ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická

BAKALÁŘSKÁ PRÁCE

2015 **Vojtěch Roztočil**

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra telekomunikační techniky

Víceúčelový server pro školu

Multipurpose server for school

bakalářská práce

Studijní program: Komunikace, multimédia a elektronika
Studijní obor: Síťové a informační technologie

Vedoucí práce: Ing. Pavel Troller, CSc

Vojtěch Roztočil

Praha 2015

Čestné prohlášení

Prohlašuji, že jsem zadanou bakalářskou práci zpracoval sám s přispěním vedoucího práce a konzultanta a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé bakalářské práce nebo její části se souhlasem katedry.

Datum: ……

Podpis bakalanta

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra telekomunikační techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Roztočil Vojtěch**

Studijní program: Komunikace, multimédia a elektronika
Obor: Síťové a informační technologie

Název tématu: **Víceúčelový server pro školu**

Pokyny pro vypracování:

Navrhněte a zrealizujte víceúčelový server, použitelný v resortu základního a středního školství tak, aby pokryl co možná nejvíce potřeb školy. Zejména se soustřeďte na tyto funkce - spolehlivé datové úložiště (pracující pod protokolem SMB/CIFS), pokročilý router a firewall pro řešení konektivity školy, hostování webových stránek školy, e-mailový server, doménový server (DNS) a další služby dle aktuální potřeby školy (např. server pro podporu výuky, laboratorní servery atd.). Bezpečné oddělení jednotlivých funkcí řešte metodou virtualizace. Při řešení využijte v maximální míře svobodný software s otevřeným kódem.

Seznam odborné literatury:

[1] Kolektiv: Linux Dokumentační projekt, 4. aktualizované vydání. Computer Press, Praha 2012. ISBN: 978-80-251-1525-1.

Vedoucí: Ing. Pavel Troller, CSc.

Platnost zadání: do konce letního semestru 2015/2016

prof. Ing. Boris Šimák, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 12. 12. 2014

Anotace:

Cílem této práce je návrh a následná realizace serverové virtualizace pro oblast základního školství a poskytnutí všech výhod, které virtualizace nabízí. Virtuální servery pak budou poskytovat potřebné služby pro oblast školství. Zejména se jedná o datové úložiště, směrovač, doménový server, řadič Active Directory, web server a poštovní server. V celém řešení bude použit svobodný software s otevřeným kódem, čímž dojde k nemalé finanční úspoře.

Klíčová slova:

virtualizace, QEMU, KVM, DNS server, Bind, DHCP server, poštovní server, školní server, doménový řadič

Summary:

The aim of this bachelor thesis is a scheme and a consequent realization of a server virtualization in the primary school area and providing all the benefits that are offered by the virtualization. Virtual servers will then provide necessary services for the area of education, in particular: network attached storage, router, domain server, Active Directory controller, web server and mail server. In the entire realization will be used free software with an open code which will result in considerable financial savings.

Index Terms:

virtualizaction, QEMU, KVM, DNS server, Bind, DHCP server, e-mail server, school server, domain controller

Obsah

1. Úvod

1.1 Rozbor práce

Cílem této práce je instalace a konfigurace víceúčelového serveru pro oblast základního nebo středního školství. Server bude poskytovat jak běžně používané služby typu Active Directory, DHCP server, DNS server tak i nově poštovní server, spolehlivé datové úložiště či pokročilý směrovač s firewallem. Kritické služby budou od sebe bezpečně odděleny a pravidelně zálohovány díky virtualizaci. V maximální možné míře bude využit svobodný software s otevřeným kódem, čímž dojde k nemalé finanční úspoře.

2. Virtualizace a její užití

2.1 Co je to virtualizace

O virtualizaci se začíná ve světě techniky mluvit koncem 20. století jako o metodě logického rozdělení sálových počítačů. V dnešní době vizualizaci chápeme jako nástroj, který nám umožňuje na dané hardwarové platformě, pomocí softwaru zvaného hypervizor vytvářet simulované prostředí pro běh dalších operačního systému, které se anglicky označují ”guest“. Český překlad host je zde poněkud zavádějící, protože se plete s anglickým označením pro hostitelský systém, anglicky označovaný též “host“. Proto budu v této práci používat anglické termíny host a guest. Podle úplnosti simulovaného hardware rozlišujeme několik druhů virtualizací.

2.1.1 Emulace nebo simulace

Emulace umožňuje počítačovému programu nebo operačnímu systému běžet na jiném hardwaru, než pro který je primárně napsán. Emulace simuluje fyzický počítač a je také schopna přijímat data na vstupu a zajistit korektní výstup. Tento způsob virtualizace prostředí zvyšuje možnosti přenositelnosti mezi platformami, na druhou stranu má většinou výrazně negativní vliv, především na výkon.

Tohoto způsobu virtualizace se hojně využívá v případech, kdy konkrétní hardware, například procesor, není fyzicky dostupný. Mezi nejznámější představitele této platformy patří Qemu, PeatPC či Bochs.[3]

2.1.2 Plná virtualizace

Základem plné virtualizace je, vytvoření virtuálního guest systému, který disponuje všemi prostředky fyzického stroje, jako je například plná instrukční sada, která odpovídá použitému procesoru Intel nebo AMD. Virtuální stroj navíc vůbec nepozná, že běží ve virtualizovaném prostředí. Klíčové je sdílení prostředků. O to se stará hypervizor, který řídí a přiděluje čas virtuálním strojům, dále jim alokuje jejich vlastní část paměti, řídí přístup k fyzickým síťovým jednotkám a řadičům pevných disků. Dále musí zajistit logické priorování jednotlivých fyzických zdrojů.[3]

2.1.3 Částečná virtualizace

Tento způsob virtualizace simuluje nebo emuluje vícero instancí jednoho hardwaru. Například odděluje adresní prostor, umí oddělit procesy, nebo sdílet prostředky, avšak neumí od sebe plně oddělit jednotlivé guest systémy. Prakticky proto nelze v tomto případě mluvit o plnohodnotném virtuálním stroji.[3]

2.1.4 Paravirtualizace

Operační systém si je vědom, že je virtualizován, a jeho zdrojový kód je modifikován tak, aby byla odstraněna potřeba binárního překladu. Hlavní změnou je úprava kritických operací. Tato volání jsou nahrazena tzv. hypercally, které komunikují přímo s hypervisorem. Výhodou je zrychlení systému, nevýhodou je nutnost změny jádra operačního systému. Zatímco u systému s otevřeným kódem jako je Linux nebo OpenBSB to není problém, u operačních systémů Microsoft taková změna možná není a tento typ je pro ně nedostupný.[4]

2.2 Proč virtualizovat

2.2.1 Konsolidace

Mezi nejdůležitější argumenty pro použití virtualizace bezesporu patří konsolidace. Při klasickém použití, tedy jeden fyzický počítač na jeden operační

systém, se kupuje hardwaru tak, aby výkonnostně pokryl zátěžové špičky. K těmto špičkám dochází jen občas, zbytek času je hardware nevyužit.

Podle statistik je standardní vytížení hardwaru při tomto druhu použití kolem 20%. Při použití virtualizace můžeme několik instalací serverů provozovat virtuálně na jednom fyzickém hardwaru. Docílíme tak lepší využitelnosti, neboť špičky budou rozloženy v čase a výkon hardwaru bude použit pro několik virtuálních serverů současně. Navíc tak dojde ke značnému snížení nákladů, neboť není potřeba kupovat tolik fyzických serverů. Další výhodou je nižší spotřeba elektřiny a samozřejmě i z toho plynoucí menší nároky na chlazení. Dojde také ke snížení nároků na potřebné místo v serverovně. Při použití sdílené storage, například za použití technologií SAN nebo iSCSI, je dokonce možné servery za chodu přesouvat z jednoho fyzického serveru na jiný. Dochází pouze k přesunu malého množství nezbytných dat a obrazu paměti, neboť vlastní data serveru jsou umístěna na sdílené storage. Tímto způsobem můžeme s virtuálními servery dynamicky pracovat, přesouvat je například v případě potřeby na silnější hardware. [11]

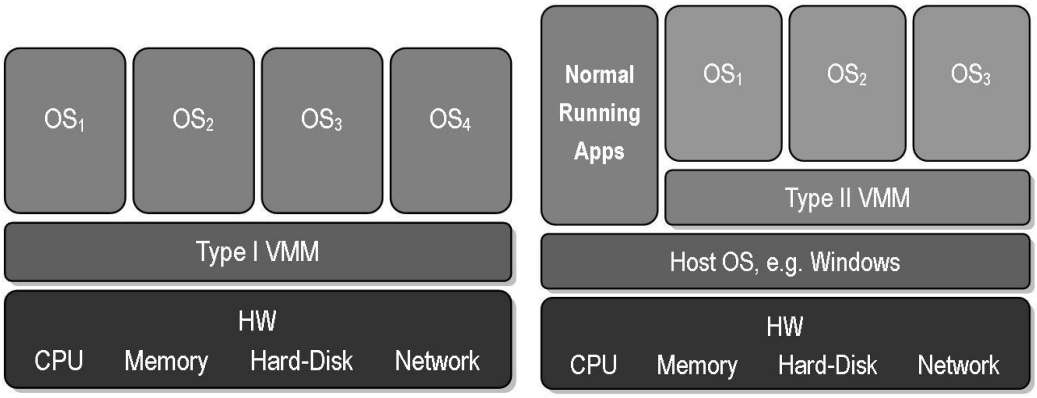
2.2.2 Oddělené prostředí

Stále více se také používají oddělená prostředí pro jednotlivé služby, hlavně z bezpečnostních důvodů. Služby velmi často sdílejí různé systémové prostředky či knihovny a navzájem se mohou ovlivňovat. Při použití virtualizace pak při pádů jedné služby nejsou ovlivněny služby jiné a dojde tedy jen k částečnému odmítnutí služeb. Případně lze použít rozdílné knihovny, které by si jinak mohly překážet. Při použití virtualizace opět není nutné služby provozovat na odděleném hardwaru, ale je možné je pustit současně na jednom fyzickém serveru v oddělených virtuálních strojích. Díky absolutní separaci virtuálních strojů pád jednoho virtuálního stroje neovlivní ty ostatní. Také při kompromitaci jedné služby útočník nemůže ovlivnit jiné služby, pokud jsou puštěny v jiných virtuálních strojích. Podobného principu lze také použít v případě, že máme odladěnou aplikaci, která nám běží na určité verzi operačního systému. Není tedy žádoucí měnit verzi operačního systému a provádět nové odladění aplikace. Může ale nastat případ, že z nějakého důvodu potřebujeme nasadit novější hardware, který ovšem není podporován starším operačním systémem. Díky virtualizaci

