Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

Studijní program : Aplikovaná informatika

Obor: Informační systémy a technologie

Metoda výběru koncepce on-premise IT infrastruktury

DIPLOMOVÁ PRÁCE

Student : Bc. Martin Šťastný

Vedoucí : Ing. Libor Gála, Ph.D.

Oponent : Ing. Jana Fortinová

2016

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze které jsem čerpal.

V Praze dne 10. prosince 2016 . . . . . .

Martin Šťastný

Poděkování

Moje poděkování patří vedoucímu práce, který mi poskytl všechnu potřebnou podporu a zároveň dostatek volnosti ve zpracování daného tématu. V začátku práce existovala celá řada nejasností, avšak především díky rozsáhlým zkušenostem pana doktora Gály se podařilo všechny překážky překonat a nejasnosti rozptýlit.

Děkuji také rodině, která mě v mém studiu po celou dobu podporovala a patří jí nemalá část zásluh za to, že nyní můžu stát na konci svého studia a odevzdávat tuto práci.

Abstrakt

Zaměřením této práce je tvorba metody pro výběr koncepce on-premise IT infrastruktury. Práce je rozdělena celkem do šesti kapitol, z nichž první se věnuje stručnému úvodu do problematiky a vymezuje cíle práce. Druhá kapitola se zaměřuje na aktuální stav problematiky, jejíž součástí je rešerše podobných prací a také vymezení používaných pojmů. Třetí kapitola se zevrubně věnuje popisu jednotlivých koncepcí jako základu pro čtvrtou kapitolu, která se zabývá konstrukcí samotné metody. Pátá kapitola je věnována ověření metody v reálném prostředí a kapitola poslední shrnuje celou diplomovou práci.

Klíčová slova

IT infrastruktura, koncepce IT infrastruktury, virtualizace, kontejnerová virtualizace, třívrstvá koncepce

Abstract

The focus of this work is to create a method for the selection of the concept of on-premise IT infrastructure. The document is divided into six chapters, where the first is dedicated to brief introduction and definition of objectives of the work. The second chapter focuses on current state of the problem, which includes research of similar works and definitions of used terms. The third chapter is devoted to describing conceptions as the basis for the fourth chapter, which deals with construction of the method. The fifth chapter is devoted to the verification of the method in a real-world environment and last chapter summarizes the whole thesis.

Keywords

IT infrastructure, IT infrastructure conception, virtualization, container virtualization, three-layer conception

Obsah

Obsah....iv

1 Úvod....1

1.1 Cíle práce....2

1.2 Pro koho je práce určena....2

1.3 Struktura práce....3

2 Aktuální stav problematiky a vymezení pojmů....4

2.1 Rešerše aktuálního stavu problematiky....4

2.2 Vymezení on-premise IT infrastruktury a jejích koncepcí....6

2.2.1 On-premise....6

2.2.2 IT infrastruktura a její koncepce....7

2.2.3 Datacentrum....8

2.3 Atributy IT infrastruktury....8

3 Koncepce a technologie IT infrastruktury....13

3.1 Fyzické servery a třívrstvá koncepce....13

3.1.1 Výhody a nevýhody nasazení třívrstvé koncepce....14

3.1.2 Dostupná technologická řešení....19

3.1.3 Dopady koncepce vzhledem atributům....22

3.2 Virtualizace celých operačních systémů....24

3.2.1 Výhody a nevýhody virtualizace celých operačních systémů....25

3.2.2 Dostupná technologická řešení....30

3.2.3 Dopady koncepce vzhledem k atributům....33

3.3 Virtualizace na úrovni operačního systému....35

3.3.1 Výhody a nevýhody virtualizace na úrovni operačního systému....36

3.3.2 Dostupná technologická řešení....40

3.3.3 Dopady koncepce vzhledem k atributům....43

4 Metoda výběru koncepce nasazení on-premise infrastruktury....45

4.1 Předpoklady pro tvorbu rozhodovací tabulky....46

4.2 Teorie konstrukce metody....47

4.3 Výpočet číselných hodnot atributů jednotlivých koncepcí....48

4.4 Rozhodovací tabulka a způsob jejího použití....51
4.5 Příklad použití rozhodovací tabulky....53

5 Ověření platnosti metody v praxi....55
5.1 Představení společnosti....55
5.2 Aplikace metody....56
5.3 Implementace testovacího provozu nové koncepce a jeho vyhodnocení....59

6 Závěr....61

7 Terminologický slovník....63

Seznam literatury....64

1 Úvod

Od počátků počítačů a jejich prvních operačních systémů existovala pouze jediná možnost, kterou koncipovat on-premise IT infrastrukturu. Tou bylo provozování aplikací přímo nad operačním systémem a fyzickým hardwarem. V posledních dvou desetiletích nicméně došlo k významnému rozvoji dalších koncepcí. Ty umožňují na jednom fyzickém počítači provozovat více nezávislých operačních systémů a nebo uzavírat aplikace do virtuálních kontejnerů v rámci jednoho operačního systému. To vše s sebou přineslo nové možnosti, jak koncipovat on-premise IT infrastrukturu.

Pojem on-premise vyjadřuje provozování aplikací potažmo IT infrastruktury ve vlastní režii organizace, čímž je myšleno vlastnictví nebo dlouhodobý pronájem technických prostředí (především počítačů) ve vlastním nebo pronajatém datacentru a to plně ve správě zaměstnanců organizace (dále vysvětleno v kapitole 2.2). Protikladem k on-premise je cloud computing, který umožňuje nakupovat prostředky (výpočetní výkon, úložný prostor, konektivitu a podobně) jako službu v rámci veřejných nebo privátních virtuálních prostředí.

I přes velký nástup cloud computingu jsou on-premise řešení stále platným odvětvím, s jehož rozvojem přímo souvisí nárůst plochy a finančních objemů v datacentrech. Dle zprávy společnosti Markets and Markets (2014) se mezi lety 2015 a 2020 očekává nárůst na dvojnásobek tempem 16,1% ročně. Tento trend jasně podtrhují investice do výstavby nových ploch významnými hráči na českém trhu, mezi které patří například společnostmi TTC a nebo VSHosting.

V práci uvažuji o třech základních koncepcích, kterými lze on-premise IT infrastrukturu utvářet. Řadí se mezi ně třívrstvá koncepce, virtualizace celých operačních systémů a virtualizace na úrovni operačního systému. Tyto koncepce jsou v práci rozebrány a na základě tohoto rozboru je navržena metoda, která pomůže ve výběru té nejvhodnější z nich.

1.1 Cíle práce

Cílem práce je navržení metody „výběru koncepce on-premise IT infrastruktury“, která by se stala nástrojem pro efektivnější rozhodování při stavbě nových datových center a plánování informačních projektů.

Tento cíl je podpořen studií zabývající se aktuálním stavem problematiky, která zkoumá všechny dostupné poznatky v oblasti rozhodování při výběru koncepce on-premise infrastruktury. Podstatnou částí této práce dále je studie zabývající se analýzou a aktuálním stavem technologií pro jednotlivé koncepce on-premise IT infrastruktur.

K naplnění cíle práce došlo navržením metody pro výběr koncepce nasazení on-premise infrastruktury. V závěru práce je vypracována studie v prostředí reálného informačního projektu, která ověřuje platnost této metody.

1.2 Pro koho je práce určena

Cílovou skupinou této práce jsou zejména IT manažeři a vedoucí IT pracovníci v malých a středních organizacích, které nemají vybudované vlastní silné technologické know-how. Do této skupiny rovněž mohou být zařazeni i jednotlivci, kteří jsou pověřeni návrhem koncepce IT infrastruktury jak v rámci výše uvedených typů organizací, tak i v rámci vlastního podnikání (do této skupiny se řadí zejména začínající technologické společnosti).

Další dimenzí, která vymezuje cílové publikum této práce, je rovina provozování IT infrastruktury on-premise, čili ve vlastní režii bez použití outsourcingu nebo veřejných cloudů. Důvodů pro vyřazení outsourcingu nebo veřejných cloudů může být mnoho. Mezi ty nejzávažnější se řadí nedostatečná konektivita směrem k off-premise IT infrastruktuře, obavy o bezpečnost, zastaralé aplikace a nebo zákonná nařízení týkající se ochrany osobních údajů.

Tato práce výše uvedenému publiku na příkladech představí základní koncepty on-premise IT infrastruktur a zároveň mu dá do rukou efektivní nástroj, který mu usnadní rozhodování v tom, jaká koncepce je pro danou situaci nejvhodnější.

1.3 Struktura práce

První část této práce se věnuje rešerši a zakotvení základních pojmů, které tato práce používá. Druhá část je zaměřena na detailní popis každého konceptu od obecné stránky až po konkrétní technologie, které se v rámci něj používají. V této kapitole je rovněž zhodnoceno, jakým způsobem daný koncept naplňuje jednotlivé atributy IT infrastruktur.

Ve čtvrté kapitole je rozebrán způsob, jakým je tvořena metoda a definuje metodou samotnou, jejíž platnost je v navazující kapitole ověřena. Poslední částí je závěr, který shrnuje a hodnotí výsledky této práce.

2 Aktuální stav problematiky a vymezení pojmů

Následující kapitola si klade za cíl prozkoumat aktuální stav problematiky formou rešerše dostupných prací, které se věnují IT infrastruktuře z pohledu metodik či metod pro výběr jejich koncepce. V další části budou vymezeny pojmy datacentrum, on-premise IT infrastruktura a co se myslí jejími koncepty. Na závěr této kapitoly budou rozebrány jednotlivé vlastnosti a jejich atributy, které IT infrastruktura může nabývat.

2.1 Rešerše aktuálního stavu problematiky

Problematiku zabývající se koncipováním IT infrastruktur řeší velké množství autorů a to jak v knihách, vědeckých pracích, tak i dizertacích.

Publikace s názvem *A Practical Guide for IT Architects* (Arrasjid, 2016) na svém začátku definuje klíčové charakteristiky, na základě kterých provádíme navrhování IT infrastruktury. Jsou zmíněny zejména dostupnost, spravovatelnost, výkon, schopnost zotavení se z chyby a bezpečnost. Tyto charakteristiky jsou ohodnoceny podle možného dopadu jejich podcenění a velikostí rizika, se kterým toto může nastat. V dalších částech kniha nastiňuje jednotlivé kroky v procesu plánování IT infrastruktury, mezi které patří tvorba designu architektury, postupu instalace, plánu implementace a testování a příručky s procedurami pro správu. Primárním zaměřeních této knihy nicméně je zevrubný popis postupu plánování, instalace a správy IT infrastruktury v pojetí virtualizace celých operačních systémů. Kniha tudíž neřeší jiné koncepce IT infrastruktur a tím pádem se ani nesnaží vytvářet metody pro jejich výběr.

Autoři studie s názvem *An empirical study of the relationships between IT infrastructure flexibility, mass customization, and business performance* (Chung a spol., 2005) se zabývali vztahem mezi flexibilitou IT infrastruktury a výkonem byznysu organizace. Stanovili několik hypotéz, z nichž nejzajímavější říká, že integrace IT

infrastruktury má pozitivní efekt na výkonnost byznysu. Na tuto hypotézu navazuje další řetězec hypotéz, z nějž lze vyzdvihnout, že modularita IT infrastruktury má vliv na lepší přizpůsobení a to má vliv na lepší výsledky společnosti. Všechny hypotézy se autorům podařilo potvrdit a konstatují, že integrace a volba správné koncepce má velmi pozitivní vliv na činnost společnosti a je jednou z možných konkurenčních výhod. Práce samotná nicméně nijak neřeší otázku, jak koncepci IT infrastruktury vybrat.

Pod hlavičkou standardizační organizace IEEE byl realizován výzkum s názvem *An IT Infrastructure Patterns Approach to Improve IT Service Management Quality* (Ferreira da Silva a Brito e Abreau, 2010) zabývající se vytvořením vzorů pro koncipování IT infrastruktury. Jeho autoři označují IT infrastrukturu za základní stavební kámen pro kvalitní IT služby, díky kterým je možné zajistit hladký běh firemních procesů. Ve své práci rovněž kladou důraz na návrh takových vzorů, které jsou nezávislé na řešení jednotlivých dodavatelů. Tento výzkum ovšem není dokončen a sami autoři výše uvedený výstup považují za první krok v tvorbě metodiky.

Další prací je bulletin s názvem *IT Infrastructure Projects: A Framework for Analysis* (Grochow, 2015), který se zaměřuje na stanovení rámce pro posouzení potřeb infrastrukturních projektů v prostředí univerzit a analýzu problémů a rizik v implementaci a správě IT infrastruktury. Infrastrukturní projekty jsou rozděleny na počáteční implementaci, údržbu a vylepšování, reimplementaci a technologickou migraci. U každé skupiny hodnotí její přínosy, rizika a náklady. Práce se celkově snaží upozornit na fakt, že je pro IT manažery velmi náročné výše zmíněná posouzení učinit, což může mít za následek podhodnocení IT infrastruktury a její selhání.

Předchozí pohled z velké části sdílí i Peter Weil, Mani Subramani a Marianne Broadbent ve svém výzkumu s názvem *IT Infrastructure for Strategic Agility* (2002), ve kterém zmiňuje, že investice do IT infrastruktury jsou pro manažery jedny z nejnáročnějších úloh, protože musí s předstihem odhadnout potřeby byznysu i nové trendy trhu. Tento výzkum nicméně vzhledem ke svému stáří nezahrnuje nové trendy a tudíž se myšlenky autorů dají aplikovat pouze obecně.

Z výše uvedeného je zřejmé, že autoři považují problematiku koncipování IT infrastruktur za velmi důležitou pro samotnou existenci organizací na straně jedné a jako velmi problematickou pro manažerské rozhodování na straně druhé.

Všechny mnou nalezené práce se zaměřují a dále rozvíjejí pouze jednu z mnou v úvodu nastíněných koncepcí on-premise IT infrastruktury a nebo nabízejí pohled z vyšší málo technologicky konkrétní úrovně bez jasného vyhranění jednotlivých koncepčních směrů.

Metodiky popisované ve výše analyzovaných pracích se zaměřují pouze na jednu koncepci a nebo na jedno konkrétní prostředí. Jednoznačně se tak otevírá prostor pro práci, která vytvoří metodu, která bude stavět na neabstraktních postupech a technologiích a vezme v potaz aktuální stav poznání.

2.2 Vymezení on-premise IT infrastruktury a jejích koncepcí

Pojem on-premise IT infrastruktura se skládá ze dvou částí - on-premise a IT infrastruktura. Význam obou těchto pojmů vysvětluje tato kapitola. Dále se tato část práce bude zabývat vysvětlením pojmu datacentrum.

2.2.1 On-premise

On-premise (či on-premises) je anglické slovní spojení, které vyjadřuje, že něco je v rámci majetku či budov nějaké organizace. Anglický výkladový slovník (Cambridge University Press, 2016), popisuje pojem *„premises“* jako *„the land and buildings owned by some, especially by a company or organization“*, což lze do češtiny přeložit jako *„země a budovy vlastněné někým, typicky firmou nebo organizací“*. Ve spojení s pojmem *„IT infrastruktura“* se tak jedná o takovou IT infrastrukturu, která se nachází v rámci majetku organizace a je provozována interně.

Pod on-premise IT infrastrukturu by mohl být zařazen i privátní cloud, nicméně ten dle definice amerického úřadu NIST (Mell a Grance, 2011) může být provozován i

mimo režii organizace jako služba někým jiným než organizací samotnou a tudíž jako off-premise. V kontextu práce nelze privátní cloud považovat za samostatnou koncepci, jelikož se tento pojem váže spíše ke službám IT infrastruktury než k tomu, jak je postavena. Z těchto důvodů nebude privátní cloud uvažován.

