

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Komparativní analýza virtualizačních technologií
serverů

Zdeněk Černocho

Bakalářská práce

2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Zdeněk Černocho**
Osobní číslo: **I09087**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Komparativní analýza virtualizačních technologií serverů**
Zadávací katedra: **Katedra informačních technologií**

Z á s a d y p r o v y p r a c o v á n í :

Autor provede komparativní analýzu vybraných virtualizačních technologií pro servery. Autor představí technologické možnosti virtualizace serverů a podrobně představí vybrané technologie. Na základě principů fungování a klíčových vlastností sestaví metodiku zátěžových testů z pohledu využívání systémových prostředků, které následně realizuje a vyhodnotí. Testovaným prostředím bude Windows server 2008 R2.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

[BY GERARD BLOKDIJK, Ivanka Menken]. **Virtualization - the Complete Cornerstone Guide to Virtualization Best Practices Concepts, Terms, and Techniques for Successfully Planning, Implementing and Managing Enterprise It Virtualization Technology - Second Edition. 2nd ed. Brisbane: Gardners Books, 2010. ISBN 978-174-2445-885.**

FINN, Aidan a Patrick LOWNDS. **Mastering Hyper-V deployment.**

Indianapolis, Ind.: Wiley, c2011, xxvi, 574 p. Serious skills. ISBN 04-708-7653-0.

MARSHALL, David, Wade A REYNOLDS a Dave MCCRORY. **Advanced server virtualization: VMware and Microsoft platforms in the virtual data center. Boca Raton: Auerbach Publications, 2006, xvi, 742 p. ISBN 978-084-9339-318.**

Vedoucí bakalářské práce:

Mgr. Josef Horálek

Katedra softwarových technologií

Datum zadání bakalářské práce: **21. prosince 2012**

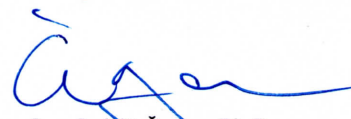
Termín odevzdání bakalářské práce: **10. května 2013**



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 29. března 2013

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 10.8.2013

Zdeněk Černoch

Poděkování

Chci poděkovat svému vedoucímu za pomoc při zpracování bakalářské práce. Svým rodičům za umožnění studia a za podporu. Stejně tak i svým přátelům, kteří mě po celou dobu studia podporovali.

Anotace

Práce obsahuje seznámení s principy virtualizace serverů. První část se zaměřuje na obecnou problematiku virtualizace. V druhé části jsou popsány tři hlavní virtualizační řešení na trhu od firem VMware, Microsoft a Citrix. Následuje zhodnocení posbíraných dat.

Klíčová slova

virtualizace, hypervizor, konsolidace

Title

Comparative analysis of server virtualization

Annotation

The work includes familiarization with the principles of server virtualization. The first part focuses on general issues of virtualization. The second part describes the three major virtualization solutions from companies VMware, Microsoft and Citrix. Following evaluation of the collected data.

Keywords

Virtualization, hypervisor, consolidation

Obsah

Seznam zkratk	8
Seznam obrázků	9
Seznam tabulek	9
Seznam grafů	9
Úvod	10
1 Virtualizace	11
1.1 Co je to virtualizace.....	11
1.2 Cíle virtualizace.....	11
1.2.1 Lepší zhodnocení HW	11
1.2.2 Snížení nákladů na provoz.....	12
1.2.3 Zvýšení flexibility	12
1.2.4 Zotavení a mobilita.....	12
1.2.5 Testování a vývoj služeb	12
1.3 Typy serverové virtualizace.....	13
1.3.1 Plná virtualizace	13
1.3.2 Paravirtualizace	14
1.3.3 Emulace	14
1.4 Vrstvy virtualizace.....	15
1.4.1 Virtualizace aplikací (AppV)	15
1.4.2 Virtualizace prezentační vrstvy (PresentV).....	15
1.4.3 Virtualizace desktopů (DeskV)	15
1.4.4 Správa virtualizace (ManageV).....	15
1.4.5 Virtualizace sítí (NetV)	15
1.4.6 Virtualizace uložení (StoreV)	15
1.4.7 Serverová virtualizace (SerV)	16
1.5 Historie serverové virtualizace	16
1.6 Architektura hypervizorů.....	17
1.6.1 Microkernel hypervizor	17
1.6.2 Monolitický hypervizor	17
1.7 Principy virtualizace důležitých součástí systému	18
1.7.1 Virtualizace procesoru.....	18

1.7.2	Virtualizace paměti.....	18
1.7.3	Bounded buffer.....	19
1.8	Hardwarová podpora virtualizace.....	19
1.8.1	Podpora virtualizace od AMD (AMD-V).....	19
1.8.2	Podpora virtualizace od Intel (VT-x).....	21
2	Virtualizační nástroje.....	23
2.1	Microsoft Hyper-V.....	23
2.1.1	Popis implementace.....	24
2.1.2	HW požadavky.....	25
2.2	VMware ESXi.....	26
2.2.1	Popis implementace.....	26
2.3	Citrix XenServer.....	27
2.3.1	Popis implementace.....	28
2.3.2	HW požadavky.....	28
3	Metodika testování.....	29
3.1	Co je powershell.....	29
3.2	Analýza skriptu.....	31
3.3	HW serveru.....	33
4	Zhodnocení nasbíraných dat.....	34
4.1	Porovnání vlastností jednotlivých hypervizorů.....	34
4.2	Zhodnocení hardwarových limitů hypervizorů.....	37
4.3	Testování efektivity využití hardwaru hypervizorem.....	37
4.3.1	Testování práce s diskem.....	37
4.3.2	Zhodnocení testování disku.....	41
4.3.3	Testování CPU.....	41
4.3.4	Zhodnocení testování CPU.....	41
4.3.5	Analýza dat ze skriptu.....	42
	Závěr.....	45
	Literatura.....	47

Seznam zkratk

RAID	Redundant array of independent disks
I/O	Input/output
CTSS	Computer Time-Sharing System
IS	Informační systém
OS	Operační systém
HW	Hardware
SW	Software
VM	Virtual machine

Seznam obrázků

Obrázek 1 - Hypervizor typ 1	13
Obrázek 2 - Hypervizor typ 2	14
Obrázek 3 – Architektura Microsoft Hyper-V	24
Obrázek 4 - Architektura ESXi	26
Obrázek 5 - XenServer architektura	27
Obrázek 6 - Ukázkový příklad: výpis běžících procesů	30
Obrázek 7 - Windows PowerShell ISE	31
Obrázek 8 - Zjištění HW konfigurace serveru	31
Obrázek 9 - Skript sběr dat o procesoru	32
Obrázek 10 - Skript sběr dat o paměti	32
Obrázek 11 - Skript sběr dat o disk	33

Seznam tabulek

Tabulka 1 – maximální hardwarové vlastnosti hypervizorů	34
Tabulka 2 - Čtení 50MB	38
Tabulka 3 - Zápis 50MB	38
Tabulka 4 - Čtení 100MB	39
Tabulka 5 - Zápis 100MB	39
Tabulka 6 - Čtení 500MB	39
Tabulka 7 - Zápis 500MB	40
Tabulka 8 - Čtení 1000MB	40
Tabulka 9 - Zápis 1000MB	40
Tabulka 10 – Čas výpočtu π	41

Seznam grafů

Graf 1 - Maximální počet virtuálních počítačů	34
Graf 2 - Max. počet logických procesorů	35
Graf 3 - Max. počet virtuálních CPU	35
Graf 4 - Max. počet vCPU/VM	36
Graf 5 - Max. velikost RAM	36
Graf 6- Max. velikost RAM/VM	37
Graf 7 - Využití disku - skript	42
Graf 8 - využití paměti RAM-skript	43
Graf 9 - využití paměti - skript	43
Graf 10 - využití procesorového času	44
Graf 11 - využití procesoru - skript	44

Úvod

Virtualizace je spolu cloud computingem¹ jedním z velkých témat informačních technologií dnešních dní. Jedná se o jeden ze způsobů efektivnějšího využívání systémových zdrojů. Důvodem k aplikaci tohoto způsobu řešení je snížení ekonomických nákladů na IT infrastrukturu. Maximalizaci využití hardwarových zdrojů spolu s přidanou hodnotou ve formě posílení zabezpečení, flexibility, dostupnosti služeb a zálohování. Další klíčovou výhodou je zjednodušení správy serveru, kde běží hypervisor a pomocí klientského softwaru je možno se k serveru připojit a spravovat vzdáleně.

Náplní této práce bude seznámení s pojmem virtualizace. Obzvlášť je kladen důraz na serverovou virtualizaci, zmíněny zde budou i další druhy virtualizace, ale jen okrajově. Dále následuje seznámení s vybranými virtualizačními nástroji Citrix XenServer 6.2, Microsoft Server Hyper-V 2008 R2 a VMware ESXi 5.1. V teoretické části bude popsán princip jejich fungování, architektura a hardwarové nároky. V praktické části pak zhodnocení hardwarových limitů, které jsou schopny hypervizory zpracovat. Praktická část obsahuje porovnání z hlediska vytížení hardwaru, na kterém běží. Sledovanými součástmi počítače je tedy využití CPU. Pomocí skriptu a vybraného testovacího softwaru. Druhou sledovanou hodnotou bude procentní využití paměti RAM za stejných podmínek jako v případě procesoru. Posledním kritériem je práce s pevným diskem. Zde budou sledovány rychlosti při práci s diskem.

¹ cloud computing – „vzdálená správa digitálních dat, aplikací a jiných IT služeb, které jsou umístěny v „mraku“ – na specializovaných (internetových) serverech“ [1]

1 Virtualizace

1.1 Co je to virtualizace

Samotný pojem virtuální vyjadřuje něco fyzicky neexistujícího nebo imaginárního [2]. Z tohoto slova se vyvinul pojem virtualizace, který vyjadřuje abstrakci výpočetních zdrojů nebo rozdělení výpočetních zdrojů fyzického systému. Jedná se o vytvoření serveru uvnitř virtuálního prostoru složeného z imaginárních součástí. Což v praxi znamená rozdělení paměti, procesoru, disku a I/O periférií na virtuální části. Které jsou následně přiděleny virtuálním počítačům. V následujících odstavcích jsou uvedeny definice virtualizace velkých firem, které se zabývají vizualizačními technologiemi.

Dle firmy VMware „virtualizace řeší nejpálčivější problém IT infrastruktury rozrůstání infrastruktury, které nutí oddělit na údržbu až 70% svého rozpočtu takže nezbývají zdroje na inovace“ [3]. Snížení nákladů na provoz, lepší zhodnocení HW a zvýšení flexibility jsou body z cílů virtualizace, které charakterizují proč VMware používá tuto definici virtualizace. Detailněji jsou rozepsány tyto body v kapitole 1.2.

Pohled konkurenční firmy Microsoft je „Virtualizace umožnila novou generaci datových center. Namísto běhu dedikovaných serverů, může běžet několik serverů na mnohem výkonnějších serverech. Přitom serverová virtualizace může pomoci dosáhnout snížením nákladů a vyšší efektivnosti.“[4]. Tato definice staví na výhodách, které přináší konsolidace serverů a opět odkazuje na cíle virtualizace. Které jsou rozepsány v kapitole 1.2.

Při virtualizaci se simuluje rozhraní fyzického objektu, vytváří se několik virtuálních objektů pomocí využití multiplexingu fyzických zdrojů. Uživateli je poskytnut simulovaný objekt poskytující stejné chování jako v případě fyzického, přitom jím není, to je důvodem, proč je nezýváno virtuálním objektem. Jedním z důležitých úkolů je ochrana fyzického rozhraní.

Mezi virtualizace nepatří pouze virtualizace celých počítačů. Zde je několik příkladů běžně využívaných technologií. Například RAID dle [5] je příkladem virtualizace obsahující agregaci výpočetních zdrojů. Určitý počet disků je sloučen do jednoho celku, který poskytuje stejné rozhraní každému disku. Dohromady však poskytují lepší služby mezi, které patří vyšší odolnost proti chybám a vyšší rychlost čtení a zápisu.

1.2 Cíle virtualizace

Tato část obsahuje základní cíle vizualizačních řešení a popis principů, kterých se využívá při plnění těchto cílů.

1.2.1 Lepší zhodnocení HW

K odůvodnění tohoto tvrzení je třeba brát na zřetel, na to že kvůli stabilitě serveru se ve většině případů používají servery pro jeden účel. Příkladem může být mail server, DNS a DHCP server, Print server a jiné, které nejsou nijak vysoce vytížené. Využití jejich CPU

se pohybuje zhruba na hodnotě 10% a tak je plýtváno hardwarovými zdroji. Cílem vizualizačního řešení je dosáhnout využití přibližně 60-70%.

Klíčem k pochopení přínosu virtualizace v rámci lepšího zhodnocení HW je vědět co znamená pojem konsolidace serverů. Na serveru časopisu Computerworld je použita tato definice: „Konsolidace představuje proces, kdy jsou sjednoceny či případně centralizovány různá prostředí IT tak, aby je bylo možné provozovat či spravovat jednotným způsobem“[6]. Ve virtuálním prostředí se tedy jedná o běh několika virtuálních serverů uvnitř jednoho fyzického. Tím pádem se zjednodušuje jejich správa, protože je možné je spravovat z jednoho bodu za pomoci klienta. Tímto klientem se ze stanice, která se nalézá v administrativní síti, přihlásí administrátor k serveru a provede na něm příslušné úpravy.

1.2.2 Snížení nákladů na provoz

I v tomto bodě se těží z konsolidace serverů viz kapitola 1.2.1.. Tím že se nemusí provozovat více serverů, klesají náklady na napájení a chlazení serverů. Je možné provozovat více virtuálních serverů na ploše stejné velikosti, aniž bychom prováděli stavební úpravy místnosti, ve které jsou servery umístěny. Centralizace správy dále může poskytnout úspory v rámci personálního vytížení, kdy se menší počet zaměstnanců musí starat o servery a mohou se věnovat dalším úkolům v rámci firmy. [6][7]

1.2.3 Zvýšení flexibility

Pokud potřebujeme přidat akutně další fyzický server, je to problém. Trvá nějakou dobu, než je server dovezen a zapojen. Poté se na něj může začít instalovat software, který potřebujeme. Virtualizace nám umožní flexibilně navýšit počet serverů a omezení jsou pouze zdroji fyzického serveru a licencemi operačního systému. [6][7]

1.2.4 Zotavení a mobilita

Velkou výhodou vizualizačních řešení je podpora snapshotů a migrace za provozu („Live Migration“). V případě snapshotu se jedná o schopnost zachycení stavu paměťového zařízení. Vytvoří se tedy návratový bod („snímek“). Toto řešení například umožňuje obnovu omylem smazaných dat. Je samozřejmé, že vytváření snapshotů vyžaduje dostatek místa na disku a výkon stanice (při tvorbě snímku se musí pozastavit IO operace, aby byla zajištěna konzistence dat). Migrace za provozu umožňuje přemístit běžící virtuální stroj z jednoho hostitele do druhého, aniž by byl vyžadován restart. [8][9]

1.2.5 Testování a vývoj služeb

Testování na virtuálních počítačích je velice jednoduché. Lze zde využít funkce klonování, která vytvoří identickou kopii virtuálního počítače. Na tomto stojí odladit testované služby či si na nečisto vyzkoušet konfiguraci. Pokud již tento stroj není třeba tak ho odstranit. Při tomto způsobu testování se vyvarujeme jakémukoli zásahu do fungujícího systému. Případně jak již bylo zmíněno výše v kapitole 1.2.4., lze využít vlastnosti snapshotů a otestovat daný program či konfiguraci a vrátit se do návratového bodu.