můžeme toto elegantně vyřešit, neboť do host systému nainstalujeme novější operační systém a skrze virtuální zařízení poskytneme zařízení starší verzi operačního systému, který pustíme ve virtuálním stroji. V některých případech je také nutné poskytnout klientům plný přístup k operačnímu systému, například skrze ssh. Pokud na jednom serveru provozujeme službu pro více klientů, je toto téměř nemožné vyřešit, neboť administrátorská práva přidělená jednomu klientovi by mohla ohrozit bezpečnost dat pro ostatní klienty. Pokud použijeme virtualizaci, můžeme pro každého klienta provozovat vlastní virtuální stroj, a tudíž mu můžeme ponechat plná práva k systému. Za prvé neuvidí data jiných klientů a za druhé si každý může provádět nastavení podle vlastní potřeby, aniž by to ovlivnilo ostatní. [11]

2.3 Možnosti virtualizace

Jak již bylo řečeno v úvodu, software nebo firmware, který vytváří a spravuje běh virtuálních strojů, nazýváme hypervizor nebo se též můžeme setkat s anglickým termínem virtual machine manager (VMM). Ten může běžet přímo na fyzickém serveru, což je původní koncept CP/CMS vyvinutý v IBM v roce 1960. Schéma tohoto modelu je na obrázku 2.3.1 vlevo. V dnešní době je asi nejznámější hypervizor tohoto typu ESXi od společnosti VMware či XenServer od společnosti Citrix. Druhou možností je běh hypervizora na běžném operačním systému, který nazýváme host. Takovéto hypervizory jsou například VirtualBox, Hyper-V a nebo QEMU-KVM.[1]



Obrázek 2.3.1: Možnosti virtualizace: (vlevo) nativní, (vpravo) host OS

3. Přehled řešení s otevřeným kódem

V této kapitole se budu věnovat nejvýznamnějším a nejpoužívanějším hypervizorům s otevřeným kódem.

3.1 Oracle VM VirtualBox

VirtualBox je multiplatformní software, který zastává funkci hypervizora na host systému, nabízí plnou virtualizaci a je k dispozici s jistým omezením zdarma ve dvou variantách.

První variantou je VirtualBox Open Source Edition (OSE) se zdrojovými kódy, skripty pro kompilaci a s licencí GNU General Public License. Tato verze ale má jistá omezení, není zde totiž podpora USB, USB over RDP, či iSCSI.

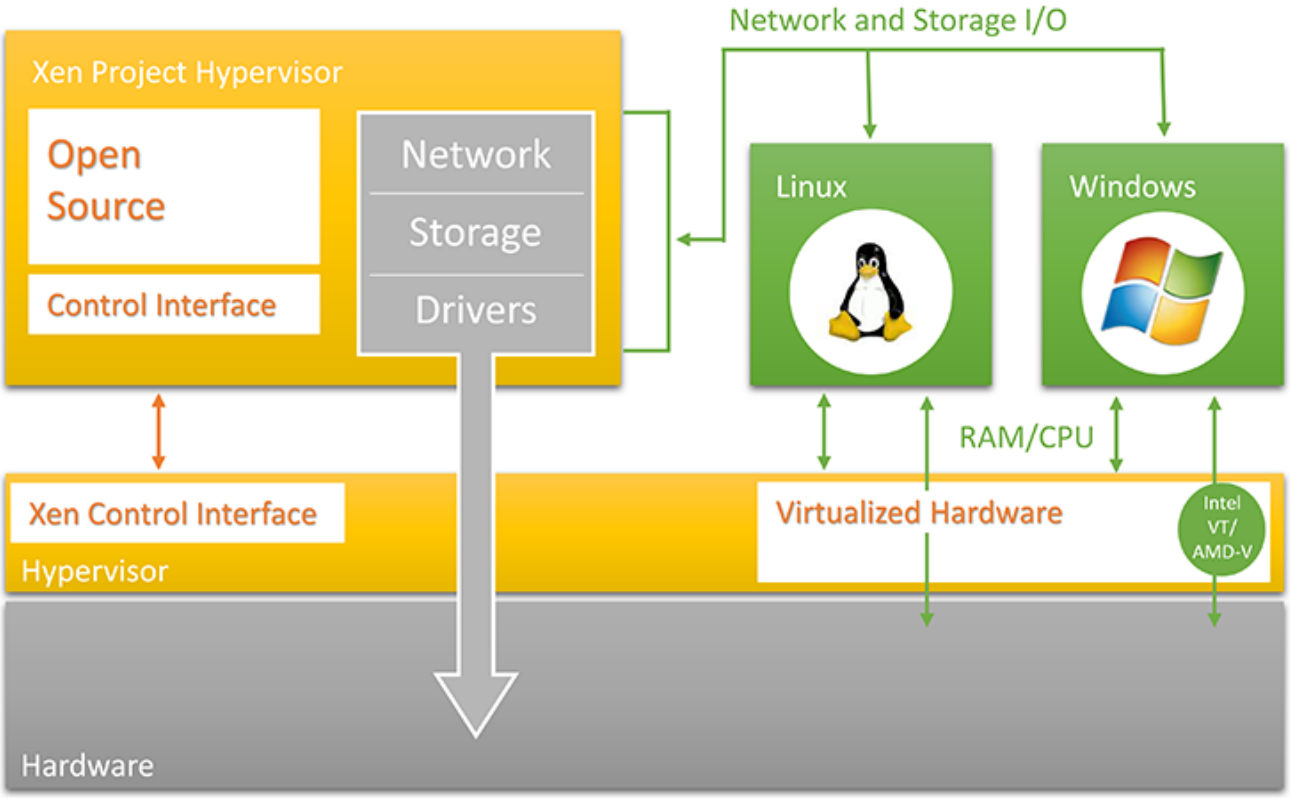
Druhou možností je VirtualBox Personal Use and Evaluation License (PUEL) s předkompilovaným binárním kódem. Avšak tato možnost je výhradně pro osobní použití nebo pro zkušební účely.

Pro rozsáhlé instalace nabízí Oracle manažera virtuálních strojů s otevřeným kódem s názvem Hyperbox.

Největší výhodou hypervizora VirtualBox bude bezesporu jednoduchost a uživatelsky přívětivé prostředí, díky čemuž je hojně využívám právě pro osobní potřebu běžných uživatelů. Nevýhodou jsou jednak restrikce verze s GNU licencí, značně omezené možnosti konfigurace virtuálních počítačů a nepříliš dobrá pověst o stabilitě a výkonu VirtualBoxu.[2]

3.2 Xen

Jedná se původně o projekt svobodného softwaru s otevřeným kódem z univerzity v Cambridge, který následně koupila společnost Citrix. Poté došlo několikrát k přejmenování, hlavně z marketingových důvodů a v současné době je tento hypervizor nabízen pod jménem XenServer, který je k dispozici zdarma s otevřeným kódem. Zpoplatněná je až verze pro rozsáhlé instalace, která obsahuje rozsáhlejším management virtuálních strojů. Z obrázku 3.3.1, kterým společnost Citrix prezentuje svůj produkt je vidět, že se jedná o nativního hypervizora s paravirtualizací. [6]



Obrázek 3.2.1: Uspořádání XenServeru.

3.3 ESXi Open Source

Dalo by se říct, že VMware ESXi je jedním z předních hráčů ve virtualizaci s poměrně dlouhou tradicí. První verze ESX Server byla představena v roce 2002 a jednalo se o proprietární řešení nativního hypervizora pro rozsáhlé instalace. Až v roce 2013 v rámci konkurenčních bojů s ostatními výrobci hypervizorů se VMware rozhodlo uvolnit ESXi jako opensource. Varianta s otevřeným kódem je nabízena zdarma ale bez technické podpory a některých pokročilejších nástrojů managementu. Pro reálné nasazení se doporučuje placená verze.[5]

3.4 QEMU-KVM

V první řadě je potřeba uvést, že se jedná o spojení dvou samostatných věcí. QEMU (Quick EMUlator) je jak již název napovídá emulátor ale zároveň i virtualizátor. Když spustíme Qemu jako emulátor, můžeme emulovat nejrůzný hardware (například ARM architekturu) na vlastním počítači pomocí dynamického překladu, který poskytuje dobrý výkon. Druhou možností je použít Qemu jako virtualizátor. K tomu je potřeba spustit Qemu buď v prostředí

hypervizora Xen, anebo mít nainstalovaný modul KVM (Kernel Virtual Machine) v jádře Linuxu, čili virtualizační stroj založený na linuxovém kernelu. KVM vyžaduje od procesoru hardwarovou podporu virtrualizace, protože KVM nepoužívá paravirtualizaci ale hardwarovou virtualizaci. Spojením hardwarové virtualizace z modulu KVM a emulátoru Qemu vzniká silný hypervizor s nepřeberným množstvím nastavení. Qemu i KVM je svobodný software otevřeným kódem volně šiřitelný pod licencí GNU GPL. [7][8][9]

4. Realizace

4.1 Analýza problému

V oblasti základní nebo střední školy je třeba investovat do nového serveru z důvodu konce životnosti stávajícího serveru. Stávající server slouží jako řadič domény, řadič active directory, DNS, DHCP server a jsou na něm uložena data většiny uživatelů. Od nového serveru se očekává, že umožní poskytovat jak všechny stávající služby, tak i další využitelné služby pro školu. Jedná se zejména o spolehlivé datové úložiště, případně s možností síťového zavádění operačního systému do bezdiskových klientů, pracující pod protokolem SMB/CIFS, pokročilý směrovač a firewall pro řešení konektivity školy, hostování webových stránek školy a e-mailový server.