2.2.2 IT infrastruktura a její koncepce

Existuje mnoho definic pojmu IT infrastruktura, které na celou problematiku nahlíží z různých úhlů a různé šíře. Jedna z nich říká, že „*IT infrastruktura je definována jako množina IT zdrojů a organizačních schopností, které jsou sdíleny napříč organizací a které poskytují základ pro vývoj IT aplikací a podporu business procesů*" (Broadbent, Weill a Neo, 1999, s. 160) a pod tento pojem tak zahrnuje nejen IT zdroje, ale i organizační schopnosti.

Jiná, rozsáhlejší, definice pod pojmem IT infrastruktura vidí „*Množinu sdílených IT zdrojů složenou z fyzické hardwarové základny, softwaru, komunikačních technologií, core aplikací a lidské komponenty dovedností, odbornosti, kompetencí, závazků, hodnot, norem a znalostí, které jsou kombinovány při vytváření IT služeb, které jsou typicky unikátní pro organizaci*“ (Masa’deh, 2013, s. 205), což lze skutečně označit za velmi extenzivní výklad pojmu, který se dotýká každého myslitelného aspektu co může pod pojem patřit.

Pro potřeby této práce je nicméně třeba pojem IT infrastruktura více zúžit pouze na technologické prostředky, na kterých jsou provozovány aplikace umožňující fungování byznysu společnosti. Tomuto pojetí pojmu IT infrastruktura nejvíce vyhovuje definice převzatá z amerického standardu NIST: „*Schopnost poskytnutá konzumentovi opatřit si výpočetní, úložné, síťové a další nezbytné počítačové prostředky, na kterých může rozmístit a provozovat libovolný software, mezi který patří operační systémy a aplikace.*“ (Mell a Grance, 2011). Tato definice byla zvolena právě pro svoje úzké zaměření na technologické prostředky, s nimiž pracuje tato diplomová práce.

Diplomová práce tedy pojem IT infrastruktura chápe jako označení pro technologické vybavení, které poskytuje prostředky jako je výkon nebo úložiště pro provoz

operačních systémů a aplikací. Takovou IT infrastrukturu je dále možné koncipovat mnoha způsoby, kde mezi hlavní tři patří již zmíněná třívrstvá koncepce, virtualizace celých operačních systémů a virtualizace na úrovni operačního systému. Popisu těchto koncepcí se bude dále věnovat třetí kapitola.

2.2.3 Datacentrum

Velmi dobrou definici pojmu *„datacentrum“* nabízí agentura Gartner ve svém webovém IT slovníku: *„The data center is the department in an enterprise that houses and maintains back-end information technology (IT) systems and data stores—its mainframes, servers and databases. In the days of large, centralized IT operations, this department and all the systems resided in one physical place, hence the name data center.“* (2016) Tuto definici lze volně přeložit tak, že datacentrum je oddělení ve společnosti, které uchová a stará se o backendové IT systémy (tj. servery a úložiště) v rámci jednoho centralizovaného místa. Jelikož se tato práce nezabývá organizační složkou IT, je význam tohoto pojmu zúžen na IT technologie samotné.

2.3 Atributy IT infrastruktury

Každou IT infrastrukturu lze definovat dle atributů, které ji popisují na základě funkčních a kvalitativních (ne funkčních) vlastností. Mezi funkční atributy se řadí například výpočetní kapacita procesorů a nebo velikost operační paměti. Jelikož se jedná o atributy, které popisují konkrétní IT infrastrukturu a nejsou přímo závislé na její koncepci, tato práce se jimi nebude zabývat. Na druhé straně ne funkční parametry ovlivňují schopnosti dané IT infrastruktury a koncepce na ně má významný vliv. Patří mezi ně například flexibilita, dostupnost a nebo bezpečnost IT infrastruktury.

Existuje spousta pohledů na atributy IT infrastruktury. Ty jsou ovlivněny jednak obdobím jejich vzniku, ale především koncepcí IT infrastruktury, který se snaží popisovat či poměřovat. Jako základní pohled byl vybrán výčet z knihy *IT*

Infrastructure Architecture - Infrastructure Building Blocks and Concepts (Laan, 2013), kde autor uvádí těchto osm základních nefunkčních atributů: d*ostupnost, škálovatelnost, spolehlivost, stabilita, testovatelnost, obnovitelnost, bezpečnost a výkon.*

Pojem „ne funkční“ výše zmíněný autor vysvětluje jako atributy, které nemají přímou souvislost s výkonem či kapacitou systémů, nicméně přímo ovlivňují kvalitu poskytovaných služeb.

Vzhledem k tomu, že tato práce pracuje se třemi relativně široce rozkročenými koncepcemi IT infrastruktur, bylo nutné jít hlouběji k významu jednotlivých atributů a některé z nich rozdělit a vnímat je samostatně. Došlo tak k doplnění atributů „elasticita“, „vertikální škálovatelnost“ a „horizontální škálovatelnost“, které vznikly rozdělením pojmu „škálovatelnost“. Druhým atributem, který byl rozdělen, je atribut „obnovitelnost“, jelikož pod sebou zahrnuje dvě oddělené problematiky „zálohování“ a „obnovení po výpadku“. Důvody tohoto rozdělení budou dále vysvětleny v této kapitole.

Atributy stabilita a spolehlivost lze naopak s ohledem na definici IT infrastruktury, se kterou pracuje tato práce, považovat téměř za synonyma a proto byly sloučeny pod jeden atribut s názvem „stabilita a spolehlivost“. Pokud IT infrastruktura není stabilní (ve smyslu, že nemá žádné výpadky), nelze ji z její povahy považovat za spolehlivou (ve smyslu, že nedochází ke ztrátě dat).

Jelikož výše uvedená práce je zaměřené pouze technicky, chybí v ní provozně-finanční atributy a z tohoto důvodu došlo k jejich doplnění. Tyto atributy sice nemají přímý dopad na fungování IT infrastruktury jako takové, nicméně mohou mít zásadní vliv na rozhodnutí o výběru koncepce IT infrastruktury. Řadí se mezi ně „jednorázové výdaje na pořízení technologického vybavení“ (CAPEX), „provozní výdaje na technologické vybavení“ (OPEX), „efektivnost“ a „spravovatelnost“ (EY, 2014).

Následující body stručně charakterizují jednotlivé atributy, protože na nich je závislá třetí kapitola diplomové práce, kde dochází k ohodnocení jednotlivých koncepcí IT infrastruktur vzhledem k těmto atributům.

- *Jednorázové výdaje na pořízení technologického vybavení.* Jedná se o takové výdaje, které musí organizace vynaložit na jednorázové pořízení hardwaru a softwaru, který umožní nasadit danou koncepci IT infrastruktury. K tíži těchto nákladů je možno rovněž zařadit případná školení pracovníků, kteří budou danou IT infrastrukturu obsluhovat.

- *Provozní výdaje na technologické vybavení.* Tento druh výdajů na rozdíl od jednorázových vyjadřuje takové náklady, které musí organizace pravidelně vynakládat na provoz IT infrastruktury. Lze sem zařadit výdaje za služby jako elektřina či správa budovy, nákup podpory na software či hardware a nebo jiné servisní služby spojené s IT infrastrukturou.

- *Rychlost nasazení a testování aplikací.* Jedná se o atribut, který vyjadřuje schopnost IT infrastruktury hbitě nasazovat a testovat velké množství aplikací. Jednotlivé koncepce se v tomto parametru mohou velmi lišit. Zatímco některé rychlé nasazování aplikací téměř znemožňují, jiné byly vytvořeny přímo s cílem tento atribut maximálně podpořit.

- *Elasticita.* Pojem elasticita definoval Nikolas Roman Herbst (2012, s. 2), jako: *„Stupeň jakým je systém schopen se adaptovat na změny v zátěži opatřením si či uvolněním dalších prostředků a to automaticky tak, že v každém bodě dostupné prostředky kopírují poptávku tak úzce jak je to možné.“*

- *Efektivnost.* Tento atribut vyjadřuje schopnost IT infrastruktury využívat interní a externí prostředky takovým způsobem, že při co nejnižších nákladech je IT infrastruktura schopna obsloužit co nejvíce rozličných požadavků (WebFinance, 2016). Význam tohoto pojmu se dále liší podle koncepce IT infrastruktury.

- *Dostupnost.* Jedná se o atribut, který pevně vyjadřuje (nejčastěji v procentech) podíl doby, po kterou byla IT infrastruktura nebo její součást v provozním stavu a schopná poskytovat služby (Techtarget, 2016). Vzhledem k povaze této práce je nicméně význam tohoto atributy míněn jako schopnost IT

infrastruktury udržet dostupnost na stanovené úrovni, na což může mít koncepce IT infrastruktury naprosto zásadní vliv.

- *Zálohování.* Pojem zálohování má mnoho významů a v různých rovinách pohledu zahrnuje rozličné přístupy a techniky. Tato práce chápe tento atribut jako přívětivost dané koncepce IT infrastruktury k zálohování obecně. Jednotlivé koncepce přináší různé postupy a techniky, které nabízí různě široké možnosti pro zálohování a to od aplikačních dat až po obrazy celých operačních systémů.

- *Obnovení po výpadku.* Pojem obnovení po výpadku je komplementem k zálohování. V tomto případě svou roli ovšem hraje nejen schopnost nasadit původní, zálohovaná, data zpět do produkčního prostředí, ale i schopnost vzpamatovat se z fatální chyby softwaru, hardwaru a nebo narušení z okolního prostředí. Nejčastějším příkladem narušení způsobeným okolním prostředím je přerušení napájení. Jednotlivé koncepce mohou být ze své podstaty více či méně schopné se z takových situací zotavit.

- *Spravovatelnost.* Jednotlivé koncepce se jednoznačně liší v tom, jak snadná či obtížná je jejich běžná správa a běžná práce s nimi. Obecně potom platí, že čím více abstrakce a automatizace koncepce zahrnuje, tím je snadněji spravovatelná. Ne vždy je nicméně vysoká míra abstrakce žádoucí, jelikož ve výjimečných situacích může způsobit obtížně řešitelné komplikace.

- *Stabilita a spolehlivost.* Pojmy stabilita a spolehlivost obecně vyjadřují dvě odlišné problematiky. Stabilita je obyčejně chápána jako vlastnost systému konstantně poskytovat své služby bez výkyvů v jejich kvalitě (výpadky, dlouhé odezvy, varující chybové hlášení, a podobně), zatímco spolehlivost označuje schopnost systému vyřídit všechny požadavky takovým způsobem, aby nedošlo k narušení způsob jejich přijetí, zpracování a uložení. V případě IT infrastruktury se nicméně jedná o dvě neoddělitelné oblasti, jelikož pokud není stabilní, nemůže být ani spolehlivá a naopak.

- *Výkon.* Každá koncepce IT infrastruktury způsobem svého návrhu ovlivňuje výkon, který poskytuje aplikacím v infrastruktuře běžícím. Z tohoto pohledu tak můžeme jednotlivé koncepce hodnotit jako takové, které nezpůsobují žádnou výkonovou penalizaci, až po takové, které kvůli dalším vrstvám abstrakce část prostředků poskytnutých hardwarem spotřebují pro svoji režii.

- *Bezpečnost.* Na bezpečnost lze nahlížet z mnoha různých hledisek a každý koncept IT infrastruktury otevírá jiný způsob pro neoprávněné vniknutí do systému.

- *Horizontální škálovatelnost.* Škálovatelnost popisuje autor André B. Bondi (2000, s. 4) jako: *„schopnost systému, sítě nebo procesu zvládnout narůstající množství práce nebo jeho potenciál být zvětšen, aby byl schopen toto množství zvládnout“.* Horizontální škálovatelnost tak vyjadřuje konkrétní způsob, jakým se IT infrastruktura s narůstající zátěží vypořádává a sice přidávám dalších uzlů. Úloha je poté paralelně zpracovávána více jinak nezávislými počítači.

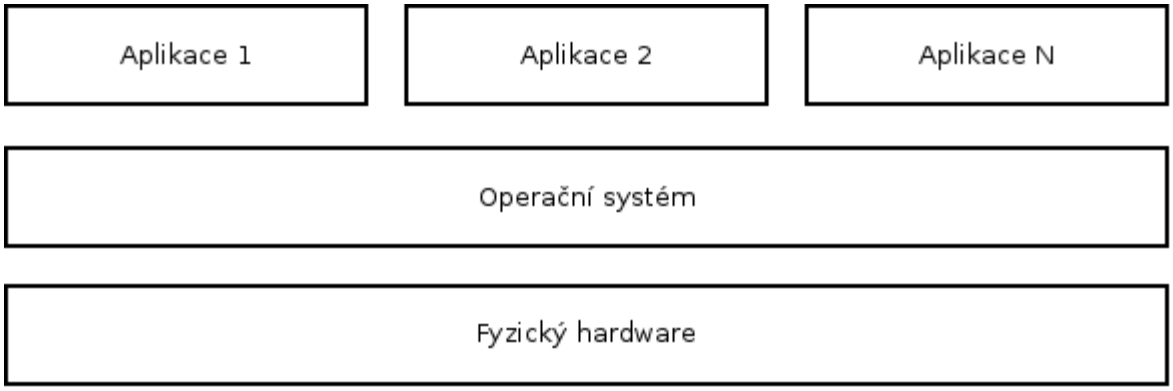
- *Vertikální škálovatelnost.* Vertikální škálovatelnost je opakem škálovatelnosti horizontální a je častěji využívána v aplikacích, které byly navrženy v dobách, kdy vývoj hardwaru dokázal uspokojit narůstající poptávku po výkonu v rámci jednoho uzlu. Vertikální škálování tak vyjadřuje takovou schopnost sytému, kdy potřebný výkon navíc je aplikaci dodám prostřednictvím navýšení výkonu jednoho uzlu.

3 Koncepce a technologie IT infrastruktury

Tato kapitola je věnována popisu tří základních koncepcí on-premise infrastruktur, které byly zmíněny hned v úvodu této práce. Jedná se o třívrstvou koncepci, virtualizaci celých operačních systémů a virtualizaci na úrovni operačního systému. U každé koncepce se tato práce bude zabývat jejím obecným popisem, historií vzniku a současným stavem. Dále vyjmenuje majoritní technologická řešení, která koncept implementují v reálném světě a popíše výhody a nevýhody daného konceptu oproti ostatním.

3.1 Fyzické servery a třívrstvá koncepce

Vznik této koncepce se váže k nástupu prvních operačních systémů, které jako mezivrstva mezi hardwarem a softwarem umožnily spouštět program, předat mu vstupní data a výstupní data zobrazit na výstupní zařízení. Takových programů mohlo běžet více najednou (známé pod anglickým termínem multitasking), operační systém dále poskytoval abstrakci hardwaru a správu zdrojů. Později operační systémy umožnily práci více uživatelů najednou a podporu pro propojení více počítačů prostřednictvím počítačových sítí (Silberschatz a Galvin, 1994).



Obrázek 1: Schéma třívrstvé koncepce

Tato práce zavádí pojem třívrstvá koncepce jako zkratku pro všechny historické i současné přístupy k řešení infrastruktur, kde jednotlivými vrstvami jsou hardware, operační systém a aplikace. Tento stav graficky znázorňuje obrázek 1. Dalším možným označením pro tyto typy infrastruktur je „nevirtualizovaná prostředí“, které bude tato práce rovněž používat. V současnosti je tento přístup postupně vytlačován konvergovanými koncepty na bázi různých virtuálních prostředí, nicméně i tak si ve spoustě aplikací zachovává mnoho klíčových výhod. Jejich popisem spolu nevýhodami se bude zabývat následující podkapitola. Pro větší přehlednost tato koncepce může být dále označována jako „koncepce 1“.

3.1.1 Výhody a nevýhody nasazení třívrstvé koncepce

Tato kapitola popisuje výhody a nevýhody třívrstvé koncepce. Syntézou publikovaných poznatků autorů Scotta Mattesona (2013), Paula Rubense (2012), Briena Poseye (2015) a osobních zkušeností autora této práce bylo identifikováno sedm výhod a sedm nevýhod třívrstvé koncepce. Tabulka 1 uvádí jejich základní výčet a přiřazuje jim jednoznačný identifikátor, který bude později dále použit. Identifikátor se skládá z označení koncepce (v tomto případě K1), a označení výhody či nevýhody (Vx či Nx).

