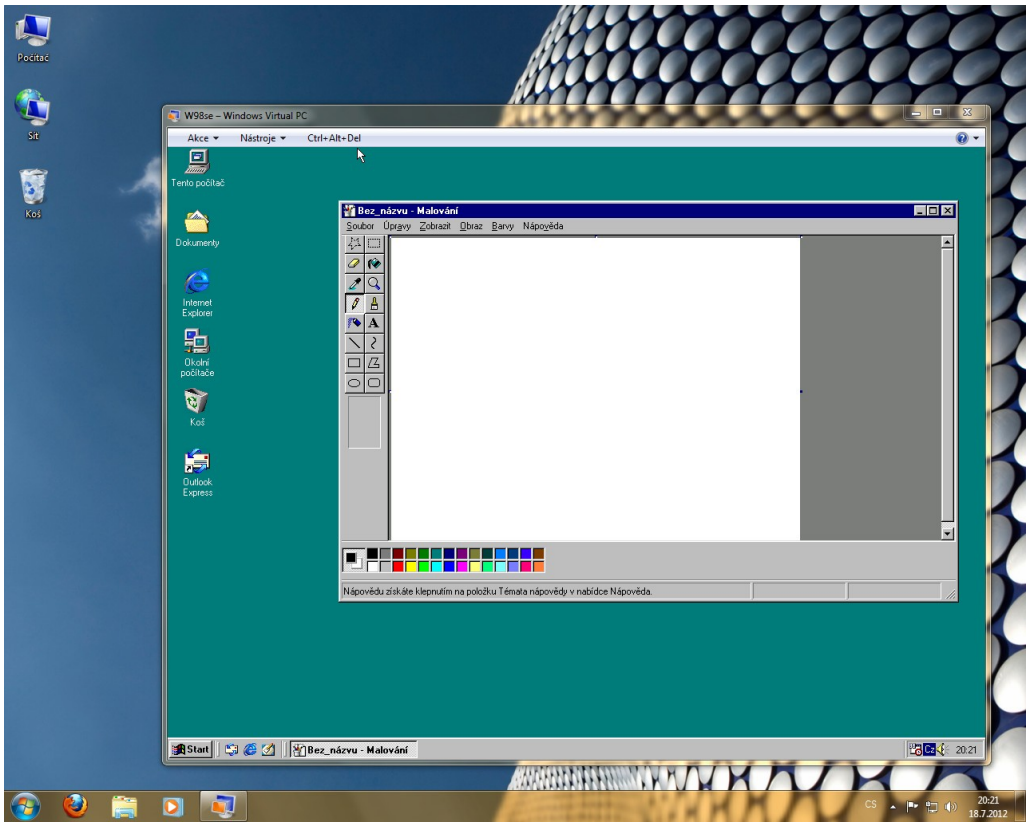
Obrázek 3.16: Režim XP ve Windows 7 Professional

3.2.3 Microsoft Virtual PC

Microsoft Virtual PC vychází z aplikace Virtual PC 2007, která byla zaměřena čistě na operační systémy Windows. Jejím cílem bylo vytvořit samostatné a nezávislé virtuální systémy. Virtual PC se stal vylepšenou verzí a je dodáván přímo s Windows 7 ve verzi Ultimate a Professional. Umožňuje konfiguraci virtuálního stroje pro instalaci a virtualizace operačního systému Windows.

Služeb Virtual PC využívá i Režim Windows XP. Jedná se o přednastavený obraz operačního systému Windows XP s integrovaným Service Packem 3. Tento režim umožňuje běh starších, s novou verzí Windows 7 nekompatibilních aplikací. V tomto případě je možné využít bezešvého režimu pro programy spuštěné ve virtuálních Windows XP. Ukázka Windows XP režimu (módu) je na obrázku 3.16.

Jelikož se jedná o nástroje z jedné dílny, jsou vysoce kompatibilní v rámci prostředí Windows. Samozřejmostí je sdílená schránka a podpora práce s USB zařízeními. Nevýhodou je dostupnost pouze pro výše uvedené verze a nemožnost virtualizovat jiné systémy než Windows [5]. Obrázek 3.17 ukazuje běh Windows 98 na virtuálním stroji pomocí nástroje Microsoft Virtual PC ve Windows 7 Professional.



Obrázek 3.17: Windows 98 ve Virtual PC

3.2.4 Parallels Desktop for Mac

Parellels Desktop byla jedna z prvních aplikací pro virtualizaci na počítačích s operačním systémem Mac OS. Mezi jeho velké výhody patří výborná podpora 3D grafiky. Z toho vyplývá, že ve virtualizovaném prostředí lze využívat akceleraci obrazu nejen pro grafické aplikace, ale i pro akceleraci grafických prostředí systému Windows nebo Linux nebo i hraní většiny her. Daní za tyto vlastnosti je vyšší náročnost na hardware hostitelského systému, zejména operační paměť. Prostředí má širokou možnost nastavení, včetně priority procesů, nastavení počtu přidělených procesorových jader nebo práce s diskovými obrazy [6]. Obrázek 3.18 ukazuje virtualizaci Windows 7 pomocí nástroje Parallels Desktop 7 for Mac na stroji s operačním (hostitelským) systémem Mac OS X.

3.2.5 Ostatní

Následující nástroje nejsou typické virtualizační aplikace jako předchozí aplikace, přesto jsou užitečné a stojí za krátké představení. Podrobněji se však jimi zabývat nebudeme.

Obrázek 3.18: Windows 7 virtualizované pomocí Parallels Desktop [10]

3.2.5.1 Sandboxie

Sandboxie je prostředí pro virtualizaci aplikací, tzv. „virtuální pískoviště". Jedná se o prostředí, v němž lze spustit libovolný program, používat jej, avšak jen definovaná data nebo procesy nechat proniknout do vlastního operačního systému. Velmi užitečný je tento nástroj při testování aplikací, jelikož není nutné instalovat celý virtuální systém, ale pouze zvolenou aplikaci [6]. Jedná se tak o jisto formu virtualizace aplikací dle kapitoly 2.2.1.

3.2.5.2 GoPC

GoPC není klasickou virtualizační aplikací. Ve své podstatě se jedná o cloudovou online službu nabízející virtuální počítač na internetu přístupný odkudkoliv. Prostředí je postavené na Linuxu a základ programové výbavy tvoří kancelářská aplikace OpenOffice.org, prohlížeč a souborový manažer Konqueror, Gimp, Firefox a další. Samozřejmostí je nahrávání i stahování vlastních souborů s možností uložení přímo ve virtuálním stroji. Nelze však instalovat vlastní aplikace [6].

3.3 Srovnání

Porovnávat vzájemně jednotlivé virtualizační systémy není snadné, neboť se zcela zásadně liší jak v přístupu, tak i v cílech a zaměření. Srovnání lze provádět pouze mezi produkty ze stejné skupiny – tedy produkty založené na hardwarové virtualizaci

a produkty založené na softwarové virtualizaci. Navíc u systémů hardwarové virtualizace (hypervisorů) není srovnání jednoduché, poněvadž každý produkt je založen na odlišných architekturách [13].

Ze tří výše uvedených hypervisorů se nabízí srovnání mezi VMware ESX/ESXi (z balíku vSphere), Hyper-V od Microsoftu a Xenem. Jak MS Hyper-V tak i Citrix XenServer provádějí všechny V/V operace virtuálních strojů pomocí rodičovské oblasti. Obecně je v rodičovské oblasti spuštěn nějaký operační systém a hypervisor využívá jeho prostředků – ovladačů a V/V zásobníku. Hyper-V takto využívá služeb Windows Server běžícího v rodičovské oblasti. Toto řešení vede k větší kompatibilitě, neboť podporuje-li Windows Server v rodičovské oblasti nějaký hardware, bude jej podporovat i hypervisor. Totéž platí i pro XenServer, avšak v jeho rodičovské oblasti, zde nazývané Dom0, běží Linux, nikoli Windows.