4.1.1 Navržení optimálního řešení

Vyšší spolehlivost nového serveru ze strany hardware by mělo zajišťovat kvalitní rackové řešení obsahující dva zdroje, RAID řadič a možnost přidat další procesor a paměť v případě potřeby. Ze strany softwaru bude lepší dostupnost a bezpečnost zajišťovat virtualizace jednotlivých služeb. V co možná nejvyšší míře bude použit svobodný software s otevřeným kódem, čímž dojde i k nemalé finanční úspoře.

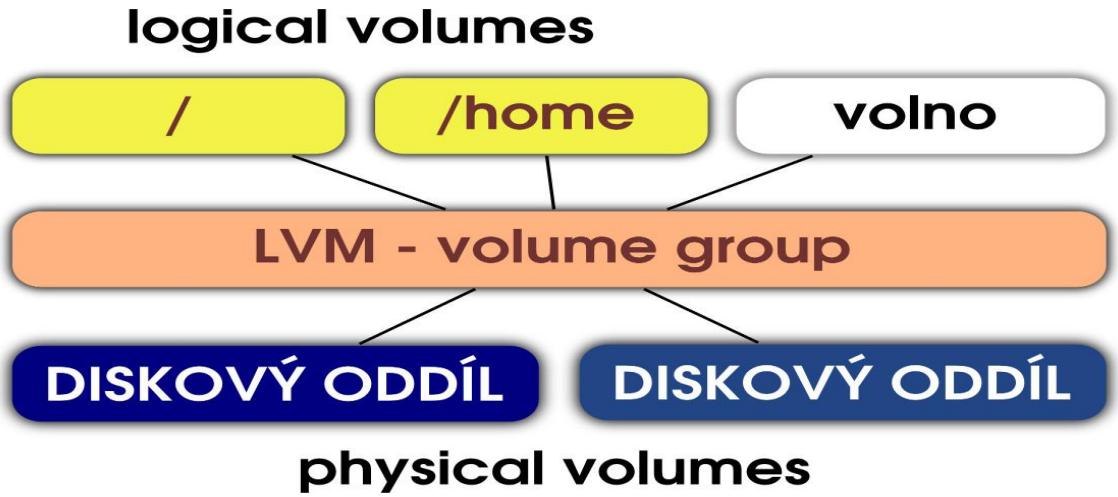
4.2 Hypervizor

Výběr hypervizora je vždy v jisté míře subjektivní záležitost a to zejména, pokud plánujeme využít svobodné řešení s otevřeným kódem a přijdeme o často důležité kritérium, kterým je cena. Pokud se podíváme na výsledky některého testu, srovnávající výkonové stránky hypervizorů, zjistíme, že se většina výsledků příliš neliší a z dílčích výsledků není patrno, který hypervizor je nejlepší. Dalším poměrně důležitým aspektem při výběru je "míra rozšířenosti" a podpora ze strany ostatních uživatelů v podobě nejrůznějších internetových návodů a diskusí, protože na rozdíl od komerčních produktů, svobodné projekty s otevřeným kódem nemívají zřízenou oficiální technickou podporu. V mém projektu jsem se nakonec rozhodl použít kombinaci Qemu a KVM, která poměrně slušně obstává ve výkonových testech, má velice dobrou podporu z řad ostatních uživatelů a jedná se o ryze svobodný software s otevřeným kódem na rozdíl od ostatních hypervizorů, které jsou jen částečně opensource.

Samotná instalace probíhá v několika etapách. První etapou je instalace operačního systému s modulem KVM v pozici host OS. V dnešní době je již jaderný modul KVM součástí distribučních jader, takže je na výběr ze spousty distribucí Linuxu. V mém případě, kdy je instalace zamýšlena jako serverové řešení na skutečném hardwaru, je vhodné zvolit takzvaný "enterprise" systém, který kromě předpřipravených balíčků pro správu je většinou uváděn výrobcem serveru jako podporovaný operační systém. V mém případě jsem vybral CentOS Linux, který sice výrobce oficiálně nepodporuje, ale jedná se o volně dostupnou linuxovou distribuci založenou na enerprise systému Red Hat Linux, který je výrobci serverů podporován.

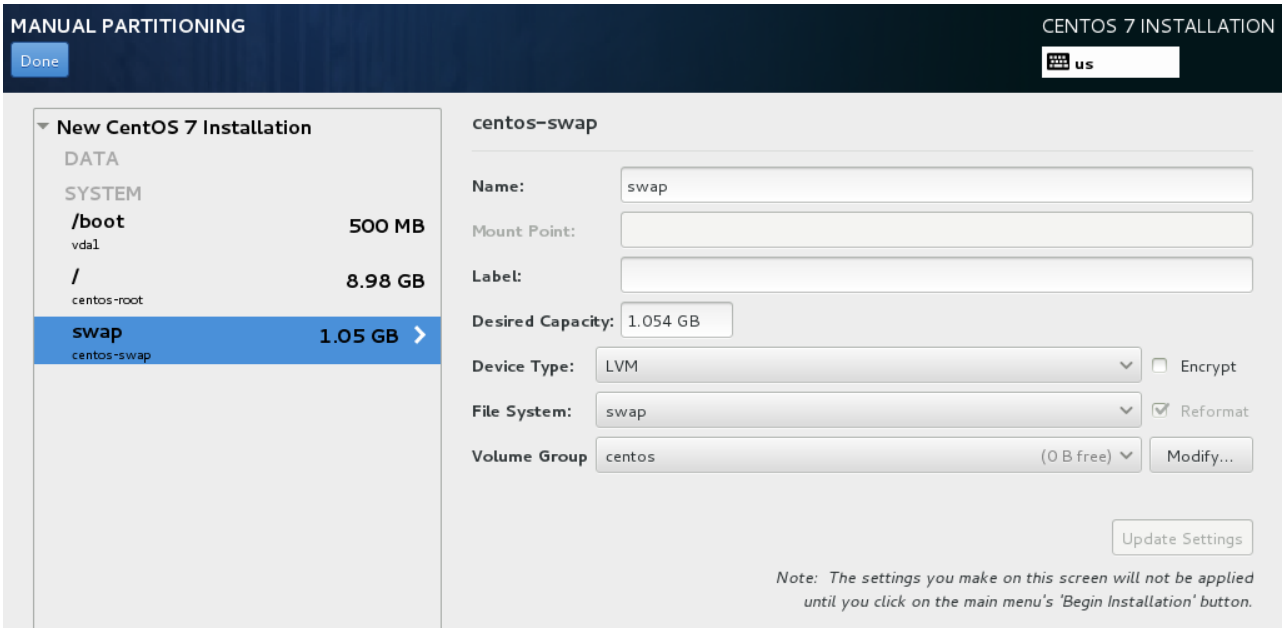
4.2.1 Instalace a konfigurace

Ať už budeme instalovat CentOS z disku DVD nebo z USB klíčenky, důležitým bodem instalace je rozdělení disků. Hardwarovým architektem serverů ze společnosti HP mi bylo doporučeno jako optimální řešení pro potřebu základní školy s ohledem na cenu a výkon požít trojici rychlých disků s kapacitou kolem 450 GB v RAID 5 pro běh operačních systémů a dva pomalejší disky s kapacitou kolem 1 TB v RAID 1 na souborový server. Protože předpokládám instalaci na server s hardwarovým RAID řadičem, nebudu se v této práci zabývat konfigurací softwarového RAID pole. Co však v sekci rozdělení disků rozhodně stojí za nastavení je LVM. Logical Volume Manager je systém pro správu logických oddílů, který přináší spoustu výhod oproti klasické práci s diskovými oddíly. LVM vytváří logické oddíly, které můžeme za běhu měnit, rušit, zvětšovat, vytvářet obrazy a jsou do jisté míry nezávislé na hardwaru. [10]



Obrázek 4.2.1.1: Schéma ilustrující princip funkce LVM

Proto v sekci nastavení disků zvolím manuální nastavení diskových oddílů. První oddíl vytvořím pro použití bootování s kapacitou 500 MB. Tento oddíl nesmí být spravován LVM. Je vhodné ještě vytvořit zvláštní oddíl pro swapování, který již může být typu LVM ale je potřeba použít speciální systém souborů swap, který je speciálně přizpůsoben pro rychlé odkládání a čtení dat vyrovnávací paměti. Zbylé místo na disku přiřadím do kořene systému označovaného / a zde rozhodně použiji LVM, který mi umožní později velikost oddílu měnit a dle aktuálních požadavků rozdělovat.



Obrázek 4.2.1.2: Manuální nastavení diskových oddílů pro instalaci.