Tabulka 1: Výčet výhod a nevýhod třívrstvé koncepce

Výhody	**Nevýhody**
Maximální efektivita využití systémových prostředků (K1V1)	Náročnější správa u rozsáhlejších instalací (K1N1)
Žádný výkonnostní dopad způsobený další vrstvou abstrakce (K1V2)	Horší využití systémových prostředků u málo vytížených aplikací (K1N2)
Nižší časové a finanční nároky implementace v malém prostředí (K1V3)	Obtížné nasazení HA pokud to nepodporuje přímo software (K1N3)
Nižší nároky na odbornost personálu(K1V4)	Obtížnější zálohování a obnova (K1N4)
Snazší pro nasazení aplikací, které škálují vertikálně (K1V5)	Pomalejší zotavení se z chyby hardwaru nebo softwaru (K1N5)
Snazší přímý přístup k hardwaru (K1V6)	Delší výpadky při aktualizacích systému (K1N6)
Nižší komplexnost celého řešení(K1V7)	Nižší flexibilita celé infrastruktury (K1N7)

Výhoda: Maximální efektivita využití systémových prostředků (K1V1)

Pokud pomineme dnes již prakticky nepoužívané řešení, kdy aplikace běží za absence operačního systému přímo na fyzickém hardwaru, je tato koncepce optimální z hlediska efektivity využití systémových prostředků. Je to způsobeno minimální mírou abstrakce, kdy fyzický hardware a aplikaci dělí pouze operační systém a nevznikají tak dalšími vrstvami vyžadované nároky na operační paměť, čas procesoru a nebo diskové úložiště. V prostředích s nízkými požadavky na konvergenci, které zároveň

nepotřebují těžit z výhod virtualizovaných prostředí (a nebo si je umí zajistit jinak), mohou tyto úspory systémových prostředků vést k nemalým úsporám finančním.

Výhoda: Žádný výkonostní dopad způsobený další vrstvou abstrakce (K1V2)

Podobně jako u předchozí výhody koncepce nevyužívá další vrstvu, přes kterou musí proudit všechny požadavky softwaru na vstup či výstup nebo jakékoliv další systémové či hardwarové prostředky. To s sebou opět přináší nižší nároky na celkový výkon hardwaru, ale i lepší odezvu a propustnost celého systému. Pro software, u kterých je klíčová co nejnižší odezva na vstup, mezi které patří například systémy pro obchodování na burze, je jakákoliv virtualizovaná koncepce nevhodná. To stejné platí u systémů s požadavkem na maximální prostupnost, jako jsou například aplikace pro výpočty všeho druhu.

Výhoda: Nižší časové a finanční nároky implementace v malém prostředí (K1V3)

V případě stavby menších infrastruktur, mohou časové a finanční nároky na implementaci virtualizovaných prostředí být v porovnání s přínosy příliš vysoké. Virtualizace v malém nasazení může neúměrně prodloužit instalaci a testovací provoz infrastruktury a rovněž zatížit rozpočet dalšími poplatky za práci techniků a případné další softwarové licence.

Výhoda: Nižší nároky na odbornost personálu (K1V4)

Tato výhoda úzce souvisí s předchozí, jelikož v mnoha případech virtualizované prostředí zavádí oproti třívrstvé koncepci spoustu nových nároků nejen na nasazení, ale i na provoz. To s sebou přináší vyšší nároky na kvalifikaci obslužného personálu i vyšší finanční požadavky na zajištění podpory ze strany výrobce či dodavatele řešení. V malých prostředích mohou tyto více náklady vést k absolutnímu smazání všech výhod, které by virtualizovaná koncepce přinesla.

Výhoda: Snazší pro nasazení aplikací, které škálují vertikálně (K1V5)

Typickým příkladem softwaru, který škáluje vertikálně, jsou databázové systémy. Ty vyžadují velké množství operační paměti a procesorových jader, které jsou schopny nejefektivněji využít v rámci jedné instance softwaru a nikoliv více spolupracujících uzlů. Nasazení takových aplikací je z důvodu jejich těžkosti v rámci virtualizovaných prostředí velmi nákladné až nemožné a absolutně popírá všechny výhody takových prostředí. Třívrstvá architektura se tak jeví jako nejvhodnější pro tento druh aplikací.

Výhoda: Snazší přímý přístup k hardwaru (K1V6)

U nasazení, která vyžadují přímý přístup k hardwaru, jakým může být například grafická karta nebo jiné speciální obvody, je přímé použití fyzického serveru často jediným možným řešením, jelikož přístup přes další vrstvy může být velmi komplikovaný až nemožný. Jako příklad takových nasazení mohou sloužit servery využívající GPGPU (General-purpose computing on graphics processing units) a podobné obvody pro výpočty náročných úloh.

Výhoda: Nižší komplexnost výsledného řešení (K1V7)

Třívrstvá architektura obecně obsahuje menší počet softwarových komponent, tudíž i menší počet míst, ve který může dojít k selhání jak softwaru tak k lidské chybě. Absence další vrstvy tak umožňuje rychlejší odhalení chyby a její nápravu.

Nevýhoda: Náročnější správa u rozsáhlejších instalací (K1N1)

V případě rozsáhlejšího nasazení na větší počet fyzických serverů je obecně obtížnější spravovat jednotlivé nevirtualizované operační systémy. Mezi operačním systémem a hardwarem neexistuje žádná další vrstva, která by usnadňovala přístup a zároveň poskytovala nástroje pro správu hardwaru a operačního systému a zároveň byla oddělená od aplikací, které v operačním systému běží. V případě restartu operačního systému nebo jeho pádu je operátor odkázán pouze na vzdálený přístup ke konzole pomocí BMC (Baseboard Management Console) daného serveru. Pokud žádná taková

a nebo podobná technologie není dostupná, musí operátor zasáhnout fyzicky přímo na místě.

Nevýhoda: Horší využití systémových prostředků u málo vytížených aplikací (K1N2)

V dnešní době existuje v organizacích spousta aplikací, které vyžadují pro svůj běh tak málo systémových prostředků, že i ten nejlevnější dostupný hardware je pro jejich provoz předimenzovaný. Pokud takové aplikace vyžadují oddělené prostředí (z důvodu odlišných operačních systémů a nebo z důvodu bezpečnosti), nezbývá, než je provozovat na více fyzických serverech, což je značně neefektivní. Společnost musí nakoupit a provozovat větší množství fyzického hardwaru, než je z hlediska výkonu a ostatních systémových prostředků potřeba a to s sebou nese větší náklady jak na akvizice tak i na provoz.

Nevýhoda: Obtížné nasazení HA pokud to nepodporuje přímo software (K1N3)

Systémy s vysokou dostupnosti (HA – High Availability) lze obecně rozdělit do dvou kategorií. První z nich tento problém řeší přímo na aplikační úrovní a tudíž není závislá na nižších vrstvách. Druhá kategorie aplikací nicméně vysokou dostupnost neumí řešit sama a spoléhá se na různá clusterová řešení, která monitorují běh aplikace potažmo celého operačního systému a v případě chyby se postarají o její spuštění jinde, na jiném uzlu clusteru. Z toho je zřejmé, že klasická třívrstvá koncepce neumí u aplikací nepřipravených na vysokou dostupnost zajistit vysokou dostupnost a v případě selhání hardwaru a nebo operačního systému dojde i k selhání celé služby, kterou aplikace poskytuje.

Nevýhoda: Obtížnější zálohování a obnova (K1N4)

Podobně jako u předchozích nevýhod i zde absence další vrstvy mezi operačním systémem, na kterém běží aplikace, a hardwarem, přináší menší objem možností, jak provádět zálohování a obnovu. Pravidelné bezvýpadkové zálohování obrazu celého operačního systému je v takovýchto případech velmi složité až nemožné. V opačném případě, kdy je potřeba ze zálohy obnovit celý operační systém, vznikají dlouhé časové

prodlevy, které bez dalšího vybavení (například externího diskového pole) znamenají výpadek v řádu hodin.

Nevýhoda: Pomalejší zotavení se z chyby hardwaru nebo softwaru (K1N5)

Každá chyba hardwaru a nebo operačního systému znamená v drtivé většině nutnost provést restart celého fyzického serveru. Fyzický hardware (na rozdíl od virtuálního) nicméně startuje velmi pomalu a to v řádech jednotek až desítek minut, což má za následek delší nedostupnost poskytované služby.

Nevýhoda: Delší výpadky při aktualizacích systému (K1N6)

U každého operačního systému je potřeba instalovat aktualizace, které vyžadují jeho restart. Některé operační systémy (například Microsoft Windows) navíc instalují aktualizace před vypnutí systému (ovšem v době, kdy už neběží ostatní aplikace) a nebo naopak po startu ještě před spuštěním aplikací. U nevirtualizovaných prostředí neexistuje žádný způsob jak tento výpadek zkrátit a ten tak spolu s delší dobou inicializace hardwaru může narůst až do řádu desítek minut.

Nevýhoda: Nižší flexibilita celé infrastruktury (K1N7)

U třívrstvé koncepce neexistuje snadná cesta jak přemigrovat jeden operační systém či aplikaci z jednoho fyzického hardwaru na druhý. Stejně tak je náročnější až nemožné zvýšit výkon procesoru serveru, přidat operační paměť a nebo diskové úložiště. Všechny tyto zásahy znamenají jednak fyzickou přítomnost operátora a taky výpadek v řádu desítek minut. Podobný problém nastává v případě potřeby klonování serveru pro účely testování a nebo vykrytí výkonové špičky.

3.1.2 Dostupná technologická řešení

Tato kapitola podrobněji rozvine dostupná technologická řešení, která se v současné době uplatňují při nasazování a správě jednotlivých vrstev nevirtualizovaných prostředí. V případě hardwaru se bude věnovat především možnostem jeho snazšího

nasazení a správy a u operačních systémů zacílí na jejich instalaci a vzdálenou správu. Poslední část se bude věnovat správě a orchestraci aplikací instalovaných v těchto prostředích. U každé oblasti se bude odkazovat na předchozí kapitolu s komentářem, jak dané řešení buď prohlubuje nevýhody a nebo vylepšuje výhody tohoto konceptu.

Hardware

Moderní servery obsahují integrovaný, na operačním systému nezávislý, obvod, který umožňuje pomocí počítačové sítě vzdálenou správu samotného hardwaru. Tomuto řešení se říká Baseboard Management Console a u různých výrobců je implementováno pod různými názvy (příklad: HP iLO, Lenovo IMM, atd.), nicméně každý výrobce se řídí jednotnou specifikací IPMI (Intelligent Platform Management Interface). Přístup do tohoto rozhraní probíhá v drtivé většině případů přes web pomocí integrovaného webového serveru a nebo přes protokol SSH (Secure Shell Protocol).

Obvyklou strategií výrobců je nabízet více verzí tohoto vzdáleného managementu, z nichž ty nejnižší obvykle umí pouze monitorovat stav serveru (teploty, napětí, otáčky ventilátorů) a provádět akce s napájením (restart, vypnutí a podobně). Vyšší verze se umí připojit na konzolu serveru (klávesnice, monitor, myš) a zprostředkovat tak přístup k serveru vzdáleně bez nutnosti fyzické přítomnosti. Stejně tak je možné připojit obraz instalačního média k virtuální mechanice. Tato řešení nám tedy umožňují plně ovládat hardware způsobem velmi blízkým fyzické přítomnosti.

Velkou nevýhodou takovýchto řešení je jejich roztříštěnost (různé verze různých výrobců) a jejich velmi složitá integrace do jednoho řešení. Dalším problémem je bezpečnost takovýchto řešení, jelikož jejich zabezpečení je plně závislé na výrobci serveru, které nikdy nebývá na takové úrovni, jakým bývají zabezpečeny operační systémy. Z výše uvedeného je zřejmé, že v malých prostředích, nejlépe s jedním typem hardwaru (od jednoho výrobce), lze díky těmto technologiím některé nevýhody obejít, nicméně s o to větším důrazem na bezpečnost.

Operační systémy

Mezi běžně používané operační systémy se řadí různé verze Microsoft Windows, distribuce systému GNU/Linux a případně UNIX či jeho klony. Všechny tyto operační systémy obsahují nástroje pro hromadnou instalaci a správu.

Prvním krokem je instalace operačního systému, u které si v masovém nasazení můžeme vše předdefinovat tak, aby proběhla automaticky. V případě operačních systémů z rodiny Windows existuje řešení známé jako Microsoft Deployment Toolkit, ve kterém je možnost nastavit širokou škálu parametrů instalace od rozdělení pevného disku přes vložení licenčního klíče až po konfiguraci sítě. Rovněž je možné do instalačního obrazu vložit další ovladače nebo software. Instalátory distribucí GNU/Linux přistupují k celé věci trochu jinak, kdy je možno celou konfiguraci instalace popsat jedním konfiguračním souborem známým jako Kickstart.

Vzdálená správa je u každého operačního systému řešena rovněž odlišně. Microsoft Windows primárně využívá Vzdálenou plochu, která, jak už název napovídá, zobrazí obsah pracovní plochy na vzdáleném počítači pomocí počítačové sítě. Ověřování uživatele probíhá buď pomocí hesla, čipovou kartu a nebo skrz Bránu vzdálené plochy. Distribuce operačního systému GNU/Linux primárně používají přístup přes SSH, což je zabezpečený protokol skrze který je možné spustit shell a nebo tunelovat jakákoliv jiná data. Možností jak ověřit uživatele je zde mnohem více. Patří mezi ně ověření uživatelským jménem a heslem, privátním klíčem nebo čipovou kartou.

Správa aplikací

Poslední vrstvou třívrstvé koncepce jsou aplikace a jejich správa. Operační systémy z rodiny Microsoft Windows standardně obsahují nástroje pro správu skupinových politik, pomocí kterých je možné konfigurovat nejen operační systém, ale i aplikace. V případě většího nasazení je možné politiky distribuovat skrz Active Directory, čímž správce získává jeden centrální bod pro konfiguraci. Přes skupinové politiky je rovněž možné instalovat software, nicméně se spoustou omezení (například software musí být zabalen jako MSI balíček). Pro centrální instalaci softwaru proto existuje nástroj Microsoft System Center, který umožňuje instalovat jakýkoliv software v rámci

rozlehlých sítí. Důležitou součástí tohoto nástroje je rovněž komponenta, která umožňuje monitorovat průběh instalace v rámci všech spravovaných počítačů.

U operačních systémů z rodiny GNU/Linux existují nástroje pro hromadnou správu operačních systémů, které většinou integrují správu i instalaci aplikací v rámci jednoho řešení. Prvním z nich je software Spacewalk (komerční verze známa jako RedHat Satellite), která byl primárně vytvořen pro distribuci aktualizací a instalaci softwaru v rámci spravovaných serverů. Funkcionalita byla nicméně později rozšířena i o správu konfigurace (konfiguračních souborů). Druhým zástupcem je software Ansible, který je více zaměřen na hromadnou konfiguraci. Tu udržuje ve formě takzvaných playbooků, což jsou soubory, které obsahují informaci jakým způsobem dostat systém do požadovaného stavu (ať už instalací programu a nebo změnou konfigurace). Podstatným rozdílem oproti Spacewalku je, že nevyžaduje běh jako serverová aplikace a ani klienta na spravovaných počítačích.