Systém VMware však využívá jiného principu – V/V operace se zpracovávají v samotném hypervisoru. Tím je dosaženo většího výkonu (průchodnosti operací) a menší režie avšak na úkor hardwarové kompatibility. V případě aktualizace ovladačů je nutno aktualizovat celý hypervisor, protože ovladače i V/V zásobník jsou jeho součástí.

Z výše uvedeného vyplývá, že rozdíl v architekturách systémů jsou zcela zásadní. Je třeba podotknout, že každé řešení má své výhody i nevýhody a je třeba pečlivě zvážit, který produkt je pro dané nasazení nejvhodnější [13].

Nicméně, přes výše uvedené, se v následující kapitole pokusím o objektivní porovnání některých virtualizačních systémů představených v kapitolách 3.1 a 3.2.

Kapitola 4

Porovnání jednotlivých řešení

Tato kapitola popisuje, jakým způsobem jsem přistoupil k testování výkonu jednotlivých virtualizačních systémů, jaké nástroje jsem použil a jakých výsledků jsem dosáhl.

4.1 Testovací nástroje a metodika

Mým cílem bylo porovnat virtualizační systémy po výkonnostní stránce v několika oblastech, a to zejména výpočetní výkon procesoru, přenosové rychlosti pevného disku a síťové karty. Zaměřil jsem se na tyto tři prvky virtuálního počítače, jelikož jsou pro jeho výkon zásadní. Rychlost virtuálního CPU závisí především na počtu a rychlosti CPU v hostitelském systému. Podobně i rychlost pevného disku je závislá na celkovém řešení diskového úložiště. Virtuální stroj nejčastěji komunikuje s okolím právě pomocí síťové karty, jejím prostřednictvím je i ovládán, nebo dokonce je připojen na síťové úložiště. Z tohoto pohledu je výkon síťového rozhraní zcela zásadní Naopak výkon grafického systému je nepodstatný.

K testování virtualizačních systémů jsem jako server použil počítač s procesorem Intel i3-540 (2 fyzická jádra, 64bitový, 3,06 GHz, technologie VT-x), operační pamětí 4 GB, základní deskou s chipsetem Intel H55 a síťovou kartou Realtek 10/100/1000 Mbit/s a pevný disk Maxtor (SATA150, 120 GB). K tomuto serveru byl připojen notebook (přímo nebo přes gigabitový switch SMCGS16) s gigabitovou síťovou kartou k vzdálenému spravování serveru a pro měření propustnosti síťového rozhraní serveru.

Pro testování byly využity následující verze virtualizačních nástrojů:

- VMware ESXi 5.0.0.update01, sestavení 623860, pro správu VMware vSphere Client 5.0.0, sestavení 623373,
- Citrix XenServer 6.0.201, pro správu Citrix XenCenter 6.0.2,

- Microsoft Hyper-V Server 2012 RC 64-bit, pro správu Remote Server Administration Tools,
- Oracle VM VirtualBox 4.1.18, sestavení 78361,
- VMware Workstation 8.0.4, sestavení 744019.

Při testování serverových virtualizačních nástrojů byl nejprve nainstalován na server samotný hypervisor. Poté byl pomocí vzdálené správy hypervisor nakonfigurován včetně dvou virtuálních strojů. Do těchto virtuálních strojů byly posléze nainstalován operační systém Windows 7 Professional a to jak 64bitová verze, tak i 32bitová verze. Pro testování desktopových systémů, byl nejdříve na počítač nainstalován OS hostitele Windows 7 Professional x64, poté samotný virtualizační systém a nakonec byly zprovozněny hostované operační systémy, resp. virtuální stroje s operačním systémem Windows 7 Professional, opět v obou verzích. Do všech hostovaných systémů byly nainstalovány rozšiřující balíčky pro daný virtualizační systém. Všechny testy probíhaly tak, že pro testovaný stroj bylo k dispozici jedno jádro procesoru (virtuální stroj měl nakonfigurováno právě jedno jádro). Některé konfigurace byly navíc testovány s vypnutou podporou virtualizace Intel VT-x. Na těchto strojích pak byly provedeny testy pomocí následujících programů.

Pro celkové otestování virtuálního stroje jsem použil nástroj NovaBench 3.0.4[1], pro otestování přenosové rychlosti pevného disku nástroj CrystalDiskMark 3.0.1c[2]. Dále pro testování rychlosti síťového rozhraní jsem použil nástroje PCATTCP ve verzi V2.01.01.14[3] a Apache Benchmark Tool 2.2[4]. Pro výpočetní testy zaměřené na rychlost CPU, jsem použil CPUMathMark 3[5] a HyperPI 0.99b[6]. Jako poslední, ryze praktický test, jsem vytvářel archiv pomocí programu 7zip 9.20[7]. Podrobněji tyto nástroje včetně použitého nastavení popíši v následujících kapitolách.

Jako srovnávací etalon posloužilo otestování samotného serveru s operačním systémem Windows 7 Professional x64 i x86 a jedním jádrem (omezeno v BIOSu).

4.1.1 NovaBench

NovaBench je program, který slouží k rychlému otestování celého počítače. Test trvá maximálně dvě minuty a přináší detailní informace o systému včetně celkového skóre. To také bylo využito jako srovnávací parametr. Oblasti, které test pokrývá:

[1] www.novabench.com
[2] www.crystalmark.info
[3] www.pcausa.com/Utilities/pcattcp.htm
[4] http://httpd.apache.org/docs/2.2/programs/ab.html
[5] http://majorgeeks.com/CPU_Math_Mark_d123.html
[6] www.virgilioborges.com.br/hyperpi
[7] www.7-zip.org

- Floating Point test – testuje rychlost CPU v operacích s plovoucí čárkou,
- Integer test – testuje rychlost CPU v operacích s celými čísly,
- rychlost vytváření MD5 Hash – univerzální test CPU,
- 3D grafický test – testování GPU pomocí výrazně stínované 3D scény (v mém testování nevyužito),
- rychlost RAM – testuje rychlost zápisu a čtení paměti RAM,
- rychlost zápisu na disk – testuje rychlost zápisu na primární nebo zvolený pevný disk.

Pro testování jsem použil variantu testu bez testování GPU.

4.1.2 CrystalDiskMark

CrystalDiskMark testuje rychlost pevného disku a řadiče pomocí několika testů. První z nich je sekvenční čtení a zápis, dále pak čtení a zápis souborů o velikosti 512 kB resp. 4 kB a 32 současně zapisovaných/čtených 4 kB souborů. Poslední možnost je zaměřena na testování NCQ[8] funkce. Rozhraní umožňuje volbu, zda budou zapisována náhodná data nebo samé jedničky nebo samé nuly. Dalšími volbami je objem zapisovaných dat od 50 MB do 4000 MB, počet průběhů testu a nakonec volba disku k testování. Výsledkem testů je přenosová rychlost disku v MB/s.

Nastavení pro testování: 1 průběh, 50 MB, náhodná data. Pro každý operační systém (x64, x86) jsem použil odpovídající verzi programu.

4.1.3 PCATTCP

PCATTCP je port původního Test TCP (TTCP) pro operační systém BSD. Ovládá se z příkazové řádky, podporuje IPv4 i IPv6, komunikuje pomocí TCP nebo UDP paketů a má implementovanou vícevláknovou komunikaci. Program zobrazí rychlost spojení v určených jednotkách (např. kB/s).