K dokončení instalace již stačí pouze zadat heslo správce a případně vytvořit dalšího uživatele. Po prvním přihlášení je potřeba zapnout připojení k síti, abychom si mohli stáhnout a nainstalovat další potřebné programy. To se provede přepsáním položky ONBOOT=no na ONBOOT=yes v souboru /etc/sysconfig/network-scripts/ifcfg-eth0. Nyní nám již nic nebrání provést update systému pomocí příkazu yum update a následně začít s instalací samotného hypervizora. Jak již bylo řečeno, jaderný modul KVM je v dnešní době standardně zahrnut v jádře linuxových distribucí, čímž se nám práce zjednodušuje. QEMU můžeme stáhnout ze stránek projektu a zkompilovat, anebo využít pohodlnější cesty s využitím takzvaného balíčkovacího systému, kdy instalaci provedeme příkazem:

```
yum install qemu-kvm.
```

Pro kontrolu je dobré překontrolovat, zdali je modul kvm načtený následujícím příkazem:

```
lsmod | grep kvm.
```

Pokud se nám zobrazí kvm_intel, případně kvm_amd, je server připraven virtualizovat další guest systémy, avšak zejména pro méně zkušené uživatele je vhodné využít takzvaného správce virtuálních počítačů, případně správce virtuálních počítačů s grafickým rozhraním. Dobré zkušenosti jsem získal se správcem virtuálních počítačů libvirt, který má možnost grafického rozhraní a dokáže používat větší množství hypervizorů. Navíc nabízí další pokročilé nástroje pro práci s virtuálními stroji. Instalaci lze opět provést pomocí balíčkovacího systému příkazem

```
yum install libvirt
```

Tímto je systém připraven vytvářet nové virtuální stroje, avšak konkrétní práci s virtuálními servery se budu věnovat až o několik kapitol později.

4.2.2 Nastavení sítě

Virtuální stroje mají stejně jako fyzické počítače řadu vstupních i výstupních rozhranní, který nám zajišťují komunikaci. U stolních počítačů to typicky bývá monitor, klávesnice a myš, avšak u serverů je situace poněkud jiná. Monitor, klávesnici ani myš často u serveru vůbec nenajdeme, veškerá komunikaci totiž probíhá přes síťové rozhraní.

V případě, že jsme nainstalovali správce virtuálních počítačů Libvirt, při instalaci se vytvořilo automaticky nové síťové rozhraní s názvem vibr0. Toto rozhraní založené na NAT je však určeno spíše pro snadné a rychlé vyzkoušení virtualizace. Pro reálný provoz virtuálních serverů je zapotřebí vytvořit nové síťové rozhraní, takzvaný síťový most, který bude sdílený mezi více virtuálních serverů najednou. Pokud pracujeme v systému CentOS, budeme muset nejdříve vypnout výchozího správce síťových připojení, který by nám blokoval vytvoření síťového mostu. K tomu lze použít následující příkazy:

```
chkconfig NetworkManager off
chkconfig network on
service NetworkManager stop
service network start
```

Dále upravíme v adresáři /etc/sysconfig/network-scripts/ soubor ifcg-eth0, který se stará o síťovou kartu tak, aby byl součástí síťového mostu. Toho můžeme docílit například pomocí následujících příkazů:

```
# cat > ifcfg-eth0 <<EOF
DEVICE=eth0
HWADDR=00:11:22:33:44:55
ONBOOT=yes
BRIDGE=br0
NM_CONTROLLED=no
EOF
```

Příkaz HWADDR je vhodné doplnit vlastní MAC adresou.

Druhý soubor bude vytvářet nové virtuální zařízení typu síťový most. Ve stejném adresáři vytvoříme soubor ifcg-br0 a použijeme tyto parametry:

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=dhcp
ONBOOT=yes
DELAY=0
NM_CONTROLLED=no
```

Následně můžeme restartovat síťová připojení a vypnout netfilter pro síťový most.

```
service network restart
cat >> /etc/sysctl.conf <<EOF
 net.bridge.bridge-nf-call-ip6tables = 0
 net.bridge.bridge-nf-call-iptables = 0
 net.bridge.bridge-nf-call-arptables = 0
 EOF
 # sysctl -p /etc/sysctl.conf
```

Nyní by měl být síťový most aktivní a připraven k použití pro virtuální servery.[12]

4.2.3 DHCP server

Zkratka DHCP pochází z anglického Dynamic Host Configuration Protocol, což se běžně překládá jako protokol pro konfiguraci síťových klientů. DHCP je již poměrně dlouhou dobu standardizován, ale původně byl publikován jako doporučení RFC 2131 v roce 1997 a jak již název napovídá, DHCP server určitým způsobem může konfigurovat TCP/IP parametry jednotlivých klientů. Kromě IP adresy může DHCP server přidělovat například i masku sítě, implicitní bránu, DNS servery, jméno hostitele a domény, může hostitelům nastavit chování podle určitých norem, změnit jeho MAC adresu, informovat ho o SMTP, POP, NTP a dalších serverech, zapnout, nebo vypnout IP forwarding, bezdiskovým stanicím určí cestu k bootovacímu obrazu, apod.

V mém případě, kdy mám nainstalován hypervizor na běžné verzi CentOS Linux, můžu si dovolit nainstalovat DHCP server paralelně s hypervizorem, tedy přímo na host systém. Zdaleka nejrozšířenější software pro tuto službu je DHCP server

od ISC (Internet Systems Consortium), který je poskytován svobodně s otevřeným kódem. Pro instalaci pomocí balíčkovacího systému bude příkaz v našem případě vypadat takto:

```
yum install dhcp
```

Instalací se nám vytvoří konfigurační soubory, manuálové stránky, init skripty a mimo jiné také program dhcpd, kerý běží na serveru jako démon a poskytuje vlastní DHCP službu.

Po instalaci je třeba nejprve dhcp server správně nakonfigurovat a teprve poté spustit dhcp démona. Umístění konfiguračních souborů závisí na distribuci operačního systému, v mém případě je tím /etc/dhcp/, kde se nachází již předpřipravené dva soubory. Prvním souborem je dhcpd.conf, který slouží k nastavení parametrů dhcp serveru pro přidělování IPv4 adres a druhý soubor dhcpd6.conf, který přiděluje IPv6 adresy. V mém případě budu konfigurovat pouze IPv4 dhcp server a to již zmíněným souborem dhcpd.conf, jehož obsah uvádím níže.

```
option domain-name "tyrsova.cz";
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
default-lease-time 3600;
max-lease-time 7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
        range 192.168.1.100 192.168.1.200;
}

host guru {
    hardware ethernet 08:00:4b:2c:22:23;
    fixed-address 192.168.1.100;
}
```

Následně soubor uložíme a spustíme dhcp démona pomocí příkazu

```
systemctl start dhcpd
```

a pomocí příkazu

```
systemctl enable dhcpd
```

nastavíme démona, aby se spouštěl automaticky při startu serveru. V tento okamžik je dhcp server funkční a je připraven přijímat žádosti od klientů. [13][14][15]

4.2.4 DNS server

Jmenný server je jedním ze základních služeb nejen pro přístup k internetu ale i pro místní síť. Stará se o překlad názvů počítačů na síťové adresy a opačně. Mezi nejrozšířenější DNS servery patří Bind, a proto se budu i v této práci zabývat právě jím. Bind verze 9 je vyvíjen stejně jako DHCP server v minulé kapitole společností Internet Systems Consortium a jedná se o univerzální dns server, který může být nastaven jak do role rekurzivního, tak i do role autoritativního dns serveru.

Pokud bychom se rozhodli mít dns server přístupný z internetu například jako autoritativní server naší domény, pak by bylo dobré z bezpečnostních důvodů vytvořit nový virtuální stroj vyhrazený pro tuto službu. Tím by se dalo alespoň částečně zabránit vyřazení celého fyzického serveru například při DoS nebo DDoS útocích. Dns server bychom mohli pravidelně zálohovat a bezpečně oddělit od ostatních služeb běžících na stejném fyzickém stroji.

V mém případě zcela nevidím důvod ke správě vlastní domény ze strany internetu, a proto můj dns server bude přístupný pouze pro lokální síť, čímž podstatně zmenším bezpečnostní rizika a Bind nainstaluji na hos systém.

Bind 9 se v CentOS Linux instaluje pomocí balíčkovacího systému příkazem:

```
yum -y install bind bind-utils
```

Hlavní konfigurační soubor /etc/named.conf nastavuje umístění ostatních konfiguračních souborů, porty pro komunikaci, dnssec, pro které zóny bude rekurzorem a obsahuje jména a odkazy na konfigurační soubory jednotlivých zón. V mém případě vypadá konfigurační soubor následovně:

```
options {
directory"/var/named";
dump-file"/var/named/data/cache_dump.db";
statistics-file"/var/named/data/named_stats.txt";
memstatistics-file"/var/named/data/named_mem_stats.txt";
allow-query{ localhost; 192.168.1.0/24; };
allow-transfer { localhost; 192.168.1.0/24; };
recursion yes;
dnssec-enable yes;
dnssec-validation yes;
dnssec-lookaside auto;
bindkeys-file "/etc/named.iscdlv.key";
managed-keys-directory "/var/named/dynamic";
pid-file "/run/named/named.pid";
session-keyfile "/run/named/session.key";
};
logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};
view "internal" {
        match-clients {
                localhost;
                192.168.1.0/24;
        };
        zone "." IN {
                type hint;
                file "named.ca";
        };
        zone "tyrsova.cz" IN {
                type master;
```

```
                file "local";
                allow-update { none; };
        };
        zone "1.168.192.in-addr.arpa" IN {
                type master;
                file "local.rev";
                allow-update { none; };
        };
};
```

Soubory definující jednotlivé zóny jsem vytvořil ve výchozím adresáři, kterým je /var /named/ . Celkem se tedy jedná o tři soubory. První se jmenuje named.ca a určuje kořenové dns servery, kterých se můj dns server zeptá v případě, že sám nezná odpověď. Soubor local definuje zónu tyrsova.cz, tedy tu zónu, ve které se dns server sám nachází a pro kterou vrací klientům autoritativní odpovědi. Poslední soubor local.rev slouží ke zpětnému překladu IP adres v lokální síti na síťové jméno.

Pokud máme všechny soubory správně nakonfigurovány, můžeme Bind spustit.