3.1.3 Dopady koncepce vzhledem atributům

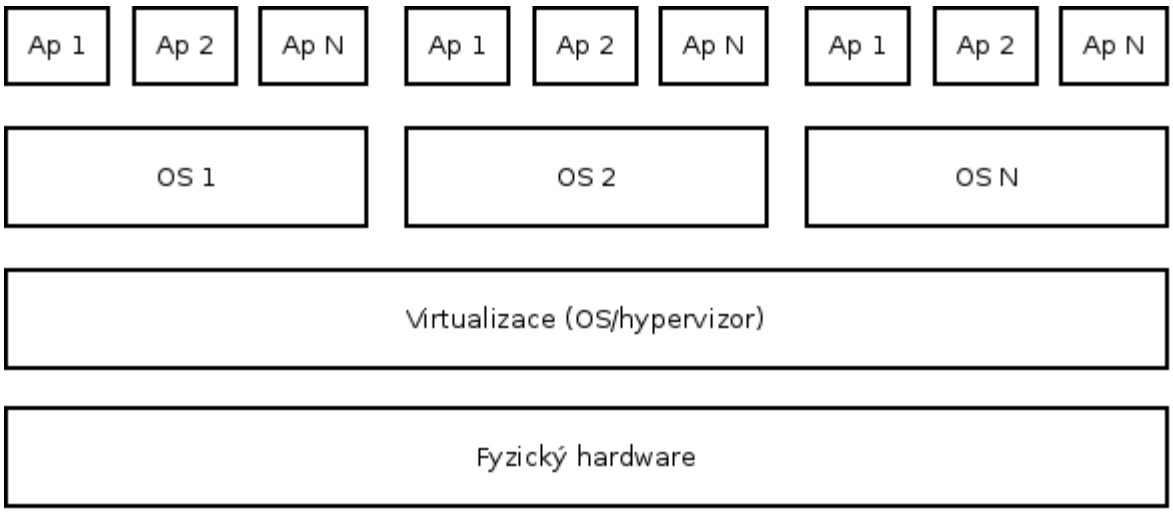
Cílem této kapitoly je stručně, formou tabulky, shrnout charakteristiky třívrstvé koncepce a na tomto základě formulovat dopady vzhledem k atributům definovaným ve druhé kapitole této práce. První sloupec tabulky 2 obsahuje atribut, ve druhém sloupci se nachází jeho charakteristika formou odkazu na výše popsanou výhodu či nevýhodu a v posledním sloupci je vyhodnocen dopad. Dopad je stanoven na základě subjektivního hodnocení autora a může nabývat hodnot *negativní, spíše negativní, neutrální, spíše pozitivní* a *pozitivní* a nebo jejich rozsahu.

Tabulka 2: Dopady třívrstvé koncepce vzhledem k atributům IT infrastruktury

Atribut	**Charakteristika**	**Dopad**
Jednorázové výdaje na pořízení technologického vybavení	K1V3, K1V4	Neutrální až pozitivní
Provozní výdaje na technologické vybavení	K1V1, K1V4	Neutrální až pozitivní
Rychlost nasazení a testování a aplikací	K1N1, K1N7	Spíše negativní až negativní
Elasticita	K1N1, K1N7	Negativní
Efektivnost	K1V1, K1N2	Neutrální až spíše negativní
Dostupnost	K1N4, K1N6	Spíše negativní
Zálohování	K1N4, K1N5	Spíše negativní
Obnovení po výpadku	K1N4	Spíše negativní
Spravovatelnost	K1N1	Spíše negativní
Stabilita a spolehlivost	K1V7, K1N3	Neutrální až spíše negativní
Výkon	K1V1, K1V6	Pozitivní
Bezpečnost	K1N1,	Neutrální až spíše negativní
Horizontální škálovatelnost	K1N3, K1N7	Spíše negativní
Vertikální škálovatelnost	K1V5	Pozitivní

3.2 Virtualizace celých operačních systémů

Virtualizace je technologie, která vytváří vrstvu nad fyzickým hardwarem a abstrahuje jeho fyzické zdroje a dále je poskytuje hostovanému (virtuálnímu) počítači, jehož operační systém je vnímá, jako by byly fyzické. Mezi tyto zdroje patří procesor, operační paměť, úložiště a nebo síťová karta. Koncept jako takový byl popsán již v roce 1974 autory Popekem a Goldbergem (1974), nicméně jeho masivní rozvoj umožnil až nárůst výkonu procesorů v první polovině nultých let tohoto století spolu s uvedením hardwarové podpory v procesorech Intel Pentium 4 s označeními 672 a 662 (Všetečka, 2005). Na poli x86 procesorů mají podporu virtualizace v současné době implementovánu oba výrobci, Intel pod názvem VT-x a AMD pod zkratkou AMD-v. Základní schéma virtualizace celých operačních systémů znázorňuje obrázek 2.



Obrázek 2: Schéma koncepce virtualizace celých operačních systémů

Autoři Popel s Goldbergem rozlišují dva základní typy, které označili jako nativní a hostovaný. Zatímco v případě prvního z nich je hypervizor spuštěn přímo nad fyzickým hardwarem a poskytuje k němu přímý přístup, druhý typ v sobě zahrnuje jako mezivrstvu i hostitelský operační systém jehož je hypervizor součástí. V dnešní době se nicméně rozdíly mezi oběma typy prakticky slévají a tak hypervizory prvního

typu je díky jejich komplexnosti možné považovat téměř za plnohodnotné operační systémy a hypervizory druhého typu naopak běží na úrovni jádra a tudíž mají přímý přístup k hardwaru (mimo ostatní vrstvy operačního systému). Vzhledem k těmto faktům tato práce mezi nimi nebude dále rozlišovat. Pro snazší orientaci může být tato koncepce označována jako „koncepce 2“

3.2.1 Výhody a nevýhody virtualizace celých operačních systémů

Cílem této kapitoly je popsat výhody a nevýhody koncepce virtualizace celých operačních systémů. Vznikla za pomoci syntézy prací autorů Sary Angeles (2014), Jacka Wallena (2013), Puala Maha (2012) a vlastního pozorování autora této práce. Bylo identifikováno celkem sedm výhod a šest nevýhod koncepce virtualizace celých operačních systémů. Tabulka 3 uvádí jejich základní výčet a přiřazuje jim jednoznačný identifikátor, který bude později dále použit. Identifikátor se skládá z označení koncepce (v tomto případě K2), a označení výhody či nevýhody (Vx či Nx).

Tabulka 3: Výčet výhod a nevýhod koncepce virtualizace celých operačních systémů

Výhody	**Nevýhody**
Snazší správa u rozsáhlejší instalací (K2V1)	Nižší efektivita využití systémových prostředků (K2N1)
Lepší využití fyzických hardwarových prostředků u málo vytížených aplikací (K2V2)	Výkonnostní dopad (K2N2)
Snadné nasazení HA na úrovni virtualizace (K1V3)	Vyšší nároky na implementaci a personál (K2N3)
Snadné zálohování a obnova (K2V4)	Nevhodné pro aplikace škálující vertikálně (K2N4)
Rychlé zotavení se z chyby hardwaru nebo softwaru (K2V5)	Obtížnější přímý přístup k hardwaru (K2N5)
Minimalizace výpadků při údržbě IT infrastruktury (K2V6)	Vyšší komplexnost výsledného řešení (K2N6)
Vysoká flexibilita IT infrastruktury(K2V7)	

Výhoda: Snazší správa u rozsáhlejších instalací (K2V1)

V případě rozsáhlejších a nebo heterogenních instalací (různý hardware, operační systémy a aplikace) je v případě použití virtualizace snazší správa celé IT infrastruktury. Mezi fyzickým hardwarem a operačním systémem existuje další vrstva, která sjednocuje základní nástroje pro správu fyzických i virtuálních počítačů a tudíž pro základní operace není nutné používat specifické nástroje toho kterého hardwaru a nebo operačního systému. Rovněž pro zásahy na úrovni hardwaru virtuálního

počítače není třeba fyzická přítomnost a vše je možné provést skrz nástroje virtualizačního softwaru.

Výhoda: Lepší využití fyzických hardwarových prostředků u málo vytížených aplikacích (K2V2)

Virtualizace umožňuje provozovat více nezávislých virtuálních počítačů v rámci jednoho fyzického serveru, což je obzvlášť výhodné u aplikací, které vyžadují vlastní prostředí, avšak nespotřebovávají mnoho systémových prostředků. Virtualizace tak může uspořit mnoho nákladů na nákup a provoz velkého množství málo vytížených serverů, místo kterých může postačovat menší množství, u nějž dochází k optimálnímu využití hardwarových prostředků.

Výhoda: Snadné nasazení HA na úrovni virtualizace (K2V3)

Virtualizace otvírá možnosti pro nasazení řešení vysoké dostupnosti nezávislých na podpoře operačního systému a nebo aplikací. Takováto řešení pracují s virtuálním počítačem jako celkem, tudíž operační systém ani aplikace není třeba jakkoliv upravovat. V případě, že dojde k selhání virtuálního počítače (ať už přímo na úrovni virtualizovaného operačního systému a nebo celého virtualizačního uzlu), virtualizační vrstva se postará o jeho opětovné spuštění na jiném uzlu virtualizačního clusteru. Nejpokročilejší řešení jsou navíc schopny v reálném čase zrcadlit i obsah operační paměti hostovaných virtuálních strojů na více uzlů.

Výhoda: Snadné zálohování a obnova (K2V4)

Díky přítomnosti další vrstvy mezi hostovaným operačním systémem a hardwarem se otevírá celá škála možností pro zálohování a obnovu. Předně je možné zálohovat virtuální počítače jako celek (což je díky blokovému čtení výrazně rychlejší než záloha po souborech) a to i přímo za běhu. Díky tomu nedochází k výpadku poskytovaných služeb a zálohování je možné řešit centrálně a nezávisle na hostovaných operačních systémech a aplikacích. Analogicky k zálohování probíhá obnova, kdy pro návrat do předešlého stavu stačí nastartovat virtuální počítač z požadovaného obrazu. Díky

klonování je tak možné provádět s virtuálním počítačem všechny myslitelné operace bez rizika ohrožení produkčního prostředí.

Výhoda: Rychlé zotavení se z chyby hardwaru nebo softwaru (K2V5)

Vzhledem k abstrakci fyzického hardwaru startují virtuální počítače velmi rychle, tudíž je zotavení se z chyby také velmi rychlé. V případě jakýchkoliv potíží hostovaného operačního systému je rovněž velmi snadné okamžitě a vzdáleně přistoupit na konzoly virtuálního serveru nebo rovnou do jeho souborového systému. Výše zmíněné technologie tak rapidně snižují dobu výpadku.

Výhoda: Minimalizace výpadků při údržbě IT infrastruktury (K2V6)

Díky možnosti provádět živou bezvýpadkovou migraci (přenos) běžících virtuálních počítačů na jiné uzly virtualizačního clusteru, je možné provádět servis hardwaru nebo softwaru fyzických serverů bez přerušení poskytovaných služeb. Přínosem virtualizace je krátká doba startu virtuálního počítače. Tím dochází k minimalizaci výpadků při údržbě IT infrastruktury. Rovněž je možné využít techniky klonování a do produkčního prostředí nasazovat aktualizované obrazy operačních systémů s minimální výpadkem.

Výhoda: Vysoká flexibilita IT infrastruktury (K2V7)

Vzhledem k tomu, že hostovaný operační systém je kompletně oddělen od fyzického hardwaru, je možné provádět celou škálu operací umožňujících pružně spravovat IT infrastrukturu. Jak již bylo zmíněno výše, je možné provádět klonování virtuálních počítačů, jejich živou migraci a nebo navyšování či ponižování dostupných hardwarových prostředků (operační paměť, procesorová jádra, diskové úložiště a podobně). Není potřeba fyzická přítomnost operátora a pokud jsou požadované hardwarové prostředky fyzicky dostupné, jejich přidělení je otázkou okamžiku. Vzhledem k tomu, že virtualizační uzly typicky disponují velkým množstvím systémových prostředků, existuje spousta prostoru pro vykrytí výkonových špiček, na které by u fyzického serveru dimenzovaného pro typické zatížení jedné aplikace nebyl prostor.

Nevýhoda: Nižší efektivita využití systémových prostředků (K2N1)

Virtualizace jako každá jiná softwarová vrstva má svoji režii, která sice pro většinu aplikací může být zanedbatelná, nicméně ve specifických případech (zejména tehdy, když nelze uplatnit žádnou z výše uvedených výhod) může zbytečně spotřebovávat hardwarové prostředky (čas procesoru, operační paměť, diskové úložiště, atd.), které by jinak našly využití pro chod provozované aplikace.

Nevýhoda: Výkonnostní dopad (K2N2)

Vzhledem k tomu, že všechna práce s hardwarovými prostředky probíhá skrz další vrstvu, dochází k delším odezvám a potenciálně nižší propustnosti. U aplikací, které jsou na tyto parametry citlivé tak může dojít i k omezení jejich funkčnosti a zásadnímu výkonnostnímu dopadu. Pokud potřebujeme využít maximálního potenciálu hardwaru pro chod jedné aplikace, je virtualizace nevhodná.

Nevýhoda: Vyšší nároky na implementaci a personál (K2N3)

Implementace a správa virtualizace je obecně náročnější na odbornost než třívrstvá koncepce, což s sebou přináší zvýšené nároky jak na návrhové a implementační schopnosti tak na obslužný personál obecně. Virtualizace u velmi malých infrastruktur tak může přinést více nákladů (provozních i jednorázových) než přínosů.

Nevýhoda: Nevhodné pro aplikace škálující vertikálně (K2N4)

Aplikace, které škálují vertikálně (mezi ně patří například databázové systémy) je obecně nevhodné provozovat v rámci virtuálních prostředí, jelikož při škálování do výšky se vytrácí většina jejich výhod. Manipulace (klonování, zálohování, migrace) s velmi velkými virtuálními stroji je obecně velmi náročná až nemožná.

Nevýhoda: Obtížnější přímý přístup k hardwaru (K2N5)

Pokud aplikace vyžaduje přímo přistupovat ke konkrétnímu fyzickému hardwaru, představuje virtualizace nežádanou překážku, kterou může být složité až nemožné

obejít. Může to být způsobené nedostatkem fyzicky dostupných prostředků (například počet fyzických síťových karet je omezen hardwarovými možnostmi serveru) a nebo nepodporováním přístupu k danému hardwaru (například TPM (Trusted Platform Module) čip nebo grafická karta).

Nevýhoda: Vyšší komplexnost výsledného řešení (K2N6)

Virtualizace obecně obsahuje větší množství softwarových komponent, které mohou selhat a nebo způsobit chybu. Kvůli tomu je typicky náročnější najít příčinu problému (extrémní příklad je vadný firmware fyzického serveru, který způsobuje snížený výkon virtuálních počítačů) a jeho následné řešení.

3.2.2 Dostupná technologická řešení

Tato kapitola se bude věnovat představení dostupných technologických řešení, mezi něž patří KVM, XEN, Hyper-V a ESXi. První dvě z nich jsou virtualizační systémy s otevřeným (svobodným) zdrojovým kódem, které vyvíjí velké technologické společnosti spolu s dobrovolníky z celého světa. Produkty Hyper-V a ESXi jsou komerční, jež vyvíjí společnosti Microsoft potažmo VMWare.

KVM

Zkratka KVM vyjadřuje pojem Kernel-based Virtual Machine, což je virtualizační řešení, jež je součástí jádra operačního systému Linux. Jedná se o nejmladší řešení, které plně staví na hardwarové podpoře virtualizace moderních procesorů. V uživatelském prostoru je virtualizace implementována pomocí upraveného emulátoru QEMU (Quick EMUlator) a virtualizačního API libvirt.

Hypervizorem je tak samotné jádro Linux, v němž jsou jednotlivé virtuální počítače implementovány jako standardní procesy. Součástí celého ekosystému okolo KVM jsou jak emulované ovladače pro základní hardware (síťová karta, řadič pevného disku, atd.), tak i paravirtualizované (pomocí ovladačů virtio), které mají implementován přímý přístup k hardwaru. Existuje podpora pro širokou škálu typů

úložišť pro virtuální počítače od souborů (RAW či QCOW), přes logické správce diskových oddílů (LVM nebo Btrfs), softwarově definované úložiště (Ceph, Gluster) až po přímý přístup k fyzickým úložištím (přímo připojené disky, iSCSI, Fibrechannel, atd). Stejně široké možnosti jsou i u podpory sítí, kdy je možné vytvářet virtuální síťové karty, softwarově definované sítě nebo switche, pokročile routovat, používat virtuální sítě a nebo přímo přistupovat k fyzickým síťovým kartám.