Pro test jsem nastavil počet odesílaných bufferů na 32 768 (z původních 2048), abych síťové rozhraní více zatížil.

4.1.4 Apache Bench

Apache Bench, neboli ab, je nástroj pro testování výkonu Apache serveru. Je určen k otestování výkonu aktuální instalace Apache serveru a ukazuje, kolik požadavků

[8]NCQ (Native Command Queuing) umožňuje zařízení, aby si samo určilo optimální pořadí, ve kterém bude přistupovat k požadovaným informacím na médiu.

dokáže během jedné sekundy server vyřídit. Tento nástroj jsem použil pro testování více spojení skrze síťovou kartu současně. Výsledkem je doba potřebná k uskutečnění určeného počtu přenosů.

Testování probíhalo ve třech variantách – pokaždé bylo nastaveno odeslání 10 000 požadavků, a to 10, 100, resp. 200 požadavků současně, aby byla využitá celá šířka síťového rozhraní. Ze serveru byla stahována stránka o velikosti 994 bajtů.

4.1.5 CPUMathMark

CPUMathMark spouští několik dílčích testů k otestování výkonosti CPU v matematických výpočtech. Obsahuje test sčítání, odčítání, násobení, dělení, z pokročilých operací je to pak test prvočísla, výpočet čísla π. Hodnotícím faktorem je doba potřebná pro všechny uvedené výpočty.

4.1.6 HyperPI

HyperPI je grafickou nadstavbou programu SuperPI. Program vypočítá číslo π na určený počet míst. Hodnotícím faktorem je opět doba výpočtu.

Pro testování jsem měřil dobu výpočtu čísla π na 512 000 míst. Tato hodnota představuje kompromis mezi časem testování a náročností výpočtu.

4.1.7 7zip

7zip je open-source archivační program. Vytváří archivy v několika formátech, jedním z nich je 7z a zip, které jsem použil pro srovnání. Hodnotícím faktorem byla doba potřebná k vytvoření archivu z testovacích dat. Program existuje pro 64 bitové i 32 bitové operační systémy, proto jsem použil při testování odpovídající verzi.

Jako testovací data posloužily fotografie, pdf soubory, hudba a textové dokumenty o celkové velikosti 408 MB. Formát 7z i zip byly ve výchozím nastavení.

4.1.8 Metodika

Všechna měření probíhala na čisté instalaci operačního systému Windows 7 Professional CZ získaného z programu MSDN AA. Operační systém byl bez Service Packu 1.

Testování pomocí 7zipu proběhlo jednou. Měření pomocí programu CrystalDiskMark proběhlo pětkrát a celkový výsledek v jednotlivých oblastech byl dán aritmetickým průměrem naměřených hodnot. Měření ostatními programy probíhalo desetkrát a celkový výsledek je dán jako aritmetický průměr, přičemž byl vyřazen jeden nejlepší a jeden nejhorší výsledek.

testovací program	podíl na hodnocení			
PCATTCP	20 %		sekvenční čtení	5 %
Apache Bench (10)	4 %		sekvenční zápis	7 %
Apache Bench (100)	5 %		512k čtení	2 %
Apache Bench (200)	1 %		512k zápis	4%
CrystalDiskMark	30 %	z toho	4k čtení	2 %
NovaBench	10 %		4k zápis	4%
CPUMathMark	10 %		4k QD32 čtení	2 %
HyperPI	10 %		4k QD32 zápis	4%
7zip (7z)	5 %			
7zip (zip)	5 %			

Tabulka 4.1: Podíly jednotlivých testů na celkovém hodnocení

Celkové hodnocení jsem stanovil na základě důležitosti jednotlivých kategorií. Nejdříve byly stanoveny poměrové výsledky pro každý virtualizační systém jako:

$$v = \frac{x_i}{\max(x_0 \dots x_n)} \tag{4.1}$$

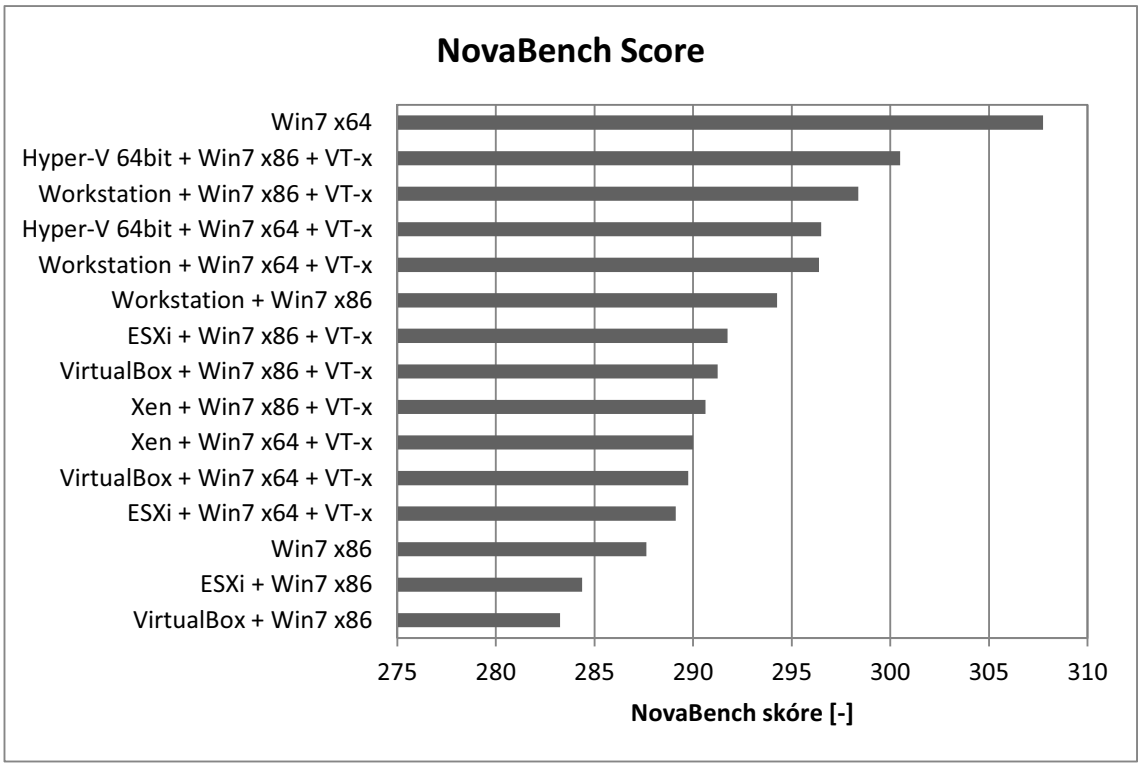
pro testy, ve kterých je cílem maximální hodnota, resp.