1.3 Typy serverové virtualizace

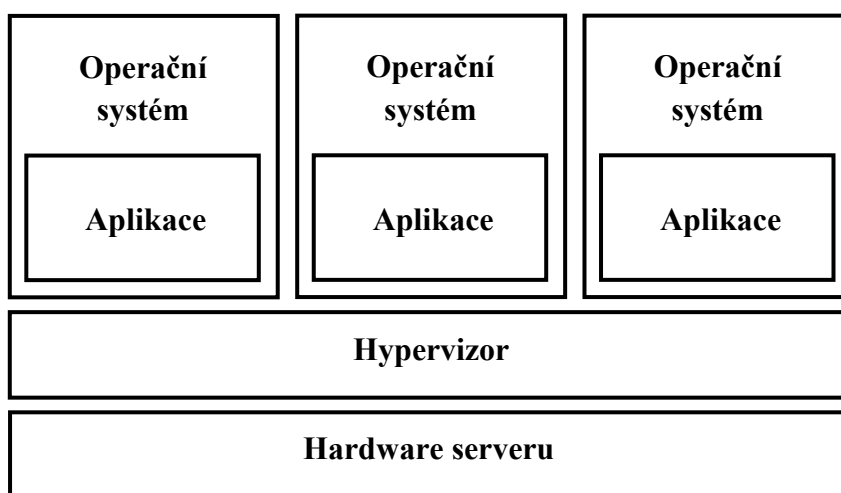
Serverovou virtualizaci dále můžeme rozdělit na plnou virtualizaci (nativní), paravirtualizaci a emulaci. V podkapitolách je vysvětlen přístup těchto řešení k problému.

1.3.1 Plná virtualizace

Pokud virtualizujeme všechny komponenty počítače, jedná se o plnou virtualizaci. V tomto případě je virtualizovaný počítač ve styku pouze s prostředím, které interpretuje jeho požadavky a pracuje s fyzickými zdroji. Tomuto softwaru se říká hypervizor. Operační systém nemůže poznat, že nepřistupuje přímo k hardwaru. Operační systém ani aplikace nemusí být nijak upraveny. Výhoda tohoto řešení spočívá v plné přenositelnosti virtuálního počítače, protože nemusí existovat žádná přímá vazba na konkrétní hardware, na kterém je virtualizace provozována. [10]

1.3.1.1 Hypervizor typu I – Virtualizace s hardwarovou podporou

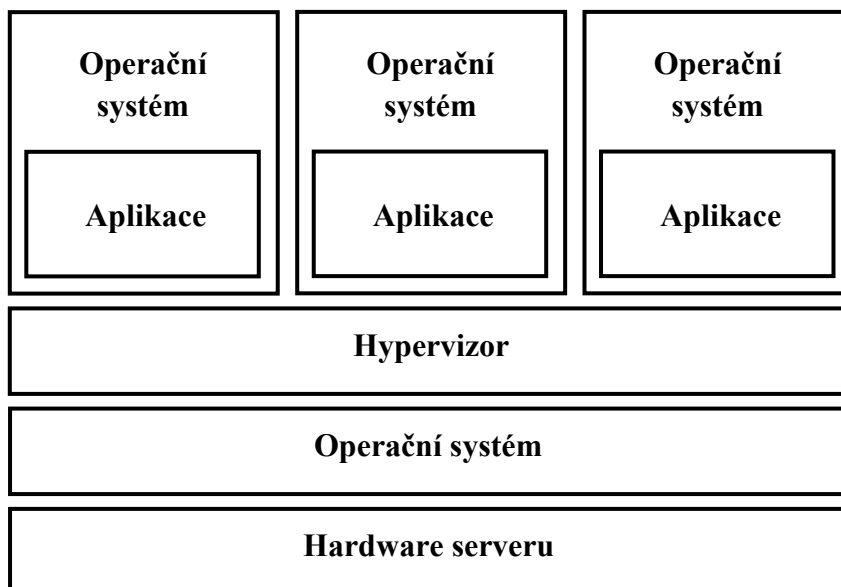
Tento typ hypervizoru běží přímo na hardwaru systému. Hardware je poté poskytnut virtuálním počítačům. Je plýtváno pouze minimálním množstvím zdrojů na hypervizor samotný, aby bylo možné využít výpočetní výkon pro virtuální počítače. Hypervizor tohoto typu je efektivnější než typu 2. Hypervizor je navržen tak, aby monitoroval události, ke kterým dochází uvnitř virtuálního počítače, když je vyžadováno poskytnout nebo zamítnout přidělení zdroje. [9][10]



Obrázek 1 - Hypervizor typ 1, Zdroj: Vlastní

1.3.1.2 Hypervizor typu II – Softwarová virtualizace

V tomto případě je virtualizace spuštěna až nad operačním systémem. Operační systém poskytuje virtualizaci podporu I/O zařízení, správu paměti a další. Tento způsob je vhodný použít pokud není efektivita na prvním místě. Bezpečnost a stabilita tohoto řešení jde ruku v ruce s vlastnostmi operačního systému. Dále je třeba počítat s pravidelnými instalacemi oprav operačního systému, který tvoří mezivrstvu.[8][6]



Obrázek 2 - Hypervizor typ 2, Zdroj: vlastní

1.3.2 Paravirtualizace

Paravirtualizace vyžaduje, aby se některé zdroje, které má systém k dispozici shodovaly. Zaměřuje se na úpravu jádra operačního systému ke zvýšení efektivity. Takto upravené jádro zajistí mnohem hladší běh a tím vyšší výkon. Protože všechny zdroje nejsou virtuální a protože ví, že je počítač virtualizován. [9][10]

Kladem tohoto řešení je bezesporu vyšší rychlost, která pramení z toho, že má jádro OS přístup k fyzickému hardwaru. Hypervizor se v tomto případě pouze kontroluje, kdo dostane přístup ke zdrojům. V porovnání s plnou virtualizací není omezen výběr ovladačů zařízení. Místo toho využívá virtuálního počítače, který má k HW přístup. Nevýhodou je, že pouze modifikované operační systémy mohou být použity při implementaci. Mezi tyto systémy patří modifikovaný Linux, Solaris nebo FreeBSD. Mezi zástupce patří hypervizor Xen. [9][10]

1.3.3 Emulace

Definice slova emulace je „modelování operací jednoho počítače interpretačním programem na jiném počítači, obyč. na bázi mikroprogramů“ [1]. Problémem je, že ve většině případů při tvorbě emulátoru, nejsou k dispozici detailní specifikace architektury systému, programátor si musí vystačit s metodami reverzního inženýrství. Podle nichž je vytvořen emulátor. Z toho plyne, že systém takto emulovaný nemusí být funkční na 100% a samozřejmě bude pomalejší než originál.[10][11]

Důvodem k emulaci může být testování při vývoji softwaru. Zajištění zpětná kompatibility staršího softwaru na novějším hardwaru či podpora softwaru jiné platformy. Příkladem může být DOSbox sloužící k emulaci MS DOS nebo Bochs. [10][11]

1.4 Vrstvy virtualizace

Virtualizaci je možné rozdělit na sedm samostatných vrstev. Každá z těchto vrstev představuje odlišný způsob využití informačních technologií, která je virtualizována. Zde jsou uvedeny spolu s charakteristikou.

1.4.1 Virtualizace aplikací (AppV)

Principiálně se podobá softwarové virtualizaci serverů. Namísto virtualizace celého systému se zaměřuje pouze na jednotlivé aplikace. Služba běžící na serveru, obsahuje aplikace, které se do ní nainstalují. Klient se připojí k serveru a spustí aplikaci. To znamená, že nemusí být u klienta vůbec nainstalovány. Tato služba poskytuje prostředí, ve kterém může spouštět aplikace izolovaně od operačního systému. Výhody jsou redukce konfliktů mezi nainstalovanými aplikacemi, provoz různých verzí aplikací a nejspíše nejdůležitější centralizace administrace. [12][13] [14]

1.4.2 Virtualizace prezentační vrstvy (PresentV)

Typickým zástupcem jsou terminálové služby. Přináší centralizovanou správu poskytovaných aplikací uživatelům. Velkou výhodou je možnost přístupu k terminálové službě z libovolného zařízení a centralizace administrace na několik serverů poskytující tyto služby. [13][14]

1.4.3 Virtualizace desktopů (DeskV)

V tomto případě na serverové infrastruktuře je hypervizor spojen s aplikací na virtualizaci desktopů. Klient se tedy připojuje do datového centra, kde se nalézá jeho softwarové vybavení počítače. A na stole tedy stačí mít pouze tenkého klienta. Výhody, které se dají získat tímto řešením, jsou bezpečnost, centralizovaná správa a jednodušší údržba systémů. [13][14]

1.4.4 Správa virtualizace (ManageV)

Technologie pro správu datového centra jak fyzického tak virtuálního. Základní dělení této vrstvy je fond zdrojů a nabídky virtuálních služeb. Fond zdrojů obsahuje hardwarové zdroje. Nabídky virtuálních služeb obsahují virtuální počítače. Klíčové je zde oddělení těchto vrstev a jejich dostatečné zabezpečení. [13][14]

1.4.5 Virtualizace sítí (NetV)

Rozděluje využití síťových zdrojů pomocí logické segmentace sítě. Umožňuje řízení dostupné šířky pásma dělením na nezávislé kanály a jejich přidělení. Pracuje se všemi zdroji jako s jedním zdrojem prostředků. Jedním ze zástupců je virtuální LAN (VLAN), které segmentují síť. [14]

1.4.6 Virtualizace uložišť (StoreV)

Řeší slučování pevných disků do celku, který se zvětší jeví jednotně. Standardně se jedná o disková pole, která nejsou připojena přímo k serverům a jsou přístupna z více serverů. Principiálně se vytvoří vysokorychlostní pole pro sdílení dat. Uložiště dále lze rozdělit na přímo připojené (DAS), síťově připojené (NAS) nebo síť (SAN). Poskytuje spolu se

serverovou virtualizací možnost thin provisioning² a přiřazení logické jednotky (LUN). Přináší výhody v rámci rychlosti archivace, zálohy a obnovení systémů. [14]

1.4.7 Serverová virtualizace (SerV)

Rozděluje fyzické zdroje serveru na virtuální počítač. Poskytuje možnost virtualizovat platformy x86 a x64. Dělí se dále na softwarovou virtualizace (SoftV) a hardwarovou virtualizaci (HardV). [13][14]

1. Softwarová virtualizace (SoftV): Spuštění operačního systému vizualizovaného systému nad existujícím operačním systémem.
2. Hardwarová virtualizace (HardV): Spuštění přímo nad hardwarem počítače. Tomuto softwaru, nad kterým běží, se říká hypervizor. Úkolem je nabídnout hardwarové zdroje virtualizovanému počítači.

1.5 Historie serverové virtualizace

Historie virtualizace sahá do šedesátých let dvacátého století. V roce 1961 započal vývoj CTSS (počítačové systémy se sdílením času). Supervizor poskytoval určitý počet virtuálních počítačů. Jeden z nich běžel v pozadí a byl připojen k páskovému zařízení. Každý z nich byl počítač IBM 7094 a mohl spravovat jeho regulérní strojový kód navíc jednu instrukci, kterou požívalo velké množství volání supervizoru. Bylo to první požití virtuálních počítačů tehdy nazvané pseudopočítači vůbec. Klíčové vylepšení zde bylo hardwarové oddělení uživatele od supervioru a od každého dalšího.[15]

V roce 1964 použití virtualizace při rozdělování svých sálových počítačů na jednotlivé oddíly (partitioning). Hardwarovou platformou byl IBM System 360 a operačním systémem byl CP-40. Podporoval až 14 souběžně běžících virtuálních počítačů. Každý z těchto počítačů běžel ve stavu, kdy každá I/O operace způsobila výjimku, která byla odchycena kontrolním programem a simulována. Stejným způsobem byl řešen i přístup do paměti. Jednalo se o první hypervizor. [15]

Dalším velkým mezníkem bylo Logical partition (LPAR) v roce 1987 použitým na mainframe IBM. Jedná se o rozdělení hardwaru mezi virtuální stroje. Například pro 2 LPARy je možno každému přidělit jeden nebo více procesorů případně každý z procesorů sdílet. Paměť je rozdělena do rozsahu adres pro každou část a nepřekrývá se, ovšem nepřímo se do paměti druhého LARPu dostat lze. [15]

Spolu se vznikem architektury x86 a modelu klient-server byla koncepce virtualizace potlačena. Se zvyšujícím se výkonem serverů se vynořili následující otázky.

1. Jak efektivně využít dostupné zdroje?
2. Jak co možná nejvíce snížit nároky na řízení a údržbu?
3. Jak zajistit ochranu před výpadkem?

² thin provisioning - Způsob přidělování kapacity disku, součet předělené kapacity může být větší než celková fyzická kapacita.

Došlo se k závěru, že optimálním řešením by mohl být návrat k myšlence použité na mainframe v šedesátých letech. Roku 1999 představila svoje řešení firma VMware se svým VMware Workstation. Toto řešení v sobě spojilo binární překlad a přímé vykonávání instrukcí na procesoru. Společnost si uvědomila potenciál a aplikoval ji na úroveň serverů. [15]

1.6 Architektura hypervizorů

1.6.1 Microkernel hypervizor

Microkernel je minimalistický operační systém poskytující pouze základní služby jako je řízení procesů, mezi-procesní komunikace a paměťová podpora. Zatímco způsoby (správa paměti, plánování, ovladače zařízení) jsou implementována za použití user-mode procesů, tyto komponenty jsou vzájemně od sebe odděleny. Jsou též nazývány jako multi-server systémy. Výhodou tohoto řešení je jeho malá velikost. Mezi zástupce této architektury Microsoft Hyper-V. [16]

1.6.2 Monolitický hypervizor

Monolitické hypervizory obstarávají všechny hardwarové přístupy pro jejich oddíly. Hypervizor musí obsahovat ovladače pro veškerý hardware, ke kterému má přístup. Jsou vyžadovány specializované ovladače napsané přímo pro hypervizor. Tímto požadavkem je limitováno množství hardwaru použitelné na serveru. Ale je zde možnost dosáhnout vyššího stupně optimalizace a tím i vyšší rychlosti. Mezi zástupce této architektury patří Xen a VMware ESX. [16]

1.7 Principy virtualizace důležitých součástí systému

1.7.1 Virtualizace procesoru

Hlavní výzvou při implementaci virtuálních počítačů je najít správné abstrakce dle [5]. Klíčové je pojetí odesílání (SEND) a přijímání (RECIIVE) nad omezenými zásobníky (virtualizované komunikační linky), virtualizovanou paměť a vlákny (virtualizované procesory). Běžně jsou tyto tři abstrakce implementovány Operačním systémem. Každému virtuálnímu stroji je přiřazena vlastní virtuální paměť, virtuální procesor a část datového uložení sloužícího jako jeho virtuální disk.[5]

Prvním krokem při virtualizaci počítače je virtualizace procesoru. Aby byl propůjčen modul s virtuálním procesorem, vytvoří se vykonávací vlákno (thread of execution). Toto vlákno je abstrakcí, která obaluje stav vykonávání aktivního výpočtu. Tento status v sobě uchovává vnitřní proměnné překladače (registry procesoru) které v sobě uchovávají: [5]

1. Odkaz na další část programu
2. Odkaz na prostředí

Překladači je umožněno, aby vlákno kdykoli zastavil a později se k němu vrátil. Tato schopnost je nutná pro multiplexing zdrojů fyzického procesoru.

Postup při spuštění procesu pro vlákno:

1. Načtení programového textu a dat uložené v souborovém systému do paměti
2. Alokace vlákna a start na specifické adrese. Alokace vlákna obsahuje alokování stacku k umožnění vláknu použití procedurálních volání, nastavení SP registru na vrchol stacku a nastavení PC registru na počáteční adresu.