Mezi pokročilé funkce KVM patří například KSM (Kernel Samepage Merging), které umožňuje v operační paměti slučovat stejné stránky pocházející z různých virtuálních počítačů a nebo možnost přidělit virtuálním počítačům dohromady více procesorových jader a operační paměti, než je na hostitelském počítači dostupné. KVM/libvirt rovněž podporuje živou migraci (přenos) virtuálních počítačů a ukládání jejich stavu na disk.

Všechny výše zmíněné technologie jsou dostupné zdarma například v distribucích CentOS nebo Debian, nicméně společnosti jako Redhat nebo SuSE na nich staví řešení s komerční podporou. Jelikož KVM nabízí plnou virtualizaci, je na něm možné provozovat rozličné operační systémy od Unixu přes Linux až po Windows.

XEN

Druhý zástupce open source řešení je hypervizor XEN, který je podobně jako KVM součástí linuxového jádra. Jelikož tento systém vznikl ještě v době, kdy procesory přímo nepodporovaly virtualizaci, umí pracovat ve dvou režimech. Prvním z nich je paravirtualizace, která od hostovaného operačního systému vyžaduje pro tento režim podporu (avšak od procesoru nikoliv). Druhým režimem je plná virtualizace, která vyžaduje podporu procesoru, ale chová se stejně jako KVM (emuluje hardware pomocí QEMU a tudíž nevyžaduje podporu ze strany hostovaného operačního systému).

Všechny možnosti okolo úložišť a síťování jsou téměř totožné jako u KVM. XEN je stejně jako KVM dostupný zdarma v rámci distribucí jako je CentOS nebo Debian, nicméně není příliš podporován ze strany dodavatelů softwaru s podporou, jelikož došlo k jeho obecnému nahrazení právě systémem KVM. Obecně se tak jedná o řešení,

které je spíše v útlumu a pokud uživatel nenajde specifický důvod pro jeho volbu, je KVM jednoznačně lepší volba.

Hyper-V

První zástupce čistě komerčních řešení je Hyper-V vyvíjené společností Microsoft, které je součástí serverových operačních systémů Windows. Podobně jako u KVM jeho vývoj započal až po integraci podpory virtualizace do procesorů a tudíž nabízí plnou virtualizaci. Mezi oficiálně podporované operační systémy patří samozřejmě Windows, ale i SuSE Enterprise Linux nebo Redhat Enterprise Linux.

Možnosti úložišť jsou obecně menší než u KVM, jelikož Hyper-V primárně jako úložiště virtuálních počítačů používá soubory VHD (Virtual Hard Disk), které jsou uloženy na souborovém systému NTFS nebo ReFS. Tyto souborové systémy nicméně mohou spočívat (být uloženy) na takzvaných Cluster Shared Volumes, které jsou online replikované v rámci více serverů v clusteru a Hyper-V díky tomu podporuje živou migraci (přenos). Základní možnosti sítí jsou podobné jako u KVM.

Výhodou a zároveň nevýhodou Hyper-V je jeho integrace s ostatními produkty společnosti Microsoft. Díky tomu velmi dobře fungují technologie jako je například App-V (virtualizace aplikací) a nebo vzdálená virtualizace desktopu skrz Vzdálenou plochu (Remote Desktop Connection). Uzavřenost celého systému na druhou stranu podstatně znemožňuje systém používat jinak, než přímo k účelu na který byl navržen, snadno jej rozšiřovat a integrovat s jinými softwarovými systémy.

ESXi

Druhým komerčním, řešením je ESXi, které je vyvíjeno společností VMware. ESXi (někdy označováno jako vSphere) je produkt nezávislý na jiném operačním systému a je tak vlastně sám o sobě operačním systémem stvořeným pouze pro potřeby virtualizace, což je rozdíl oproti předchozím řešením, které jsou postaveny na obecných operačních systémech. Výčet podporovaných hostovaných operačních systému je podobný jako u předchozích řešení.

Podobně jako KVM i ESXi podporuje živou migraci v rámci více fyzických serverů, přidělení více operační paměti než je fyzicky dostupné a nebo technologii Transparent Page Sharing (obdoba KSM u KVM) umožňující sdílet stejné stránky v paměti různých virtuálních počítačů. Velkou výhodou ESXi je široká škála řešení od jiných dodavatelů, které rozšiřují možnosti síťování, úložišť a nebo zálohování.

3.2.3 Dopady koncepce vzhledem k atributům

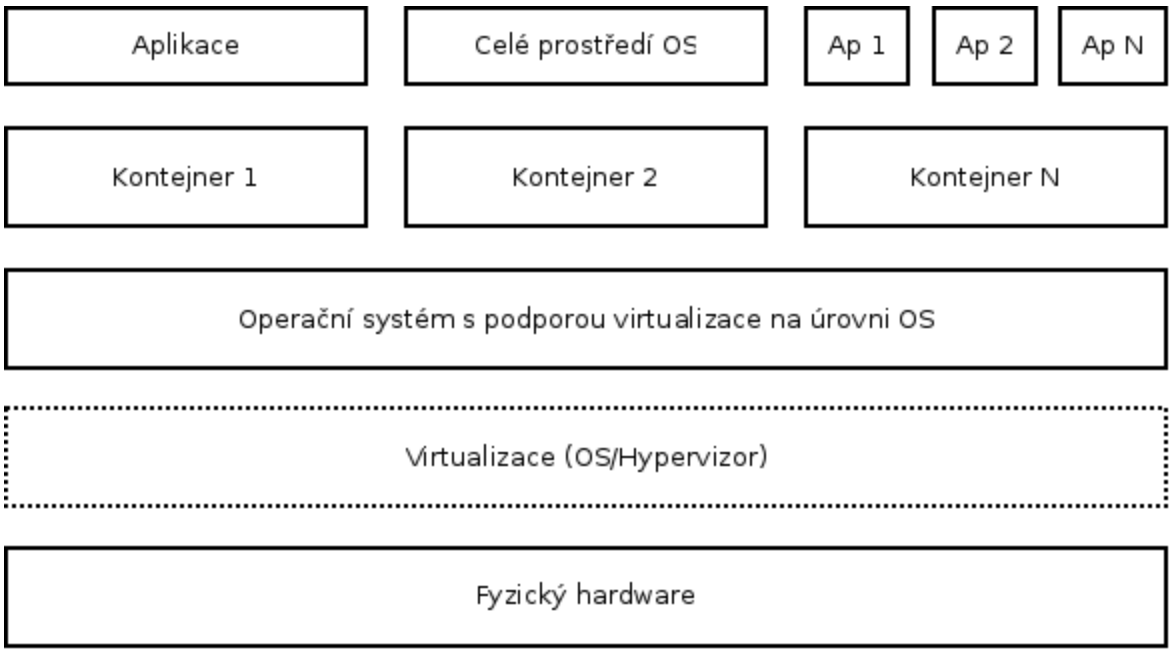
Účel této kapitoly je stejný jako v případě kapitoly 3.1.3. Pro rekapitulaci uvádím, že jsou zde shrnuty charakteristiky a dopady koncepce vzhledem k atributům definovaným ve druhé kapitole této práce. První sloupec tabulky 4 obsahuje atribut, ve druhém sloupci se nachází jeho charakteristika formou odkazu na výše popsanou výhodu či nevýhodu a v posledním sloupci je vyhodnocen dopad. Dopad je stanoven na základě subjektivního hodnocení autora a může nabývat hodnot *negativní, spíše negativní, neutrální, spíše pozitivní* a *pozitivní* a nebo jejich rozsahu.

Tabulka 4: Dopady koncepce virtualizace celých operačních systémů vzhledem k atributům IT infrastruktury

Atribut	**Charakteristika**	**Dopad**
Jednorázové výdaje na pořízení technologického vybavení	K2V2, K2N1	Neutrální až negativní
Provozní výdaje na technologické vybavení	K2V1, K2N3	Neutrální
Rychlost nasazení a testování a aplikací	K2V1, K2V7	Spíše pozitivní
Elasticita	K2V7	Spíše pozitivní až pozitivní
Efektivnost	K2V2	Spíše pozitivní až pozitivní
Dostupnost	K2V5, K2V6	Pozitivní
Zálohování	K2V4	Pozitivní
Obnovení po výpadku	K2V4, K2V5	Pozitivní
Spravovatelnost	K2V1	Spíše pozitivní
Stabilita a spolehlivost	K2V3, K2V6	Neutrální až spíše pozitivní
Výkon	K2N1, K2N2	Spíše negativní
Bezpečnost	K2V1	Neutrální až spíše pozitivní
Horizontální škálovatelnost	K2V7	Spíše pozitivní
Vertikální škálovatelnost	K2N4, K2N5	Negativní

3.3 Virtualizace na úrovni operačního systému

Virtualizace na úrovni operačního systému je někdy označována přesnějším označením kontejnerová virtualizace. Od virtualizace celých operačních systémů se liší tím, že neemuluje oddělené virtuální počítače, nýbrž vytváří uzavřené prostory (dále bude označováno jako kontejnery) v rámci jednoho jádra operačního systému. Z toho plyne podstatná nevýhoda, že není možné v rámci kontejnerů provozovat jiný operační systém či jiný software než ten, který podporuje hostitelské jádro. Na druhou stranu tento typ virtualizace přináší celou řadu výhod, mezi něž se řadí nižší výkonnostní dopad, větší flexibilita kontejnerů a absolutní nezávislost na platformě (virtualizace na úrovni operačního systému může být provozována i v rámci virtuálního počítače). Základní schéma virtualizace na úrovni operačního systému popisuje obrázek 3.



Obrázek 3: Schéma virtualizace na úrovni operačního systému

Tento typ virtualizace je implementován přímo v jádře operačního systému formou izolovaných jmenných prostorů, které v rámci nich umožňují oddělit procesy a virtualizovat systémové prostředky. V případě jádra operačního systému Linux se

jedná o celkem šest jmenných prostorů, mezi něž se řadí virtuální připojení souborových systémů, virtuální strom procesů, virtuální síťové adaptéry, oddělený subsystém pro komunikaci mezi procesy, virtuální jméno počítače a oddělený virtuální seznam uživatelů a skupin. Důležitou součástí jsou takzvané kontrolní skupiny (cgroups), které v rámci jednotlivých jmenných prostorů umožňují limitovat systémové prostředky jako jsou čas procesoru, operační paměť a nebo počet operací s pevným diskem.

Na vyšší úrovni dále existují dva základní principy, jak ke kontejnerové virtualizaci přistupovat. První z nich umožňuje v rámci kontejnerů spouštět celé operační systémy, které na první pohled vypadají jako plnohodnotné (tj. s vlastním jádrem). Druhý princip se orientuje na spouštění pouhých aplikací (a jejich minimalizovaného prostředí). Vzhledem k tomu, že první přístup se příliš neliší od virtualizace celých operačních systémů, bude tato práce veškeré výhody a nevýhody vztahovat k přístupu druhému. Tato koncepce může být dále označována jako „koncepce 3“.

3.3.1 Výhody a nevýhody virtualizace na úrovni operačního systému

V této kapitole jsou popsány výhody a nevýhody koncepce virtualizace na úrovni operačního systému. Ty byly čerpány ze syntézy prací autorů Vineeta Badola (2015), Joea Brockmeiera (2010), Stevena J. Vaughan-Nicholse (2015) a osobní zkušeností autora této práce. Bylo identifikováno celkem sedm výhod a tři nevýhody koncepce virtualizace na úrovni operačního systému. Tabulka 5 uvádí jejich základní výčet a přiřazuje jim jednoznačný identifikátor, který bude později dále použit. Identifikátor se skládá z označení koncepce (v tomto případě K3), a označení výhody či nevýhody (Vx či Nx).

Tabulka 5: Výčet výhod a nevýhod koncepce virtualizace na úrovni operačního systému

Výhody	**Nevýhody**
Snadná správa a nasazení u rozsáhlých instalací (K3V1)	Možnost použít pouze jeden operační systém, omezený výběr (K3N1)
Maximální využití hardwarových prostředků u málo vytížených aplikacích (K3V2)	Nutnost změnit přístup k IT infrastruktuře a vývoji aplikací (K3N2)
HA řešeno na úrovni kontejnerů (K3V3)	Nevhodné pro zastaralé aplikace (K3N3)
Snadné zálohování, přenos a obnova kontejnerů (K3V4)	
Maximálně rychlé zotavení se z chyby (K3V5)	
Velmi vysoká flexibilita IT infrastruktury (K3V6)	
Minimální režie a vysoká efektivita využití systémových prostředků (K3V7)	

Výhoda: Snadná správa a nasazení u rozsáhlých instalací (K3V1)

Vzhledem k tomu, že u této koncepce nedochází k manipulaci s celými operačními systémy, ale s odlehčenými kontejnery, odpadá spousta konfiguračních úloh spojených s operačními systémy. Technologická řešení jsou přímo navržena pro snadnou a rychlou správu a nasazení aplikací v masovém měřítku. Z tohoto hlediska je kontejnerová virtualizace ze všech tří virtualizací hodnocena jako nejlepší.

Výhoda: Maximální využití hardwarových prostředků u málo vytížených aplikacích (K3V2)

V rámci kontejnerové virtualizace můžeme nezávisle oddělit aplikace do vlastních uzavřených prostředí, které se výkonově ani bezpečnostně nijak neovlivňují. To umožňuje umístit na jeden server více aplikací, které vyžadují vlastní prostředí, avšak nespotřebovávají příliš mnoho systémových prostředků. Navíc vzhledem k tomu, že každý kontejner si s sebou nese pouze nutné minimální prostředí a sdílí jádro operačního systému, zůstává drtivá většina systémových prostředků k dispozici pro aplikace.

Výhoda: HA řešeno na úrovni kontejnerů (K3V3)

Podobně jako u virtualizace celých operačních systémů je i zde možné řešit vysokou dostupnost na úrovni samotných kontejnerů. V případě selhání kontejneru dojde k jeho automatickému restartování a to buď v rámci stejného fyzického serveru a nebo i jiného uzlu clusteru. Vzhledem k tomu, že v drtivé většině případů po selhání startuje pouze aplikace samotná, je obnovení činnosti velmi rychlé.

Výhoda: Snadné zálohování, přenos a obnova kontejnerů (K3V4)

Samotné kontejnery jsou typicky velmi malé a umožňují snadno oddělit samotnou aplikaci od jejich dat. Vzhledem k tomu, že většina technologických řešení obsahuje nástroje pro automatické sestavení obrazu kontejneru, není nutné tyto obrazy zálohovat a v případě jakéhokoliv selhání lze snadno sestavit kontejner nový. S tím souvisí i velmi snadná přenositelnost kontejnerů, není nutné přenášet celý obsah, ale stačí pouze jeho definice. Samotná data aplikací mohou být uložena velkým množstvím standardizovaných způsobů a vzhledem k tomu, že hostitelský operační systém na ně vidí přímo a může je propagovat i do více kontejnerů, otevírá se široká škála možností pro jejich zálohování a obnovu.

Výhoda: Maximálně rychlé zotavení se z chyby (K3V5)

Kontejnerová virtualizace představuje s ohledem na rychlost zotavení se z chyby to nejvhodnější řešení. V prvé řadě dochází ke startu pouze samotné aplikace, tudíž po

selhání hardwaru či hostitelského operačního systému nedochází k paralelnímu startu několika těžkých virtuálních počítačů. V případě chyby na úrovni samotného kontejneru je jeho restart velmi rychlý a pokud ten nepomůže, lze rovněž velmi rychle na základě nového obrazu sestavit kontejner nový.

Výhoda: Velmi vysoká flexibilita IT infrastruktury (K3V6)

Od virtualizace celých operačních systémů představuje kontejnerová virtualizace další abstrakční krok, jímž odděluje nejen operační systém od hardwaru, ale samotnou aplikaci od operačního systému. To umožňuje flexibilně rozdělovat komponenty jedné služby do mnoha kontejnerů a ty podle potřeby rozmísťovat v rámci fyzické infrastruktury. Kontejnery mohou na základě poptávky velmi rychle vznikat a zanikat a rovněž je možné v reálném čase měnit jejich limity pro využití systémových prostředků.

