ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA RIADENIA A INFORMATIKY

DIPLOMOVÁ PRÁCA

Študijný program: Systémy pre podporu rozhodovania

Bc. Radoslav Franko

Riešenie dátových úložísk pre virtualizované servery

Vedúci práce: Ing. Peter Palúch, PhD.

Reg. č.: 375/2013
Ministerské číslo práce: 28360320142375

ŽILINA, 2014

ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA RIADENIA A INFORMATIKY

DIPLOMOVÁ PRÁCA

Bc. Radoslav Franko

RIEŠENIE DÁTOVÝCH ÚLOŽÍSK PRE VIRTUALIZOVANÉ SERVERY

Študijný odbor: Informačné systémy
Študijný program: Systémy pre podporu rozhodovania

Vedúci práce: Ing. Peter Palúch, PhD.

Stupeň kvalifikácie: Inžinier (Ing.)
Dátum zadania práce: 29.10.2014
Dátum odovzdania práce: 17.4.2014

ŽILINA, 2014

Zadanie

Abstrakt

FRANKO, Radoslav: Riešenie dátových úložísk pre virtualizované servery. [diplomová práca] – Žilinská univerzita v Žiline. Fakulta riadenia a informatiky, Katedra informačných sietí. – Vedúci práce: Ing. Peter Palúch, PhD. – Stupeň odbornej kvalifikácie: Inžinier v odbore Informačné systémy. Žilina: FRI ŽU v Žiline, 2014. – 78 s.

Riešenie dátových úložísk pre virtualizované servery

Cieľom diplomovej práce je vytvorenie a nasadenie riešenia prideľovania úložného miesta na diskových poliach virtuálnym serverom s ohľadom na flexibilitu jeho správy. Základom práce je oboznámenie sa s protokolom iSCSI pre prístup k sieťovým úložiskám, s manažérom logických zväzkov, manažmentom diskových polí a zvolenými súborovými systémami v Linuxe. Práca analyzuje a rieši možnosti prideľovania úložného miesta a toto riešenie implementuje pre používanie v dátovom centre FRI.

Kľúčové slová: SAN sieť. Virtuálne servery. Protokol iSCSI. Diskové polia. Manažér logických zväzkov. Zálohovanie a migrácia virtuálnych serverov.

Abstract

FRANKO, Radoslav: Data Storage Solution for Virtualized Servers [Diploma thesis] – The University of Žilina. Faculty of Management Science and Informatics, Department of Information Network. – Tutor: Ing. Peter Palúch, PhD. – Qualification level: Engineer in field Information Network. Žilina: FRI ŽU v Žiline, 2014. - 78 p.

Data Storage Solution for Virtualized Servers

The aim of Diploma thesis is to create and implement solutions of storage allocation on disk arrays by virtual server, considering flexibility of its management. The base of thesis is to become familiar with iSCSI protocol which enables access to the network storage, with Logical Volume Management, disk arrays management, and selected file systems in Linux. Thesis analyzes and solves possibilities of storage allocation and implements this solution for using in FRI data centre.

Keywords: SAN network. Virtual servers. iSCSI protocol. Disk arrays. Logical Volume Management. Backup and migration of virtual servers.

Čestné vyhlásenie

Vyhlasujem, že som diplomovú prácu vypracoval samostatne, pod odborným vedením vedúceho Ing. Petra Palúcha, PhD. a používal som len literatúru uvedenú v práci.

V Žiline, dňa 17.4.2014

Poďakovanie

V prvom rade by som sa chcel poďakovať svojmu vedúcemu diplomovej práce Ing. Petrovi Palúchovi, PhD. za jeho odborné rady, vedenie a cenné pripomienky, ktoré dopomohli k vypracovaniu práce. Tiež by som chcel poďakovať Ing. Romanovi Hajtmánkovi za jeho pomoc, návrhy a pripomienky, ktoré boli rovnako cenné pri písaní diplomovej práce.

V neposlednom rade patrí moja vďaka rodine a priateľke za ich podporu a trpezlivosť počas vypracovania práce a tiež počas celého štúdia na vysokej škole.

V Žiline dňa 17.4.2014

ZOZNAM ILUSTRÁCIÍ A TABULIEK

ZOZNAM SKRATIEK

CPU	Central Processing Unit
FC	Fibre Channel
Gb	Gigabit
GB	GigaByte
GPL	General Public License
CHAP	Challenge Handshake Authentication Protocol
IEEE	Institute of Electrical and Electronics Engineers
IMG	Disk Image
I/O	Input/Output
IP	Internet Protocol
KB	KiloByte
KVM	Kernel-based Virtual Machine
LAN	Local Area Network
LUN	Logical Unit Number
LV	Logical Volume
MB	MegaByte
ms	milisekunda
NFS	Network File System
PV	Physical Volume
RAID	Redundant Arrays of Independent Disk
RAM	Random Access Memory
RDP	Remote Desktop Protocol
SAS	Serial Attached SCSI
SATA	Serial Advanced Technology Attachment
SCSI	Small Computer System Interface
TB	TeraByte
TCP	Transmission Control Protocol
tzv.	takzvaný
UUID	Universally Unique Identifier
VG	Volume Group
WWID	World Wide IDentification

OBSAH

ÚVOD

V súčasnej dobe čoraz viac napreduje technológia virtualizácie serverov. Využívaním jedného alebo viacerých fyzických serverov, na ktorých môže bežať niekoľko desiatok a v niektorých prípadoch dokonca až stoviek virtuálnych, je možné značne znížiť hlavne počiatočné, ale samozrejme aj dlhodobé prevádzkové náklady. Rovnako veľkým prínosom je zjednodušenie celkovej administrácie. S virtualizáciou sa otvárajú nové možnosti rovnako ako vo veľkých, tak aj v malých spoločnostiach bez ohľadu na veľkosť použitej infraštruktúry.

Smerom virtualizácie sa uberá aj dátové centrum na fakulte FRI ŽUŽ. Virtualizácia na fakulte je využívaná už dlhšiu dobu takým spôsobom, že všetky virtuálne servery sú uložené lokálne na danom serveri. Uvedený spôsob prináša rad problémov, ktoré bolo potrebné vyriešiť, preto vznikla myšlienka ukladania obrazov virtuálnych serverov na sieťovo dostupné diskové pole. Jeho využitím by bolo zabezpečené oddelenie týchto obrazov od konkrétneho fyzického servera a tým by bol minimalizovaný prípadný výpadok virtuálneho servera prevzatím jeho obrazu iným fyzickým serverom.

Cieľom diplomovej práce bolo vytvorenie a nasadenie riešenia prideľovania miesta na sieťových diskových poliach virtuálnym serverom s ohľadom na flexibilitu jeho správy. Téma virtuálnych serverov a ich možností uloženia je veľmi zaujímavá a má široké využitie v dnešnej dobe plnej komerčných platených riešení od veľkých spoločností, akými sú VMware alebo Microsoft. Open-source riešenie založené na Linuxe úplne nenahradí všetky ponúkané funkcie veľkých platených riešení, ale prinajmenšom dokáže zastrešiť základnú funkcionalitu potrebnú pre používanie na fakulte. Ďalším stanoveným cieľom práce bolo porovnanie zvolených súborových systémov a ich vhodnosť použitia na diskovom poli.