Modul může mít jedno i více vláken k dispozici. Moduly s jedním vláknem jsou běžné a jednoduše implementovatelné, protože pak lze předpokládat, že se bude zpracovávat sériově. Navíc je pro pochopení a implementaci jednodušší pokud má běžet několik vláken současně. Vlákno prochází čtyřmi stádii start, zpracování a výstup vlákna, ukončení. Při použití více vláken je možné přiřadit každému zařízení vlákno. Nebo je možné použít vícebodové spojení k jednomu zařízení kvůli minimalizaci odezvy operací zároveň. Vlákna jsou řízena správcem vláken. Jeho cílem je multiplexing určitého počtu vláken na omezený počet procesorů. Správa je řešena takovým způsobem, aby chyba v jednom vlákně neovlivnila ostatní vlákna. Díky možnosti pozastavení vlákna je správce vláken schopen zastavit vlákno, alokovat procesor jinému vláknu obsloužit ho a následně se vrátit k vláknům odloženému. [5]

1.7.2 Virtualizace paměti

Všechna vlákna sdílejí stejnou fyzickou paměť. Každý procesor používající vlákno odesílá požadavky pro čtení (READ) a zápis (WRITE) po sběrnici spolu s adresou identifikující místo v paměti ze kterého se bude číst nebo se tam bude zapisovat. Sdílení paměti má své

výhody, ale bez kontroly by mohlo snadno dojít k chybám. Jestliže několik vláken má uložená data na stejném místě pak malá chyba, při které se špatně vypočítá adresa, může způsobit ovlivnění dat cizího vlákna je nepřijatelná. Dalším úskalím je velikost samotné paměti a adresního prostoru pro aplikace vyžadující opatrné přidělování kvůli velikosti. K zajištění ochrany paměti se musíme ujistit, že vlákno jednoho modulu nemůže přepsat data jiného modulu. Proto se přiděluje každému z modulů vlastní virtuální paměť. Virtuální paměť poskytuje modulům vlastní adresní prostor. Virtuální správce paměti se pak stará o překlad adres virtuálních na adresy fyzické.[5]

1.7.3 Bounded buffer

pro umožnění komunikace mezi klienty (VM) a službami (VMM) na virtuálních procesorech se využívá odesílání (SEND) a přijímání (RECEIVE) s pomocí tohoto zásobníku. Při zavolání SEND vláknem dojde k pokusu o vložení do zásobníku, pokud je zásobník plný čeká se na uvolnění. Podobná je i situace při RECEIVE, ale s tím že pokud je zásobník prázdný tak se čeká.[5]

1.8 Hardwarová podpora virtualizace

Technologie pro podporu virtualizace obsahují komponenty, které rozšiřují funkčnost v některých oblastech ve virtualizaci. Umožňují jednodušší řešení běhu virtuálních počítačů na fyzické platformě a zajišťují jejich izolaci, ochranu a efektivnější běh. V následujících podkapitolách budou popsány principy technologie firem AMD a Intel, které v rámci svých procesorů poskytují hardwarovou asistenci při virtualizaci.

1.8.1 Podpora virtualizace od AMD (AMD-V)

Rozšíření pro podporu virtualizace od AMD bylo vyvinuto pod kódovým označením „Pacifica“ v roce 2004. Původní označení tohoto rozšíření bylo AMD Secure Virtual Machine (AMD SVM) následně se přejmenovala na AMD-Virtualization. Jedná se o procesory Athlon 64 a 64 X2 pro socket AM2(+) revize F a G, Turion 64 X2, Opteron 2. a 3. generace a novější.

Rozšíření, které Pacifica poskytuje, můžeme rozdělit na dvě doplňující se kategorie a těmi jsou podpora virtualizace a podpora zabezpečení. Do těchto kategorií spadá mechanismus pro rychlé přepínání mezi VM a hypervizorem. Paměťový řadič je integrován do CPU, tím je vyřešeno oddělení jednotlivých systémů hardwarově na rozdíl od konkurenčního řešení od Intelu, kde je oddělení řešeno softwarovou cestou. Dalšími rozšíření budou rozepsány v samostatných podkapitolách. Jsou to tyto rozšíření. Schopnost zachytit vybrané instrukce a události uvnitř VM. Externí ochrana proti (DMA) přístupu do paměti. Asistence pro zachytávání přerušování a přepínání mezi hypervizorem a VM. A poslední podporou virtualizace je TLB se značením VM/host pro redukci virtualizační režie. Poslední úpravou je přidání podpory takzvaného „guest mode“. Do tohoto módu přechází procesor při zavolání instrukce VMRUN. Když se nachází v tomto módu některé x86 instrukce se mění, aby usnadnily virtualizaci. [17]

Dalším zlepšení, které bylo zapracováno do procesorů, za účelem zlepšení virtualizace, je Rapid Virtualization Indexing (RVI), která umožňuje akceleraci výkonu aplikací umožnění hardwarové správy paměti a AMD-V Extended Migration slouží k migraci virtuálních počítačů mezi procesory AMD Opteron. [17]

1.8.1.1 Externí přístupová ochrana

VM mohou dostat přímý přístup do vybraného I/O zařízení. Hardwarová podpora je navržena tak, aby se zamezilo přístupu do paměti, kterou si vybuoval jiná VM nebo hypervizor. Zabezpečením mechanismu virtuálního překladu adres, hypervizor může omezit CPU virtuálního počítače. Avšak pokud má VM k dispozici zařízení s přímým přístupem (DMA) je vyžadována další ochrana. SVM poskytuje několik ochranných domén, které omezují přístup k zařízení na bázi stránek. Toho je dosaženo za pomoci kontrolní logiky v severním můstku. Výchozí hodnotou jsou čtyři domény a každá doména je přiřazena na základě exkluzivním vektoru zařízení (DEV) které specifikuje práva zařízení v doméně. Zařízení samotné je identifikováno za pomoci HyperTransport sběrnice/ID jednotky. DEV je kontingenční pole bitů ve fyzické paměti a každý bit v tomto poli odpovídá 4KB stránce ve fyzické paměti. Takto je fyzická paměť rozdělena po úsecích a informace o ní jsou uchovávány v DEVBASE registrech. Do kterých se nepřímo přistupuje přes funkční blok DEVCTL.[17]

1.8.1.2 Značované TLB (Tagged Translation Lookaside Buffer)

Využívá se zde způsobu, při kterém je hypervizor mapován v odlišném adresovém prostoru než je tomu u VM. Aby se redukovala režie při přepínání, je TLB značkována identifikátory pozice v paměti (ASID), který rozlišuje mezi zápisy hypervizoru a zápisy VM. Hypervizor může používat při tomto postupu několik stínových stránkovacích tabulek a pro každou stránku je vyhrazen pouze jeden ASID. Tak je umožněno přepínání mezi procesy bez toho, aby se tabulky zahazovali při přepnutí. [17]

1.8.1.3 Podpora přerušení

Pro zajištění efektivní virtualizace a přerušení je zavedena podpora následujících rozšíření pod kontrolou VMCB příznaků. [17]

Zachycení doručení fyzického přerušení je funkce, při které hypervizor zažádá o fyzické přerušení. Toto volání je provedeno, aby ukončilo běžící proces VM. Tato funkce dovoluje hypervizoru pomocí hardwarového přerušení přerušit libovolný proces VM. [17]

Virtuální přerušení je funkce, při které hypervizor může vložit virtuální přerušení do CPU virtuálního počítače. Pod kontrolou hypervizoru použije virtuální kopii příznaku EFLAG.IF s přerušovací maskovacím bitem a s pomocí APIC registru priority úkolů je transparentně řízen namísto toho, aby se používalo HW přerušení. [17]

Sdílení fyzických APIC je funkce, při které hypervizor umožní sdílet stejnou fyzickou APIC. Řeší to chování škodlivých či závadných VM, které by tak mohli zanechat přerušení s vysokou prioritou bez odpovědi a tím zablokovat použití přerušení pro ostatní. [17]

1.8.2 Podpora virtualizace od Intel (VT-x)

Rozšíření pro podporu virtualizace od Intelu bylo vyvinuto pod kódovým označením „Vanderpool“. Toto vylepšení bylo poprvé použito u procesoru Pentium 4 (model 662 a 672). [18]

Cíly Intel-VT je redukovat komplexnost hypervizoru, uzavřením hardwarových „vizualizačních děr“ v designu. Redukovat potřebu specifického zařízení navázané na hypervizor. Zlepšení rentability, zabezpečení a ochrany. Poskytnutí nové kontroly nad zařízením DMA a přerušení. Zlepšení funkcionality poskytováním podpory pro starší (nemodifikované) OS. Povolením přístupu skrz přístup do I/O zařízení. Zlepšením výkonu eliminací nepotřebných přepínání u hypervizoru. Přidání nového překladového mechanismu adres (pro CPU a ostatní zařízení). Redukování nároků na paměť (překladem kódu a použitím stínových tabulek).[18][19]

1.8.2.1 Virtual Machine Extension (VMX)

Podpora virtualizace je poskytnuta za pomoci integrace VMX operací. Jedná se o balíček deseti specifických instrukcí určené pro virtualizaci. Tyto operace lze rozdělit na dva druhy operací VMX root operace a VMX non-root operace. Hypervizory využívají VMX root operace a VM používají VMX non-root operace. Přechod mezi root a non-root operacemi se nazývá VMX transition. Jsou dva druhy VMX transition první se jmenuje VM enter přepínající z VMX non-root do VMX root a druhým je VM exit přepínající z VMX root do VMX non-root.[19]

1.8.2.2 Extended Page Tables (EPT)

Umožňuje každému virtuálnímu počítači mít vlastní stránkovací tabulku, která obsahuje adresy paměti. Bez tohoto rozšíření hypervizor musí přerušit práci s VM, aby došlo k překladu adresy. Toto řešení snižuje režii, která by byla jinak vyšší. Při použití TLB (Translation Lookaside Buffer) cache se předpokládá další roli k hlídání virtuální paměti a fyzické paměti OS virtuálního počítače. Každý virtuální počítač je držen v TLB s přiděleným identifikátorem adresního prostoru. Při použití tohoto identifikátoru může za pomoci TLB vystopovat VM a nemusí se vyprazdňovat TLB cache při přepnutí VM. Výhodou tohoto řešení je snížení potřeby synchronizace stínových stránek. Počet stínových stránek je pak závislý na počtu VM běžících na serveru. Eliminace synchronizace produkuje zvýšení výkonu serveru.[19]

1.8.2.3 Intel Virtualization Technology for Directed I/O (Intel VT-d)

Klíčovým úkolem této technologie je oddělení přístupu VM na zařízení, které má k dispozici jiná VM. Zde jsou čtyři klíčové schopnosti, které nabízí. Přidělování I/O zařízení. Tato podpora dovoluje administrátorovi přidělit libovolné I/O zařízení k libovolné VM. Dále je to DMA přemapování podporující adresní překlad pro zařízení podporující DMA datový transfer. Přemapování přerušení. Poskytující směrování a izolaci přerušení od zařízení. Podpora spolehlivosti, která nahlašuje a zaznamenává chyby DMA a přerušení které by jinak mohli způsobit komplikace v paměti. [19]

1.8.2.4 Intel Virtual Machine Control Structure Shadowing (Intel VMCS Shadowing)

Slouží pro přímé volání instrukcí procesoru virtuálním strojem, není nutné přepínat na hypervisor pro obsluhu volání. Přístupy na hardware jsou předem vytvořeny a tím se zkracuje čas, podle Intelu je náročnost na čas 15x menší než v případě přepínání.[18][19]

2 Virtualizační nástroje

Tento oddíl obsahuje základní informace o vizualizačních nástrojích použitých při testování a jejich charakteristiku. Vybrány jsou nástroje firem, které považuje analytická skupina Gartner za leadery v oblasti virtualizace [20].

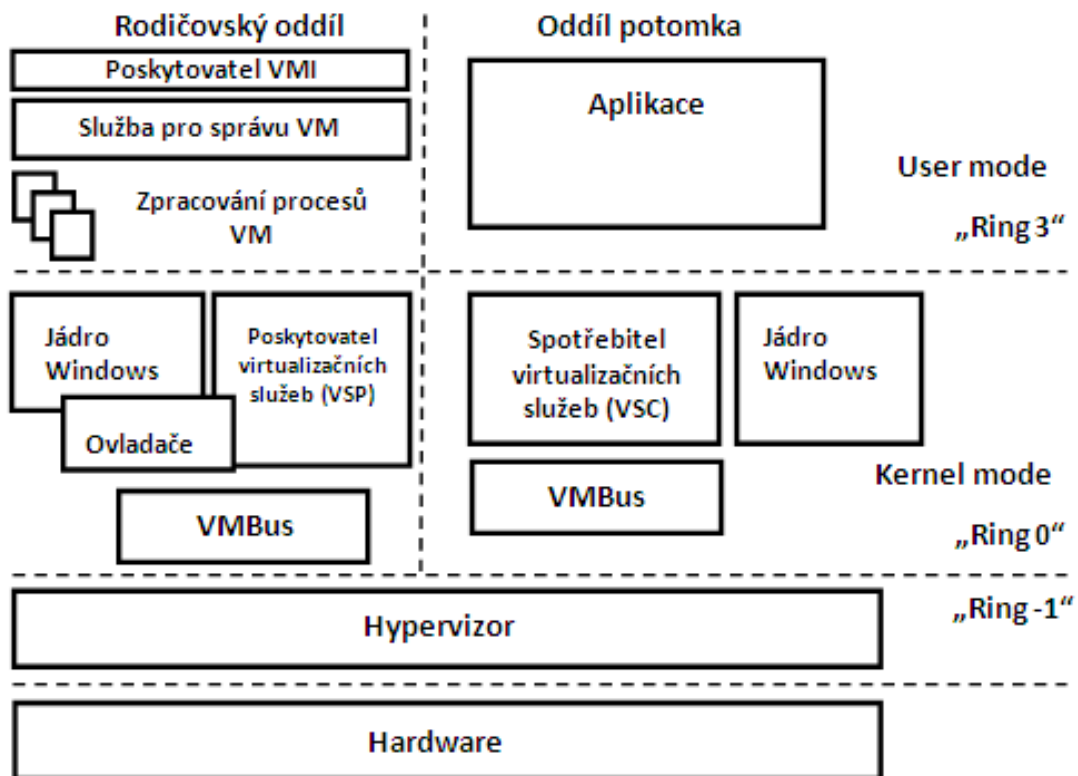
Nástroje jsou vybrány napříč typy serverové virtualizace. Microsoft Hyper-V a VMware ESXi jsou zástupci plné virtualizace prvního typu. Citrix XenServer využívá hypervizoru Xen, který je zástupcem paravirtualizace viz kapitola 1.3. Hypervizory se liší i v architektuře kterou jsou implementovány. Microsoft Hyper-V je postaveno na microkernel architektuře. VMware ESXi je spolu s hypervizorem Xen naopak zástupcem monolitického hypervizoru viz kapitola 1.6. Rozdílnost, pojetí jakým způsobem přistupují k virtualizaci, bude dobře vidět na obrázcích v následujících podkapitolách, kde na nich bude popsána jejich implementace.

2.1 Microsoft Hyper-V

Vydán v roce 2008 jako update Windows Server 2008. K dispozici i jako samostatný produkt (Microsoft Hyper-V Server 2008 R2) nebo spolu s Windows Server 2008 nebo Windows Server 2008 R2. Na trh virtualizace Microsoft vstoupil koupením firmy Connectix v roce 2003, která stála za projektem Virtual PC. Využívá hypervizor typu 1 a microkernel architekturu. Nespornou výhodou je provázanost s produkty Windows. [21]

Při instalaci tohoto produktu je třeba počítat s tím, že se instaluje celý upravený operační systém Microsoft Windows Server 2008 R2 či Microsoft Windows Server 2012 (podle zvolené instalace). Který bude sloužit, jako rodičovský oddíl viz obrázek 3. Samotná instalace hypervizoru tedy je v tomto případě asi nejpomalejší.

Produkt umožňuje správu serveru za pomoci sady nástrojů pro konfiguraci serveru. Lokálně pomocí Hyper-V Manager konzole z jiného Hyper-V Server 2008 R2 nebo pomocí balíčku Remote Server Administration Tools (RSAT), která je určena pro Microsoft Windows 7.