Výhoda: Minimální režie a vysoká efektivita využití systémových prostředků (K3V7)

Vzhledem k tomu, že kontejnerová virtualizace nevytváří další vrstvu nad jádrem hostitelského operačního systému, ale pouze uzavřené prostory v rámci něj, nedochází téměř k žádnému výkonnostnímu dopadu a režii samotné virtualizace. Efektivita využití systémových prostředků je tak téměř rovna třívrstvé koncepci a to s většinou výhod virtualizace celých operačních systémů. Samotné kontejnery jsou rovněž velmi malé, obsahují pouze nezbytné součásti prostředí aplikace. Rovněž vedle sebe neběží několik plnohodnotných operačních systémů, což významně šetří operační paměť.

Nevýhoda: Možnost použít pouze jeden operační systém, omezený výběr (K3N1)

První a největší nevýhodou je omezená možnost výběru operačních systémů a jejich vzájemné kombinace. Potřebné schopnosti mají pouze jádra operačních systémů Linux, BSD a dalších komerčních UNIXů. Tyto operační systémy déle mezi sebou není možné kombinovat, tudíž na jádře jednoho operačního systému nemůže běžet operační systém jiný. Toto omezení vychází přímo z návrhu kontejnerové virtualizace, jelikož hostované kontejnery se dělí o jedno společné jádro operačního systému.

Nevýhoda: Nutnost změnit přístup k IT infrastruktuře a vývoji aplikací (K3N2)

Aby se využily všechny možnosti této koncepce IT infrastruktury na maximum, je nutné změnit celý pohled na IT infrastrukturu a vývoj aplikací pro ni. To s sebou přináší nemalé nároky na školení personálu a transformaci všech přístupů pro nasazování a provoz aplikací. Zejména je nutné rozdělit aplikace do menších funkčních jednotek, začít chápat kontejner jako jednu službu a již při vývoji brát ohled na to, že aplikace bude v takovém prostředí nasazena.

Nevýhoda: Nevhodné pro zastaralé aplikace (K3N3)

Aplikace se zastaralým návrhem, které není možné rozdělit do menších služeb a pracovat s nimi víceméně bezstavově (například velké relační databázové stroje) se pro nasazení v tomto typu infrastruktury příliš nehodí, jelikož podobně jako u virtualizace celých operačních systémů nejsou schopny využít téměř žádnou výhodu dané koncepce.

3.3.2 Dostupná technologická řešení

V rámci této kapitoly jsou představena vybraná technologická řešení, mezi které patří Docker, LXC, OpenVZ a FreeBSD jail. Všechna čtyři řešení mají otevřený (svobodný) kód, které kromě dobrovolníků vyvíjí velké softwarové společnosti. Docker, LXC a OpenVZ pracují s jádrem operačního systému Linux, FreeBSD jail je technologie operačního systému FreeBSD.

Docker

Ze všech čtyř technologií je nejmladší a ke kontejnerům má odlišný přístup než zbylá technologická řešení. Docker vznikl jako interní projekt ve společnosti dotCloud jako její proprietární řešení, avšak v březnu roku 2013 byl uvolněn jako open source. Pro svůj běh vyžaduje jádro operačního systému Linux, nicméně v době psaní této práce existují i předprodukční verze podpory pro Windows. Využívat Docker je možné v rámci libovolné linuxové distribuce, nicméně existují speciální distribuce upravené

přímo pro Docker. Patří mezi ně CoreOS, Project Atomic a nebo komerční Redhat Enterprise Linux Atomic Host.

Jak již bylo zmíněno na začátku této podkapitoly, Docker se vyznačuje odlišným přístupem ke kontejnerům než většina řešení. To spočívá v tom, že v rámci kontejnerů neběží celé (byť upravené) prostředí operačních systémů, ale pouze jedna aplikace či její součásti. Primární spouštění více procesů v rámci kontejneru systému Docker není oficiálně podporováno. Samotný obraz, ze kterého se kontejner spouští je díky tomu velmi malý (typicky okolo 200 MB) a díky tomu, že se nemusí spouštět procesy operačního systému, je start kontejneru velmi rychlý. V praxi tak mohou existovat kontejnery, které obsahují třeba i velmi jednoduchý program, který se spustí v podstatě stejně rychle, jako by běžel nativně na operačním systému (třívrstvá koncepce). Obraz kontejneru bývá složen z více vrstev, které mohou být společné pro více obrazů a samotný kontejner si při svém spouštění vytváří další zapisovatelnou vrstvu. To velmi výrazně šetří místo na disku a zároveň zrychluje jakoukoliv manipulaci s kontejnery, protože například pro start nového kontejneru není potřeba kopírovat celý obraz, ale stačí vytvořit pouze novou zapisovatelnou vrstvu, do které se uloží všechny změny oproti obrazu.

Výše zmíněné specializované distribuce operačního systému Linux se od běžných odlišují tím, že mají ve výchozím stavu uzpůsobené celé prostředí pro provoz kontejnerů a nelze do nich doinstalovat další software (adresáře kam se instaluje jsou připojené pouze pro čtení). Jejich aktualizace probíhá rovněž odlišným způsobem, kdy dojde k vytvoření celého nového kořene souborového systému, ze kterého se po restartu operační systém načte. V případě chyby je možné se vrátit ke kořenu původnímu. Tím je dosaženo absolutní konzistence a velmi snadné spravovatelnosti hostitelských operačních systémů.

LXC

Linux Containers (zkracováno jako LXC) představují pojetí kontejnerové virtualizace, které je bližší koncepci virtualizace celých operačních systémů než Docker. Podobně jako virtualizace celých operačních systémů i LXC umožňuje provozovat (téměř) plnohodnotné operační systémy, které mají svůj vlastní souborový systém, síťové

karty a jiný virtuální hardware, nicméně nemají vlastní jádro. To znemožňuje provozovat jiné operační systémy než ty, které jsou založené na jádře operačního systému Linux.

V současné době se jedná o řešení, které je aktivně vyvíjené a je součástí v podstatě všech linuxových distribucí, avšak většina zájmu směřuje spíš k Dockeru a nebo OpenVZ, které přišlo dříve než LXC a spoustu součástí mu předalo.

OpenVZ

Počátky vývoje OpenVZ sahají až do roku 2000, kdy byl společností Parallels pod názvem Virtuozzo uvolněn jeden z prvních systémů pro virtualizaci na úrovni operačního systému. V roce 2005 došlo k jeho uvolnění jako open source pod názvem OpenVZ (Open Virtuozzo). Ve svých začátcích se jednalo o průlomovou technologii, jelikož neexistovaly hardwarové prostředky pro plnou virtualizaci celých operačních systémů a servery nebyly ani dostatečně výkonné, aby na nich bylo efektivní provozovat paravirtualizaci. Jednalo se tak o de facto jediné řešení, které umožňovalo provozovat více nezávislých operačních systémů v rámci jednoho fyzického počítače.

OpenVZ spoustu součástí předalo jak jádru operačního systému Linux tak i podobné, novější, technologii LXC. Z toho důvodu jsou možnosti LXC a OpenVZ velmi podobné, avšak vzhledem k delší historii OpenVZ pro něj existuje mnohem širší podpora. Za celým řešením stojí primárně jedna konkrétní společnost (Virtuozzo), tudíž na rozdíl od LXC lze na OpenVZ snadno získat komerční podporu.

FreeBSD jails

Poslední představovaný zástupce technologií pro virtualizaci na úrovni operačního systému bude ze světa operačních systémů z rodiny UNIX, konkrétně FreeBSD. Vznik FreeBSD jails se váže ke společnosti R&D Associates, která financovala implementaci potřebných technologií a před jejich veřejným vydáním v roce 1999 na ně měla roční exkluzivitu. Jednalo se o první implementaci myšlenky „nedokonalé virtualizace“, která teprve definovala pojem virtualizace na úrovni operačního systému.

Mezi hlavní cíle FreeBSD jails patří oddělené souborové systémy, oddělené procesy, oddělené uživatelské účty, kompletní oddělení kontejneru od okolí a snadná delegace globálních oprávnění do kontejnerů. Jedná se o technologii velmi robustní, nicméně v praxi nepříliš rozšířenou, ke které navíc nelze snadno získat komerční podporu. Možnosti oproti LXC nebo OpenVZ jsou menší, v rámci FreeBSD jails například nelze snadno provádět živou migraci nebo připojovat fyzická zařízení.

3.3.3 Dopady koncepce vzhledem k atributům

V této podkapitole jsou formou tabulky shrnuty charakteristiky a dopady koncepce virtualizace na úrovni operačního systému vzhledem k atributům definovaným ve druhé kapitole této práce. První sloupec tabulky 6 obsahuje atribut, ve druhém sloupci se nachází jeho charakteristika formou odkazu na výše popsanou výhodu či nevýhodu a v posledním sloupci je vyhodnocen dopad. Dopad je stanoven na základě subjektivního hodnocení autora a může nabývat hodnot *negativní, spíše negativní, neutrální, spíše pozitivní* a *pozitivní* a nebo jejich rozsahu.

Tabulka 6: Dopady koncepce virtualizace na úrovni operačního systému vzhledem k atributům IT infrastruktury

Atribut	**Charakteristika**	**Dopad**
Jednorázové výdaje na pořízení technického vybavení	K3V1, K3N2	Neutrální až spíše negativní
Provozní výdaje na technické vybavení	K3V2	Neutrální
Rychlost nasazení a testování a aplikací	K3V4	Pozitivní
Elasticita	K3V4, K3V6	Spíše pozitivní až pozitivní
Efektivnost	K3V7	Pozitivní
Dostupnost	K3V3, K3V5	Spíše pozitivní až pozitivní
Zálohování	K3V4	Pozitivní
Obnovení po výpadku	K3V5	Pozitivní
Spravovatelnost	K3V1	Spíše pozitivní až pozitivní
Stabilita a spolehlivost	K3V3, K3V5	Spíše pozitivní až pozitivní
Výkon	K3V2, K3V7	Spíše pozitivní až pozitivní
Bezpečnost	K3V2	Neutrální až pozitivní
Horizontální škálovatelnost	K3N3, K3N1	Spíše negativní až negativní
Vertikální škálovatelnost	K3V6, K3V4	Spíše pozitivní

4 Metoda výběru koncepce nasazení on-premise infrastruktury

Pro konstrukci metody připadají v úvahu dvě možné cesty, mezi které patří rozhodovací stromy a rozhodovací tabulky.

V případě využití rozhodovacích stromů kořen stromu reprezentuje výchozí stav a každý uzel (včetně kořene) stromu představuje rozhodování podle jedné vlastnosti (atribut nebo skupina atributů) IT infrastruktury. Pro konstrukci rozhodovací otázky jednoho uzlu mohou být využity atributy definované ve druhé kapitole této práce. Rozhodovací otázka může být konstruována jako uzavřená a to typicky se dvěma možnými odpověďmi *ano* a *ne.* Vznikají tak větve, které mohou končit dvěma možnými způsoby. Prvním možným koncem větve je chybový stav, kdy uživatel metody zvolí takovou kombinaci odpovědí, která vylučuje doporučení konkrétní koncepce IT infrastruktury. Druhým koncovým stavem je doporučení konkrétní koncepce IT infrastruktury včetně zdůvodnění.

Využití rozhodovacích stromů s sebou nicméně přináší několik nevýhod. Tou první a největší je, že při doporučení výběru koncepce IT infrastruktury rozhodovací strom není schopen dát váhu (důležitost) jednotlivým otázkám. Ve výsledku tak může na základě odpovědi na nepodstatné otázky dojít k doporučení nevhodné koncepce IT infrastruktury. Doprovodným jevem tohoto problému je rovněž i to, že na celkovém výsledku není patrné jak moc jsou od sebe jednotlivé koncepce ve vhodnosti daleko. Druhou nevýhodou je nevyvážitelnost vzniknuvšího stromu a velké množství chybových stavů (listů, které by nevedly k doporučení koncepce).

Z výše uvedených důvodů se tak jako vhodnější způsob konstrukce rozhodovací metody jeví rozhodovací tabulka. Umožňuje přiřadit jednotlivým atributům váhu, má stejný počet rozhodovacích kroků, žádné chybové stavy a na konci zřetelně seřadí všechny koncepce od nejvhodnější po tu nejméně vhodnou.

4.1 Předpoklady pro tvorbu rozhodovací tabulky

Než se práce dostane k popisu konstrukce samotné rozhodovací tabulky, je nutné vyjasnit a definovat pojmy, se kterými bude tabulka pracovat.

- *Atribut.* Atributem se myslí atribut IT infrastruktury definovaný v kapitole 2.3 této práce.

- *Váha atributu.* Váha přiřazuje atributu důležitost. Ta může mít tři stupně označované jako *nepodstatné, neutrální, důležité.* Tyto úrovně jsou v rozhodovací tabulce reprezentovány celým číslem od nuly po dvojku (viz. tabulka 7). Váha vyjadřuje jak moc je daný atribut pro konkrétní rozhodování důležitý a uživatel metody si ho stanovuje sám.

- *Dopad atributu.* Dopad pro každý atribut je definován pro každou koncepci ve třetí kapitole, kde může nabývat hodnot *negativní, spíše negativní, neutrální, spíše pozitivní a pozitivní.* Převod na numerické hodnoty uvádí tabulka 8. Jelikož je v rámci hodnot přípustný i jejich rozsah, k jeho převedení na číslo dojde pomocí zprůměrování číselných reprezentací hodnot všech členů rozsahu. Z tohoto důvodu jsou tyto hodnoty v rozhodovací tabulce reprezentovány desetinným číslem od mínus dva po plus dva (konkrétně -2, -1,5, -1, -0,5, 0, 0,5, 1, 1,5, 2).

- *Vážený dopad atributu.* Vážený dopad atributu vzniká kombinací váhy atributu s dopadem atributem pomocí součinu.

Dále následují dvě tabulky, které explicitně definují převod slovních pojmenování na jejich číselnou reprezentaci.

Tabulka 7: Mapování slovního názvu váhy atributu na číslo

Slovní reprezentace	**Číselná reprezentace**
Nepodstatné	0
Neutrální	1
Důležité	2

Tabulka 8: Mapování slovního názvu dopadu nad číslo

Slovní reprezentace	**Číselná reprezentace**
Negativní	-2
Spíš negativní	-1
Neutrální	0
Spíše pozitivní	1
Pozitivní	2

4.2 Teorie konstrukce metody

Cílem metody je určit nejvhodnější koncepci IT infrastruktury s ohledem na její silné a slabé stránky vážené vůči jejich důležitosti pro uživatele metody. Metoda pracuje se sadou čtrnácti atributů, které jsou definovány ve druhé kapitole této práce. Uživatel metody si pro každý atribut nezávisle určí jeho důležitost buď jako nepodstatnou, neutrální a nebo důležitou (tyto pojmy se převádí na číselnou reprezentaci, viz. předchozí podkapitola). Důležitost se dále kombinuje pomocí součinu s předem definovaným dopadem (ve třetí kapitole této práce) daného atributu dané koncepce.

Tato operace se provádí na všech kombinacích atributů s koncepcemi, čímž vznikne kompletní matice vážených dopadů u všech třech koncepcí.

V posledním kroku dojde k sečtení všech hodnot vážených dopadů u jednotlivých koncepcí, čímž vznikne součet vážených dopadů pro každou koncepci. Čím vyšší součet je, tím je koncepce vhodnější, respektive nejvyšší součet označuje nejvhodnější koncepci, druhý nejvyšší součet označuje druhou nejvhodnější koncepci a nejnižší součet označuje nejméně vhodnou koncepci. Jelikož jednotlivé dopady mohou nabývat hodnot od mínus dva po plus dva, může být výsledný součet jak kladný tak i záporný.