Úvodných päť kapitol diplomovej práce sa venuje teoretickému priblíženiu celej problematiky. V skratke sú vysvetlené pojmy ako virtualizácia, SAN sieť, súborové systémy v Linuxe, správca logických zväzkov a protokol iSCSI. Ku každej oblasti sú vytýčené základné vlastnosti, fungovanie a delenie.

Šiesta kapitola sa zaoberá súčasnou situáciou v dátovom centre, aktuálnymi problémami a možnosťami ich riešenia. Ďalej je priblížené testovacie prostredie a jeho príprava, pridelený hardvér a možnosti jeho zapojenia.

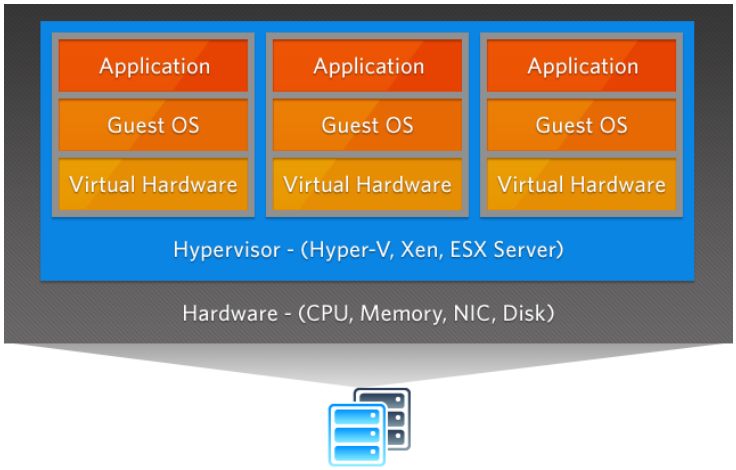
Siedma kapitola je analytická, venuje sa otázkam výberu vhodného súborového systému, porovnáva spôsoby ukladania virtuálnych strojov, rieši výber vhodného spôsobu prideľovania úložného miesta diskového poľa jednotlivým serverom a s tým spojené možné problémy. Záver kapitoly je venovaný otázkam a možnostiam zálohovania virtuálnych serverov.

Posledné dve kapitoly približujú implementáciu celého riešenia od pripojenia k diskovému poľu, cez vytvorenie a pridelenie zväzkov fyzickým serverom až po migráciu virtuálnych serverov. Rovnako je uvedených niekoľko ďalších možných konfiguračných nastavení. Záver je venovaný možnému zlepšeniu a rozšíreniu vytvoreného riešenia.

1 VIRTUALIZÁCIA

Virtualizácia je technológia, ktorá v súčasnej dobe čoraz viac napreduje a doslova mení pohľad na celkovú IT infraštruktúru. V súčasnosti sú počítače stavané primárne pre beh jedného operačného systému, ale s využitím virtualizácie je možné spustiť na jednom fyzickom počítači viacero navzájom izolovaných operačných systémov a aplikácií. Izolovaných z toho dôvodu, pretože virtuálne stroje sa správajú ako samostatné stanice a vzájomne sa neovplyvňujú. Jednotlivé operačné systémy sa spolu delia o hardvérové prostriedky ako procesor, pamäť, úložný priestor pevných diskov, vymeniteľné zariadenia, či napríklad sieťové pripojenie. Delenie umožňuje na jednom serveri beh niekoľkých desiatok a v niektorých prípadoch až stoviek rôznych operačných systémov. Keďže sú tieto systémy navzájom izolované, možný pád jedného neovplyvní beh ďalších, čo znamená, že je dodržaná aj bezpečnosť. Bez virtualizácie nie sú využívané všetky hardvérové prostriedky, čo zanecháva fyzické počítače zbytočne nevyužité. S použitím virtualizácie dochádza k ich efektívnemu využívaniu, preto si mnohé firmy bez nej nevedia predstaviť ďalšie fungovanie.

Virtualizovať je možné rovnako ako pracovné stanice, tak aj servery. Najnovší trend v tejto oblasti predstavuje práve virtualizácia serverov, ktorá prináša zásadné úspory pri zriaďovaní a celkových prevádzkových nákladoch. Rovnako zjednodušuje ich správu, vytváranie a nasadzovanie. Vďaka virtualizácii je umožnený prenos virtuálnych staníc medzi jednotlivými fyzickými servermi, čo je veľmi užitočné pri vyvažovaní výkonu ako aj pri potrebe odstávky fyzického servera bez prerušenia jeho poskytovaných služieb. [1]



Obrázok 1: Virtualizácia
(Podľa: [1])

Medzi najväčšie výhody virtualizácie patrí:

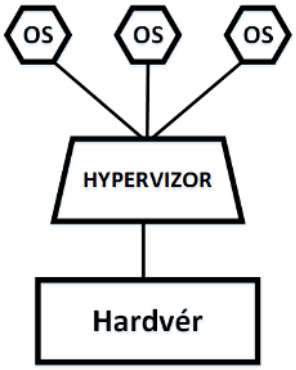
- jednoznačná úspora prevádzkových nákladov zefektívnením využívania výkonu jedného fyzického stroja namiesto viacerých, čo zároveň prináša aj úsporu miesta v dátových centrách,
- možnosť prevádzky rôznych operačných systémov na jednom fyzickom počítači (Windows a Linux),
- nasadenie, spustenie, administrácia a zálohovanie nových operačných systémov na virtualizovaných staniciach je niekoľkonásobne jednoduchšie a rýchlejšie,

- využívanie virtuálneho hardvéru operačným systémom umožňuje jednoduchú migráciu tohto systému na iné fyzické zariadenie (server) bez potreby ďalšej konfigurácie,
- takzvaná živá migrácia alebo tiež migrácia za behu umožňuje prenos bežiaceho systému z jedného fyzického zariadenia na druhé bez výpadku služieb,
- jednoduchá prenositeľnosť dát medzi rôznymi virtualizovanými systémami a takisto medzi fyzickými stanicami,
- zjednodušenie prevádzky firemných aplikácií s ohľadom na požiadavky vysokej dostupnosti a výkonu,
- jednoduchšie testovanie nových firemných riešení pred samotným nasadením do produkčného prostredia. [2]

1.1 VIRTUALIZAČNÝ SOFTVÉR

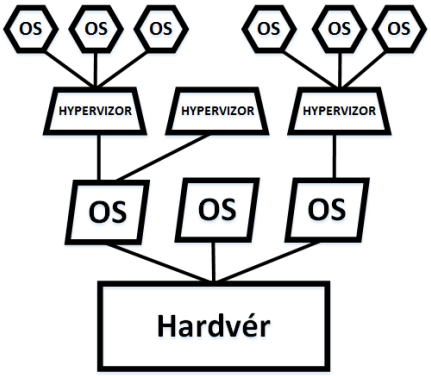
Základom virtualizovaného prostredia je virtualizačný softvér, takzvaný **hypervízor**. Jeho hlavnou úlohou je transformácia fyzických komponentov na virtuálne, ktoré potom prideľuje konkrétnej virtuálnej stanici. Hypervízor, tiež nazývaný **správca virtuálnych serverov**, je medzivrstvou medzi fyzickým hardvérom a virtuálnymi servermi. Riadi prístup virtuálnych serverov k fyzickému hardvéru, riadi ich beh a vzájomne ich oddeľuje. Po pridelení hardvérových zdrojov je stále možnosť neskoršej zmeny podľa potreby. Poznáme dva typy hypervízorov: [3]

- **Natívny (Typ 1)** – typ hypervízora, ktorý je spustený priamo na hardvéri počítača. Odtiaľ riadi beh virtuálnych strojov a monitoruje ich. Je kompletne nezávislý od operačného systému a spúšťa sa ešte pred jeho štartom. Tento typ využívajú všetci hlavní hráči na poli virtualizácie – VMware ESXi, Microsoft Hyper-V, Citrix a XEN.