Obrázek 3 – Architektura Microsoft Hyper-V, Zdroj: vlastní

2.1.1 Popis implementace

Implementace izolace virtuálního počítače je zde vyřešena pomocí oddílů. V tomto případě oddíl je logická jednotka podporovaná hypervizorem ve které běží virtualizovaný systém. Existuje minimálně jeden rodičovský oddíl, ve kterém běží virtualizovaný zásobník a má přímý přístup k hardwarovým zařízením. Rodičovský oddíl může vytvořit oddíl potomka obsahující OS virtuálního počítače. Tato tvorba probíhá pomocí hypercallů API. Potomek nemá přímý přístup k fyzickému procesoru a nedokáže tedy obsloužit přerušení. Namísto toho vidí virtuální adresy, které obsahují přidělený adresní prostor paměti. Hypervizor obstarává přerušení procesoru a přesměrovává je k odpovídajícím oddílům za použití „Synthetic Interup Controler“ (SynIC). Oddíly potomků vidí zdroje jako virtuální zařízení. Jakýkoli požadavek o virtuální zařízení je přesměrováno pomocí virtuální sběrnice VM Bus k rodičovskému zařízení. VM Bus je logický kanál umožňující mezi-oddílovou komunikaci. Rodičovské oddíly obsahují Poskytovatel vizualizačních služeb (VSP), který přesměrovává požadavky k VSP pomocí virtuální sběrnice. Celý tento proces je transparentní k hostujícímu operačnímu systému. [21][22]

Virtuální zařízení může využít vylepšení nazývané „Enlightened I/O“. Je to specializovaná implementace, která bere ohled na virtualizaci vysoké úrovně komunikačního protokolu jako je SCSI k přímé výhodě umožňující překročit emulační vrstvu tohoto zařízení. Umožní tak efektivnější komunikaci vyžaduje však přímou podporu v OS. Mezi systémy kde je tato podpora obsažena patří Windows Server 2008 R2, Windows Server 2008, Windows 7, Windows Vista, Red Hat Enterprise Linux a SUSE Linux. [21][22]

2.1.2 HW požadavky

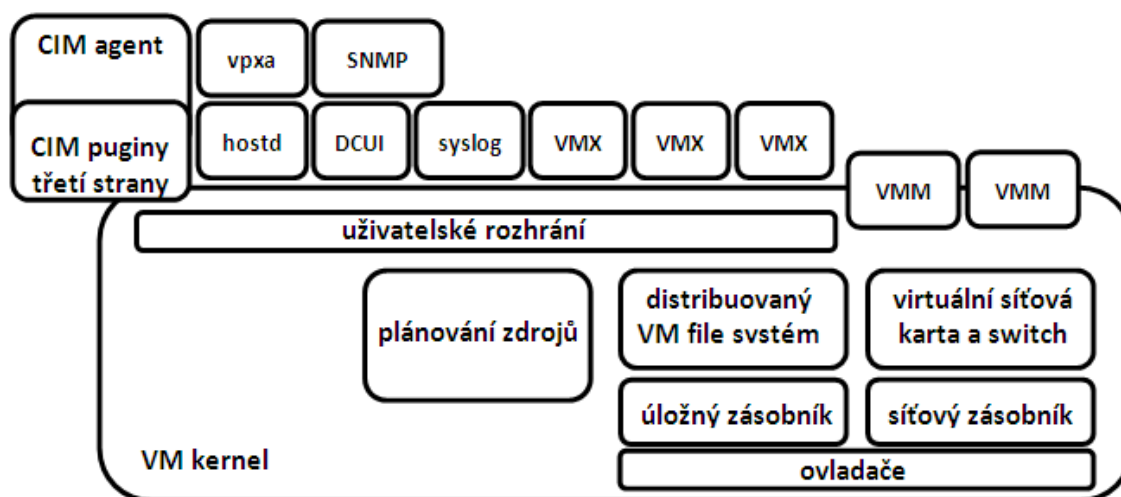
Požadavky na hardware serveru jsou rozděleny do čtyř kategorií, těm jsou CPU, RAM, HDD a HW podpora virtualizace. V rámci nároku na CPU je vyžadován 64-bitový procesor architektury x86 s podporou jako je, Intel VT nebo AMD Virtualization (AMD-V, "Pacifica"). Procesor musí mít NX bit a musí být povolena Hardware Data Execution Prevention (DEP). Doporučením je podpora překladu adres druhé úrovně (EPT). Naprosto minimální nároky na paměť RAM pro běh hypervizoru jsou 2GB, ale každá VM vyžaduje svoji vlastní paměť. Minimální množství paměti je tedy reálně vyšší. Maximálním celkovému množství RAM, které je schopno obhospodařit, jsou 2 TB a to v případě Windows Server 2008 Hyper-V. Maximum pro běžící VM pod Windows Server 2008 Standard (x64) Hyper-V full GUI nebo Core jsou 31GB pro VM a 1GB pro rodičovský OS. Minimální nároky na diskové zdroje jsou 8GB doporučená hodnota je od 20GB výše.

2.2 VMware ESXi

Jedná se o řešení leadera na poli vizualizačních technologií. První verze ESX byla vydána v roce 2001. Stejně jako u ostatních dvou se jedná o nativní hypervizor typu 1. Zatím poslední stabilní verze je 5.1.0a. ESX využívá monolitické architektury hypervisoru.

Instalace hypervizoru je u tohoto produktu velmi rychlá. Instalátor během instalace požaduje pouze určení diskového prostoru, do kterého se bude hypervizor instalovat, několik upozornění, licenční ujednání, ale nejdůležitější částí je konfigurace root hesla. Zbytek je pak plně automatický. Instalována je pouze tenká vrstva hypervizoru bez konzole a tím pádem velice kompaktní. Po dokončení instalace je spuštěn hypervizor, na kterém se nastaví pouze síť pro správu a síťový název serveru a může se začít pracovat.

Při práci se serverem se využívá klienta VMware vSphere Client, který požaduje identifikační označení serveru a jeho nastavené heslo pro přístup k hypervizoru. Po přihlášení je zde přehledné prostředí, které umožňuje vytvářet VM, sledovat vytížení hardwaru, vytvářet snapshoty a jiné operace.



Obrázek 4 - Architektura ESXi. Zdroj: vlastní

2.2.1 Popis implementace

ESXi se využívá upraveného jádra OS na kterém je hypervisor postaven, firmou označován též jako VM kernel. Linuxové jádro startuje jako první a využívá rozsah specializovaných vizualizovaných komponent včetně VM kernel komponenty. Dříve se u produktů VMware využívalo přístupu, při kterém se Linuxové jádro se stávalo prvním virtuálním počítačem a byla volána servisní konzole. VMware ESXi, přináší změnu architektury při, které již není potřeba servisní konzole. Využívá se výše zmíněného VM kernelu, který běží přímo na hardwaru serveru. VM kernel vytváří instance VMM (Virtual Machine Monitor) pro každý virtuální počítač, který obsluhuje. Tímto způsobem se izolují od sebe VM. VM kernel pak využívá zásobníků, které jsou plněny požadavky VMM

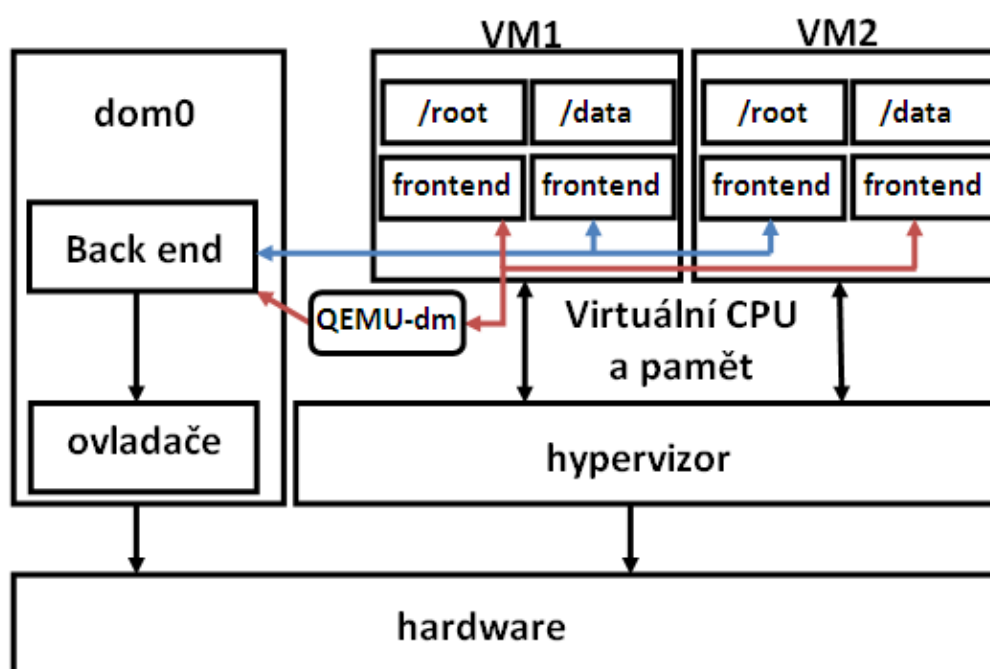
a plánováno jejich obsluhu pomocí funkčního bloku „plánování zdrojů“, který můžeme vidět na obrázku 4. [23][24]

2.3 Citrix XenServer

Xen je hypervisor typu 1, který vznikl na universitě v Cambridge. Jeho první verze vyšla v roce 2003. Od roku 2010 publikován pod licencí GNU GPLv2. Je základem nejen pro Citrix XenServer ale i pro Oracle VM, Virtual Iron, Sun xVM.[25]

Instalace tohoto produktu byla ze všech tří produktů nejrychlejší. Instalátor si v první fázi ohledal osazený hardware serveru. Následovala možnost umístění, hesla a časové zóny a začala instalace. Po instalaci samotného hypervisoru byla možnost doinstalování rozšíření pro optimalizaci. Po startu hypervisoru opět stačí pouze nastavit síťový název a síťovou adresu, poté je připraven server pro práci s hardwarem.

Správa serveru probíhá pomocí klienta Citrix XenCenter, který se chová podobně jako VMware vSphere Client s rozdílem, že se do fondu zdrojů přidávají servery a disková pole. Ty se poté dají dále spravovat. I zde je možnost sledovat využití hardware, měnit konfiguraci, vytvářet snapshoty a další funkce.



Obrázek 5 - XenServer architektura, Zdroj: vlastní

2.3.1 Popis implementace

Architekturu hypervizoru Xen, tvoří privilegované domény (dom0) a neprivilegované domény (domU). Privilegovaná doména je modifikovaným jádrem založena na Linuxu (není to striktně dáno, existuje alternativa v modifikované variantě NetBSD nebo Solaris) fungujícím jako servisní konzole. Má přístupová práva pro I/O zdroji a schopnost interakce s ostatními virtuálními počítači. Obsahuje ovladače pro podporu sítě a virtuálních disků pro neprivilegované domény. V případě síťové podpory se jedná o „Network backend driver“ komunikující přímo mezi síťovým hardwarem a virtuálními počítači. Pro disky o „Block backend driver“ starající se o čtení a zápis na fyzické disky. Neprivilegované domény nemají přímý přístup k hardwaru. Všechny paravirtualizované systémy jsou označeny jako „PV Guests“ zde běží modifikované systémy umožňující tento běh (Linux, Solaris, FreeBSD a jiné UNIXové systémy). V tomto případě virtuální stroj ví, že nemá přímý přístup k hardwaru a že běží další na stejném počítači. Obsahuje „PV Network Driver“ a „PV Block Driver“. Všechny plně virtualizované systémy jako „HVM Guest“ jedná se o nijak neupravené systémy jako Windows. Virtuální stroje nevědí, že sdílí hardware s dalšími. Zde nejsou žádné ovladače, ale pro každou doménu se spustí démon uvnitř Dom0 pro požadavky od neprivilegovaných domén.[25][26]

2.3.2 HW požadavky

Zde jsou hardwarové požadavky hypervizoru Xen. Požadavky na hardware serveru jsou rozděleny do čtyř kategorií, těmi jsou CPU, RAM, HDD a HW podpora virtualizace. V rámci nároků na CPU to je jeden nebo více 64-bitových procesorů architektury x86, minimálně na frekvenci 1,5GHz doporučený je však procesor s rychlostí 2GHz s více jádry. Dalšími nároky na CPU je podpora Intel VT a AMD-V. Minimální požadavky na RAM jsou 1GB doporučené jsou 2GB a více. Diskové zdroje serveru by měli být tvořeny lokálním uložištěm s minimálním množstvím místa 16 GB a doporučeným 60 GB a více. Při instalaci se vytvoří dva 4GB oddíly pro XenServer Host Control Domain; zbývající místo je k dispozici pro virtuální počítače.[27][28]

3 Metodika testování

Cílem testování je objektivní zhodnocení nároků jednotlivých virtualizačních řešení na hardwarové zdroje a efektivnosti práce s nimi. Zvoleny byly položky, které je možno porovnat a které tvoří páteřní funkčnost serveru. V první fázi jsou porovnány maximální velikosti fondu zdrojů, které jsou schopny pojmout, a pracovat s nimi viz kapitola 4.1. Důvodem rozebrání této části je zobrazení hranice, které mohou dosáhnout konfigurace serveru, v rámci vizualizačního řešení. Druhou část tvoří sada testů práce s pevným diskem pomocí programu Crystal Disk Mark. Detailněji bude tento postup popsán v kapitole 4.3.1. Důvodem tohoto testu je zobrazení rozdílů v efektivitě práce s diskem. Třetí částí je testování výkonu CPU za pomoci programu SuperPi viz kapitola 4.3.3. Při kterém by měla být vidět optimalizace hypervizoru a jeho režie při práci s procesorem. Posledními třetí částí obsahují výsledky a sumarizaci dat posbíraných pomocí powershell skriptu. Způsoby, kterými tyto data jsou posbírány, jsou rozebrány v této kapitole. Která nejdříve představí nástroj, následovaným analýzou částí skriptu.