Metoda nerozlišuje mezi různými výchozími stavy IT infrastruktury, jelikož jejím cílem není zabývat se transformací mezi různými koncepcemi, ale určit koncepci nejvhodnější a to především z technického hlediska.

Ze shrnutí výše uvedeného jednoznačně vyplývá, že metoda dokáže na základě vážených dopadů pro každý atribut přesně určit pořadí vhodnosti jednotlivých koncepcí včetně jejich rozdílu. Díky tomu má uživatel metody přehled o tom, jak moc jsou pro něj jednotlivé koncepce IT infrastruktury vhodné a v případě těsných výsledků může provést zpětnou revizi určení důležitosti jednotlivých dopadů.

4.3 Výpočet číselných hodnot atributů jednotlivých koncepcí

Primárním obsahem této podkapitoly je tabulka, ve které je proveden výpočet číselných hodnot dopadů atributů u jednotlivých koncepcí. Pokud je dopad definovaný přesně (tj. bez užití rozsahu), převede se pouze slovní popis na číselný pomocí tabulky definované v kapitole 4.1. V případě, že se jedná o rozsah možných dopadů, dojde k převodu všech součástí rozsahu na číselnou reprezentaci a výpočtu jejího průměru. Každý řádek tabulky je rozdělený na dva podřádky, kde v prvním z nich je uvedena výsledná hodnota a ve druhém z nich její výpočet.

Tabulka 9: Výpočet číselných hodnot atributů jednotlivých koncepcí

Atribut	**Koncepce 1**	**Koncepce 2**	**Koncepce 3**
Jednorázové výdaje na pořízení technologického vybavení	**1**	**-1**	**-0,5**
	(0 + 1 + 2)/3	(0+(-1)+(-2)/3	(0+(-1))/2
Provozní výdaje na technologické vybavení	**1**	**0**	**0**
	(0 + 1 + 2)/3	0	0
Rychlost nasazení a testování aplikací	**-1,5**	**1**	**2**
	(-2+(-1))/2	1	2
Elasticita	**-2**	**1,5**	**1,5**
	-2	(1+2)/2	(1+2)/2
Efektivnost	**-0,5**	**1,5**	**2**
	(0+(-1))/2	(1+2)/2	2
Dostupnost	**-1**	**2**	**1,5**
	-1	2	(1+2)/2
Zálohování	**-1**	**2**	**2**
	-1	2	2
Obnovení po výpadku	**-1**	**2**	**2**
	-1	2	2
Spravovatelnost	**-1**	**1**	**1,5**
	-1	1	(1+2)/2

Atribut	**Koncepce 1**	**Koncepce 2**	**Koncepce 3**
Stabilita a spolehlivost	**-0,5**	**0,5**	**1,5**
	(0+(-1))/2	(0+1)/2	(1+2)/2
Výkon	**2**	**-1**	**1,5**
	2	-1	(1+2)/2
Bezpečnost	**-0,5**	**0,5**	**0,5**
	(0+(-1))/2	(0+1)/2	(0+1)/2
Horizontální škálovatelnost	**-1**	**1**	**1**
	-1	1	1
Vertikální škálovatelnost	**2**	**-2**	**-1,5**
	2	-2	(-1+(-2))/2

Pro kompletnost je přiložen příklad dovysvětlující, jakým způsobem byly hodnoty dopadů vypočteny. U prvního atributu (Jednorázové výdaje na pořízení technologického vybavení) koncepce 1 je uvedena hodnota 1 s výpočtem (0 + 1 + 2)/3. Tento atribut u této koncepce nabývá rozsahu hodnoty dopadu neutrální až pozitivní, tudíž došlo k rozvedení celého rozsahu na kompletní výčet jeho prvků (tj. neutrální, spíše pozitivní, pozitivní). Dále tyto hodnoty byly převedeny na číselnou reprezentaci (tj. 0, 1, 2) a v posledním kroku došlo k výpočtu aritmetického průměru těchto hodnot jejich sečtením a vydělením jejich počtem. Výsledkem bylo číslo 1.

U atributů, které mají dopad definovaný právě jedním slovem je situace podstatně jednodušší, dochází pouze k převodu slovního označení na numerické a v části tabulky určené pro výpočet je uvedeno pouze toto číslo.

4.4 Rozhodovací tabulka a způsob jejího použití

Předchozí části této kapitoly definovaly všechny potřebné pojmy a konstanty pro samotnou konstrukci rozhodovací tabulky (tabulka 10). Ta je tvořena záhlavím, čtrnácti řádky (pro každý atribut jeden) a posledním řádkem pro vyplnění součtu. První sloupec označuje atribut a do druhého sloupce uživatel metody vpisuje celé číslo od nuly po dvojku, které reprezentuje jakou má pro něj atribut váhu (viz. předchozí podkapitoly). Dále následují tři další sloupce, z nichž každý reprezentuje jednu koncepci IT infrastruktury. Tyto sloupce jsou vertikálně rozděleny na dva podsloupce, kde v prvním se nachází předdefinovaná hodnota dopadu (v tabulce zkráceno jako „D“) daného atributu a ve druhém je volné místo pro vepsání součinu váhy a dopadu (váženého dopadu, zkráceno jako „VD“).

Tabulka 10: Prázdná rozhodovací tabulka

Atribut	**Váha**	**Koncepce 1**		**Koncepce 2**		**Koncepce 3**	
		D	**VD**	**D**	**VD**	**D**	**VD**
Jednoráz. výd. na pořízení tech. vyb.		1		-1		-0,5	
Provozní výd. na tech. vybavení		1		0		0	
Rychlost nasaz. a test. a aplikací		-1,5		1		2	
Elasticita		-2		1,5		1,5	
Efektivnost		-0,5		1,5		2	
Dostupnost		-1		2		1,5	
Zálohování		-1		2		2	
Obnovení po výpadku		-1		2		2	
Spravovatelnost		-1		1		1,5	
Stabilita a spolehlivost		-0,5		0,5		1,5	
Výkon		2		-1		1,5	
Bezpečnost		-0,5		0,5		0,5	
Horizontální škálovatelnost		-1		1		1	
Vertikální škálovatelnost		2		-2		-1,5	
Součet vážených dopadů							

Postup vyhodnocení rozhodovací tabulky je následující: V prvním kroku je nutné vyplnit první volný sloupec určující váhu. Ve druhém kroku se na každém řádku provede součin váhy atributu s dopadem atributu a ten se zapisuje do volného pole. V posledním kroku se sečtou všechny vážené dopady pro každou koncepci a výsledek se zapíše do posledního řádku.

4.5 Příklad použití rozhodovací tabulky

Poslední část čtvrté kapitoly na příkladu ilustruje způsob použití rozhodovací tabulky (viz. tabulka 11). Fiktivní uživatel označil jako nepodstatné atributy *jednorázové výdaje na pořízení technického vybavení, elasticita, zálohování, stabilita a spolehlivost* a *horizontální škálovatelnost.* Jako neutrální označil *provozní výdaje na technické vybavení, efektivnost, dostupnost, obnovení po výpadku, výkon* a *vertikální škálovatelnost.* Důležitými označil *rychlost nasazení a testování aplikací, dostupnost, spravovatelnost* a *bezpečnost.* Tyto skutečnosti zapsal do patřičného sloupce a provedl jejich součin s dopady. Na závěr všechny vážené dopady sečetl. První koncepce získala -4,5 bodů, druhá 9,5 a třetí 15. V tomto ukázkovém případu by byla doporučena jako nejvhodnější virtualizace na úrovni operačního systému, druhá nejvhodnější virtualizace celých operačních systémů a jako nevhodná byla označena třívrstvá koncepce.

Tabulka 11: Příklad vyplnění rozhodovací tabulky

Atribut	**Váha**	**Koncepce 1**		**Koncepce 2**		**Koncepce 3**	
		D	**VD**	**D**	**VD**	**D**	**VD**
Jednoráz. výd. na pořízení tech. vyb.	*0*	1	*0*	-1	*0*	-0,5	*0*
Provozní výd. na tech. vybavení	*1*	1	*1*	0	*0*	0	*0*
Rychlost nasaz. a test. a aplikací	*2*	-1,5	*-3*	1	*2*	2	*4*
Elasticita	*0*	-2	*0*	1,5	*0*	1,5	*0*
Efektivnost	*1*	-0,5	*-0,5*	1,5	*1,5*	2	*2*
Dostupnost	*2*	-1	*-2*	2	*4*	1,5	*3*
Zálohování	*0*	-1	*0*	2	*0*	2	*0*
Obnovení po výpadku	*1*	-1	*-1*	2	*2*	2	*2*
Spravovatelnost	*2*	-1	*-2*	1	*2*	1,5	*3*
Stabilita a spolehlivost	*0*	-0,5	*0*	0,5	*0*	1,5	*0*
Výkon	*1*	2	*2*	-1	*-1*	1,5	*1,5*
Bezpečnost	*2*	-0,5	*-1*	0,5	*1*	0,5	*1*
Horizontální škálovatelnost	*0*	-1	*0*	1	*0*	1	*0*
Vertikální škálovatelnost	*1*	2	*2*	-2	*-2*	-1,5	*-1,5*
Součet vážených dopadů			***-4,5***		***9,5***		***15***

5 Ověření platnosti metody v praxi

Metoda výběru koncepce on-premise IT infrastruktury byla ověřena v reálném podniku, který používá středně velkou on-premise IT infrastrukturu a tudíž již nějakou koncepci nasazení IT infrastruktury má. Cílem ověření bylo zjistit, zda-li je tato koncepce pohledem navržené metody ta nejlepší a pokud tomu tak není, navrhnout koncepci lepší. Navrženou koncepci dále implementovat do testovacího provozu a na základě jeho vyhodnocení ověřit, zda-li skutečně oproti stávající koncepci přináší benefity a metoda tedy je funkční.

5.1 Představení společnosti

Společnost, v níž došlo k ověření metody, působí jako internetový obchod s oblečením a doplňky. Jedná se o českou společnost (vlastněnou českým kapitálem), která má své ústředí v Praze a eshop provozuje pro Českou republiku, Slovensko a Rumunsko. V Praze se nachází i její datové centrum, které je pro všechny země společné. Společnost si celé IT řešení vyvíjí ve vlastní režii a toto řešení je postaveno na open source technologiích.

Autor práce, v době jejího psaní, v této společnosti pracuje jako správce IT infrastruktury, tudíž má přehled o jejím rozsahu a použité koncepci. V provozu je okolo třiceti fyzických serverů rozličné konfigurace (od čtyř procesorových jader až po šestnáct, od šestnácti gigabajtů operační paměti po sedmdesát dva), z nichž každý je dimenzovaný pro svůj účel. Existuji tak servery aplikační, databázové, vyvažovací a nebo vyhledávací. Postupem času nicméně došlo k tomu, že kvůli uspokojení okamžitých potřeb získaly různé servery více rolí, což s sebou přináší spoustu negativní dopadů na poskytované služby. S ohledem na výše uvedené tak aktuálně používanou koncepci lze jednoznačně označit jako třívrstvou se všemi jejími výhodami a nevýhodami.

5.2 Aplikace metody

Ve spolupráci s vedoucím IT infrastruktury této společnosti došlo k vyhodnocení metody. V prvním kroku bylo nutné určit váhy jednotlivých atributů IT infrastruktury:

- *Jednorázové výdaje na pořízení technologického vybavení.* Vzhledem k tomu, že se jedná o společnost, která se silně rozvíjí a má pro tento rozvoj dostatek kapitálu, byl tento atribut označen jako *nepodstatný.*

- *Provozní výdaje na technologické vybavení.* Podobně jako u předchozího atributu i tento byl vyhodnocen jako *nepodstatný*.

- *Rychlost nasazení a testování aplikací.* Tento atribut je pro společnost velmi *důležitý*, jelikož probíhá její rozvoj, který souvisí s neustálým vývojem nových verzí aplikací, které pohání její podnikání.

- *Elasticita.* Postoj k tomu atributu je hodnocen jako *neutrální*, jelikož společnost nečelí (až na výjimky) velkým výkyvům poptávky po jejich službách. Vzhledem k růstu jejích obchodních aktivit pro zajištění elasticity stačí průběžně IT infrastrukturu rozšiřovat.

- *Efektivnost.* U tohoto atributu došlo rovněž k vyhodnocení jako *neutrální*, protože finanční situace společnosti neklade nejvyšší nároky na co dosažení co nejvyšší efektivnosti.

- *Dostupnost.* Je pro podnikání společnosti naprosto kritická, jelikož i sebemenší výpadek znamená pro společnost obrovské ztráty. Tyto ztráty jsou jednak finanční (neuskutečněné objednávky, nepracující sklad, a podobně), ale také na dobré pověsti společnosti. Z těchto důvodu byl tento atribut ohodnocen jako *důležitý*.

- *Zálohování.* Společnost uchovává dva základní typy dat. Prvním z nich jsou zdrojové kódy aplikací, které jsou nicméně spravované v externích nástrojích mimo IT infrastrukturu. Druhým typem jsou informace o nabízeném zboží,

objednávkách a zákaznících. Ztráta těchto dat by na podnikání společnosti měla velmi vážný dopad, tudíž i tento atribut byl ohodnocen jako *důležitý*.

- *Obnovení po výpadku.* Podobně jako u zálohování i zde společnost spoléhá na svoje data, nicméně jejich okamžitá dostupnost není ve všech případech pro běž byznysu nutná. Proto byl atribut ohodnocen jako *neutrální*.

- *Spravovatelnost.* V současném stupni rozvoje společnosti, kdy ještě není definována celá řada procesů, neexistuje velký tlak na spravovatelnost IT infrastruktury. S ohledem na budoucnost je nicméně vhodné tento atribut ohodnotit jako *neutrální.*

- *Stabilita a spolehlivost.* Ačkoliv je tento atribut pro společnost důležitý, přílišný tlak na stabilitu a spolehlivost by pro společnost mohl znamenat zaškrcení rozvoje kreativity, což je jeden z hlavních motorů jejího růstu. Atribut byl tedy vyhodnocen jako *neutrální.*

- *Výkon.* Tento atribut byl ohodnocen jako *nepodstatný*, protože pro společnost je výhodnější investovat do nákupu hardwarových prostředků než věnovat omezené úsilí do implementace výkonnějších řešení.

- *Bezpečnost*. Vyhodnocení tohoto atributu probíhalo podobně jako u stability a spolehlivosti, tudíž byl označen jako *neutrální.*

- *Horizontální škálovatelnost.* Postoj k horizontální škálovatelnosti byl označen jako *neutrální*. Aplikace sice škálují horizontálně, nicméně není potřeba zvláštní podpory ze strany IT infrastruktury.

- *Vertikální škálovatelnost*. Tento atribut byl vyhodnocen jako *nepodstatný*, jelikož společnost neprovozuje žádnou aplikaci, která potřebuje škálovat vertikálně.

Dalším krokem, který navázal na vyhodnocení důležitosti atributů bylo vyplnění rozhodovací tabulky:

Tabulka 12: Rozhodovací tabulka vyplněná rámci aplikace metody v reálném prostředí

Atribut	**Váha**	**Koncepce 1**		**Koncepce 2**		**Koncepce 3**	
		D	**VD**	**D**	**VD**	**D**	**VD**
Jednoráz. výd. na pořízení tech. vyb.	*0*	1	*0*	-1	*0*	-0,5	*0*
Provozní výd. na tech. vybavení	*0*	1	*0*	0	*0*	0	*0*
Rychlost nasaz. a test. a aplikací	*2*	-1,5	*-3*	1	*2*	2	*4*
Elasticita	*1*	-2	*-2*	1,5	*1,5*	1,5	*1,5*
Efektivnost	*1*	-0,5	*-0,5*	1,5	*1,5*	2	*2*
Dostupnost	*2*	-1	*-2*	2	*4*	1,5	*3*
Zálohování	*2*	-1	*-2*	2	*4*	2	*4*
Obnovení po výpadku	*1*	-1	*-1*	2	*2*	2	*2*
Spravovatelnost	*1*	-1	*-1*	1	*1*	1,5	*1,5*
Stabilita a spolehlivost	*1*	-0,5	*-0,5*	0,5	*0,5*	1,5	*1,5*
Výkon	*0*	2	*0*	-1	*0*	1,5	*0*
Bezpečnost	*1*	-0,5	*-0,5*	0,5	*0,5*	0,5	*0,5*
Horizontální škálovatelnost	*1*	-1	*-1*	1	*1*	1	*1*
Vertikální škálovatelnost	*0*	2	*0*	-2	*0*	-1,5	*0*
Součet vážených dopadů			***-13,5***		***18***		***21***

Výsledek metody tedy dopadl jednoznačně v neprospěch třívrstvé koncepce, která získala -13,5 bodů a byla vyhodnocena jako absolutně nevhodná. Ostatní dvě koncepce dopadly výrazně lépe, kdy jako nejvhodnější byla doporučena virtualizace na úrovni operačního systému a jako druhá nejvhodnější virtualizace celých operačních systémů. Tyto dvě koncepce se umístily velmi blízko, tudíž nelze žádnou z nich vybrat jako jednoznačného vítěze a do finálního rozhodnutí může promluvit individuální posouzení technických možností společnosti.