$$v = \frac{\min(x_0 \dots x_n)}{x_i} \tag{4.2}$$

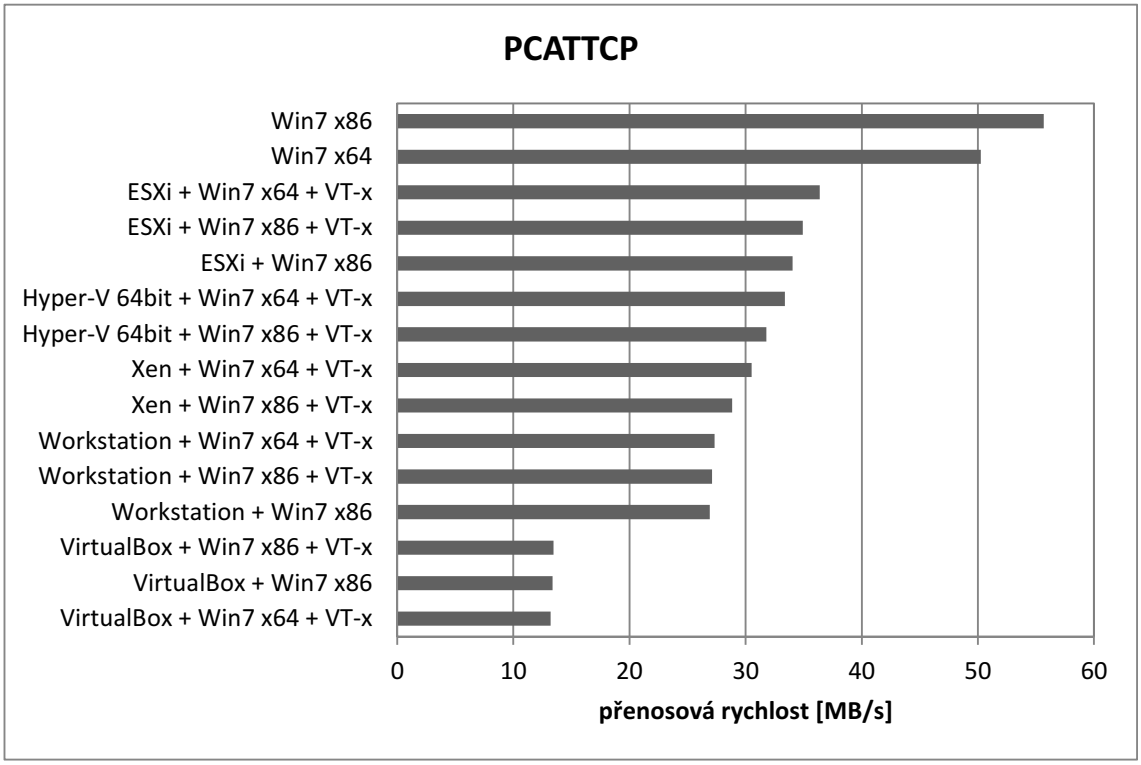
pro testy, ve kterých je cílem hodnota minimální a kde v je poměrový výsledek, x_i je původní výsledek pro daný virtualizační systém a $x_0 \dots x_n$ jsou všechny výsledky v dané kategorii. Celkové hodnocení pro virtualizační systém se rovná součtu dílčích výsledků násobených příslušným koeficientem dle tabulky 4.1. Tabulka 4.1 nám tak ukazuje, které testy se jakým podílem podílejí na celkovém hodnocení.

4.2 Výsledky testů

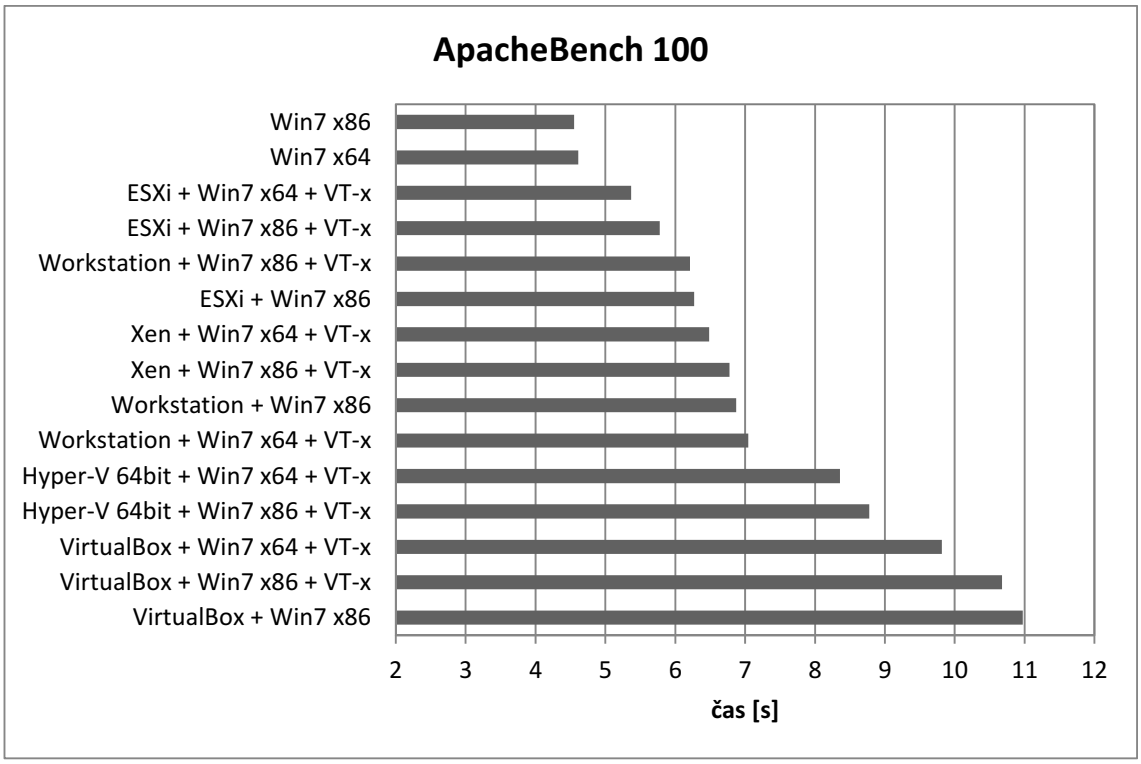
V této kapitole uvedu výsledky testů v grafické podobě, všechny naměřené hodnoty jsou uloženy v příloze na CD disku.



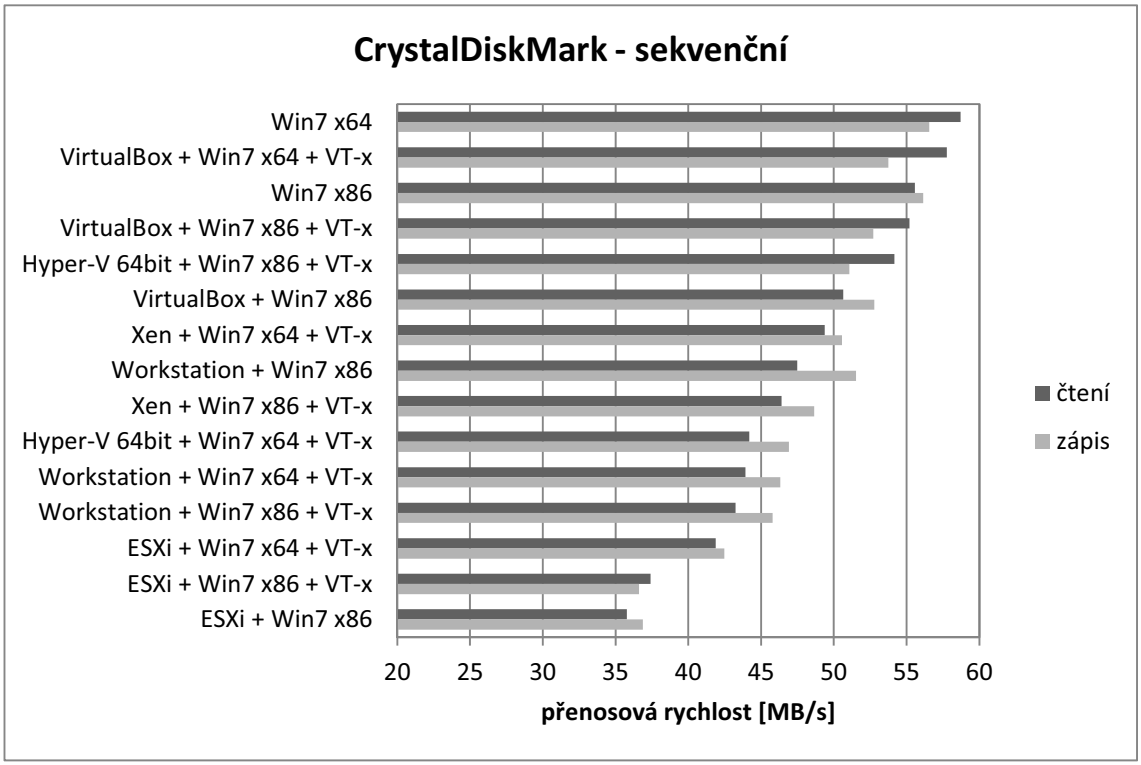
Obrázek 4.1: Graf výsledků měření z NovaBench