3.1 Co je powershell

Powershell je rozšiřitelný shell s podporou skriptovacího jazyku od společnosti Microsoft. Je založen na platformě .NET Framework. Na rozdíl od textových rour v shellu, který podporují unixové systémy je zde podpora rour objektových. Díky tomu že je postaven na .NET Framework jsou v powershellu přístupné veškeré informace ke kterým má přístup OS. Různé druhy informací a specializovaných činností jsou přístupné pomocí takzvaných cmdletů. Jedná se o specializované třídy .NET pro různé operace. Jejich názvy se skládají z <slovesa>-<podstatného jména> jako například Get-Service, který bude v ukázkovém příkazu pro powershell níže. Na výstupu je vždy objekt. Který je možná nasměrovat pomocí roury na vstup dalšího cmdletu a zřetěžit tak příkazy. Samozřejmě je zde možnost si vytvářet vlastní cmdlety. [29]

Zde je ukázkový příklad, jeho cílem je získání výpisu služeb, které právě běží. Syntaxe bude následující: `Get-Service | Where-Object {$_.Status -eq „Running“}`

```

Administrator: Windows PowerShell
PS C:\Documents and Settings\sidonius> Get-Service * | Where-Object {$_.Status -eq "Running"}

Status Name DisplayName
-----
Running AcrSch2Svc Acronis Scheduler2 Service
Running AdvancedSystemC... Advanced SystemCare Service
Running afcdpsrv Acronis Nonstop Backup service
Running ALG Služba brány aplikačního rozhraní
Running AudioSrv Zvuk systému Windows
Running avast! Antivirus avast! Antivirus
Running BBUpdate BBUpdate
Running Browser Prohledávání počítačů
Running btwins Bluetooth Service
Running CryptSvc Šifrování
Running DcomLaunch Spouštěč procesů serveru DCOM
Running Dhcp Klient DHCP
Running Dnscache Klient DNS
Running Eventlog Protokol událostí
Running EventSystem Systém událostí modelu COM+
Running FastUserSwitchi... Kompatibilita pro rychlé přepínání ...
Running HidSrv HID Input Service
Running JavaQuickStarte... Java Quick Starter
Running LanmanServer Server
Running lanmanworkstation Pracovní stanice
Running MSSQL$SQLEXPRESS SQL Server (SQLEXPRESS)
Running Netman Síťová přípojení
Running Nla Sledování umístění v síti (NLA)
Running PlugPlay Plug and Play
Running PolicyAgent Služby IPSEC
Running ProtectedStorage Chráněné úložiště
Running RasMan Správce vzdáleného přístupu
Running RpcSs Uzdálené volání procedur (RPC)
Running SamSs Správce zabezpečení účtů
Running seclogon Sekundární přihlašování
Running SENS Oznamování systémových událostí
Running SharedAccess Brána Firewall / Sdílení připojení ...
Running ShellHWDetection Rozpoznávání hardwaru
Running Schedule Plánovač úloh
Running Spooler Zařazování tisku

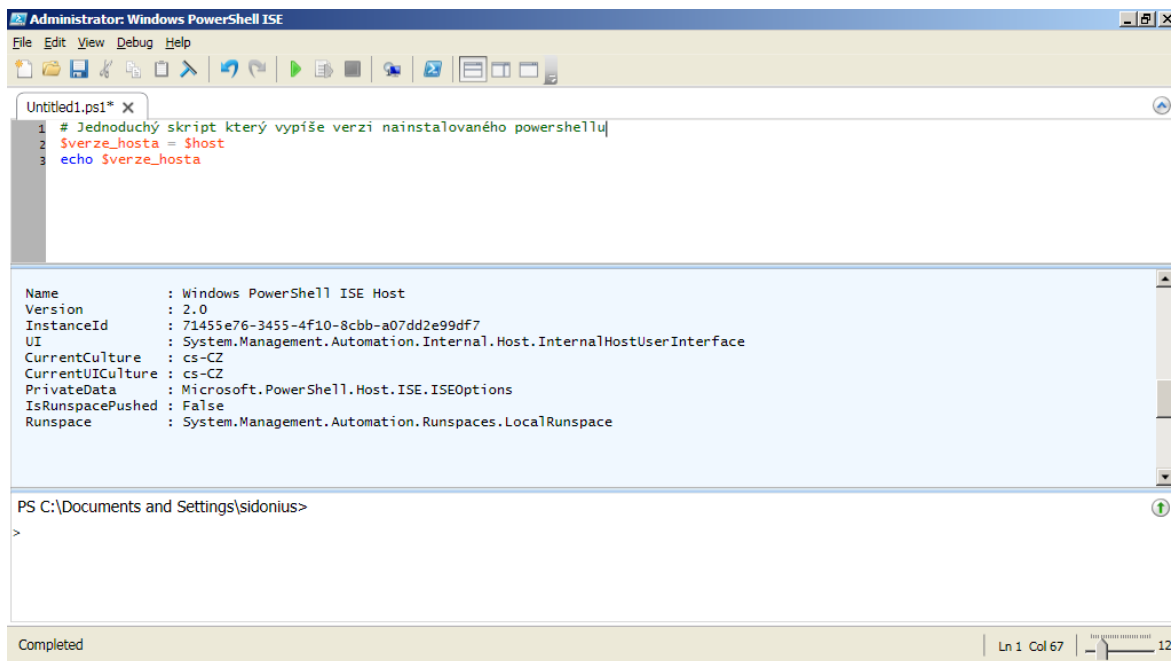
```

Obrázek 6 - Ukázkový příklad: výpis běžících procesů, Zdroj: vlastní

Tento příkaz se skládá z volání cmdletu Get-Service, který předá na vstup Where-Object veškeré běžící procesy. Ty jsou následně filtrovány na rovnost s řetězcem „Running“. Na Obrázku výše je vidět výstup.

Verze 1.0 byla vydána v roce 2006 pro Windows XP SP2/SP3, Windows Server 2003 a Windows Vista. A přinesla základní funkčnost, která je zmíněná na začátku toho oddílu.[24]

Verze 2.0 je integrována ve Windows 7 a Windows Server 2008 R2 také je k dispozici pro Windows XP se Service Pack 3, Windows Server 2003 se Service Pack 2 a Windows Vista se Service Pack 1. Přináší změny ve skriptovacím jazyku a API také přidává více než 240 cmdletů. Nové operátory, podpora modulů, do kterých lze organizovat skripty, podpora debugingu a několik dalších vylepšení. A dle mého názoru nejdůležitější je vývojové prostředí Windows Powershell ISE.[24]



Obrázek 7 - Windows PowerShell ISE, Zdroj: vlastní

Ve výchozím nastavení je rozložení okna tak jak je vidět na Obrázku 7. V horní části je samotné editační pole skriptu. Uprostřed je výstup powershellu. A ve třetí části se nalézá klasická příkazová řádka, jejíž výstup je zobrazen uprostřed. Lze tedy jednoduše si vypsat nápovědu k příslušnému cmdletu aniž by bylo nutné přepínat mezi okny. Zároveň je zde zavedena podpora debuggeru. Který usnadňuje ladění složitějších skriptů.

3.2 Analýza skriptu

Níže jsou části skriptu, které slouží ke sběru dat o hostitelském systému. Postupně zde budou uvedeny části skriptu s podrobným popisem funkčnosti.

```
$System = Get-WmiObject -Class win32_computersystem
$processor = Get-WmiObject -Class win32_processor
$video = Get-WmiObject -Class win32_videocontroller
$disk = Get-WmiObject -Class win32_LogicalDisk -Filter "DeviceID='C:'"

$informace = ""
$informace+="CPU: `n"
$informace+="Název procesoru:"+$processor.Name+"`n"
$informace+="Rychlost procesoru:"+$processor.CurrentClockspeed+"`n"
$informace+="Pocet jader:"+$processor.NumberOfCores+"`n"

$informace+="Grafická pamet (MB):"+$video.AdapterRAM/1mb+"`n"
$informace+="RAM(MB):"+$System.TotalPhysicalMemory/1mb+"`n"
$informace+="Disk(MB):"+$disk.Size/1mb+"`n"

echo $informace
```

Obrázek 8 - Zjištění HW konfigurace serveru, Zdroj: vlastní

Do proměnné \$System je pomocí cmdletu Get-WmiObject získána instance třídy win32_computersystem. Která v sobě uchovává základní informace o systému jako je jeho název, model výrobce, síťovou skupinu a hlavně celkovou velikost paměti RAM. Obdobně je do proměnné \$processor pomocí cmdletu Get-WmiObject získána instance třídy

win32_processor. Zde se nalézají informace jako je aktuální frekvence procesoru, velikost L2 a L3 cache, název, velikost adresního prostoru, počet jader a jiné informace týkající se procesoru. Do proměnné \$video je pomocí cmdletu Get-WmiObject získána instance třídy win32_videocontroller. Obsahuje informace o grafické kartě jako je frekvence, velikost paměti, verze ovladačů, datum jejich vydání a další užitečné informace. Do proměnné \$disk je pomocí cmdletu Get-WmiObject získána instance třídy Win32_LogicalDisk. Jak napovídá název, jsou zde uchovány informace o disku, jeho velikost, volné místo na disku. Pomocí těchto dat se získá představa o HW vybavení počítačem, na kterém je spuštěn skript.

```
# vytizeni procesoru
echo "Sbírám data o využití procesoru - ukládáno do souboru c:\scripts\vytizeni_procesoru.csv"
Get-counter -Counter "\\win-c9zwxj8dhjn\procesor(_total)\% času procesoru" -MaxSamples 200 |
convertto-csv -useculture | Out-File c:\scripts\vytizeni_procesoru.csv
```

Obrázek 9 - Skript sběr dat o procesoru, Zdroj: vlastní

Nejdříve se vypíše na obrazovku řetězec "Sbírám data o využití procesoru - ukládáno do souboru c:\scripts\vytizeni_procesoru.csv". Cmdlet Get-Counter získá čítač ze systému parametr -Counter definuje, který z čítačů bude vybrán. Za tímto parametrem následuje řetězec specifikující, který ze systémových čítačů se jedná zde je to " \\win-c9zwxj8dhjn\procesor(_total)\% času procesoru". WIN-C9ZWXJ8DHJN je názvem systému. Následuje výběr typu čítače, zde je vybrán poměr celkového využití procesoru děleno časem procesoru. MaxSamples omezuje počet vzorků, které budou zpracovány před dokončením příkazu zde je to 200 vzorků. Na vstup convertto-csv jsou předány hodnoty čítače vyjádřené v procentech a zde jsou data dle formátu .csv oddělovány středníkem. Parametrem -useculture se specifikuje oddělovač. Poslednímu zřetězení, které zpracovává tento příkaz je Out-File sloužící k uložení dat na vstup do souboru specifikovaného dále. Zde je cesta souboru c:\scripts\vytizeni_procesoru.csv.

```
# vyuziti pameti
echo "Sbírám data o využití přidělené paměti - ukládáno do c:\scripts\vytizeni_pameti.csv"
Get-counter -Counter "\\win-c9zwxj8dhjn\pamet%\% využití svěřených bajtů" -MaxSamples 200 |
convertto-csv -useculture | Out-File c:\scripts\vytizeni_pameti.csv
```

Obrázek 10 - Skript sběr dat o paměti, Zdroj: vlastní

Nejdříve se vypíše na obrazovku řetězec “ Sbíráám data o využití přidělené paměti - ukládáno do c:\scripts\vytizeni_pameti.csv“. Cdmllet Get-Counter získá čítač ze systému parametr –Counter definuje, který z čítačů bude vybrán. Za tímto parametrem následuje řetězec specifikující, který ze systémových čítačů se jedná zde je to “ \\win-c9zwxj8dhjn\pamět\% využití svěřených bajtů“. WIN-C9ZWXJ8DHJN je názvem systému. Následuje výběr typu čítače, zde je vybráno k zjištění procentní využití paměti RAM. MaxSamples omezuje počet vzorků, které budou zpracovány před dokončením příkazu zde je to 200 vzorků. Na vstup convertto-csv jsou předány hodnoty čítače vyjádřené v procentech a zde jsou data dle formátu .csv oddělovány středníkem. Parametrem –useculture se specifikuje oddělovač. Poslednímu zřetězení, které zpracovává tento příkaz je Out-File sloužící k uložení dat na vstupu do souboru specifikovaného dále. Zde je cesta souboru c:\scripts\vytizeni_pameti.csv.

```
# využití disku
echo "Sbíráám data o aktivitě disku - ukládáno do c:\scripts\vytizeni_disku.csv"
Get-counter -Counter "\\win-c9zwxj8dhjn\fyzičký disk(_total)\% času disku" -MaxSamples 200 |
convertto-csv -useculture | Out-File c:\scripts\vytizeni_disku.csv
```

Obrázek 11 - Skript sběr dat o disk, Zdroj: vlastní

Nejdříve se vypíše na obrazovku řetězec „Sbíráám data o aktivitě disku - ukládáno do c:\scripts\vytizeni_disku.csv“. Cdmllet Get-Counter získá čítač ze systému parametr – Counter definuje, který z čítačů bude vybrán. Za tímto parametrem následuje řetězec specifikující, který ze systémových čítačů se jedná zde je to “ \\win-c9zwxj8dhjn\fyzičký disk(_total)\% času disku“. WIN-C9ZWXJ8DHJN je názvem systému. Následuje výběr typu čítače, zde se zjišťuje procentuální hodnota využití času disku. MaxSamples omezuje počet vzorků, které budou zpracovány před dokončením příkazu zde je to 200 vzorků. Na vstup convertto-csv jsou předány hodnoty čítače vyjádřené v procentech a zde jsou data dle formátu .csv oddělovány středníkem. Parametrem –useculture se specifikuje oddělovač. Poslednímu zřetězení, které zpracovává tento příkaz je Out-File sloužící k uložení dat na vstupu do souboru specifikovaného dále. Cesta k souboru je c:\scripts\vytizeni_disku.csv.

3.3 HW serveru

První část skriptu získává základní informace o HW vybavení počítače. Je zde zjištěn název procesoru, počet jader, maximální frekvence. Procesorem v serveru je Intel Xeon E5335 (8M Cache, 2.00 GHz, 1333 MHz FSB). Server byl vybaven 8GB paměti RAM. Dále byl server vybaven 6-ti SATA disky zapojených do řadiče RAID s konfigurací RAID 5 o celkové kapacitě 2TB.

4 Zhodnocení nasbíraných dat

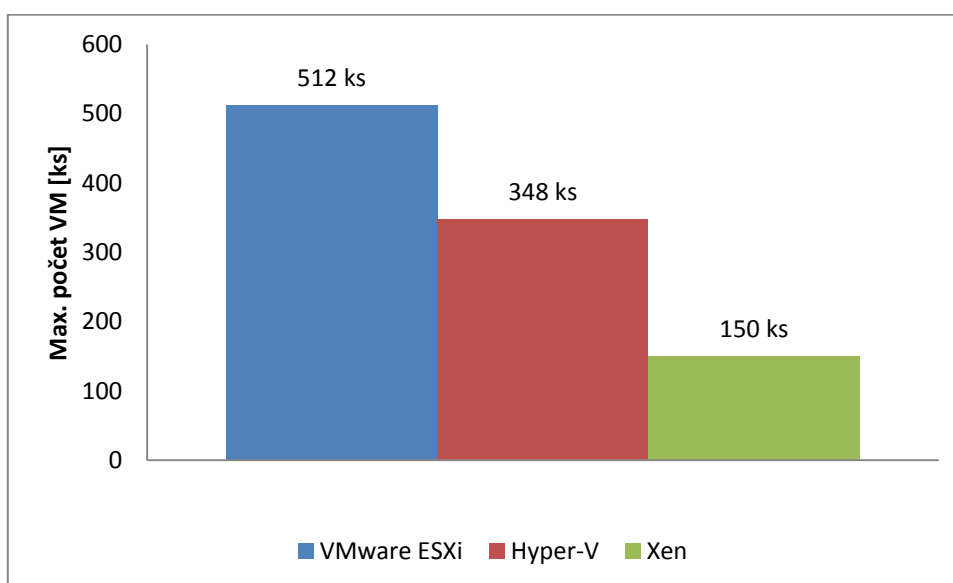
4.1 Porovnání vlastností jednotlivých hypervizorů

Byly vybrány vlastnosti hypervizorů, které jsou společné pro všechny tři hypervizory. A ohraničují maximální vlastnosti, kterými mohou disponovat. Dále omezení maximálního množství zdrojů, které lze přidělit jedné VM. Těmi je maximální počet virtuálních počítačů. Maximální počet logických procesorů. Maximální počet virtuálních CPU. Maximální počet virtuálních CPU na jeden virtuální počítač. Maximální množství podporované paměti RAM. Maximální množství paměti RAM na jeden virtuální počítač. Posledním kritériem je maximální počet virtuálních disků. Porovnání bude provedeno na základě hodnot, které uvádějí výrobci jednotlivých virtualizačních řešení.