Obrázok 2: Natívny hypervízor
(Podľa: [3])

- **Hosťovaný (Typ 2)** – hypervízor, ktorý je spustený priamo v prostredí operačného systému a teda sa naň úplne spolieha. Pokiaľ operačný systém nenaštartuje, nie sú dostupné ani vytvorené virtuálne stanice. Hlavným nedostatkom tohto typu je, že ak zhavaruje operačný systém sú ovplyvnené všetky virtuálne stanice. Príkladom je Oracle VM VirtualBox, KVM atď.



Obrázok 3: Hosťovaný hypervízor
(Podľa: [3])

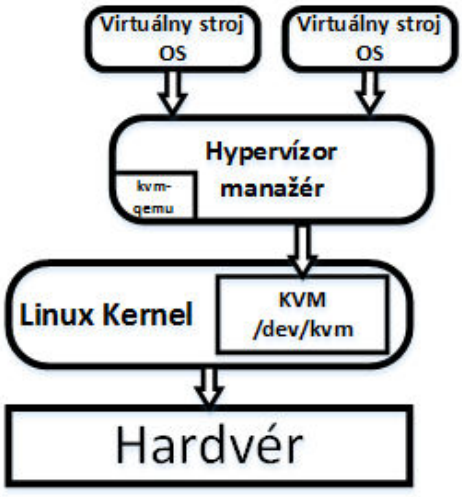
1.1.1 Typy hypervízorov

Najznámejšie a najrozšírenejšie voľne dostupné virtualizačné riešenia pre linuxové systémy sú:

KVM (Kernel-based Virtual Machine)

Je to virtualizačná infraštruktúra pre hardvérovú platformu architektúry x86. KVM je prvý hypervízor, ktorý sa stal súčasťou Linuxového jadra (zahrnutý od verzie 2.6.20). Poskytuje úplnú virtualizáciu, ktorá pre svoj beh potrebuje hardvérovú podporu procesora (Intel VT pre procesory Intel a AMD-V pre procesory AMD). KVM taktiež podporuje istú formu paravirtualizácie tým, že nepodporuje skutočnú paravirtualizíciu vrátane procesora, ale iba pre ovládače zariadení na zvýšenie I/O výkonu ako napríklad pre sieť alebo blokové zariadenia. [4]

Technológia KVM je implementovaná do dvoch zložiek. Prvou je samotný KVM modul, ktorý poskytuje správu virtualizovaného hardvéru. Druhou je modifikácia verzie QEMU, ktorá poskytuje emuláciu počítačového hardvéru. [5] Po nainštalovaní KVM hypervízora je v Linuxe vytvorený hardvérový súbor */dev/kvm*, ktorý pôsobí ako interpreter medzi fyzickým hardvérom a hypervízor manažérom. Keď príde požiadavka na hardvérové zmeny od hypervízor manažéra, KVM okamžite začína alokovať požadované zdroje pre konkrétny virtuálny stroj zo skutočného fyzického hardvéru. [6]

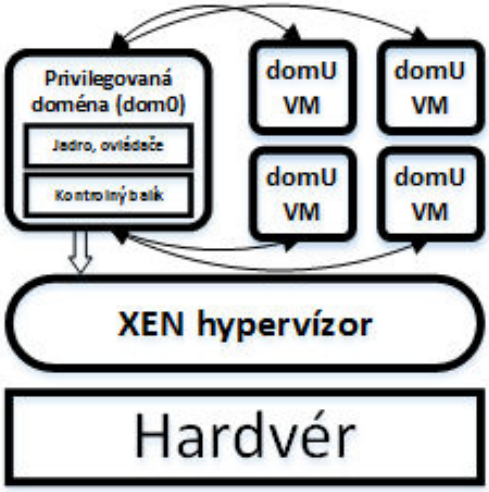


Obrázok 4: Virtualizačná architektúra KVM
(Podľa: [6])

XEN

Ide o jediný dostupný open-source natívny hypervízor (typ 1) bežiaci priamo na hardvéri, za ktorý je aj sám zodpovedný. Xen podporuje úplne virtualizované hosťovské systémy, ale aj paravirtualizované, ktoré sú rýchlejšie ako virtuálne stroje s hardvérovým rozšírením (úplná virtualizácia) a navyše nevyžadujú hardvérovú podporu na procesore.

XEN je spustený ešte pred zavedením systému. Umožňuje paralelné spustenie veľkého množstva virtuálnych strojov. Bežiaca inštancia virtuálneho stroja je nazývaná ako doména. Vždy musí byť dostupná špeciálna doména zvaná dom0 (doména 0) [7], bez ktorej nie je XEN hypervízor použiteľný. Často je nazývaná ako hosťujúca alebo privilegovaná doména, ktorá sa spúšťa ako prvá, hneď po zavedení hypervízora. Ako jediná má špeciálne práva ako priamy prístup k fyzickému hardvéru, zabezpečenie prístupu ku všetkým vstupno-výstupným funkciám systému, poskytovanie ovládačov pre všetky zariadenia v systéme. Obsahuje aj kontrolný balík pre vytváranie, konfiguráciu, odstránenie a celkovú správu a kontrolu ostatných virtuálnych strojov nazývaných ako domU. Domény domU, označované ako hosťovaná alebo neprivilegovaná doména, vyžadujú, aby bola základná doména, dom0 spustená, inak nie sú schopné existencie. Všetky ich požiadavky sú smerované na privilegovanú doménu, ktorá ich spracováva a zasiela odpovede.



Obrázok 5: Virtualizačná architektúra XEN
(Podľa: [7])

1.2 MIGRÁCIA VIRTUÁLNYCH SERVEROV

Pojem migrácia predstavuje proces presunutia virtuálneho servera z jedného fyzického servera na druhý. Tento proces je umožnený vďaka kompletnému virtualizovanému prostrediu, v ktorom bežia virtuálne servery. Funguje jednoduchým zaslaním stavu uloženej pamäte hosťovaného virtuálneho stroja cieľovému fyzickému serveru. Na rozdiel od kopírovania virtuálneho stroja, pri migrácii nie je vytváraný nový stroj. Poznáme dva typy migrácie:

- **Migrácia pri vypnutom systéme**, tzv. studená migrácia – prenáša vypnutý virtuálny stroj na nového hostiteľa.
- **Migrácia za behu**, tzv. live migrácia – umožňuje presun zapnutého virtuálneho stroja na nového hostiteľa bez výpadku dostupnosti služieb, ktoré sú na ňom spustené.