```
systemctl start named
systemctl enable named
```

Těmito příkazy nejprve spustíme démona named a poté zajistíme, aby se démon spouštěl automaticky při startu systému. [16]

4.2.5 Směrovač a firewall

Existují dvě možnosti, jak připojit fyzický server k místní síti a internetu. První možností typickou pro většinu menších sítí, je využít samostatný směrovač se zapnutým překladem adres a server umístit až za směrovač do místní sítě s privátními IP adresy. Výhodou je relativní bezpečí v místní síti, avšak velikou daní je omezená komunikace s internetem způsobená zmíněným překladem adres.

Druhá možnost, je umístit server tak, aby sám plnil funkci směrovače a tedy byl zároveň připojen do Internetu tak i do místní sítě. Toto řešení odstraňuje nevýhodu omezené komunikace s internetem a zároveň pro lokální síť zachováme

bezpečí pomocí překladu adres. Ve skutečnosti existuje ještě třetí možnost. Pokud máte k dispozici dostatečné množství veřejných IP adres, můžete každému počítači přiřadit jednu veřejnou adresu a není potřeba řešit problém s překladem adres. Avšak vhledem k faktu, že volné IPv4 adresy již nejsou k dispozici, pravděpodobně nedostanete od poskytovatele připojení více než jednu veřejnou IP adresu.

V mém případě má škola k dispozici dvě nezávislá připojení k internetu, kdy u primárního je k dispozici jedna veřejná IP adresa a u záložního dokonce veřejná IP adresa není poskytována. Pro řešení celkové konektivity využiji serveru, jako pokročilého směrovače, který bude sám vybírat připojení k internetu dle dostupnosti a zároveň se bude starat o překlad IP adres pro lokální síť.

Prvním krokem je nastavení IP adres, přidělených od obou poskytovatelů připojení k internetu na síťovém rozhraní WAN. Primární připojení nastavíme kompletně včetně výchozí brány, u záložního připojení vyplníme pouze IP adresu a masku sítě. Tím docílíme, že při normálním stavu budou všechna data do internetu odcházet přes primární připojení. Pro případ výpadku jsem vytvořil skript, který kontroluje dostupnost připojení k internetu a v případě poruchy změní IP adresu výchozí bránu na záložní a zároveň mi pošle email s upozorněním na výpadek primárního připojení.

```
#!/bin/sh
# skript pro hlidani konektivity


while true
do sleep 30
check=$( ping -c 1 -W 10 8.8.8.8 | grep icmp* | grep
bytes | wc -l )
if [ $check -eq 0 ];
then
        route add default gw xxx.xxx.xxx.xxx eth1
         echo Výpadek internetu v ZS Tyrsova | mail -s
    "test" "vojta.roztocil@tyrsova.cz"
        sleep 7200
```

```
        route del default gw xxx.xxx.xxx.xxx eth1
else
fi
done
exit 0
```

Po dvou hodinách se skript přepne zpět na primární výchozí bránu, a pokud je konektivita opravena, zůstane tak. V opačném případě se celý proces periodicky opakuje až do opravení primárního připojení.

Další důležitou funkcí směrovače je NAT a firewall. K obojímu využiji rozsáhlý firewall Netfilter, který je obsáhlý v jádře systému a ovládá se pomocí nástroje Iptables. Pomocí následujících příkazů nainstalujeme nástroj Iptables, vypneme výchozí firewall systému, povolíme a následně zapneme Iptables.

```
yum install iptables-services iptables-utils
systemctl stop firewalld.service
systemctl disable firewalld.service
systemctl enable iptables.service
systemctl start iptables.service
```

Veškerá pravidla a nastavení iptables můžeme kontrolovat a měnit v souboru /etc/sysconfig/iptables. Druhou možností je psaní pravidel přímo do konzole a následné uložení do souboru. Ať už zvolíme kterýkoliv způsob, výsledek by měl být stejný. Na začátku je potřeba nastavit výchozí politika firewallu. Tedy jinými slovy co se má stát s pakety, pokud nebude v sekci pravidel výjimka.

Výchozí politika může být různá v závislosti na použití firewallu a například jeho umístění v síti. V mé práci jsem zvolil přísnější politiku a odmítám všechna spojení, která přichází na vstup směrovače nebo chtějí projít. Povoleny jsou jen pakety směrem z počítače.

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
```

Dále je sekce, která nastavuje překlad adres. První příkaz se úplně netýká Iptables ale povoluje předávání paketů skrz jádro systému, které je ve výchozím stavu zakázané. Další příkaz zapíná samotný překlad adres z vnitřní sítě LAN (eth0) do WAN (eth1). Následně se povoluje průchozí spojení, které je ve výchozí politice zakázáno a poslední pravidlo povolí přeposílání z rozhraní WAN na rozhraní LAN u existujících, a nebo souvisejících spojení.

```
echo "1" > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
iptables -A FORWARD -i eth1 -o eth0 -m state --state
ESTABLISHED,RELATED -j ACCEPT
```

Poslední sekcí jsou samotná pravidla firewallu. Povolují spojení, která byla iniciována z místní sítě, vstup pro místní smyčku a dále výčet portů, na kterých se pakety nebudou zahazovat. Pro vzdálený přístup pomocí SSH je dobré mít zapnutý port 22, port 80 pro web server, porty 25, 110, 143 pro poštovní server a odpovídat na icmp echo-request žádosti. To se může hodit při diagnostice problému v síti příkazem ping.

```
#FW pravidla
iptables -A INPUT -m state --state RELATED,ESTABLISHED -
j ACCEPT
iptables -A INPUT -i lo -j ACCEPT #propust na vstup vše
pokud je vstupní rozhraní loopback
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 25 -j ACCEPT
iptables -A INPUT -p udp --dport 110 -j
iptables -A INPUT -p tcp --dport 143 -j
iptables -A INPUT -p ICMP --icmp-type echo-request -j
ACCEPT
```

Tímto by měla být jak lokální síť, tak samotný server chráněn před případnými útočníky. Vnitřní síť i server jsou z internetu nedostupný, pokud uživatel sám neinicializuje spojení z vnitřní sítě. Vyjmuty jsou důležité porty poskytující služby, u kterých je žádoucí, aby byly dostupné z internetu. [22]

4.3 Web server

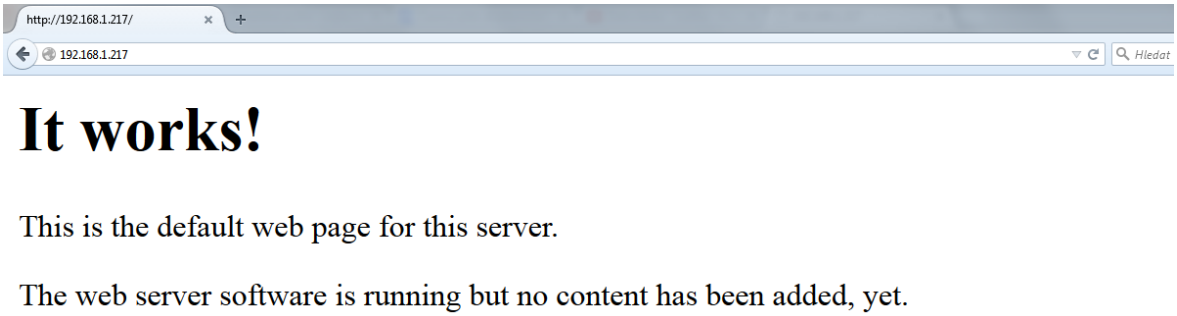
První virtuální server, který jsem nainstaloval, byl web server, který bude sloužit pro prezentaci webových stránek školy. V současné době je velké množství systémů pro http server, avšak od roku 1996 je nejrozšířenějším http serverem Apache, jehož vývoj začal v roce 1993 v jedné univerzitě původně pod jménem NCSA HTTPd. Avšak po několika letech odešel hlavní programátor, a došlo k zastavení projektu. Protože ale tou dobou už spousta správců používala NCSA HTTPd, vznikla komunita, která vytvářela opravné patche a z toho vznikl název Apache (A patche).[17] V dnešní době můžeme Apache nainstalovat na téměř všechny operační systémy, avšak nejpoužívanější a nejobvyklejší je pro tyto účely Linux. Já mám dobré zkušenosti s Debian Linuxem, proto jsem nainstaloval Apache HTTP server na Debian Linux.

Co se týče konkrétních parametrů virtuálního stroje, je potřeba vyjít z předpokládané návštěvnosti webových stránek. Maximální počet návštěvníků, kteří budou na webu základní školy současně, nebude asi nikdy v řádu tisíců ani stovek. Proto bude i s rezervou stačit přidělit pro virtuální stroj 2GB RAM.