	ESXi	Hyper-V	XEN
Podpora Intel VT-x,AMD-V	ANO	ANO	ANO
Max. počet VM	512 ks	348 ks	150 ks
Max. log. procesorů	160 ks	64 ks	160 ks
Max. vCPU	2048 ks	512 ks	900 ks
Max. vCPU/VM	64 ks	64 ks	16 ks
Max. RAM	2 TB	2 TB	1 TB
Max. RAM/VM	1 TB	1 TB	128 GB
Max. počet virtuál. disků	2048 ks	omezeno OS	512 ks
Formáty diskových obrazů	VMDK	VHD, VHDX	KVM, XEN 3/XEN 4,XVA

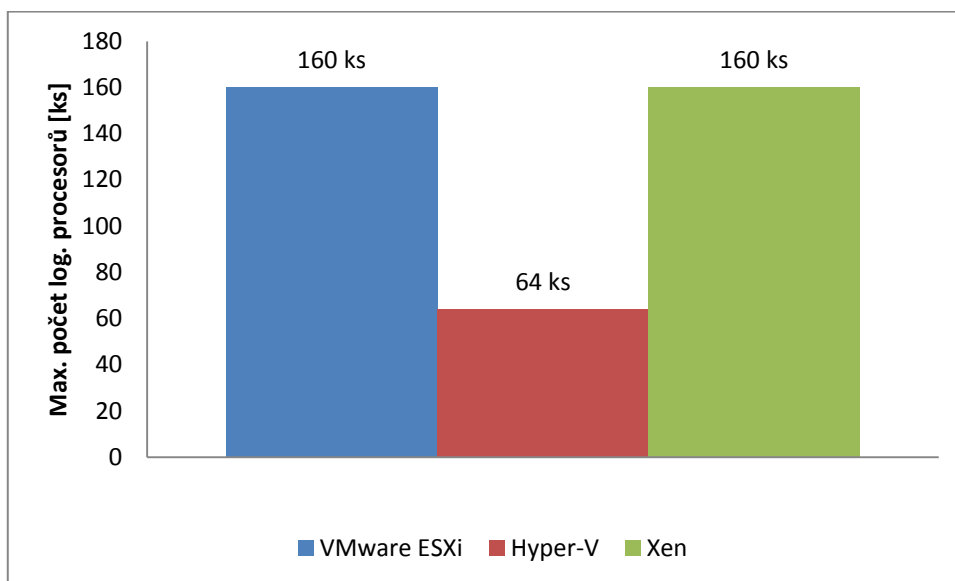
Tabulka 1 – maximální hardwarové vlastnosti hypervizorů, Zdroj:

Tabulka 1 poskytuje přehled vlastností hypervizorů, které byli posbírány z manuálů hypervizorů. Rozdíly hodnot uvedené v tabulce budou uvedeny dále a rozebrány u grafů, které patří k jednotlivým položkám tabulky.



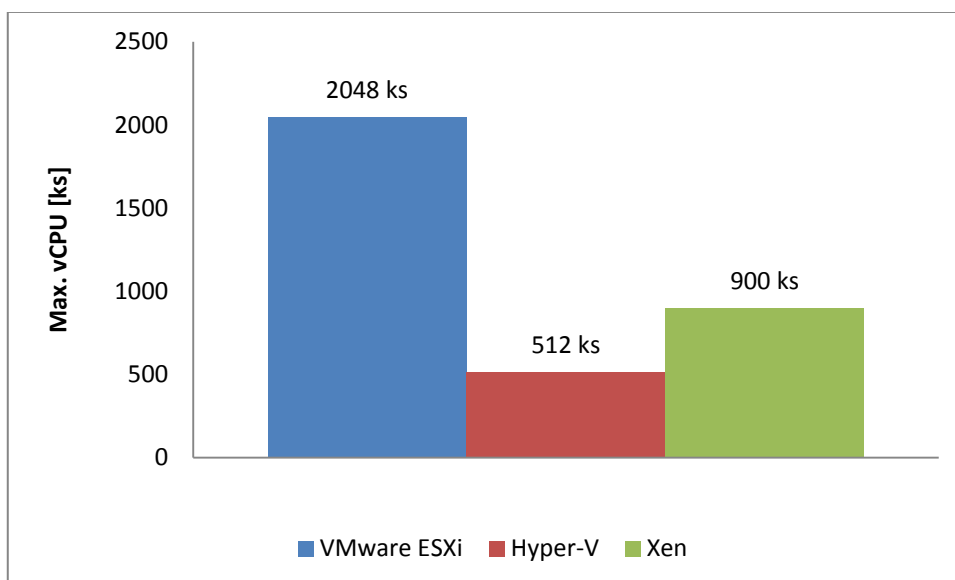
Graf 1 - Maximální počet virtuálních počítačů, Zdroj: vlastní

Tento sloupcový graf zobrazuje grafické vyjádření množství virtuálních strojů, které lze provozovat naráz na řešeních jednotlivých firem. VMware ESXi zde má náskok 165 virtuálních strojů oproti druhému Hyper-V a o 362 oproti Xenu.



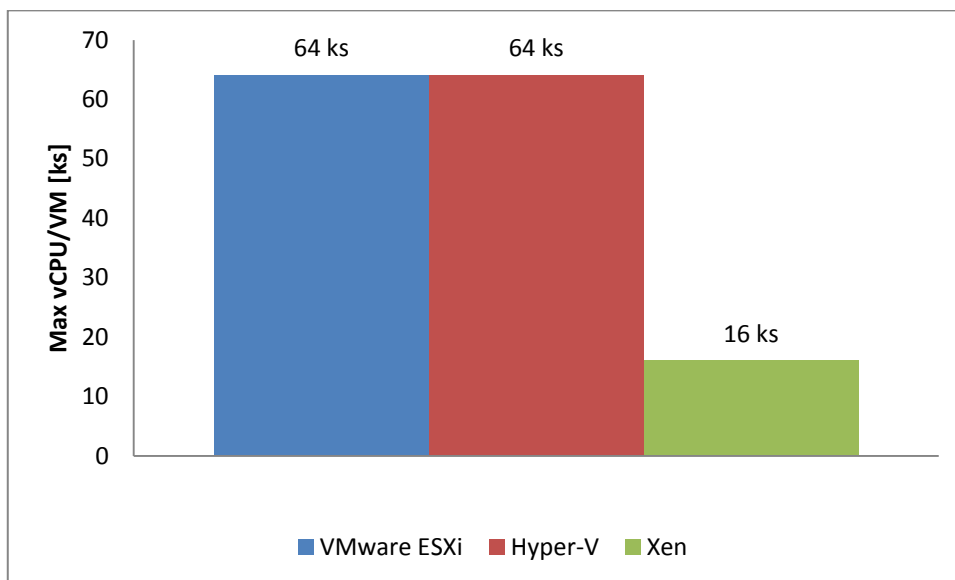
Graf 2 - Max. počet logických procesorů, Zdroj: vlastní

Tento sloupcový graf zobrazuje grafické vyjádření maximálního počtu logických procesorů, které mohou podporovat. Xen má v tomto případě vyrovnanou hodnotu 160 log. procesorů s VMware ESXi. Hyper-V ztrácí oproti oběma o počet 96 logických procesorů.



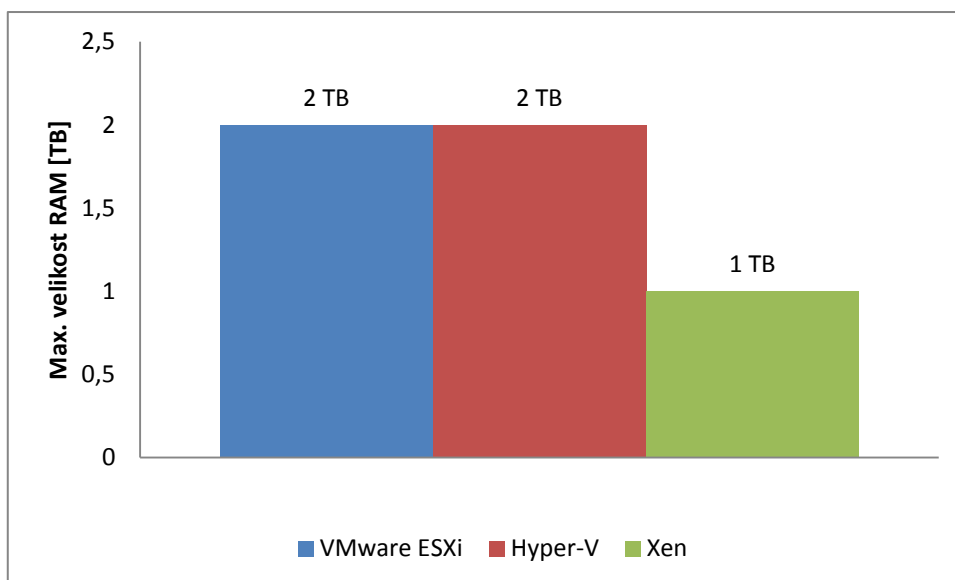
Graf 3 - Max. počet virtuálních CPU, Zdroj: vlastní

Tento sloupcový graf zobrazuje grafické vyjádření maximálního počtu virtuálních CPU, které zvládají distribuovat mezi jednotlivé virtuální počítače. VMware ESXi zde má náskok 1148 od Xenu a o 1536 od Hyper-V. Zde je rozdíl mezi množstvím naprosto propastný.



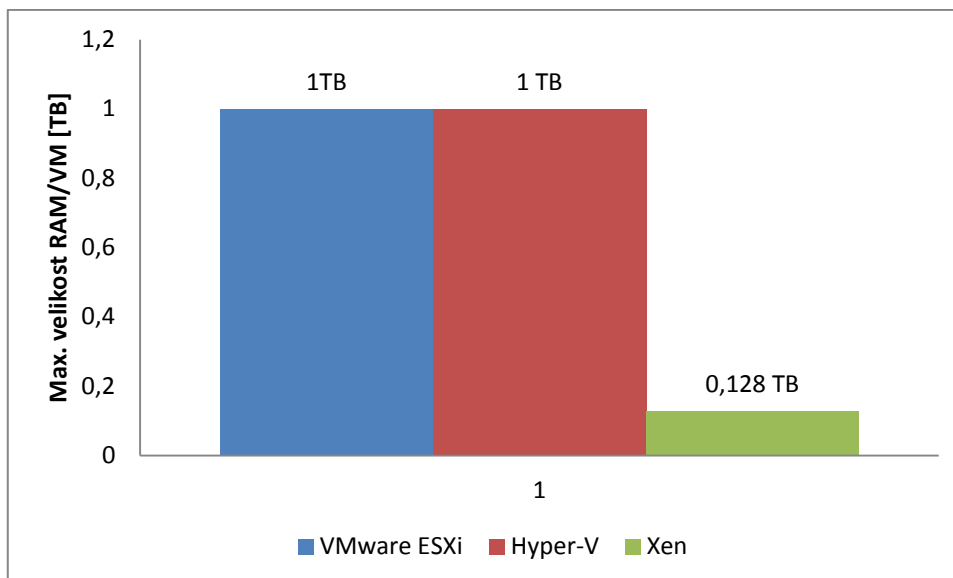
Graf 4 - Max. počet vCPU/VM, Zdroj: vlastní

Tento sloupkový graf zobrazuje grafické vyjádření maximálního počtu virtuálních CPU, které lze přidělit jednomu virtuálnímu počítači. Hyper-V zde drží krok s VMware ESXi s hodnotou 64 virtuálních CPU na jeden virtuální stroj. Xen podporuje pouze třetinovou hodnotu 16 virtuálních procesorů.



Graf 5 - Max. velikost RAM, Zdroj: vlastní

Tento sloupkový graf zobrazuje grafické vyjádření maximálního množství RAM, které umí používat řešení jednotlivých firem. VMware ESXi stejně jako Hyper-V zvládá využití 2TB RAM paměti. Xen pak poloviční hodnotu 1TB.



Graf 6- Max. velikost RAM/VM

Tento sloupcový graf zobrazuje grafické vyjádření maximálního množství RAM, které umí používat řešení jednotlivých firem pro jeden virtuální počítač. VMware ESXi stejně jako Hyper-V zvládá využití 1TB RAM paměti pro jeden virtuální stroj. Xen pak poloviční hodnotu 128GB.

4.2 Zhodnocení hardwarových limitů hypervizorů

Vlastnosti hypervizoru leadera na trhu VMware ESXi je nejvyšší ve 3 z 7 sledovaných vlastností je nejlepší a u ostatních vlastností je hodnota shodná s nejvyšší hodnotou. Ve 3 vlastnostech drží Hyper-V krok s ESXi. Xen drží krok s ESXi pouze v jednom případě jinak mírně ztrácí. Dle hodnot lze tvrdit, že VMware ESXi je možno doporučit pro největší virtualizační řešení. Je tu tedy i největší prostor pro růst výkonu serveru dodatečnými rozšířeními. Ať už dokoupení větší diskové kapacity, nebo dokoupení RAM disku.

4.3 Testování efektivity využití hardwaru hypervizorem

Testování je sestaveno tak, aby se otestovala práce hypervizoru s diskem, procesorem a pamětí. Testování disku probíhá za pomoci Crystal Disk Mark 3.0.2 x64. Pro testování práce s pamětí a procesorem byl vybrán program SuperPI. Při testování byly nainstalovány pouze hypervizory a žádné další optimalizační balíčky. VMware ESXi bylo nainstalováno ve verzi 5.1. XenServer ve verzi 6.2. Microsoft Hyper-V byl zvolena verze Microsoft Server Hyper-V 2008 R2. Testování bylo spuštěno po čerstvém startu systému.

4.3.1 Testování práce s diskem

Cílem testů zaměřených na práci s diskem bylo zjistit rozdíly mezi rychlostmi při práci s diskem. Pro tento účel byl zvolen program Crystal Disk Mark. Umožňuje sadu 4 testů práce s diskem a volitelné množství dat, která se budou zapisovat a číst. Metody testů které podporuje jsou: sekvenční čtení/zápis (velikost úseku je 1024KB), náhodný čtení/zápis

(velikost úseku 512KB), náhodný čtení/zápis (velikost úseku 4KB), náhodný čtení/zápis (velikost úseku 4KB, s délkou fronty 32). Výchozí počet měření před výpočtem aritmetického průměru je 5. Rozsah, který lze nastavit v programu, je mezi hodnotami 1 až 9. Pro potřeby testování je použita výchozí hodnota, což znamená 5 opakování. Volitelné množství dat, které se při těchto postupech zapíše je 50MB, 100MB, 500MB, 1000MB, 2000MB a 4000MB. Pro potřeby zhodnocení jsou použity první 4 hodnoty.

	ESXi	Hyper-V	XenServer	Server bez hypervizoru
Seq. 1024K	550,7 MB/s	507,3 MB/s	174,8 MB/s	601,5 MB/s
Náh. 512K	399 MB/s	396,6 MB/s	174,2 MB/s	399 MB/s
Náh. 4K	26,49 MB/s	22,48 MB/s	6,914 MB/s	38,33 MB/s
4K QD32	96,88 MB/s	90,77 MB/s	7,527 MB/s	94,78 MB/s

Tabulka 2 - Čtení 50MB, Zdroj: vlastní

Tato tabulka obsahuje rychlosti čtení při způsobech přístupu, které jsou popsány na začátku podkapitoly. Se souborem o celkové velikosti 50MB. Hypervizory skončili v pořadí VMware ESXi, Microsoft Hyper-V a Citrix XenServer. V rámci segmentového čtení byl nejbližší výkon operačního systému běžícímu na serveru bez virtualizační mezivrstvy VMware ESXi který dosáhl 91,6% efektivity využití výkonu. V rámci čtení náhodně rozděleného souboru po 512KB byl znova nejrychlejší se 100% efektivitou VMware ESXi. Stejně tak v případě čtení náhodně rozděleného souboru po 4KB s efektivitou 69,2%. Žádná změna na první pozici ani u poslední položky kde byl výkon dokonce vyšší než u řešení bez hypervizoru.

	ESXi	Hyper-V	XenServer	Server bez hypervizoru
Seq. 1024K	148,1 MB/s	126,2 MB/s	111,2 MB/s	156,6 MB/s
Náh. 512K	114,9 MB/s	106,4 MB/s	155 MB/s	119,4 MB/s
Náh. 4K	2,215 MB/s	2,234 MB/s	5,935 MB/s	2,274 MB/s
4K QD32	3,863 MB/s	4,099 MB/s	7,426 MB/s	3,365 MB/s

Tabulka 3 - Zápis 50MB, Zdroj: vlastní

Zápis se souborem o celkové velikosti 50MB. Hypervizory skončili v pořadí Citrix XenServer, VMware ESXi a Microsoft Hyper-V. V rámci segmentového zápisu byl nejbližší výkon operačního systému běžícímu na serveru bez virtualizační mezivrstvy VMware ESXi který dosáhl 94,6% efektivity využití výkonu. V rámci zápisu náhodně umístěného souboru po 512KB byl nejrychlejší XenServer. Stejně tak v případě čtení náhodně rozděleného souboru po 4KB s dvojnásobnou efektivitou. Žádná změna na první pozici ani u poslední položky kde byl výkon dokonce vyšší než u řešení bez hypervizoru.