Z pohľadu efektivity je zaujímavejšia live migrácia, keďže nie je potrebné ukončovať bežiaci operačný systém prenášaného virtuálneho stroja. Pri prenášaní obsahu pamäte na cieľového hostiteľa prenášaný virtuálny server pokračuje vo svojej činnosti na zdrojovom hostiteľovi pokým nie je prenesený kompletne. KVM monitoruje obsah, ktorý sa zmenil po prenesení a tieto zmeny prenáša ako posledné. Počas migrácie taktiež odhaduje prenosovú rýchlosť, na základe ktorej je schopný pozastaviť zdrojový virtuálny stroj v prípade, že prenos zostávajúcich dát bude trvať presne určenú nastaviteľnú dobu, ktorá štandardne predstavuje 10 ms. Po jeho pozastavení presunie zostávajúce dáta a pokračuje v behu na cieľovom fyzickom serveri. Využitie migrácie je nasledovné:

- **Vyvažovanie výkonu** – v prípade nedostatku hardvérových prostriedkov fyzického stroja (pri preťažení), môžu byť jeho virtuálne stroje presunuté na iné menej zaťažené fyzické servery.
- **Hardvérová nezávislosť** – v prípade potreby výmeny akéhokoľvek hardvéru fyzického servera je k dispozícii možnosť presunutia všetkých jeho virtuálnych strojov na iné hosťujúce fyzické servery, čím je zabezpečený ich nepretržitý beh v čase vypnutia servera.
- **Energetická úspora** – počas prevádzky kedy sú fyzické servery málo zaťažené, môžu byť ich virtuálne stroje popresúvané takým spôsobom, aby sa mohli niektoré servery úplne odstaviť z dôvodu šetrenia energie.

2 SÚBOROVÉ SYSTÉMY V LINUXE

Súborový systém predstavuje **spôsob ukladania** a **organizácie dát** na disku tak, aby s nimi bolo možné jednoducho pracovať a pristupovať k nim pomocou hierarchicky usporiadaného systému súborov a adresárov. Operačný systém ich využíva pre sledovanie a organizovanie všetkých súborov na disku.

Obvykle sú pevné disky rozdelené na partície (oddiely), čo znamená, že súborový systém sa rozprestiera len na konkrétnej z nich a nie na celom disku. Takýmto spôsobom je umožnené mať na pevnom disku viacero súborových systémov, ktoré môžu byť rôzneho typu. Informácie, ktoré sú na nich uložené môžeme rozdeliť na: [8]

- **Dáta** – obsahujú reálne uložené užívateľské dáta (súbory).
- **Metadáta** – dáta, ktoré popisujú iné dáta. Obsahujú informácie o štruktúre súborového systému a jednotlivých súborov (napr. kedy a kým bol daný súbor vytvorený, prístupové práva, veľkosť atď.). [9]

Každý súbor v súborovom systéme je reprezentovaný svojim názvom a tzv. číslom „inode“. [10] Je to dátová štruktúra, ktorá uchováva metadáta pre každý súbor a adresár. V skratke je teda možné povedať, že „inode“ identifikuje súbor a jeho atribúty.

Linux pozná veľmi veľa súborových systémov. Medzi najvýznamnejšie patria EXT4, XFS, JFS, REISERFS, BTRFS, ZFS a mnoho ďalších. Rozdiely medzi nimi sú dané rôznymi obmedzeniami pre konkrétny súborový systém, z ktorých najvýznamnejšie sú:

- veľkosť úložného priestoru, ktorý je schopný pokryť,
- veľkosť súborov,
- dĺžka mien súborov,
- počet vnorených adresárov,
- podporovaná znaková sada,
- atď.

2.1 ŽURNÁLOVACÍ SÚBOROVÝ SYSTÉM

Je to súborový systém, ktorý udržuje špeciálny súbor nazvaný **žurnál**. Zápis dát a metadát prebieha v niekoľkých krokoch, preto dáta nie sú vždy v konzistentnom stave. Tento súbor je využívaný na opravu nekonzistentného stavu, ktorý sa môže vyskytnúť po nevhodnom alebo náhlom vypnutí systému spôsobeného výpadkom elektrického prúdu alebo softvérovým problémom. Princípom žurnálu je vedenie chronologického záznamu vykonávaných operácií, do ktorého sa zapisujú všetky vykonávané činnosti (logovanie).

V prípade systémového výpadku môže dôjsť k dvom situáciám. Ak po obnove žurnál hlási konzistentný stav a teda všetky zmeny v dátach boli úspešne uložené do súborového systému (zapísane korektne na pevný disk), problém nenastáva a nie je potrebná obnova dát. Ak sa však dáta nestihli uložiť korektne, žurnál nemá úspešný posledný záznam a teda nie je v konzistentnom stave, súborový systém vráti žurnál do posledného známeho konzistentného stavu a tým môže dôjsť k strate dát, ku ktorým bolo práve pristupované. Obnova dát je zabezpečená do takej miery, v akej bol hlásený posledný korektný stav pred výpadkom.

Súborové systémy, ktoré nepodporujú žurnálovanie, je vždy potrebné po výpadku skontrolovať špeciálnym programom kvôli integrite dát, čo je veľmi časovo náročná operácia. Taktiež môže pri tejto kontrole dôjsť k strate aj tých dát, ktoré neboli ovplyvnené výpadkom priamo. Preto sú žurnálovacie súborové systémy oveľa rýchlejšie a zabezpečujú,

že interná štruktúra súborového systému je vždy v konzistentnom stave. To znamená, že umožňuje mnohonásobne rýchlejšie spustenie počítača po systémovej havárii, hoci na druhej strane v niektorých prípadoch taktiež nezabraňuje strate dát. [11]

2.2 VÝZNAMNÉ SÚBOROVÉ SYSTÉMY

Významnými zástupcami linuxových súborových systémov sú **EXT4, XFS** a **BtrFS**, ktoré budú v tejto sekcii bližšie popísané.

2.2.1 XFS (eXtended File System)

Je to vysoko výkonný žurnálovací súborový systém vytvorený spoločnosťou Silicon Graphics v roku 1993. Prvýkrát bol pridaný do jadra Linuxu verzie 2.4 v roku 2002 a odvtedy je dostupný vo všetkých linuxových jadrách.

Základné vlastnosti: [12]

- maximálna veľkosť súborového systému: 16 TB/32-bit a 18 EB/64-bit,
- maximálna veľkosť súborov: 16 TB/32-bit a 9 EB/64-bit,
- maximálna veľkosť bloku súborového. systému: 4 KB/32-bit a 64 KB/64-bit,
- dĺžka mien súborov: 255 znakov,
- počet vnorených adresárov: niekoľko miliónov,
- podporovaná znaková sada: všetky okrem NULL.

Najvýznamnejšie funkcie:

- **Veľmi rýchla obnova** – umožňuje veľmi rýchly reštart po neočakávaných výpadkoch bez ohľadu na počet spravovaných súborov.
- **Rýchle transakcie** (operácie) – používa efektívne stromové štruktúry pre rýchle vyhľadávanie a alokáciu úložného priestoru spojené s veľmi rýchlou dobou odozvy.
- **Obrovská škálovateľnosť** – je to plne 64-bitový adresný súborový systém, a preto umožňuje prácu so súbormi o veľkosti v miliónoch TB.
- **Výborná priepustnosť** – poskytuje veľmi rýchly prenos veľkých súborov, ktorý je rýchlosťou blízko vstupno-výstupného výkonu v prípade prázdneho disku. Taktiež je veľmi rýchly pri formátovaní a pripájaní k systému.
- **Online spáva** – umožňuje zväčšovanie pripojených zväzkov, podporuje snímky na úrovni zväzkov a taktiež online defragmentáciu.