5.3 Implementace testovacího provozu nové koncepce a jeho vyhodnocení

Vyhodnocení metody v předchozí podkapitole doporučilo dvě vhodné koncepce. Výsledek byl konzultován s vedoucím celého IT a vedoucím IT infrastruktury. Na základě tohoto jednání došlo k jednoznačnému závěru, že v bodě vyhodnocení vhodnosti současné koncepce metoda zafungovala správně, jelikož současný stav nikdo nepovažoval za vyhovující. Zejména bylo vyzdviženo, že současná koncepce je nevhodná v klíčových atributech, jako je rychlost nasazení a testovaní aplikací a nebo dostupnost, což metoda správně zohlednila.

V dalším jednání probíhala diskuse o výběru nové koncepce a způsobu realizace testovacího provozu. Ačkoliv byla virtualizace na úrovni operačního systému doporučena jako nejvhodnější, existovaly pádné důvody, proč upřednostnit druhou doporučeno variantu. Hlavním z nich byl fakt, že přechod na tuto koncepci by vyžadoval rozsáhlé, časově náročné, změny na všech úrovních IT, na což bohužel nebyla podpora ze strany vedení společnosti, které po IT vyžaduje především plnění požadavků byznysu. Metoda jako taková nemá za cíl zohledňovat tyto okolnosti, avšak v budoucnu je možné na ni navázat metodu jinou, která tyto okolnosti zohlední. V případě virtualizace celých operačních systémů je přechod mnohem snazší, jelikož na fungování IT jako celku není potřeba příliš mnoho měnit a z fyzických serverů lze dělat virtuální jedna ku jedné.

Došlo nicméně ke shodě, že metoda dokázala správně určit vhodné koncepce a že virtualizace na úrovni operačního systému by z technického hlediska byla nejvhodnější. Virtualizace celých operačních systémů však byla upřednostněna z hledisek, které pravděpodobně žádná metoda nemůže postihnout a individuální zhodnocení výsledků je vždy na místě. Došlo tedy k rozhodnutí, že bude zřízena virtualizační platforma na technologii KVM, na níž budou postupně přesunuty všechny aplikační servery, které budou navíc rozděleny podle svých rolí.

Na celkovém počtu sedmi kusů fyzických serverů byl nainstalován operační systém CentOS 7 s podporou virtualizace KVM. V dalším kroku byly vytvořeny obrazy jednotlivých typů aplikačních serverů (frontend, backend, atd.), které byly postupně spuštěny v rámci všech sedmi nodů virtualizačního clusteru. Během dvou měsíců byly postupně všechny fyzické aplikační servery vypnuty a společnost přešla z testovacího provozu virtualizace na provoz ostrý. Zkušební provoz byl vyhodnocován jednak z hlediska reálného naplnění dopadů atributů IT infrastruktur definovaných v této práci, ale také z hlediska interních měřítek společnosti.

U atributů, které dle metody mohly mít nějaký negativní dopad (patří mezi ně například výkon) se nepotvrdilo, že by byl v praxi překážkou pro nasazení této koncepce infrastruktury (ačkoliv tyto obavy panovaly). Naopak se potvrdilo, že díky izolaci aplikačních serverů dle jednotlivých rolí (servery, které odpovídají přímo na požadavky zákazníků vs. servery, které dávkově zpracovávají data) došlo ke zlepšení odezev a rychlosti načítání stránek. Velmi pozitivně byla hodnocena lepší dostupnost a rychlost nasazení a testování aplikací.

Všichni zúčastnění se shodli, že metoda velmi kladně přispěla k modernizaci IT společnosti a že její výstupy byly velmi dobrým a přesným podkladem pro další rozhodování, jehož správnost se potvrdila v praxi.

6 Závěr

Cílem této práce bylo vytvořit metodu umožňující vybrat nejvhodnější koncepci on-premise IT infrastruktury. Na začátku práce byla vytvořena rešerše, ze které vyplynula opodstatněnost tvorby takové metody stejně jako fakt, že oblast metod či metodik pro výběr různých typů koncepcí IT infrastruktur je velmi široká a existuje spousta volného prostoru pro další výzkum. Ukázalo se také, že existuje spousta různých definic pojmů „IT infrastruktura“ nebo „on-premise“ a že vždy záleží na pohledu, ke kterému se tato definice váže. Jednou z výzev práce bylo určit základní tzv. ne funkční atributy IT infrastruktur, jelikož žádný autor se nevěnoval jejich rigoróznímu definování. To se nicméně podařilo na základě syntézy prací několika různých autorů i zkušeností autora diplomové práce.

Třetí kapitola se věnovala popisu jednotlivých koncepcí, jež byly celkem tři (třívrstvá koncepce, virtualizace celých operačních systémů a virtualizace na úrovni operačního systému) a u každé z nich byl uveden její základní popis a příklady konkrétních technologický produktů. Podstatnou součástí byl soupis výhod a nevýhod každé z koncepcí, který vznikl na základě syntézy prací různých autorů i osobních zkušeností autora diplomové práce. U každé koncepce se na základě těchto výhod a nevýhod podařilo definovat míru dopadu dané koncepce na atribut IT infrastruktury definované ve druhé kapitole práce.

Všechny poznatky z předešlých kapitol zužitkovala kapitola čtvrtá, která se zabývala definicí samotné metody. Nejdříve bylo třeba určit nejvhodnější způsob její konstrukce, kde v úvahu připadaly rozhodovací stromy nebo rozhodovací tabulky. Z několika důvodů byly vybrány rozhodovací tabulky, které dokáží určit pořadí, nemají chybové stavy a je patrný i rozptyl vhodnosti jednotlivých koncepcí. Tvorba samotné rozhodovací tabulky byla doplněna nejen o zevrubný popis její konstrukce či o výpočet předdefinovaných hodnot, ale i o vzorový příklad jejího vyplnění.

Předposlední kapitola práce byla věnována ověření metody v prostředí reálné společnosti. Prokázalo se, že metoda, tak jak byla navržena, funguje správně a je schopná identifikovat vhodnou koncepci pro dané potřeby. Objevilo se nicméně jedno slabé místo, které spočívá v tom, že metoda nezohledňuje rozličné překážky spojené s

přechodem z jedné koncepce do druhé a i když doporučí jako vítěznou jednu koncepci, může se druhá v pořadí ukázat jako snáze realizovatelnější a to s podobnými přínosy. Otvírá se tak nové pole pro další zájemce o dané téma, kteří mohou na tuto práci navázat a metodu doplnit.

Celkově se podařilo naplnit cíle práce a její výsledek je funkčním přínosem v oblasti metod pro určování vhodných koncepcí IT infrastruktur. Jak již bylo zmíněno v úvodu této kapitoly, existuje spousta volného prostoru pro další výzkum, který se může ubírat směrem off-premise IT infrastruktur, infrastruktur stojících na pomezí on-premise a off-premise či jejich spojení se serverless řešeními.

7 Terminologický slovník

Termín	Význam
IT infrastruktura	Servery, úložiště, síťové prvky či software sloužící ke společnému účelu
on-premise	V budovách či majetku organizace

Seznam literatury

ANGELES, Sara. 2014. *The Pros and Cons of Virtualization.* Bussiness News Daily. Dostupné z: http://www.businessnewsdaily.com/6014-pros-cons-virtualization.html [cit 2016-10-22]

ARRASJID, John Yani, Mark GABRYJELSKI, Chris MCCAIN. 2016. *It Architect: Foundation in the Art of Infrastructure Design: A Practical Guide for It Architects.* IT Architect Resource. ISBN 978-0-9966-4770-0.

BADOOLA, Vineet. 2015. *Container Virtualization: what makes it work so well?* Cloud Academy. Dostupné z: http://cloudacademy.com/blog/container-virtualization/ [cit 2016-10-28]

BONDI, André B. 2000. *Characteristics of scalability and their impact on performance.* ACM. ISBN 1-58113-195-X. Dostupné z: http://dl.acm.org/citation.cfm?doid=350391.350432

BROADBENT, Marianne, Peter WEILL a B.S. NEO. 1999. *Strategic context and patterns of IT in-frastructure capability.* The Journal of Strategic Information Systems, roč. 8, č. 2, s. 157-187. DOI: 10.1016/S0963-8687(99)00022-0. Dostupné z: http://linkinghub.elsevier.com/retrieve/pii/S0963868799000220

BROCKMEIER, Joe. 2010. *Containers vs. Hypervisors: Choosing the Best Virtualization Technology.* Linux.com. Dostupné z: https://www.linux.com/news/containers-vs-hypervisors-choosing-best-virtualization-technology [cit 2016-10-28]

CAMBRIDGE UNIVERSITY PRESS. 2016. *Cambridge Dictionary.* Dostupné z: http://dictionary.cambridge.org/dictionary/english/premises [cit 2016-10-10]

ERNST AND YOUNG. 2014. *Optimize network OPEX and CAPEX while enhancing the quality of service.* Ernst and Young. Dostupné z: http://www.ey.com/Publication/vwLUAssets/EY_-

_Optimize_network_OPEX_and_CAPEX_while_enhancing_the_quality_of_service/$FILE/EY-optimize-network-opex-and-capex.pdf [cit 2016-10-10]

HERBST, Nikolas Roman, Samuel KOUNEV, Ralf REUSSNER. 2012. *Elasticity in Cloud Computing: What It Is, and What It Is Not.* Institute for Program Structures and Data Organisation, Karlsruhe Institute of Technology. Dostupné z: https://sdqweb.ipd.kit.edu/publications/pdfs/HeKoRe2013-ICAC-Elasticity.pdf

GARTNER. 2016. *Gartner IT Glossary.* Gartner. Dostupné z: http://www.gartner.com/it-glossary/data-center/ [cit 2016-10-21]

FERREIRA DA SILVA Luis, Fernando BRITO E ABREU. 2010. *An IT Infrastructure Patterns Approach to Improve IT Service Management Quality.* IEEE. ISBN 978-1-4244-8539-0. Dostupné z: http://ieeexplore.ieee.org/document/5655264/

GROCHOV, Jerrold M., 2015. *IT Infrastructure Projects: A Framework for Analysis.* Educase. Dostupné z: http://er.educause.edu/articles/2015/1/it-infrastructure-projects-a-framework-for-analysis [cit 2016-10-15]

CHUNG, Sock H., Terry Anthony BYRD, Bruce R. LEWIS a F. Nelson FORD. 2005. *An empirical study of the relationships between IT infrastructure flexibility, mass customization, and business performance.* ACM. ISSN 0095-0033. Dostupné z: http://dl.acm.org/citation.cfm?id=1080394

LAAN, Sjaak. 2013. *IT Infrastructure Architecture - Infrastructure Building Blocks and Concepts, Second Edition.* Lulu Press. ISBN 978-1-291-25079-4

MAH, Paul. 2012. *Six Reasons Small Businesses Need Virtualization.* CIO. Dostupné z: http://www.cio.com/article/2400612/virtualization/six-reasons-small-businesses-need-virtualization.html [cit 2016-10-22]

MARKETS AND MARKETS. 2014. *Data Center Colocation Market by Service Types (Retail, Wholesale), Industry Verticals (Banking & Insurance, IT & Telecom, Healthcare, Government & Public, Energy), End Users (SMEs, Large Enterprises) & by Regions - Global Forecast to 2020.* Markets and Markets. Dostupné z:

http://www.marketsandmarkets.com/PressReleases/colocation.asp. [cit 2016-10-10]

MASA'DEH, Ra'ed Moh'd Taisir. 2013. *The Impact of Information Technology Infrastructure Flexibility on Firm Performance: An Empirical Study of Jordanian Public Shareholding Firms.* Jordan Journal of Business Administration, roč. 9, č. 1, s. 204-224. DOI: 10.12816/0002054. Dostupné z: https://www.academia.edu/9251042/The_Impact_of_Information_Technology_Infrastructure_Flexibility_on_Firm_Performance_An_Empirical_Study_of_Jordanian_Public_Shareholding_Firms

MATTESON, Scott. 2013. *10 things you shouldn't virtualize.*Techrepublic. Dostupné z: http://www.techrepublic.com/blog/10-things/10-things-you-shouldnt-virtualize/ [cit 2016-10-15]

MELL, Peter, Timothy GRANCE. 2011. *The NIST Definition of Cloud Computing.* National Institute of Standards and Technology. Dostupné z: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

POPEK, Gerald J., Robert P. GOLDBERG. 1974. *Formal Reguirements for Virtualizable Third Generation Architectures.* Communications of the ACM, ročník 17, číslo 7. Dostupné z: http://www.logos.ic.i.u-tokyo.ac.jp/~tau/lecture/operating_systems/gen/papers/p412-popek.pdf

POSEY, Brien. 2015. *To virtualize or not: Workloads that should stay physical.* Techtarget. Dostupné z: http://searchservervirtualization.techtarget.com/tip/To-virtualize-or-not-Workloads-that-should-stay-physical [cit 2016-10-15]

RUBENS, Paul. 2012. *Top 10 Reasons Not to Virtualize.* Serverwatch. Dostupné z: http://www.serverwatch.com/server-trends/top-10-reasons-not-to-virtualize.html [cit 2016-10-15]

SILBERSCHATZ, Avi, Peter GALVIN. 1994. *Operating system concepts.* Addison-Wesley. Dostupné z:

http://www.cs.du.edu/~cag/courses/CS/DU/comp3361/Lectures/tm01-05.pdf [cit 2016-10-15]

TECHTARGET. 2016. *Network software glossary.* Techtarget. Dostupné z: http://whatis.techtarget.com/definition/Reliability-Availability-and-Serviceability-RAS [cit 2016-10-28]

VAUGHAN-NICHOLS, Steven. J. 2015. *Containers vs. virtual machines: How to tell which is the right choice for your enterprise.* IT World. Dostupné z: http://www.itworld.com/article/2915530/virtualization/containers-vs-virtual-machines-how-to-tell-which-is-the-right-choice-for-your-enterprise.html [cit 2016-10-28]

VŠETEČKA, Roman. 2005. *Windows i Linux na PC najednou? Zeptejte se Intelu.* Technet.cz. Dostupné z: http://technet.idnes.cz/windows-i-linux-na-pc-najednou-zeptejte-se-intelu-f7i-/hardware.aspx?c=A051115_114351_hardware_vse [cit 2016-10-20]

WALLEN, Jack. 2013. *10 benefits of virtualization in the data center.* Techrepublic. Dostupné z: http://www.techrepublic.com/blog/10-things/10-benefits-of-virtualization-in-the-data-center/ [cit 2016-10-22]

WEBFINANCE. 2016. *Business Dictionary.* WebFinance. Dostupné z http://www.businessdictionary.com/definition/efficiency.html [cit 2016-10-28]

WEIL, Peter, Mani SUBRAMANI, Marianne Broadbent. 2002. IT Infrastructure for Strategic Agility. MIT Sloan. Dostupné z: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=317307