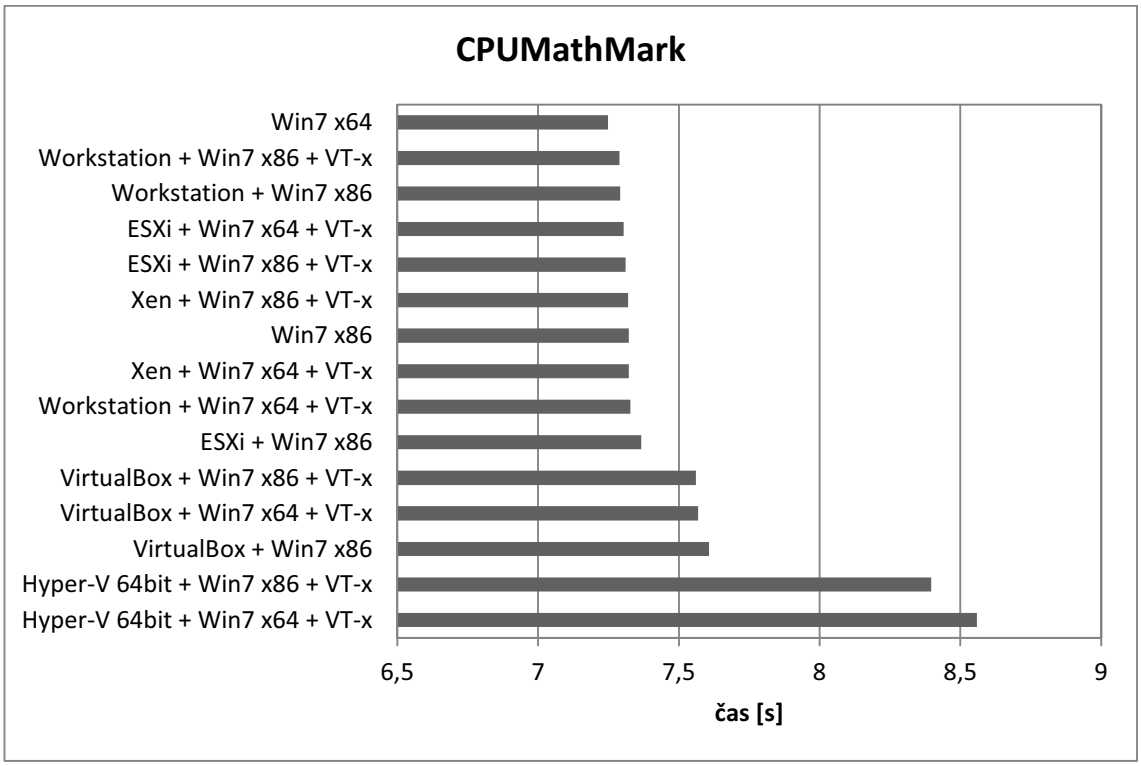
Obrázek 4.2: Graf výsledků měření z PCATTCP



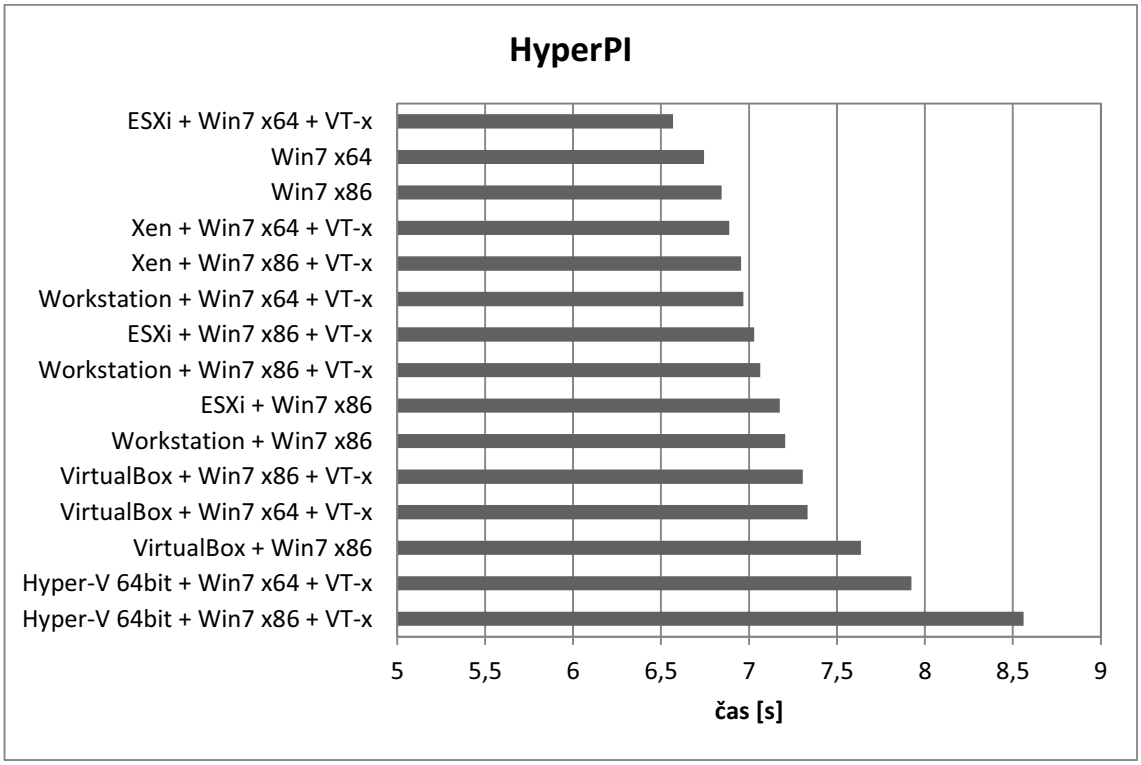
Obrázek 4.3: Graf výsledků měření z ApacheBench