Nevýhodami môže byť pomalšia práca s veľkým počtom malých súborov, preto nie je vhodný pre databázy, email atď. Nevýhodou je aj nemožnosť zmenšenia zväzku, na ktorom sa nachádza.

2.2.2 EXT4 (Fourth Extended File System)

Žurnálovací súborový systém, ktorý vznikol ako rozšírenie staršieho EXT3 s cieľom zvýšenia výkonu, stability, spoľahlivosti a tiež rozšírenia o nové funkcie. Jeho stabilná verzia bola zavedená v Linuxovom jadre verzie 2.6.28 v roku 2008. Využíva 48-bitový adresný systém z čoho vyplývajú tieto základné vlastnosti: [13]

- maximálna veľkosť súborového systému: 1 EB,
- maximálna veľkosť súborov: 16 TB,
- dĺžka mien súborov: 255 znakov,
- počet vnorených adresárov: nelimitovaný počet,
- podporovaná znaková sada: všetky okrem NULL a /.

Najvýznamnejšie funkcie:

- **Spätná kompatibilita** - existujúci EXT3 súborový systém je možné zadaním niekoľkých príkazov jednoducho migrovať na EXT4 a teda nie je potrebná reinštalácia bežiaceho operačného systému. Na druhej strane ale nie je umožnená migrácia späť na EXT3.
- **Zvýšenie výkonu** – využívanie tzv. „extents“, čo je vlastnosť, ktorá mapuje veľký počet fyzických blokov na jeden ukazovateľ a tým umožňuje spojiť 4KB bloky až na 128MB priestor, čím zlepšuje výkon a redukuje fragmentáciu.
- **Online defragmentácia.**
- **Vylepšená kontrola súborového systému** – kontrola sa sústreďuje iba na tie časti, kde by mohla byť chyba, čím sa výrazne zrýchli kontrola celého súborového systému.
- **Rozšírenosť** – je podporovaný všetkými Linuxovými distribúciami.

2.2.3 BtrFS (Better File System)

Je to nový súborový systém s GPL licenciou, ktorý implementuje pokročilé funkcie. Niektoré z nich prebral od súborového systému ZFS a niektoré od ReiserFS. Jeho vývoj začal v roku 2007 spoločnosťou Oracle. Na rozdiel od ostatných dvoch nepoužíva žurnál. V dnešnej dobe je stále považovaný za experimentálny, pretože ešte nebol nasadený do produkčného prostredia. Jeho základné vlastnosti sú:

- maximálna veľkosť súborového systému: 16 EB,
- maximálna veľkosť súborov: 16 EB,
- dĺžka mien súborov: 255 znakov,
- počet vnorených adresárov: nelimitovaný počet,
- podporovaná znaková sada: všetky okrem NULL a /. [14]

Najvýznamnejšie funkcie:

- ponúka vysoký výkon, preto je veľmi vhodný pre servery,
- využíva štruktúru B-stromu pre ukladanie dát,
- online defragmentácia,
- veľmi rýchla kontrola súborového systému.

3 LOGICAL VOLUME MANAGER (LVM)

LVM je v doslovnom preklade **správca logických zväzkov** pre operačný systém Linux. V súčasnosti poznáme dve verzie:

- **LVM 1** – pôvodná verzia, obsiahnutá v jadre Linuxu verzie 2.4. Je to vyspelá verzia, stabilná počas niekoľkých rokov.
- **LVM 2** – najnovšia, vylepšená verzia LVM. Je takmer úplne spätne kompatibilná so zväzkami vytvorenými pomocou LVM verzie 1. Výnimkou sú snímky zväzkov, ktoré je potrebné pred migráciou odstrániť. Táto verzia bude používaná vo zvyšku práce. [15]

Správa logických zväzkov poskytuje vyššiu úroveň pohľadu na diskové úložisko v porovnaní s bežným pohľadom na disky a partície pomocou operačného systému.

3.1 NAJVÝZNAMNEJŠIE FUNKCIE LVM

Jednoduchá správa – s využitím LVM majú administrátori oveľa väčšiu flexibilitu pri alokácii priestoru aplikáciám a užívateľom. Zväzky vytvorené pomocou LVM je možné ďalej zväčšovať, zmenšovať, presúvať a meniť podľa potreby. Taktiež je umožnené pomenovanie logických zväzkov a skupín zväzkov, čím je uľahčené ich rozpoznávanie a teda nie je potrebné pracovať s konkrétnymi názvami fyzických zväzkov ako napríklad „sda1, sdb2“ atď.

Široké využitie - využívanie LVM je často spájané s väčšími inštaláciami obsahujúcimi veľký počet pevných diskov, avšak je tiež vhodné pre menšie systémy s dvomi, tromi a dokonca s jedným pevným diskom.

Jednoduché zálohovanie – možnosť vytvárania záloh konkrétnych logických zväzkov vytváraním ich snímok.

Zjednodušené pridávanie diskov – pri pridávaní nových diskov už viac nie je potrebné manuálne presúvanie dát. Namiesto toho sa disk pridá do existujúcej skupiny zväzkov a následne je umožnené rozšírenie konkrétnych logických zväzkov o požadovanú veľkosť, ktorá je limitovaná veľkosťou pridávaného disku. Rovnako je vylepšené odobratie starých diskov, kedy je možný jednoduchý presun dát za behu na novšie disky, bez ovplyvnenia bežiacich služieb.

3.2 ANATÓMIA LVM

Predtým ako bude priblížená samotná práca s LVM manažérom, je potrebné oboznámiť sa s nasledovnými pojmami:

Fyzický zväzok (PV) – môže to byť buď celý pevný disk, alebo len jeho časť – partícia. Je základom pre vytváranie VG a to takým spôsobom, že na začiatku tvorí VG jeden alebo viacero fyzických zväzkov, ktoré je možné v prípade potreby ďalšieho rozšírenia ľubovoľne dokladať bez potreby ukončenia práce s bežiacim operačným systémom.

Skupina zväzkov (VG) – je to základná administratívna jednotka pri práci s LVM. Označuje veľký priestor vytvorený z niekoľkých fyzických zväzkov, z ktorého je možné vytvárať jednotlivé logické zväzky uvoľnením potrebného diskového priestoru. Pokiaľ VG obsahuje dostatok nepridelenéhо priestoru, tento voľný priestor môže byť využitý pre vytvorenie ďalších LV alebo pre rozšírenie už vytvorených.

Logický zväzok (LV) – je ekvivalentom pre konkrétnu partíciu vytvorenú v systéme, kde sa nevyužíva LVM. To znamená, že je to zväzok, ktorý sa ponúka operačnému systému ako štandardné blokové zariadenie pre ukladanie dát a teda je možné ho k nemu jednoducho pripojiť. Podobne ako klasická partícia môže obsahovať súborový systém, naviac ešte ponúka možnosť pridelenia ľubovoľného názvu a jednoduchšiu spravovateľnosť.