Po spuštění virtuálního stroje můžeme přejít k instalaci všech součástí, které standardně web servery využívají. Zkratkou pro tyto programy se stává LAMP, tedy Linux, Apache HTTP server, MySQL a PHP. Instalace Debian Linuxu je velice podobná jako instalace CentOS, kterou jsem popsal v předešlých kapitolách. Avšak rozhodně doporučuji nainstalovat pouze Standard system utilities, protože instalace grafického rozhraní by zbytečně spotřebovávala systémové prostředky a navíc se nenainstalují programy, které nechceme. Instalace zbylých programů je nejjednodušší s využitím balíčkovacího systémů následujícími příkazů:

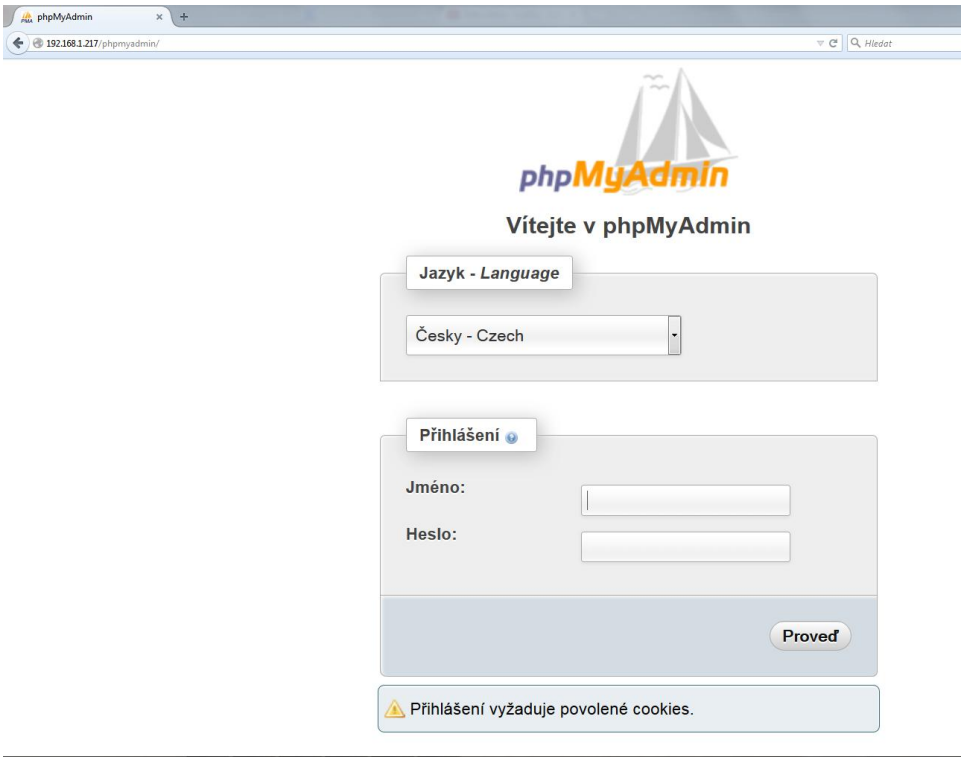
```
apt-get install apache2
apt-get install mysql-server mysql-client
apt-get install php5 php5-mysql libapache2-mod-php5
apt-get install phpmyadmin
```

Kde prvním příkazem se provede instalace Apache HTTP serveru, včetně vytvoření uživatele www-data. Další příkaz je pro instalaci mysql. Během této instalace je velice vhodné nastavit heslo pro mysql server. Ve třetím řádku instalujeme php včetně propojení s apache servrem a mysql. Poslední, spíše už volitelný řádek je instalace projektu phpmyadmin, který umožňuje spravovat mysql databáze pomocí webového rozhraní. Pro kontrolu, že se podařilo nainstalovat apache2 můžeme zadat do internetového prohlížeče z kteréhokoliv počítače ze stejné sítě IP adresu právě nainstalovaného web serveru.



Obrázek 4.3.1: Výchozí stránka http serveru Apache

Případně ještě můžeme obdobně udělat kontrolu pomocí phpmyadmin, kde za ip adresu napíšeme /phpmyadmin



Obrázek 4.3.2: Přihlašovací stránka phpMyAdmin

Tímto je hotová základní instalace web serveru, se kterou může většina webových stránek bez problémů fungovat. V záloze samozřejmě zůstává sousta pokročilé konfigurace a modulů, která by vydala na samostatnou publikaci.[18]

4.4 Samba v roli doménového řadiče Active Directory

S termínem Active Directory se poprvé setkáváme u operačního systému Microsoft Windows Server 2000, který vznikl na základě stále se zvyšující poptávky po komplexním řešení správy počítačových sítí. Active Directory funguje jako ucelené řešení pro správu počítačů s operačním systémem Microsoft Windows a je postaven z následujících základních služeb. První z nich je distribuovaná adresářová službu, která využívá protokol LDAP a je v podstatě jádrem Active Directory. Další službou je Kerberos, síťový autentizační protokol, který umožňuje komukoli komunikujícímu v nezabezpečené síti prokázat bezpečně svoji identitu. Poslední ze základních služeb Active Directory je DNS, se kterým je silně provázáno, a dokonce jsou na něm některé části založeny. [20]

Toto řešení se stalo velmi oblíbené a je dnes téměř standardem například v oblasti základních nebo středních škol. Výhodou je komplexní řešení od jednoho výrobce software a uživatelsky známe prostředí, avšak velkou nevýhodou je finanční náročnost licencování a občasné problémy s komptabilitou různých verzí Windows ze strany serveru a klientské stanice. Toto tvrzení vychází z mé zkušenosti, kdy občas selže přihlášení klientské stanice s operačním systémem novějším než Windows XP k řadiči domény AD se systémem Windows Server 2003. Nutno dodat, že Microsoft udává všechny verze Windows jako kompatibilní, avšak zároveň nabízí na svých stránkách opravné balíčky pro případ problémů.

Poměrně novou alternativou k Active Directory se stal svobodný software s otevřeným kódem s názvem Samba a to od verze 4.0, který dokáže být jak klient domény Active Direcory, tak i doménový řadič. Projekt samba vznikl s cílem svobodné implementace SMB protokolu pomocí reverzního inženýrství a dnes je důležitou součástí k propojení Linux/Unix systémů s Active Directory.

Má bakalářská práce klade za cíl mimo jiné používat svobodný software pro oblast školství, ve které je téměř zvykem používat Active Directory a klientské stanice s operačním systémem Microsoft Windows. Proto pro moje řešení použiji software Samba v roli doménového řadiče a pokusím se vytvořit stejné prostředí pro klientské počítače, které poskytuje Active Directory od společnosti Microsoft.

Prvním krokem je vytvoření nového virtuálního počítače, který bude sloužit jako doménový řadič. V pokynech projektu Samba nedoporučují používat doménový řadič jako souborový systém, proto bude stačit pro virtuální disk přidělit kolem 30GB volného místa a souborový server bude oddělený. Množství operační paměti můžu měnit dle potřeby během chodu serveru a bude se pravděpodobně odvíjet od množství připojených klientů. Operační systém jsem zvolil stejně jako na ostatních virtuálních strojích Debian Linux. Po instalaci je třeba rozšířit atributy souborového systému ext3/4, protože Samba potřebuje souborový systém, který podporuje jak “uživatele“ tak “systém“ (xattr). Dále je potřeba doinstalovat knihovny a programy, které Samba potřebuje. Toho lze docílit pomocí balíčkovacího systému příkazem:

```
apt-get install build-essential libacl1-dev libattr1-dev
libblkid-dev libgnutls-dev libreadline-dev python-dev
libpam0g-dev python-dnspython gdb pkg-config libpopt-dev
libldap2-dev dnsutils libbsd-dev attr krb5-user docbook-
xsl libcups2-dev acl
```

Pokud se podařilo vše nainstalovat bez problémů, můžeme přejít k instalaci Samby. Jsou celkem tři možnosti. První je využít opět balíčkovací systém, avšak tato možnost není doporučována, protože balíčkovací systém není úplně aktuální. Druhou možností je využít balíček SerNet EnterpriseSAMBA, což je předkompilovaný balíček pro snadnou instalaci, avšak je dostupný pouze po registraci. Poslední varianta je stažení svobodných zdrojových kódů a následná kompilace a instalace. Touto možnost získáme nejaktuálnější verzi Samby, a proto ji doporučuji. Příkazem

```
wget https://download.samba.org/pub/samba/samba-
4.2.1.tar.gz
```

uložíme nejnovější stabilní verzi do počítače, následně ji rozbalíme, zkompilujeme a nainstalujeme.

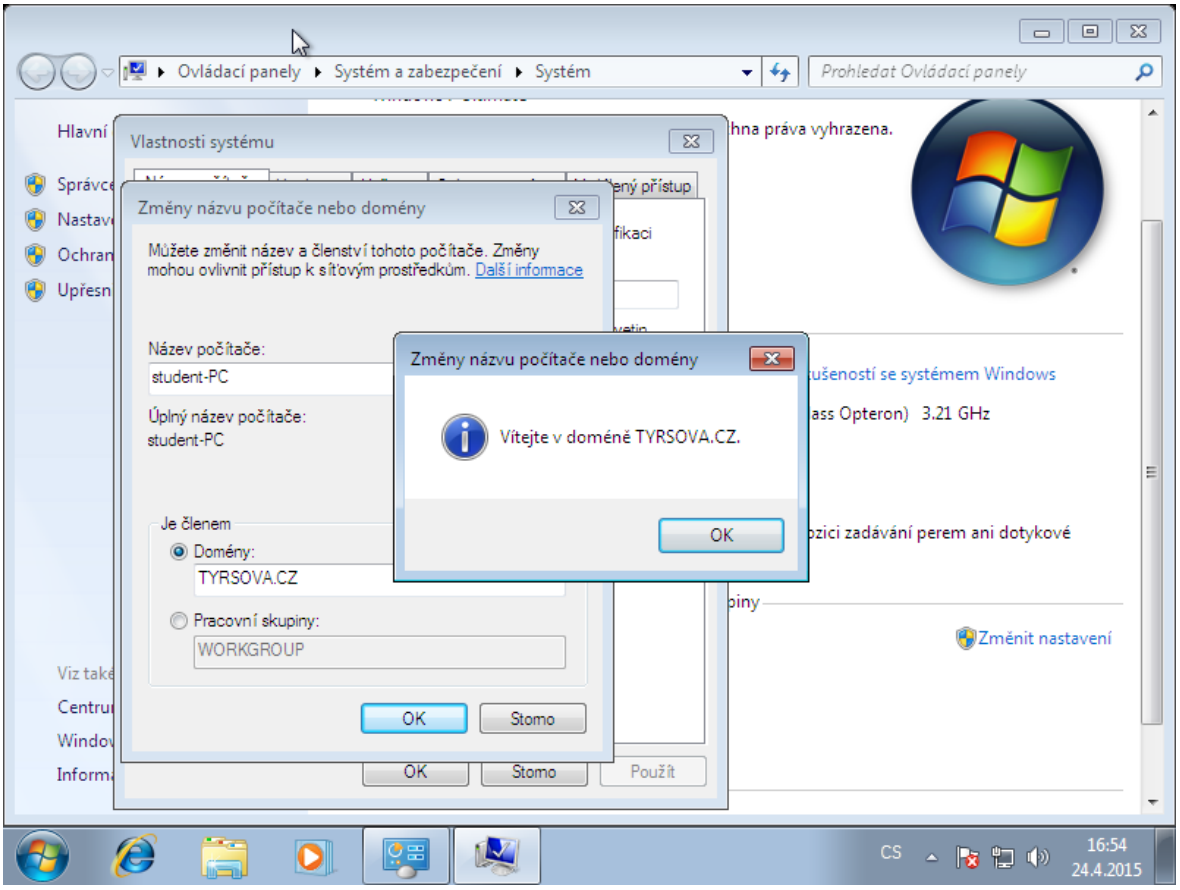
```
tar -xvzf samba-4.2.1.tar.gz
./configure
make
make install
```