	ESXi	Hyper-V	XenServer	Server bez hypervizoru
Seq. 1024K	550 MB/s	444,3 MB/s	165,4 MB/s	602,1 MB/s
Náh. 512K	407,1 MB/s	355,7 MB/s	175,7 MB/s	423,6 MB/s
Náh. 4K	18,24 MB/s	14,2 MB/s	6,998 MB/s	20,12 MB/s
4K QD32	106,3 MB/s	104,9 MB/s	7,416 MB/s	115,8 MB/s

Tabulka 4 - Čtení 100MB, Zdroj: vlastní

Čtení se souborem o celkové velikosti 100MB. Hypervizory skončili v pořadí VMware ESXi, Microsoft Hyper-V a Citrix XenServer. V rámci segmentového čtení byl nejbližší výkonu operačního systému běžícímu na serveru bez virtualizační mezivrstvy VMware ESXi který dosáhl 91,3% efektivity využití výkonu. V rámci čtení náhodně rozděleného souboru po 512KB byl znova nejrychlejší s efektivitou 96,1% VMware ESXi. Stejně tak v případě čtení náhodně rozděleného souboru po 4KB s efektivitou 90,7%. Žádná změna na první pozici ani u poslední položky kde byla efektivita 91,8%.

	ESXi	Hyper-V	XenServer	Server bez hypervizoru
Seq. 1024K	149,1 MB/s	126,6 MB/s	80,72 MB/s	155,4 MB/s
Náh. 512K	81,08 MB/s	79,05 MB/s	119,7 MB/s	77,98 MB/s
Náh. 4K	1,337 MB/s	1,313 MB/s	2,599 MB/s	1,1 MB/s
4K QD32	3,57 MB/s	3,672 MB/s	0,261 MB/s	2,945 MB/s

Tabulka 5 - Zápis 100MB, Zdroj: vlastní

Zápis se souborem o celkové velikosti 100MB. Hypervizory skončili v pořadí Citrix XenServer, VMware ESXi a Microsoft Hyper-V. V rámci segmentového čtení byl nejbližší výkonu operačního systému běžícímu na serveru bez virtualizační mezivrstvy VMware ESXi který dosáhl 95,9% efektivity využití výkonu. V rámci čtení náhodně rozděleného souboru po 512KB byl nejrychlejší Citrix XenServer s efektivitou 103,1% VMware ESXi. V případě čtení náhodně rozděleného souboru po 4KB byl nejrychlejší XenServer s efektivitou 121,6%. Na první pozici u poslední položky skončil Microsoft Hyper-V s efektivitou 124,3%.

	ESXi	Hyper-V	XenServer	Server bez hypervizoru
Seq. 1024K	213,6 MB/s	166,8 MB/s	125,3 MB/s	245 MB/s
Náh. 512K	57,74 MB/s	49,49 MB/s	97,4 MB/s	51,36 MB/s
Náh. 4K	0,499 MB/s	0,513 MB/s	0,305 MB/s	0,387 MB/s
4K QD32	3,242 MB/s	3,195 MB/s	3,326 MB/s	3,087 MB/s

Tabulka 6 - Čtení 500MB, Zdroj: vlastní

Čtení se souborem o celkové velikosti 500MB bylo pořadí následující VMware ESXi, Citrix XenServer a Microsoft Hyper-V. V rámci segmentového čtení byl nejbližší výkonu operačního systému běžícímu na serveru bez virtualizační mezivrstvy VMware ESXi který dosáhl 87,2% efektivity využití výkonu. V rámci čtení náhodně rozděleného souboru

po 512KB byl nejrychlejší s efektivitou 189,6 % XenServer. V případě čtení náhodně rozděleného souboru po 4KB s efektivitou 132,6 % Microsoft Hyper-V. Na první pozici ani u poslední položky skončil Citrix XenServer kde byla efektivita 103,7%.

	ESXi	Hyper-V	XenServer	Server bez hypervizoru
Seq. 1024K	40,99 MB/s	37,54 MB/s	37,05 MB/s	42 MB/s
Náh. 512K	25,56 MB/s	24,79 MB/s	29,04 MB/s	25,97 MB/s
Náh. 4K	0,323 MB/s	0,332 MB/s	0,285 MB/s	0,285 MB/s
4K QD32	0,585 MB/s	0,624 MB/s	0,289 MB/s	0,564 MB/s

Tabulka 7 - Zápis 500MB, Zdroj: vlastní

Zápis se souborem o celkové velikosti 500MB bylo pořadí následující VMware ESXi, Citrix XenServer a Microsoft Hyper-V. V rámci segmentového čtení byl nejbližší výkonu operačního systému běžícímu na serveru bez virtualizační mezivrstvy VMware ESXi který dosáhl 97,6 % efektivitu využití výkonu. V rámci čtení náhodně rozděleného souboru po 512KB byl nejrychlejší s efektivitou 111,8 % Citrix XenServer. V případě čtení náhodně rozděleného souboru po 4KB s efektivitou 116,5 % Microsoft Hyper-V. Na první pozici u poslední položky skončil Microsoft Hyper-V.

	ESXi	Hyper-V	XenServer	Server bez hypervizoru
Seq. 1024K	218,2 MB/s	165,7 MB/s	117,8 MB/s	262,2 MB/s
Náh. 512K	41,8 MB/s	40,55 MB/s	27,78 MB/s	41,94 MB/s
Náh. 4K	0,44 MB/s	0,467 MB/s	0,727 MB/s	0,352 MB/s
4K QD32	3,054 MB/s	3,057 MB/s	0,662 MB/s	2,988 MB/s

Tabulka 8 - Čtení 1000MB, Zdroj: vlastní

Čtení se souborem o celkové velikosti 1000MB bylo pořadí následující VMware ESXi, Microsoft Hyper-V a Citrix XenServer. V rámci segmentového čtení byl nejbližší výkonu operačního systému běžícímu na serveru bez virtualizační mezivrstvy VMware ESXi. V rámci čtení náhodně rozděleného souboru po 512KB byl nejrychlejší VMware ESXi. V případě čtení náhodně rozděleného souboru po 4KB s efektivitou Citrix XenServer. Na první pozici u poslední položky skončil Microsoft Hyper-V.

	ESXi	Hyper-V	XenServer	Server bez hypervizoru
Seq. 1024K	44,15 MB/s	38,09 MB/s	39,24 MB/s	41,48 MB/s
Náh. 512K	24,02 MB/s	22 MB/s	27,84 MB/s	23,8 MB/s
Náh. 4K	0,297 MB/s	0,32 MB/s	2,48 MB/s	0,285 MB/s
4K QD32	0,569 MB/s	0,586 MB/s	1,969 MB/s	0,558 MB/s

Tabulka 9 - Zápis 1000MB, Zdroj: vlastní

Zápis se souborem o celkové velikosti 1000MB bylo pořadí následující VMware ESXi, Microsoft Hyper-V a Citrix XenServer. V rámci segmentového čtení byl nejbližší výkonu

operačního systému běžícímu na serveru bez virtualizační mezivrstvy VMware ESXi. V rámci čtení náhodně rozděleného souboru po 512KB byl nejrychlejší Citrix XenServer. V případě čtení náhodně rozděleného souboru po 4KB s efektivitou Citrix XenServer. Na první pozici u poslední položky skončil první Citrix XenServer.

4.3.2 Zhodnocení testování disku

I zde si vedl VMware ESXi velice dobře. Sekvenční zápis po 1024KB a náhodný zápis po 512KB byl hned v několika testech nejvyšší výkon u něj. V náhodném zápisu po malých úsecích však exceloval Citrix XenServer.

4.3.3 Testování CPU

Cílem testů zaměřených na práci s CPU bylo zjistit, který z výrobců dosáhl nevyššího stupně optimalizace s nejnižší režii. CPU bylo testováno za pomoci programu Super_PI. Program vypočítává určený počet desetinných míst za celou částí čísla π . Umožňuje nastavení počtu desetinných míst na hodnoty 16K, 32K, 64K, 128K, 256K, 512K, 1M, 2M, 4M, 8M, 16M, 32M. Pro testování byly vybrány hodnoty nad jeden milion desetinných míst. Pro výpočet je použit Gauss–Legendreho algoritmus.

	32M	16M	8M	4M	2M	1M
ESXi	26m 42s	12m 31s	5m 33s	2m 29s	1m 4s	26s
Hyper-V	26m 37s	12m 14s	5m 39s	2m 25s	1m 4s	26s
XenServer	26m 8s	12m	5m 25s	2m 24s	1m 1s	25s

Tabulka 10 – Čas výpočtu π , Zdroj: vlastní

U výpočtu nízkého počtu desetinných míst jako je 1M je hodnota času vyrovnaná u VMware ESXi a Microsoft Hyper-V Citrix XenServer je rychlejší o 1s. Na 2M Citrix XenServer rychlejší než jeho konkurence o 3s. Na 4M je nejrychlejší Citrix XenServer nejrychlejší v případě Microsoft Hyper-V je rozdíl 1s a u VMware ESXi 5s. Na 8M překonává je Citrix XenServer opět nejrychlejší VMware ESXi ztrácí 8s a Microsoft Hyper-V 14s. Na 16M překonává Citrix XenServer VMware ESXi o 31s a Microsoft Hyper-V 14s. Při nejdelším výpočtu opět byl výpočet nejrychlejší u Citrix XenServeru, Microsoft Hyper-V zde ztrácí 29s a VMware ESXi 34s.

4.3.4 Zhodnocení testování CPU

Ve všech testech za pomoci SuperPi skončil nejlépe hypervizor Xen následovaný Microsoft Hyper-V a jako poslední skončil VMware ESXi. Při počtu výpočtů do 4M byl rozdíl v času výpočtu s řádů jednotek do 5s. Dále se mírně zvyšuje čas výpočtu na minimálních 8s u 8M. Na 16M se minimální doba prodlevy 14s a na 32M je rozdíl 29s a 34s.

V tomto testu se ukázalo, že Citrix XenServer dosáhl nejnižších časů. Z toho lze usuzovat, že je dosaženo vyššího stupně optimalizace než v případě konkurenčních řešení. Druhý skončil Microsoft Hyper-V a s malým odstupem poslední skončil VMware ESXi.

4.3.5 Analýza dat ze skriptu

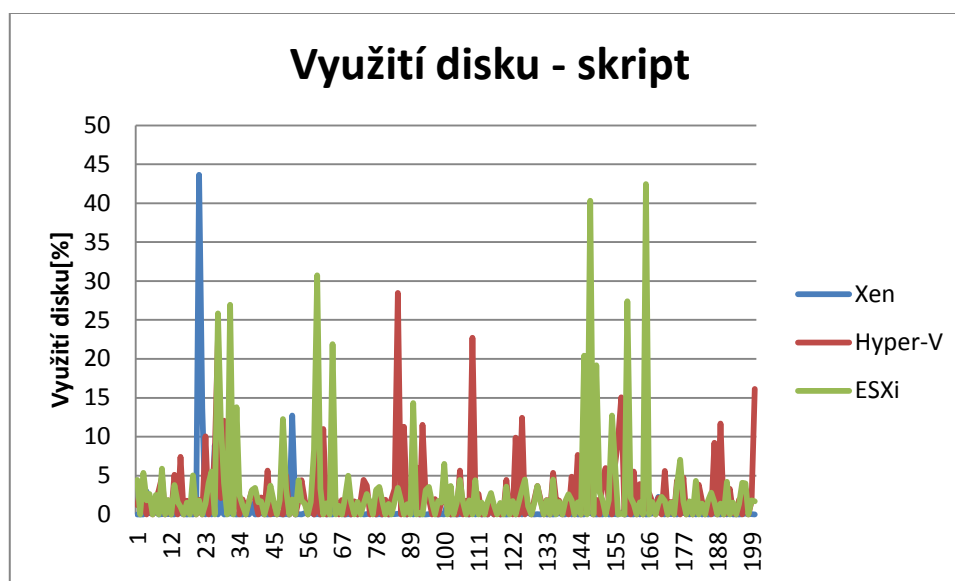
Analýza je rozdělena do tří částí. V první části jsou zhodnocena data posbírána za pomoci skriptu o disku. V druhé části jsou zhodnocena data o paměti RAM. Třetí část hodnotí využití procesorového času. detailně je popsán celý skript v kapitole 3.

4.3.5.1 Analýza dat skript využití disku

	Max	Min	Avg
ESXi	42,5%	0%	3,1%
Hyper-V	28,5%	0%	2,7%
XenServer	43,7%	0%	0,6%

Tabulka 11 - využití disku-skript, Zdroj: Vlastní

V tabulce jsou shrnuty krajní hodnoty, které byly zastoupeny ve vzorku 200 hodnot. Je zde maximální hodnota využití disku během sbírání dat. Kde nejnižší hodnota byla naměřena u Microsoft Hyper-V následován VMware ESXi a Citrix XenServer mezi nimiž byl rozdíl mezi mezní hodnotou pouze 1,2%. Minimální hodnota využití dat, která je rovna 0% u všech hypervizorů. Průměrné využití disku je založené na aritmetickém průměru sebraných hodnot. Nejnižší průměr 0,6% byl spočítán pro XenServer, následovaný Hyper-V s hodnotou 2,7% a VMware ESXi 3,1%.



Graf 7 - Využití disku - skript, Zdroj: vlastní

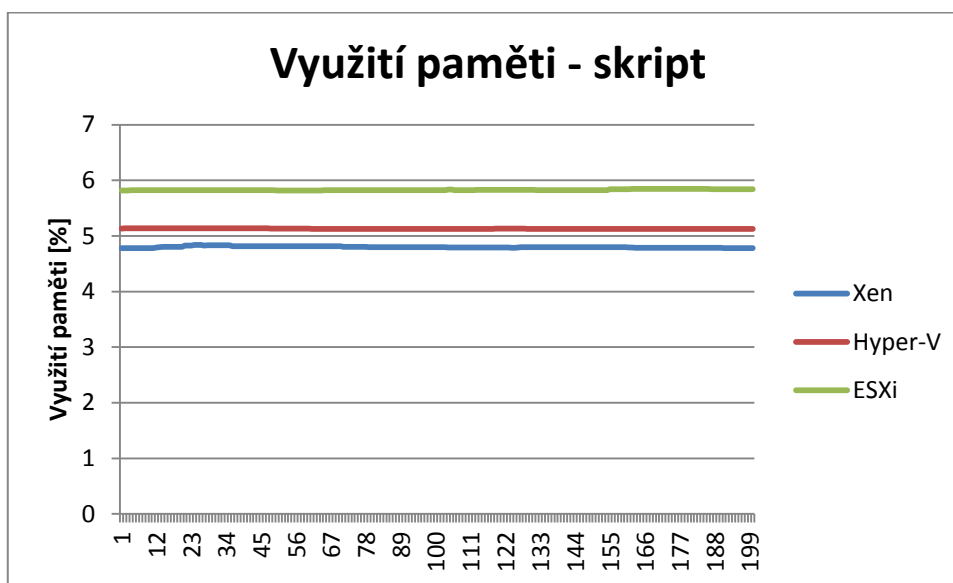
Zde v grafu jsou zaneseny hodnoty využití disku všech tří hypervizorů měnící se v čase.

4.3.5.2 Analýza dat skript využití paměti RAM

	Max	Min	Avg
ESXi	5,85%	5,82%	5,83%
Hyper-V	5,14%	5,13%	5,13%
XenServer	4,84%	4,78%	4,80%

Graf 8 - využití paměti RAM-skript, Zdroj: vlastní

V tabulce jsou shrnuty krajní hodnoty, které byly zastoupeny ve vzorku 200 hodnot. Je zde maximální hodnota využití paměti RAM během sbírání dat. Kde nejnižší hodnota byla naměřena u Citrix XenServer následován Microsoft Hyper-V a VMware ESXi mezi nimiž byl rozdíl mezi mezní hodnotou pouze 0,71%. Minimální hodnota využití dat je opět nejnižší u Citrix XenServer následován Microsoft Hyper-V a VMware ESXi. Průměrné využití disku je založené na aritmetickém průměru sebraných hodnot. Nejnižší průměr 4,8% byl spočítán pro XenServer, následovaný Hyper-V s hodnotou 5,13% a VMware ESXi 5,83%.