4 SIEŤOVÉ ÚLOŽISKÁ

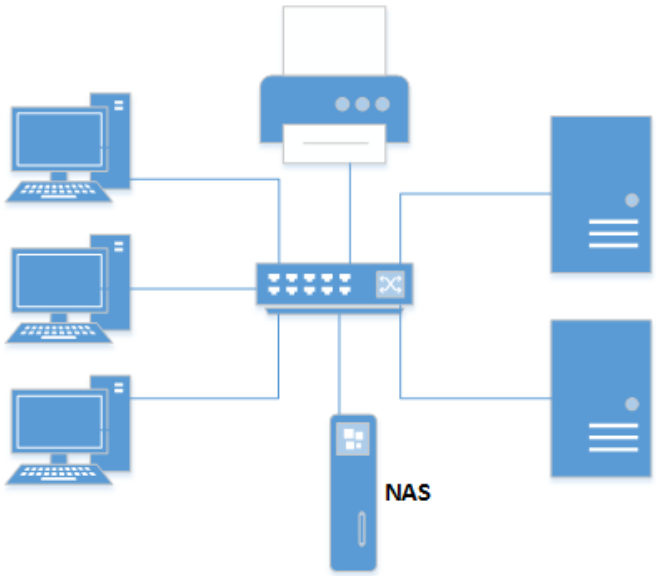
Poznáme tri základné spôsoby implementácie sieťových úložísk:

- **Network Attached Storage** (NAS),
- **Direct Attached Storage** (DAS),
- **Storage Area Network** (SAN).

4.1 NAS

Toto riešenie pozostáva zo samostatného úložného zariadenia s diskovým priestorom, ktoré sa pripája priamo do ethernetovej siete. Má vlastnú sieťovú adresu a poskytuje rôzne služby pre zdieľanie a ukladanie dát ostatným zariadeniam v spoločnej LAN sieti. Všetci pripojení užívatelia majú k nemu priamy prístup a teda sa nemusia pripájať cez iné zariadenie ako napríklad server. Technológia NAS zjednodušuje správu a zlepšuje prístupnosť dát klientom a aplikáciám bez ohľadu na to, aký operačný systém používajú. Samotné zariadenie obsahuje jeden alebo viacero pevných diskov, ktoré je možné zlúčiť a vytvoriť tak RAID pole. Na rozdiel od iných riešení, súborový systém je umiestnený a spravovaný priamo na NAS zariadení. Prenos dát po TCP/IP sieti je realizovaný využitím štandardných protokolov pre zdieľanie dát – NFS (pre unixové servery), SMB/CIFS (pre servery založené na OS Windows). [16]

Využitie NAS zariadení je rozšírené o ďalšie funkcie ako napríklad jednoduchý web server, tlačový server, PHP server, databázový server, FTP server, mediálny server (DLNA), iTunes server, download server a takisto niektoré podporujú pripojenie a nahrávanie z IP kamier. Obrázok 6 znázorňuje schému pripojenia NAS zariadenia do siete.



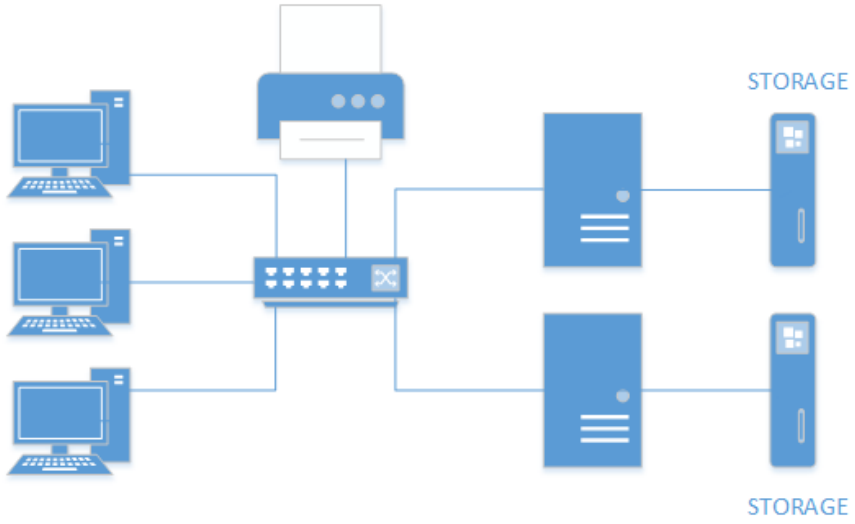
Obrázok 6: Schéma pripojenia NAS zariadenia

4.2 DAS

Je to samostatné úložisko dát, ktoré sa skladá z niekoľkých pevných diskov zapojených do RAID poľa. Hlavnou charakteristikou tohto riešenia je, že zariadenie je priamo pripojené k serveru alebo ku pracovnej stanici, čo znamená, že medzi nimi nie je žiadne iné sieťové zariadenie. Toto pripojenie zo strany servera alebo pracovnej stanice je

zabezpečené pomocou HBA adaptéra, ktorý sa pripája priamo na úložné zariadenie prostredníctvom rozhraní SATA, SAS alebo FC (Fibre Channel).

Výhodou tohto riešenia je jednoduchosť a relatívne nízka finančná náročnosť počiatočného nasadenia. Čo sa týka výkonu, riešenie ponúka koncovým užívateľom lepší výkon pre určité typy aplikácií, pretože úlohy čítania a zapisovania dát nemusia prechádzať celou sieťovou infraštruktúrou, ale iba k priamo pripojenému zariadeniu – serveru. Práve kvôli tomuto väčšina firiem nasadzuje DAS riešenie pre aplikácie, ktoré vyžadujú vysoký výkon. Príkladom môže byť databázové spracovanie atď. Na druhej strane však toto riešenie nie je schopné nezávislého a samostatného zdieľania dát a zdrojov ostatným serverom a v prípade výpadku príslušného servera sa stáva nedostupným aj samotné úložisko a teda aj všetky uložené užívateľské dáta. V súčasnej dobe firmy viac nasadzujú SAN sieť, pretože ponúka lepšiu flexibilitu, výkon a hlavne dostupnosť v porovnaní s DAS. Schéma zapojenia DAS do siete je zobrazená na Obrázku 7.