Po skončení instalace můžeme již Sambu nastavit jako primární řadič domény. K tomu slouží nástroj samba-tool domain provision. Pokud přidáme parametr --interactive spustí se průvodce, který se bude postupně ptát na Kerberos realm, NetBIOS doménové jméno, roli serveru, zda chceme využít vnitřní DNS, IP adresu pro přeposílání DNS dotazů a heslo pro administrátora dané domény. Následně se vytvoří doména a v případě, že jsme zvolili " Samba Internal DNS Server", spustí se již nakonfigurovaný DNS server pro potřeby Kerberos a adresářové služby. V /etc/resolf.conf pouze nastavíme, aby systém používal tento DNS server. V mém případě bude soubor vypadat následovně:

```
domain tyrsova.cz
nameserver localhost
```

Ke konfiguraci Kerberos slouží předpřipravený soubor /usr/local/samba/private/krb5.conf, který se vytvořil během zakládání domény. Tímto souborem můžeme přepsat konfigurační soubor Kerberos /etc/krb5.conf. Pro testování Kerberos pak slouží příkaz klist, který vrací stav spravovaných domén takzvané lístky.

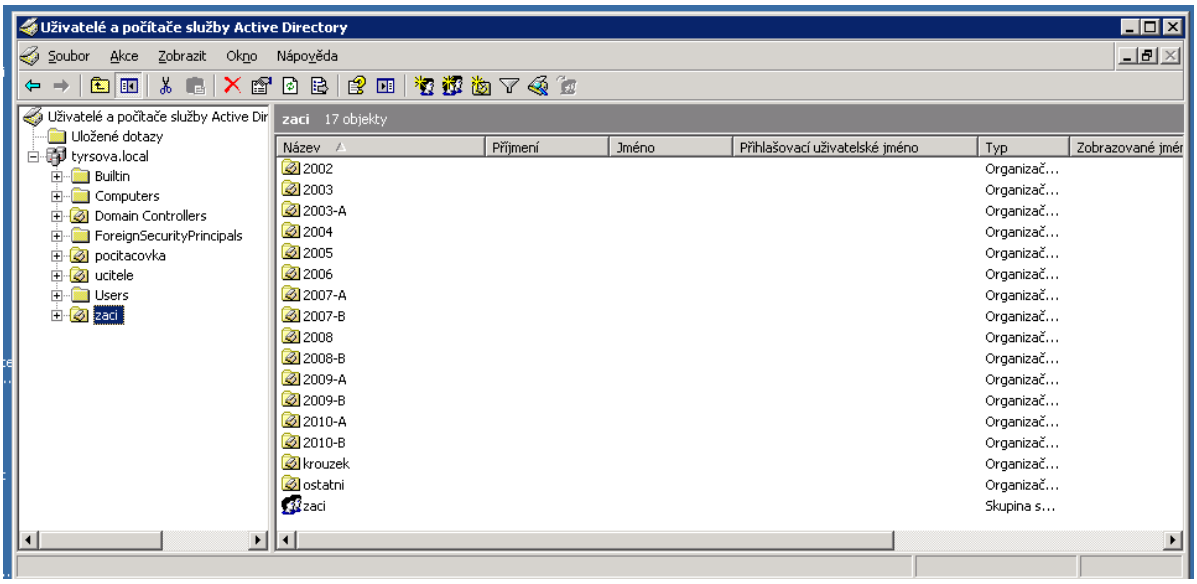
Na závěr bychom neměli zapomenout na volitelnou, avšak vysoce doporučenou službu pro korektní běh Samby, kterou je NTP server na řadiči domény. Active Directory totiž vyžaduje přesnou časovou synchronizaci mezi všemi klienty a Kerberos serverem. Posledním krokem je pak připojení uživatelských stanic do domény. Konkrétní způsob se vždy odvíjí od operačního systému, pro ukázku přikládám obrázek 4.5.1 s připojováním klientské stanice s operačním systémem Windows 7 do domény tyrsova.cz.



Obrázek 4.4.1: Připojení uživatelské stanice do domény

Pokud plánujeme doménu používat v reálných podmínkách, budeme potřebovat nějaké nástroje na správu domény. V současné době existují dvě základní

možnosti. První z nich je využít nástroje samba-tool, kterým můžeme z prostředí konzole serveru řídit chod domény. Druhou možností a programátory Samby popisovanou jako nejlehčí, je využití Microsofts RSAT (Remote Server Administration Tools) na pracovní stanici s opearčním systémem Windows.



Obrázek 4.4.2: Microsofts Remote Server Administration Tools

Jako optimální řešení, vycházející z mé zkušenosti se mi jeví kombinace obou nástrojů pro správu. Především pro drobné úpravy může být výhodnější Microsoft RSAT, který je dobře přehledný a srozumitelný, avšak pro operace s více objekty je nedostatečný a je třeba využít nástroj samba-tool.[19]

4.1 Poštovní server

Poštovní server se stará o odesílání, doručování a příjímání zpráv elektronické pošty. Pokud se rozhodneme provozovat vlastní poštovní server, získáme tím hned několik výhod. Budeme mít absolutní kontrolu nad celým serverem, zprávy budou v bezpečí bez hrozby přečtení třetí stranou a při využití svobodného software nebudeme muset platit poplatky za využívání cizích poštovních severů.

V současné době je pro oblast školství k dispozici řešení Google Apps, které je zatím nabízeno zdarma a zahrnuje v sobě mimo jiné poštovní server. Přesto se ve své práci budu věnovat konfiguraci vlastního poštovního serveru a to jak

z důvodu edukativního, tak i z důvodu stále se zhoršujících licenčních podmínek Google Apps.

Mezi dva nejrozšířenější svobodné programy s otevřeným kódem pro poštovní server se řadí Postfix a Sendmail. V mé práci využiji pro odesílání a doručování zpráv Postfix. Ke čtení skrze protokoly POP3 a IMAP bude sloužit svobodný software s otevřeným kódem Dovecot.

Pokud má Postfix sloužit jako plnohodnotný poštovní server, musí umět doručovat zprávy cílovým serverům, omezit přístup k serveru jen pro registrované uživatele, provádět formální kontrolu zpráv a mít takzvaný "blacklist" či "whitelist". V neposlední řadě se musí také starat o doručování koncovým uživatelům. Možnosti nastavení poštovního serveru jsou takřka nepřeberné a je důležité si hned na začátku ujasnit, co budeme od poštovního serveru vyžadovat. Jedním z výchozích parametrů je plánovaný počet uživatelů. Pro malé množství uživatelů v řádu jednotek je situace jiná než pro stovky uživatelů, kteří se navíc například často obměňují. Pro několik málo uživatelů stačí používat klasické systémové účty, kdy uživatel získá přístup k serveru a vytvoří se mu domovská složka v adresáři /home/ kam se budou ukládat zprávy. I když je toto řešení jednoduché, pro větší množství uživatelů je téměř nevyhnutelné vytvořit takzvané virtuální uživatele, kteří nemají v systému jako celku své uživatelské účty. Výhodou je, že můžeme oddělit využívání počítače od poštovních služeb, samostatně spravovat databázi uživatelů a hesel, kterou můžeme mít v otevřeném tvaru, protože mimo poštu se hesla nepoužívají. Do únosné meze lze mít databázi virtuálních uživatelů uloženou lokálně v souboru, avšak pro větší množství uživatelů je třeba využít některý databázový systém. Při použití kombinace Postfix a Dovecot máme na výběr z MySQL a PostgreSQL. Toto řešení je běžné na středních i velkých poštovních serverech. Dobře se spravuje a je přehledné. Ovšem v prostředí, kde máme již zavedenou centrální databázi uživatelů, kterou v případě základního či středního školství často bývá Active Directory, je většinou zbytečné mít další oddělenou databázi pro poštu. Propojení Active Directory s poštovním serverem umožňuje protokol LDAP zmíněný v předešlé kapitole.