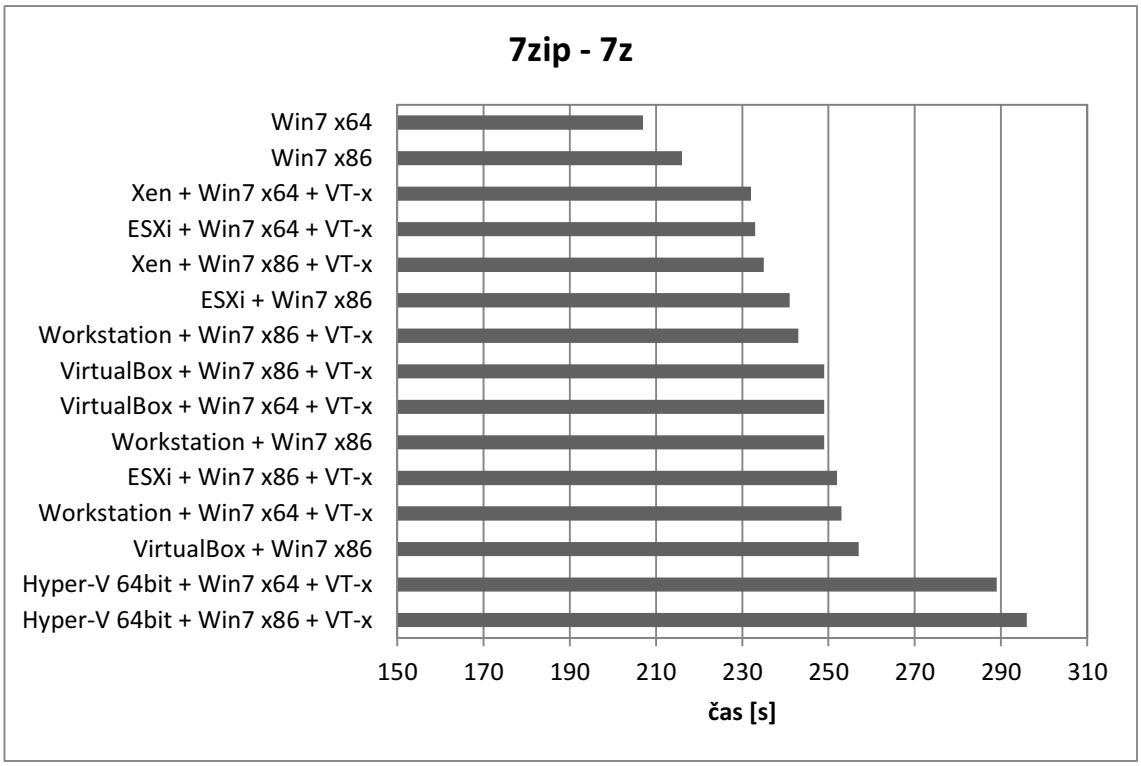
Obrázek 4.4: Graf výsledků měření z CrystalDiskMark



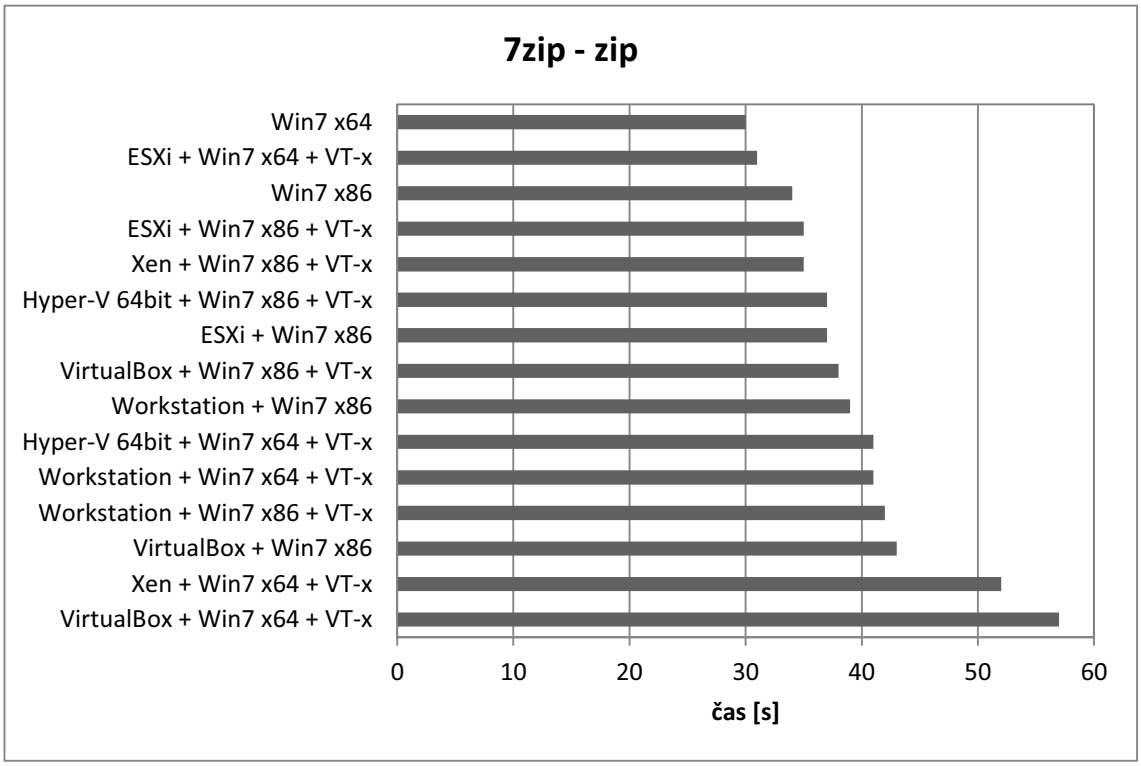
Obrázek 4.5: Graf výsledků měření z CPUMathMark



Obrázek 4.6: Graf výsledků měření z HyperPI



Obrázek 4.7: Graf výsledků měření ze 7zip (formát 7z)



Obrázek 4.8: Graf výsledků měření ze 7zip (formát zip)

4.3 Vyhodnocení testů

Výsledky testů potvrdily, že jsou dnešní virtualizační systémy mocnými a výkonými nástroji. V žádné z testovaných oblastí ani jeden systém vyloženě nepropadl, přesto zde byly patrné rozdíly. Virtualizační systémy zhodnotím jednotlivě. Nejprve však k virtualizaci 64bitových operačních systémů.

Virtualizace 64bitových operačních systémů vyžaduje podporu ze strany hardwaru, neboť je třeba, aby virtuální stroj správně detekoval 64bitový procesor a mohl tak provádět 64bitové instrukce. Hardwarovou podporu také vyžadují uzavřené operační systémy (jakým Windows 7 jsou), proto ve většině případů, vyjma ESXi, nešlo spustit ani virtuální stroj s 32bitovým operačním systémem. Tento problém se netýká desktopových (softwarových) virtualizačních systémů, ty k virtualizaci 32bitových Windows hardwarovou podporu virtualizace nevyžadují.

VMware ESXi 5 Tento hypervisor si vedl v testech dobře, zejména v testech zaměřených na výkon síťového rozhraní, ve kterých patřil k nejrychlejším. Naopak v testu výkonu pevného disku skončil na posledních místech. Jako jediný z hypervisorů dokázal hostovat operační systém Windows 7 x86 i bez zapnuté hardwarové podpory virtualizace Intel VT-x. To umožňuje zvláštní mód virtualizace vrstvy VMM (viz kapitola 3.1.1.2 a [15]).

Instalace ESXi byla bez potíží (díky široce podporovanému hardware serveru), vzdálená správa serveru také. Ovládání hostovaného systému přes konzoli VMware vSphere Client nebylo téměř použitelné pro neplynule se pohybující kurzor (a to ani po instalaci doplňků VMware). Proto jsem ovládal testovaný hostovaný operační systém přes vzdálenou plochu systému Windows.

Citrix XenServer 6 Xen hypervisor v podání společnosti Citrix si vedl dobře ve všech testovaných oblastech, podal velmi vyrovnané výsledky. Velmi dobře si vedl ve výpočtu programem HyperPI a vytváření archivu ve formátu 7z. Bez problémů virtualizoval Windows 7, avšak jedině s hardwarovou podporou virtualizace VT-x, což je očekávaná vlastnost HVM hypervisorů v kombinaci s uzavřeným hostovaným systémem.