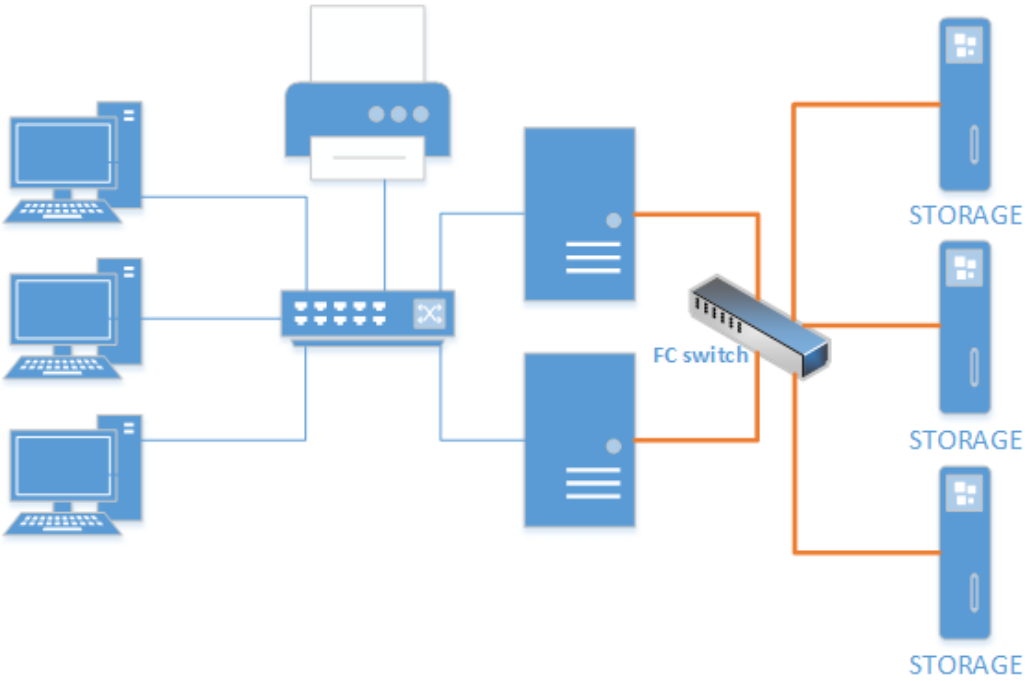


Obrázok 7: Schéma pripojenia DAS zariadení

4.3 SAN

SAN je skratkou od Storage Area Network, čo v preklade znamená sieť pre ukladanie dát, ktorá je navrhnutá pre zabezpečenie veľkých dátových prenosov. Technicky je možné SAN sieť označiť ako vysokorýchlostnú sieť počítačov (serverov), úložných zariadení, prípadne iných zariadení tvoriacich infraštruktúru určenú pre ukladanie, ochranu a zálohu dát. Je to samostatná dátová sieť, ktorá zabezpečuje pripojenie externých dátových zariadení (diskové polia, páskové knižnice a ďalšie zálohovacie zariadenia) k serverom alebo pracovným staniciam. Umožňuje zdieľanie externého úložiska všetkým zariadeniam v SAN sieti bez ovplyvnenia systémového výkonu alebo primárnej siete. Poskytuje dátové úložisko na blokovej úrovni, ktoré je prístupné pre akékoľvek aplikácie bežiace na ľubovoľnom serveri v celej SAN sieti. [17]

Každá SAN sieť obsahuje vlastnú sieť zariadení (prepínače, úložiská, servery). Úložné zariadenia pripojené pomocou SAN siete sa tvária ako zariadenia, ktoré sú priamo pripojené k danému serveru. Zvyčajne sú umiestnené v rámci jednej miestnosti, ale tiež môžu byť pripojené na veľké vzdialenosti, čím sú veľmi užitočné pre veľké firmy. Umožnená vzdialenosť pripojenia je daná použitými technológiami pre prenos dát. Schéma prepojenia v SAN sieti je zobrazená na nasledujúcom obrázku.



Obrázok 8: Schéma pripojenia do siete SAN

4.3.1 Použité technológie v SAN sieti

Výmenou SCSI príkazov prebieha komunikácia väčšiny serverov a diskových úložísk. Poznáme dve najpoužívanejšie SAN riešenia, ktoré sú dostupné ako dve rozdielne technológie pre prenos SCSI príkazov:

- **Fibre Channel Protocol**(FCP) – transportný protokol určený pre mapovanie SCSI príkazov cez optickú sieť,
- **iSCSI** – bližšie popísané v ďalšej kapitole.

Obe technológie poskytujú svoje výhody a nevýhody. Výber záleží len na firemných požiadavkách. Je možné povedať, že doposiaľ bolo dominantné pripojenie do siete SAN pomocou technológie optických vlákien (FC), avšak v jej neprospech hrajú vysoké počiatočné náklady, ktoré plynú z potreby vybudovania vlastnej samostatnej siete vrátane zariadení a hosťovských adaptérov pre pripojenie do optickej siete. Z tohto dôvodu sa v menších firmách čoraz viac využíva na prenos už existujúca sieťová infraštruktúra ethernetovej siete, ktorá využíva protokol iSCSI. Jednoduchú SAN sieť teda môžeme vytvoriť využitím iSCSI softvérových ovládačov na serveroch a iSCSI kompatibilných diskových polí cez existujúcu TCP/IP sieť. Jeho nespornou výhodou je taktiež možnosť využitia okrem lokálnej siete LAN aj samotnú sieť internet, čo umožňuje pripojenie úložných zariadení prakticky na neobmedzenú vzdialenosť.

4.3.2 Výhody použitia siete SAN

SAN sieť ponúka veľké množstvo výhod. Medzi najväčšie, ktoré pomôžu pri rozhodovaní o jeho nasadení, je možné zaradiť: [18]

- **vysokú dostupnosť a dynamickú ochranu proti výpadku** – možnosť vytvorenia úplne redundantného riešenia viacnásobnými fyzickými prepojeniami jednotlivých zariadení, čím je vytvorená ochrana proti výpadku v prípade zlyhania niektorých častí infraštruktúry,

- **zdieľanie úložného priestoru** – pretože všetky pripojené úložiská sú dostupné pre všetky servery v SAN sieti, voľný diskový priestor konkrétneho úložiska (ale aj celé úložisko) môže byť v prípade potreby alokovaný na požiadanie ktorémukoľvek serveru v ľubovoľnom čase,
- **vysoký výkon** – vysokorýchlostné pripojenie pomocou FC a 10 Gb Ethernetu a použitie vysokorýchlostných diskových technológií zvyšuje vstupno-výstupný výkon,
- **lepšie využitie diskového priestoru** – možnosť prístupu viacerých serverov k jednému fyzickému (logickému) disku, čo umožňuje efektívne využívať zostávajúci voľný priestor,
- **možnosti obnovy po výpadku** – zariadenia v SAN sieti môžu zrkadliť dáta na úložiská umiestnené mimo daného zariadenia, ktorého dáta v prípade výpadku je možné jednoducho obnoviť iným serverom, prípadne prevziať jeho funkcionalitu,
- **zvýšenie vzdialenosti medzi pripojenými zariadeniami** – v porovnaní s diskami priamo pripojenými pomocou rozhrania SCSI je umožnené pripojenie k serverom na oveľa väčšie vzdialenosti,
- **jednoduchá rozšíriteľnosť** – možnosť transparentného pridania ďalšieho úložiska bez ovplyvnenia ostatných zariadení v SAN sieti,
- **centralizované zálohovanie** – umiestnenie zálohy zariadení celej infraštruktúry je možné realizovať na jedno konkrétne zariadenie určené pre zálohu,
- **zníženie počtu administrátorov** – možnosť spravovania väčšieho množstva dát menším počtom administrátorov a tým pádom s menšími nákladmi.

4.3.3 Jednotlivé časti SAN siete

Kompletnú infraštruktúru SAN siete je možné rozdeliť do troch vzájomne na sebe závislých vrstiev: [19]

- Hosťovská vrstva,
- Stavebná (štruktúrna) vrstva,
- Úložná vrstva.