Stavbu poštovního serveru jsem začal vytvořením nového virtuálního počítače, v rámci zachování jednotnosti operačních systému jsem využil opět Debian Linux ve verzi Wheezy. V balíčkovacím systému je k dispozici Postfix ve verzi 2.9, která je dostatečně aktuální. Instalace se provede příkazem:

```
apt-get install postfix
```

Běh programu řídí komponenta s názvem master, která spouští výkonné moduly starající se o konkrétní úkoly. To je názorně vidět v konfiguračním souboru master.cf, který je umístěn stejně jako ostatní konfigurační soubory v adresáři /etc/postfix. Většina konfigurace se ale provádí pomocí souboru main.cf. V mém případě vypadá konfigurační soubor main.cf následovně:

```
myhostname = mailserver.tyrsova.cz
myorigin = $mydomain
biff = no
delay_warning_time = 3h
mynetworks = 127.0.0.0/8

smtpd_client_restrictions =
  permit_mynetworks,
  check_sender_access hash:/etc/postfix/access,
  permit

smtpd_sender_restrictions =
  permit_mynetworks,
  reject_non_fqdn_sender,
  reject_unknown_sender_domain,
  check_sender_access hash:/etc/postfix/access,
  permit

smtpd_recipient_restrictions =
  permit_mynetworks,
  reject_non_fqdn_recipient,
  reject_unknown_recipient_domain,
```

```
  check_recipient_access hash:/etc/postfix/access,
  reject_unauth_destination,
  permit

smtpd_error_sleep_time = 30
smtpd_soft_error_limit = 10
smtpd_hard_error_limit = 20

smtpd_helo_required = yes
disable_vrfy_command = yes

virtual_mailbox_domains = tyrsova.cz
virtual_mailbox_base = /var/mail/virtual
virtual_mailbox_maps =
ldap:/etc/postfix/ldap/vmailbox.cf
virtual_alias_maps = ldap:/etc/postfix/ldap/virtual.cf
virtual_minimum_uid = 100
virtual_uid_maps = static:100
virtual_gid_maps = static:100
```

První řádek určuje celý název počítače. Parametr myorigin se používá v případě, kdy odesílatel neuvede celou svoji adresu, ale pouze název uživatele bez domény. Biff předefinovává nastavení pro lokální doručování zpráv, tedy pro systémové uživatele. V mém případě je biff = no, protože používám virtuální uživatele. Delay_warning_time definuje, po jaké době dostane odesílatel zprávu, že odeslaná pošta se stále nachází ve frontě a nebyla odeslána. Parametrem mynetworks se dostáváme k složitější otázce, kterou je povolení pro odesílání pošty. Pomocí restrikčních pravidel je potřeba nastavit aby poštovní server nemohl být zneužit pro rozesílání spamu. Restrikční pravidla fungují vždy v pořadí, v jakém jsou napsána, a pokud při vyhodnocování se některé pravidlo vykoná, vyhodnocování skončí. V mém případě se tedy nejdříve zkontroluje, zdali klient splňuje parametr mynetworks a pokud ne, zkontroluje se ještě takzvaný “whitelist“, jehož hash je umístěna v /etc/postfix/access. Následují ochranné parametry, které stanovují jak dlouho čekat při chybě a kolikrát se může

chyba zopakovat, než je klient odpojen. Smtpd_helo_required = yes vyžaduje navázání komunikace příkazem HELO nebo EHLO, což je další drobná ochrana před zneužitím serveru. V poslední části konfiguračního souboru se nastavuje používání virtuálních uživatelů ale hlavně cesty k dílčím konfiguračním souborům, které nastavují propojení s Active Directory skrz protokol LDAP.

Pro přístup k poštovním schránkám jsem využil již zmíněný program Dovecot, který se stará o IMAP a POP3 server. V konfiguraci je opět potřeba nastavit spojení skrze LDAP a zajistit přístup ke schránkám, které jsou fyzicky v domovských adresářích uživatelů Active Directory.

Nyní mají všichni uživatelé Active Directory funkční vlastní e-mail, pro odesílání a přijímání pošty. Přistupovat k němu mohou buď pomocí poštovních klientů jako je například Microsoft Outlook, Mozilla Thunderbird nebo pomocí webmailu, který může být nainstalován například na web server vytvořený v minulé kapitole. Vybírat můžeme z celé množiny svobodných aplikací pro webmail. Mezi nejznámější patří Horde, Modoboa, Roundcube, Zimbra či Squirrelmail. [21]

5. Závěr

Podařilo se navrhnout víceúčelový server pro oblast základního a středního školství, který poskytuje nejvíce využívané služby v této oblasti a zároveň využívá výhradně svobodný software s otevřeným kódem. Tím je docíleno výrazné finanční úspory při licencování softwaru. Celé řešení je maximálně škálovatelné použitím virtualizace, která zároveň umožňuje jednotlivé virtuální servery pravidelně zálohovat a bezpečně oddělit jednotlivé funkce.

Celý projekt byl realizován na testovacím serveru v uzavřené síti. Nasazení do reálného provozu bylo vzhledem k chybějícím finančním prostředkům školy na nákup serveru opožděno. V současné době probíhá výběrové řízení na dodavatele hardwaru a poté proběhne realizace.

Během testování se podařilo vyzkoušet většinu funkcí, které server nabízí včetně často obávané Samby v roli řadiče Active Directory. Vypracovat se podařil jak návrh, tak realizace, čímž bylo zadání splněno.

5.1 Seznam Obrázků

5.2 Seznam použitých zkratek

CIFS - Common Internet File System

DHCP - Dynamic Host Configuration Protocol

DNS - Domain Name System

GNU - GNU not UNIX

GPL - General Public License

IMPA - Internet Message Access Protocol

iSCSI - Internet Small Computer System Interface

KVM - Kernel Virtual Machine

LDAP - Lightweight Directory Access Protocol

LVM - Logical Volume Manager

NAT - Network address translation

OS - operaction system

POP3 - Post Office Protocol verze 3

RAID - Redundant Array of Independent/Inexpensive Disks

RDP - Remote Desktop Protocol

SAN - Storage area network

SMB - Server Message Block

USB - Universal Serial Bus

VMM - virtual machine manager

5.3 Literatura

[1] Popek, Gerald J.; Goldberg, Robert P. Formal requirements for virtualizable third generation architectures. Communications of the ACM.Vol. 17, No. 7, pp. 412–421. ISSN: 0001-0782.

[2] Manual. VirtualBox. [online]. 12.5.2015 [cit. 2015-05-12]. Dostupné z: https://www.virtualbox.org/manual/ch01.html

[3] KOMÍNEK, Jiří. Porovnání virtualizačních technologií. Jihlava 2013. Bakalářská práce, VŠPJ, Katedra elektrotechniky a informatiky.

[4] KÁBRT, Tomáš. Virtualizace operačních systému v serverech. Praha: ČVUT 2012. Bakalářská práce, ČVUT, Fakulta informačních technologií, Katedra počítačových systémů.

[5] Technical Papers. Vmware. [online]. 12.5.2015 [cit. 2015-05-12]. Dostupné z: https://www.vmware.com/vmtn/resources/

[6] XenServer Tech Info. Citrix. [online], 2014 [cit. 2014-11-12]. http://www.citrix.com/products/xenserver/tech-info.html

[7] Manual. QEMU. [online]. 12.5.2015 [cit. 2015-05-12]. Dostupné z: http://wiki.qemu.org/Manual

[8]KVM. DCE [online]. 12.5.2015 [cit 2015-1-10], https://support.dce.felk.cvut.cz/mediawiki/index.php/KVM

[9] Documents. KVM. [online]. 13.5.2015 [cit. 2015-05-13]. Dostupné z: http://www.linux-kvm.org/page/Documents

[10] DOČEKAL, Michal. Správa linuxového serveru: LVM a diskové šifrování [online], 10.1.2015 [cit. 2015-1-10]. Dostupné z http://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-lvm-a-diskove-sifrovani

[11] VÁŠA, Lukáš. Virtualizace v Linuxu. Praha: VŠE 2008. Bakalářská práce, VŠE, Fakulta informatiky a statistiky.

[12] Libvirt. Bridged networking [online], 4.1.2015 [cit. 2015-1-4]. http://wiki.libvirt.org/page/Networking#Bridged_networking_.28aka_.22shared_physical_device.22.29

[13] Vladimír Žalud. DHCP – 1 . Abclinux. [online]. 13.5.2015 [cit. 2015-05-13]. Dostupné z: http://www.abclinuxu.cz/clanky/site/dhcp-1-instalace-a-konfigurace-serveru

[14] Lukáš Zapletal. Linux jako DHCP server. Root.cz. [online]. 13.5.2015 [cit. 2015-05-13]. Dostupné z: http://www.root.cz/clanky/linux-jako-dhcp-server/

[15] DROMS, R. Request for Comments: 2131 - Dynamic Host Configuration Protocol. March 1997

[16] Internet System Consortium. BIND 9 Administrator Reference [online]. 2014. [cit. 2015-05-13]. ISBN . Dostupné z: https://www.isc.org/downloads/bind/doc/bind-9-10/

[17] About. Apache HTTP Server Project?. [online]. 13.5.2015 [cit. 2015-05-13]. Dostupné z: http://httpd.apache.org/ABOUT_APACHE.html

[18] LaMp. Debian Wiki. [online]. 13.5.2015 [cit. 2015-05-13]. Dostupné z: https://wiki.debian.org/LaMp

[19] Jelmer R. The Samba Team Vernooij. The Official Samba 3.5.x HOWTO and Reference Guide. Samba. [online]. 13.5.2015 [cit. 2015-05-13]. Dostupné z: https://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/

[20] David Iseminger. Active Directory™ Services for Microsoft® Windows® 2000 Technical Reference. Microsoft Press, 2000. ISBN 9780735606241.

[21] Lukáš Jelínek. Seriál: Stavíme poštovní server. Abclinux. [online]. 13.5.2015 [cit. 2015-05-13]. Dostupné z: http://www.abclinuxu.cz/serialy/stavime-postovni-server

[22] CALETKA, Ondřej. Osobní firewall s iptables : prezentace k semináři [online]. 13.5.2015 [cit. 2015-05-13]. Dostupné z: http://xn--ondej-kcb.caletka.cz/dl/slidy/20060511-SUT-IPtables.pdf

6. Přílohy na CD

Součástí této práce je přiložený kompaktní disk, který obsahuje tuto práci v elektronické podobě ve formátu pdf. Dále se na disku nachází archiv tar s konfiguračními soubory host systému. Konfigurační soubory jsou uloženy včetně cesty k adresáři a je tak tedy možné konfiguraci popisovanou v této práci obnovit.