Instalace, konfigurace a vzdálená správa byla ze všech tří testovaných hypervisorů nejméně problémová. Instalátor umožňuje při instalaci zavést další ovladače pro hardware. Ovládání hostovaných Windows 7 skrze konzoli v aplikaci Citrix XenCenter bylo velmi rychlé. Citrix XenServer nabízel nejvíce funkcí v konzoli lokální správy.

Microsoft Hyper-V Server 2012 RC 64-bit Hypervisor z dílny Microsoftu si celkově nevedl příliš dobře. Propadl hlavně ve výpočetních testech. Naopak vcelku

obstojně si vedl v testech komunikace v síti nebo v NovaBench hodnocení. I Hyper-V si neporadil s vypnutou hardwarovou podporou virtualizace a odmítal oba nakonfigurované virtuální stroje s Windows 7 spustit.

Instalace hypervisoru byla velmi rychlá a jednoduchá, dokonce instalátor nabídl instalaci na disk připojený na přídavný řadič JMicron, pro který předchozí systémy neměly ovladače. Z toho usuzuji, že Hyper-V má širokou podporu hardware a je tedy z testovaných systémů nejvíce kompatibilní. Aplikace pro vzdálenou správu Remote Server Administration Tools a její komponenta Správce technologie Hyper-V byly poněkud těžkopádné, nicméně server i virtuální stroj šly dobře spravovat a ovládat.

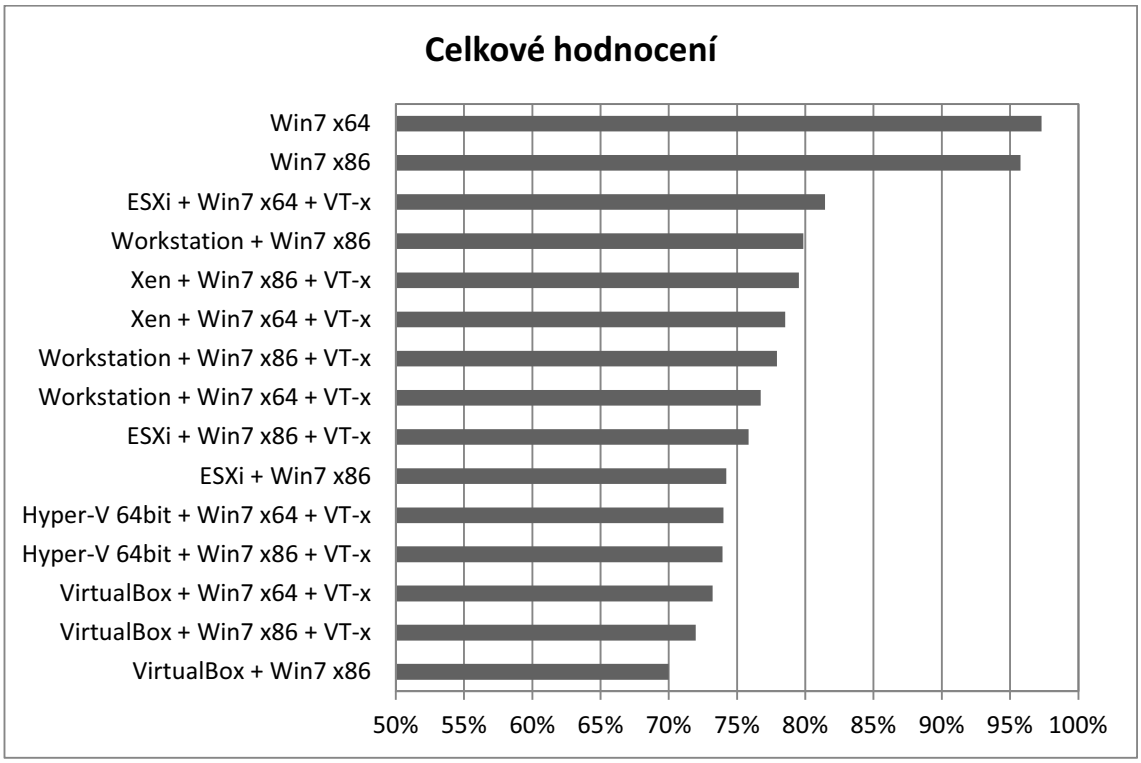
Oracle VM VirtualBox 4.1 VirtualBox si v testech vedl spíše podprůměrně. Zejména v testech zaměřených na síťové rozhraní nepodal dobrý výkon. Rychlost komunikace byla třetinová oproti ESXi, nebo poloviční oproti Workstationu. Jedinou oblastí, kde byl VirtualBox dobrý, bylo testování výkonu pevného disku, kde se dostal až na úroveň samotných Windows 7 bez virtualizace. VirtualBox umožňuje virtualizovat 32bitové Windows (a další operační systémy) i bez hardwarové podpory virtualizace, 64bitové operační systémy však tyto podporu již vyžadují.

Uživatelské rozhraní VirtualBoxu je jednoduché, intuitivní a přesto umožňuje spoustu voleb a nastavení. Z funkčního ani uživatelského pohledu mu tedy není co vytknout.

VMware Workstation 8 Workstation ve verzi 8 si vedl dobře ve všech testech. Velmi dobrých výsledků dosáhl v syntetickém testu NovaBench nebo výpočetním CPUMathMarku, naopak zaostával v testech zaměřených na síťové rozhraní, ne však nijak výrazně. Co se týče síťového rozhraní, nepodařilo se mi nastavit přímý přístup hostovaného Windows 7 k síťové kartě a nebylo možné správně ručně nastavit IP adresy pro testy. Proto testy probíhaly přes switch s automatickým přidělováním adres. Na výsledek to však vliv nemělo, jelikož switch má větší propustnost (32 Gb/s), než naměřená rychlost komunikace. Stejně jako VirtualBox, bez hardwarové podpory VT-x dokázal hostovat pouze 32bitové Windows 7.

Ovládací aplikace Workstationu obsahuje mnoho funkcí a voleb, je však dobře ovladatelná a přehledná.

Celkové zhodnocení Graf na obrázku 4.9 ukazuje celkové hodnocení jednotlivých virtualizačních systémů stanovéné dle metodiky. Výsledky všech systémů jsou v rozpětí 12 %, proto si odvážím tvrdit, že všechny systémy jsou na velmi dobré úrovni. Za nevirtualizovaným systémem zaostávají o zhruba 15 %, což ukazuje na to, že virtualizační systémy nespotřebují na svůj běh (režii) mnoho systémového výkonu. Mnohem zajímavější se tedy jeví porovnání a sledování výkonu jednotlivých



Obrázek 4.9: Celkové hodnocení virtualizačních systémů

systémů v jednotlivých kategoriích, jak jsem popsal výše.

Kapitola 5

Závěr

Cílem této práce bylo představit virtualizační systémy, jejich principy fungování a změřit jejich výkon ku vztahu nevirtualizovanému systému.