Graf 9 - využití paměti - skript, Zdroj: vlastní

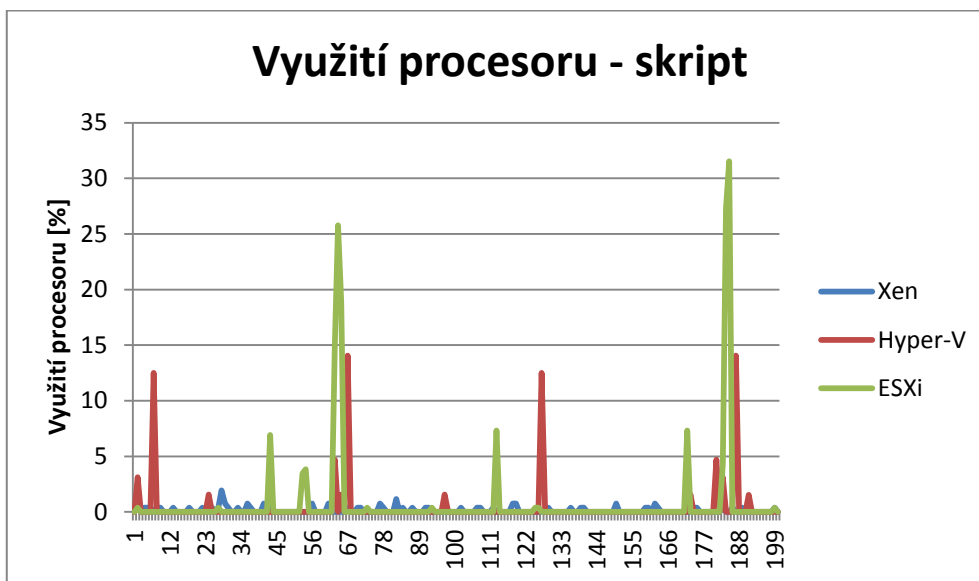
Zde v grafu jsou zaneseny hodnoty využití paměti RAM všech tří hypervizorů měnící se v čase.

4.3.5.3 Analýza dat skript využití procesorového času

	Max	Min	Avg
ESXi	31,5%	0%	0,8%
Hyper-V	14,1%	0%	0,4%
XenServer	1,9%	0%	0,1%

Graf 10 - využití procesorového času, Zdroj: vlastní

V tabulce jsou shrnuty krajní hodnoty, které byly zastoupeny ve vzorku 200 hodnot. Je zde maximální hodnota využití paměti RAM během sbírání dat. Kde nejnižší hodnota byla naměřena u Citrix XenServer následován Microsoft Hyper-V a VMware ESXi mezi nimiž byl rozdíl mezi mezní hodnotou pouze 0,71%. Zde je rozdíl mezi prvním a druhým 12,2%. Minimální hodnota využití dat je rovna u všech tří hypervizorů s hodnotou 0%. Průměrné využití disku je založené na aritmetickém průměru sebraných hodnot. Nejnižší průměr 0,1% byl spočítán pro XenServer, následovaný Hyper-V s hodnotou 0,4% a VMware ESXi 0,8%.



Graf 11 - využití procesoru - skript, Zdroj: vlastní

Zde v grafu jsou zaneseny hodnoty využití procesoru všech tří hypervizorů měnící se v čase.

Závěr

Zadáním práce bylo provést komparativní analýzu vizualizačních technologií serverů. Představení technických možností virtualizace serverů a podrobného představení vybraných technologií. Na jejichž základě se sestaví metodika zátěžových testů a z pohledu využívání systémových prostředků, která se realizuje a vyhodnotí nad prostředím Microsoft Server 2008 R2.

Práce je strukturována tak, že je uveden význam virtualizace jako takové. Následované charakterizací cílů, kterých se snaží dosáhnout. Aby bylo jasné, čeho se virtualizace snaží dosáhnout a proč ji vůbec nasazovat na server. Dalším bodem je seznámení s klasifikací serverové virtualizace. Za kterým se nalézá seznámení se sedmi vrstvami virtualizace, aby čtenář pochopil, že se jedná o velice komplexní tematiku a získal povědomí o jejích vrstvách. Následuje krátké seznámení s historií virtualizace sahající do 70. let 20. století, Poté jsou popsány hlavní architektury hypervizorů a je zde rozepsán způsob, jakým způsobem je virtualizován tak důležitý bod systému jako je procesor, paměť a sběrnice ve virtualizovaném systému. Posledním hlavním bodem první kapitoly je popis způsobu, jak je řešena hardwarová asistence virtualizace uvnitř procesorů pomocí technologie od firem Intel a AMD. Tím končí první oddíl, který se zabývá virtualizací obecně. Následuje popis vizualizačních nástrojů, které byli vybrány, aby reprezentovali hlavní zástupce vizualizačního odvětví. A těmi jsou VMware ESXi 5.1, Citrix XenServer 6.2 a Microsoft Server Hyper-V 2008. O každém z nástrojů, je uvedena základní charakteristika, hardwarové nároky, schéma architektury vizualizačního nástroje a popis jakým způsobem fungují. Ve třetí části je popis jak bude probíhat metodika testování a je zde rozebrán skript použitý pro sbírání dat o systému. Poslední část obsahuje rozebrání hodnot, které byly příslušnými způsoby posbírány a jsou zde vyjádřeny ve formě tabulek a grafů. Ty budou ještě v dalších odstavcích rozebrány.

Ve srovnání produktů v rámci maximální podpory hardwarových zdrojů dopadl nejlépe VMware ESXi 5.1. Dosáhl ve všech zkoumaných kategoriích nejvyšších hodnot. Kategoriemi ve kterých byl výsledek vyrovnaný s některým z konkurenčních hypervizorů byla maximální velikost RAM, maximální velikost RAM/VM a maximální počet vCPU/VM kde byli na stejné hodnotě s Microsoft Hyper-V. Maximální počet logických procesorů byl na stejné hladině s Citrix XenServer. Ve třech zbylých kategoriích exceloval a to tím způsobem, že nechal své soky daleko za sebou. Těmi jsou maximální počet VM, maximální počet vCPU a maximální počet virtuálních disků. Z toho plyne, že VMware ESXi 5.1 je řešení které dokáže pojmout největší množství hardwarových zdrojů naráz a dá se doporučit, pokud je v plánu virtualizace velkého množství serverů či poskytování cloudových služeb.

Sběr dat o použití paměťového prostoru RAM přinesl poznatek, že nejnižší vyžití bylo u Citrix Xenserveru s průměrnou hodnotou dosahovala 4,8%. Za ním následoval Microsoft Hyper-V s 5,13% a poslední skončil VMware ESXi 5.1 s 5,83%. Sběr dat o využití procesu přinesl poznatek, že nejnižší vytížení bylo u Citrix Xenserveru s průměrnou hodnotou dosahovala 0,1%. Za ním následoval Microsoft Hyper-V s 0,4% a poslední skončil VMware ESXi 5.1 s 0,8%. Stejného pořadí bylo dosaženo i při testu výpočtů za pomoci programu SuperPi. Ani sběr dat o využití pevných disků nepřinesl změnu v pořadí a opět skončil na prvním místě Citrix XenServer, na druhé příčce Microsoft Hyper-V a na konec VMware ESXi. Zdá se tedy, že Citrix XenServer má nejnižší režii při práci s hardwarovým vybavením serveru. Dosáhl nejnižších naměřených výsledků, jak v kategorii využití paměťového prostoru RAM, využití pevného disku, tak i zatížení procesoru jak ve skriptu, tak za pomoci SuperPi. Dá se tedy předpokládat, že by bylo vhodné pro menší virtualizační řešení použít platformu Citrix XenServer, která dosahuje velice dobrých výsledků v rámci práce s hardwarem. Ovšem na druhou stranu, rozdíly mezi hodnotami těchto hypervizorů nejsou nijak propastné. A bohužel toto vítězství sráží trochu fakt, že v rámci porovnání maximální konfigurace dopadl Citrix Xenserver naprosto nejhůře.

Možností dalšího rozšíření práce by mohla být část porovnávací nasazenou serverovou službu oproti virtuálnímu řešení. Například virtualizovaný databázový server a běžící přímo nad hardwarem. Další alternativou je porovnání konsolidovaného serveru napříč těmito třemi vizualizačními řešeními. Zároveň porovnání finančních nákladů přímo na zavedení této konsolidace. Čímž by se vyskytli další otázky týkající se efektivního výběru hardwaru serveru z hlediska efektivity, podle velikosti firmy pro kterou by se virtualizační řešení navrhovalo.

Literatura

- [1] Cloud computing - ABZ.cz: slovník cizích slov. *ABZ slovník cizích slov* [online]. © 2005- [cit. 2013-07-23]. Dostupné z: <http://slovník-cizich-slov.abz.cz/web.php/slovo/cloud-computing-klaud-kompjuting>
- [2] PETRÁČKOVÁ, Věra a Jiří KRAUS. *Akademický slovník cizích slov*. 1. vyd. Praha: Academia, 1998, 834 s. ISBN 80-200-0607-9.
- [3] VMware Virtualization - Optimize IT Resources with Virtual Technology. VMWARE. *VMware Virtualization for Desktop Server, Public Private Clouds* [online]. © 2013 [cit. 2013-08-07]. Dostupné z: <http://www.vmware.com/virtualization/what-is-virtualization.html>
- [4] Microsoft | Server Cloud | Virtualization. MICROSOFT. *Microsoft Česká Republika* [online]. © 2013 [cit. 2013-08-07]. Dostupné z: <http://www.microsoft.com/en-us/server-cloud/virtualization/default.aspx>
- [5] SALTZER, Jerome H. a M. Frans KAASHOEK. *Principles of computer system design: an introduction*. Burlington,MA 01803,USA: Morgan Kaufmann, 2009. ISBN 978-0123749574.
- [6] Rychlokurz: Cesta ke konsolidaci | Computerworld.cz. DIVIŠ, Petr. *Computerworld* [online]. Praha: IDG Czech Republic, 25.04.2007 [cit. 2013-03-18]. Dostupné z: <http://computerworld.cz/whitepapers/rychlokurz-cesta-ke-konsolidaci-2611>
- [7] HORNE, Chris. *Understanding Full Virtualization, Paravirtualization, and Hardware Assist*. Palo Alto: VMware , 2007, 14 s. Dostupné z: http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf
- [8] Snapshoty - způsob jak posunout zálohování a obnovu dále. SCHUTZ, Radek a Ondřej BAČINA. *B2B magazín o sítích, telekomunikacích a datových centrech* [online]. © 2010 – 2013 [cit. 2013-07-23]. Dostupné z: <http://www.netguru.cz/odborne-clanky/snapshoty-zpsob-jak-posunout-zalohovani-a-obnovu-dale.html>
- [9] Snapshoty. SCHUTZ, Radek a Ondřej BAČINA. DELL. *Netguru: Nezávislý odborný on-line magazín* [online]. Praha: AVERIA, 2011 [cit. 2013-03-19]. Dostupné z: <http://www.netguru.cz/odborne-clanky/snapshoty-zpsob-jak-posunout-zalohovani-a-obnovu-dale.html>

- [10] Virtualizace, Emulace, Simulace. *Http://www.root-server.cz* [online]. © 2013 [cit. 2013-07-23]. Dostupné z: <http://www.root-server.cz/virtualizace.html>
- [11] Emulátor. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2013 [cit. 2013-07-23]. Dostupné z: <http://cs.wikipedia.org/wiki/Emul%C3%A1tor>
- [12] Application Virtualization by Microsoft | Windows 8 Enterprise. MICROSOFT. *Windows Enterprise* [online]. 2013 [cit. 2013-08-08]. Dostupné z: <http://www.microsoft.com/en-us/windows/enterprise/products-and-technologies/virtualization/app-v.aspx>
- [13] TULLOCH, Mitch et al. *Understanding Microsoft Virtualization Solution From the Desktop to the Datacenter*. Redmond, Washington 98052-6399: Microsoft Press, 2010, 466 s. 2nd edition. ISBN 9780735693371. Dostupné z: http://technettoolbox.cloudapp.net/private_cloud/pages/3/dokumentation/693821e-book.pdf
- [14] RUEST, Danielle a Nelson RUEST. *Virtualizace: Podrobný průvodce*. 1. vyd. Brno: Computer Press, 2010, 393 s. ISBN 978-80-251-2676-9.
- [15] An Introduction to Virtualization. SINGH, Amit. *Kernelthread.com* [online]. © 2004 [cit. 2013-07-23]. Dostupné z: <http://www.kernelthread.com/publications/virtualization/>
- [16] Hyper-V: Microkernelized or Monolithic. *TechNet Blogs* [online]. 23.2.2011 [cit. 2013-07-23]. Dostupné z: <http://blogs.technet.com/b/chenley/archive/2011/02/23/hyper-v-microkernelized-or-monolithic.aspx>
- [17] ADVANCED MICRO DEVICES. *AMD 64 Virtualization Codenamed "Pacifica" Technology: Secure Virtual Machine Architecture Reference Manual*. 3.01. California, 2005. Dostupné z: <http://www.mimuw.edu.pl/~vincent/lecture6/sources/amd-pacifica-specification.pdf>
- [18] X86 virtualization. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2013 [cit. 2013-07-31]. Dostupné z: https://en.wikipedia.org/wiki/X86_virtualization
- [19] INTEL. *Intel Vanderpool Technology for IA-32 Processors (VT-x) Preliminary Specification*. C97063-001. Santa Clara, California, 2005. Dostupné z: http://research.cs.wisc.edu/areas/os/ReadingGroup/OS/papers/vanderpool_ia32.pdf

- [20] Magic Quadrant for x86 Server Virtualization Infrastructure. BITTMAN, Thomas J., George J. WEISS, Mark A. MARGEVICIUS. *Technology Research | Gartner Inc* [online]. Stamford: Garthner, 11. 6. 2012 [cit. 2013-03-14]. Dostupné z: <http://www.gartner.com/technology/reprints.do?id=1-1B2IRYF&ct=120626&st=sg>
- [21] Hyper-V. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-07-23]. Dostupné z: <http://en.wikipedia.org/wiki/Hyper-V>
- [22] FINN, Aidan. *Mastering Hyper-V™ Deployment*. Indianapolis: Wiley, 2011. ISBN 978-0-470-87653-4.
- [23] VMware KB: Minimum system requirements for installing ESXi/ESX. *VMware* [online]. © 2013 [cit. 2013-07-23]. Dostupné z: http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1003661
- [24] VSphere Documentation Center. *VMware* [online]. © 2013 [cit. 2013-07-23]. Dostupné z: http://pubs.vmware.com/vsphere-50/index.jsp?topic=/com.vmware.vsphere.install.doc_50/GUID-DEB8086A-306B-4239-BF76-E354679202FC.html
- [25] XenServer 6.1.0 Technical FAQ. In: [online]. 27.9.2012, 1.11.2012 [cit. 2013-03-05]. Dostupné z: <http://support.citrix.com/article/CTX134887>
- [26] Xen: How Does Xen Work?. XEN. *Welcome to xen.org: the home of the Xen* [online]. 1.0. 2009 [cit. 2013-03-19]. Dostupné z: <http://www.xen.org/files/Marketing/HowDoesXenWork.pdf>
- [27] Chapter 2. System Requirements. *Citrix XenServer* [online]. © 1999 - 2013 [cit. 2013-07-23]. Dostupné z: <http://docs.vmd.citrix.com/XenServer/4.0.1/installation/ch02.html>
- [28] CTX134887 - XenServer 6.1.0 Technical FAQ - Citrix Knowledge Center. *Powering mobile workstyles and cloud services - Citrix* [online]. 27.9.2012 [cit. 2013-07-23]. Dostupné z: <http://support.citrix.com/article/CTX134887>
- [29] Windows PowerShell. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-07-23]. Dostupné z: http://en.wikipedia.org/wiki/Windows_PowerShell#Cmdlets