Virtualizační systémy pracují v zásadě na dvou principech. Prvních z nich je softwarová virtualizace tvořícího prostředníka mezi nainstalovaným operačním systémem a vytvářenými virtuálními stroji. Tato vrstva vytvoří z prostředků hostitelského operačního systému virtuální zařízení a to zpřístupní virtuálním strojům – hostům. Vše běží nad hostitelským systémem. Výhodou je jednoduchá implementace a nezávislost virtualizačního softwaru na ovladačích k hardwaru. Toto řešení je ideální pro testování a vývoj aplikací na dalších operačních systémech, či ke spouštění aplikací určených pro starší verze aktuálně používaného operačního systému. Druhým principem je hardwarová virtualizace, kdy virtualizační systém – hypervisor – běží přímo nad hardwarem a nahrazuje tak operační systém. Opět zpřístupňují prostředky fyzického stroje jako virtuální hardware pro virtuální stroje. Nyní se však musí spolehnout na vlastní ovladače. Ty jsou implementovány spolu s nástroji správy přímo v hypervisoru či v jeho nadstavbě. Toto řešení není vždy výkonnější než softwarová virtualizace, jak ukázala provedená měření.

Z naměřených výsledků vyplývá, že dnešní virtualizační systémy jsou velmi dobře připraveny poradit si s nelehkým úkolem, který bezpochyby virtualizace je. Všechny testované systémy si celkově poradily s virtualizací velmi dobře. Jsou oblasti, ve kterých by se jednotlivé produkty mohly zlepšit, to však nekazí celkově dobrý výsledek. Vlivem virtualizačních systémů se sníží výkon virtuálního stroje oproti fyzickému o cca 15%, na druhou stranu však získáme neocenitelnou možnost provozovat na jednom fyzickém hardwaru několik virtuálních systémů, vzdáleně je spravovat včetně infrastruktury a zajistit tak vysokou dostupnost služeb.

Virtualizace přináší zejména v serverové oblasti nemalé úspory, jak energetické, tak prostorové a ve svém důsledku i finanční. Ve virtualizaci vidím přítomnost a zejména budoucnost podnikových infrastruktur. Pro domácí uživatele vidím ve virtualizaci možnost, jak nebýt vázán pouze na jeden operační systém.

Literatura

[1] RUEST, Danielle; RUEST, Nelson. *Virtualizace : Podrobný průvodce.* Brno : Computer Press, 2010. 394 s. ISBN 978-80-251-2676-9.

[2] VILE, Dale, et al. *Desktop Virtualization for dummies.* Chichester (UK) : John Wiley & Sons Ltd, 2010. 32 s. ISBN 978-0-470-97364-6.

[3] SCHEFFY, Clark. *Virtualization for dummies: AMD special edition.* Hoboken: Wiley Publishing, 2007. 42s. ISBN 978-0-470-13156-5.

[4] BROUKALOVÁ, Monika. *Virtualizace* [online]. 2010. 12 s. Přednáška. JČU. Dostupné z WWW: <http://eamos.pf.jcu.cz/amos/kat_inf/externi/kat_inf_53595/kolarp01_5/task_6/virtualizace.ppt>.

[5] KRAUS, Josef. Virtuální systém v každém počítači. *Computer.* Praha: Mladá fronta, 2012, roč. 19 (č. 1), 32-34.

[6] JANEČEK, V. Více slupek v jednom. *Computer.* Praha: Mladá fronta, 2011, roč. 18 (č. 5), 34-36.

[7] Oracle VM VirtualBox® User Manual. ORACLE CORPORATION. *VirtualBox* [online]. 2011 [cit. 2012-02-05]. Dostupné z: https://www.virtualbox.org/manual/UserManual.html

[8] SU, Honglin. ORACLE CORPORATION. *Oracle VM 3: Architecture and Technical Overview* [online]. 2011 [cit. 2012-07-30]. Dostupné z: http://www.oracle.com/us/technologies/virtualization/ovm3-arch-tech-overview-459307.pdf

[9] VICTOR, Jeff. *Oracle Solaris 10 system virtualization essentials.* Upper Saddle River, NJ: Prentice Hall, 2011, xxiv, 358 p. ISBN 01-370-8188-X.

[10] PARALLELS HOLDINGS LTD. *Parallels Desktop® 7 for Mac* [online]. 1999-2012 [cit. 2012-07-17]. Dostupné z: http://www.parallels.com/eu/products/desktop/

[11] Virtualization Products. *VMware* [online]. 2012 [cit. 2012-02-12]. Dostupné z: http://www.vmware.com/products/

[12] Virtual Machines & VMware, Part I. *ExtremeTech* [online]. 2001 [cit. 2012-08-01]. Dostupné z: http://www.extremetech.com/computing/72186-virtual-machines-vmware-part-i

[13] LOWE, Scott. *Mistrovství ve VMware vSphere 4: kompletní průvodce profesionální virtualizací*. Vyd. 1. Brno: Computer Press, 2010, 662 s. ISBN 978-80-251-2915-9.

[14] CHAUBAL, Charu. VMWARE INC. *The Architecture of VMware ESXi* [online]. 2007 [cit. 2012-07-27]. VMware white paper. Dostupné z: http://www.vmware.com/resources/techresources/1009

[15] BHATIA, Nikhil. VMWARE INC. *Virtual Machine Monitor Execution Modes in VMware vSphere 4.0* [online]. 2010 [cit. 2012-07-27]. VMware white paper. Dostupné z: http://www.vmware.com/resources/techresources/10060

[16] New to Xen Guide. CITRIX SYSTEMS, Inc. *Xen.org* [online]. 2012 [cit. 2012-07-03]. Dostupné z: http://xen.org/files/Marketing/NewtoXenGuide.pdf

[17] SUCHÝ, Miroslav. Úvod do virtualizace pomocí XENu. *Root.cz* [online]. 2007 [cit. 2012-07-03]. Dostupné z: http://www.root.cz/clanky/uvod-do-virtualizace-pomoci-xenu/

[18] BARHAM, Paul, et al. *Xen and the Art of Virtualization* [online]. Cambridge, 2003 [cit. 2012-07-30]. Dostupné z: http://www.cl.cam.ac.uk/research/srg/netos/papers/2003-xensosp.pdf. Research. University of Cambridge.

[19] Why Xen ?. CITRIX SYSTEMS, Inc. *Xen.org* [online]. 2012 [cit. 2012-07-30]. Dostupné z: http://xen.org/files/Marketing/WhyXen.pdf

[20] VISWANATHAN, Arun. *Virtualization with XEN*. Los Angeles, USA, 2007. Dostupné z: http://www.arunviswanathan.com/content/ppts/xen_virt.pdf. Přednáška. University of Southern California.

[21] TULLOCH, Mitch. MICROSOFT. *Understanding Microsoft Virtualization Solutions: From the Desktop to the Datacenter*. 2. vyd. Redmond, WA: Microsoft Press, 2010, 480 s.

[22] *KVM* [online]. 2011 [cit. 2012-07-08]. Dostupné z: http://www.linux-kvm.org/

[23] KVM Overview. IBM. *Linux information for IBM systems* [online]. 2007 [cit. 2012-07-29]. Dostupné z: http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/index.jsp?topic=%2Fliaat%2Fliaatkvmover.htm

[24] *QEMU: Open source processor emulator* [online]. 2011- [cit. 2012-07-08]. Dostupné z: wiki.qemu.org/

[25] CITRIX SYSTEMS, Inc. *Citrix Systems* [online]. 1999-2012 [cit. 2012-07-17]. Dostupné z: www.citrix.cz

[26] Understanding and exploiting snapshot technology for data protection: Part 1: Snapshot technology overview. *DeveloperWorks* [online]. 2006 [cit. 2012-07-22]. Dostupné z: http://www.ibm.com/developerworks/tivoli/library/t-snaptsm1/index.html

[27] JELÍNEK, Lukáš. POSIX. *ABC Linuxu* [online]. 2005 [cit. 2012-07-26]. Dostupné z: http://www.abclinuxu.cz/slovnik/posix

Seznam obrázků

Seznam tabulek