Hosťovská vrstva

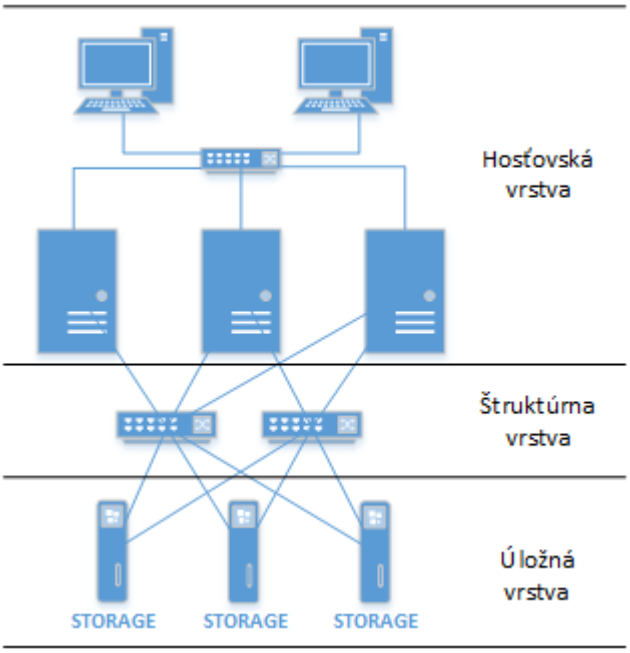
Hosťovská vrstva je vrchnou vrstvou. Medzi jej hlavné komponenty patria samotné servery, hosťovské adaptéry (HBA) a NIC karty, ktoré umožňujú serverom fyzické pripojenie do SAN siete. Tiež zahŕňa bežiaci softvér vrátane ovládačov, ktorý umožňuje komunikáciu medzi adaptérmi a štruktúrnou vrstvou.

Štruktúrna vrstva

Je prostrednou vrstvou, ktorá je v podstate sieťovou časťou SAN siete. Tvoria ju káble pripájajúce všetky zariadenia, rozbočovače, prepínače a smerovače, ktoré spolu s káblami tvoria fyzickú sieť.

Úložná vrstva

Je spodnou vrstvou, kde sú uložené všetky dáta a teda obsahuje všetky diskové jednotky, páskové jednotky a ďalšie úložné zariadenia. Súčasťou sú taktiež technológie na ochranu dát ako napríklad RAID polia a snímky zväzkov.



Obrázok 9: Jednotlivé časti SAN siete

(Podľa: [19])

5 PROTOKOL ISCSI

ISCSI je skratkou od Internet Small Computer System Interface. Je to **spojovo orientovaný sieťový protokol**, ktorý zabezpečuje prenos dát medzi úložnými zariadeniami (diskové pole, pásková knižnica atď.), servermi a klientmi takým spôsobom, že prenáša SCSI príkazy prostredníctvom TCP/IP siete. Poskytuje riešenie, ktoré má veľkú výhodu v tom, že sa dá nasadiť na už vytvorenú sieťovú infraštruktúru a takisto nie je obmedzené vzdialenosťou. Koncepcia vychádza z technológií rozhrania SCSI a protokolu TCP/IP.

5.1 ROZHRANIE SCSI

Je to **súbor štandardov a príkazov** pre fyzické pripojenie zariadení, najmä pevných diskov a páskových mechaník a taktiež pre prenos dát medzi počítačom a periférnymi zariadeniami. Tento štandard môžeme najčastejšie vidieť vo výkonných pracovných staniciach alebo serveroch. Servery pripojené pomocou RAID systému používajú takmer vždy SCSI pre pripojenie pevných diskov. SCSI sa vyslovuje ako „skazi“. Poznáme rozličné verzie štandardu SCSI. Nižšie sú popísané len tie najznámejšie:

- **Parallel SCSI** - prvá verzia, ktorá používala dizajn paralelnej zbernice. Vývoj tohto rozhrania skončil v roku 2003.
- **Serial Attached SCSI (SAS)** - je vývojovým nasledovníkom parallel SCSI. Riadiaca jednotka je pripojená priamo k diskovým jednotkám. Hlavný rozdiel je ten, že SAS zariadenia obsahujú dva dátové porty, čo znamená že ich je možné pripojiť k dvom rôznym SAS doménam. Takéto riešenie sa používa kvôli redundancii - ak jedna cesta zlyhá, pre komunikáciu je automaticky zvolená druhá cesta.

5.2 ISCSI KOMUNIKÁCIA

Rovnako ako pri SCSI, aj pri protokole iSCSI je využívaná architektúra typu klient/server. Klientskou časťou rozhrania iSCSI je takzvaný **iniciátor**, ktorý iniciuje komunikáciu odosielaním SCSI príkazov pre získanie určitého typu služby logickej jednotky servera. Túto serverovú časť reprezentuje **target,** ktorý prijaté požiadavky spracováva, vykonáva a zasiela späť odpoveď. Nikdy nezačína spojenie, ale čaká na požiadavky od iniciátora. Target môže obsahovať niekoľko logických jednotiek, kde je každej priradená vlastná adresa nazvaná číslo logickej jednotky – LUN. Výsledkom každého SCSI príkazu je dátová fáza a k nej požadovaná fáza odpovede. Informácia v dátovej fáze môže putovať v troch smeroch a to buď od iniciátora k targetu, inak nazývané ako zápis, od targetu k iniciátorovi, nazývané čítanie alebo obojsmerne. Target vo fáze odpovede odovzdáva konečný stav o požadovanej operácii a to aj vrátane prípadných chýb.

Komunikácia je medzi iniciátorom a targetom vykonávaná prostredníctvom jedného alebo viacerých TCP spojení. Každé TCP spojenie je nositeľom SCSI príkazov, kontrolnej správy, parametrov a dát v rámci výmeny iSCSI správ medzi iniciátorom a targetom. Zoskupením viacerých TCP spojení je vytvorená relácia, ktorá má definované vlastné unikátne číslo relácie (SSID). Jednotlivé TCP spojenia v rámci danej relácie sú tiež identifikované vlastným ID číslom spojenia (CID) a je ich možné odstrániť, prípadne podľa potreby pridať. Poznáme dva typy relácií:

- **Objavenie** (discovery session) – nastáva ešte pred vytvorením samotnej relácie. Je to relácia otvorená iba pre objavenie konkrétneho dostupného targetu. Počas nej iniciátor zadáva konkrétny target portal, ktorým sa pýta na jeho zoznam dostupných targetov.

- **Prevádzková relácia** – prebieha po objavení a vytvorení spojenia.

Pri používaní iSCSI protokolu sa veľmi často využíva výraz „**portal**“, ktorý označuje spojenie IP adresy targetu a TCP portu, ktorý je štandardne nastavený na číslo 3260. Príklad:

10.2.0.41:3260

Priebeh komunikácie je možné začleniť do troch fáz:

1. Nadviazanie spojenia medzi iSCSI iniciátorom a targetom prostredníctvom TCP protokolu, nazývané tiež ako iSCSI prihlásenie (login). Pokiaľ je úspešné, target toto spojenie potvrdí, čím sa vytvorí relácia.
2. Prenos dát, zasielanie príkazov, od iniciátora k targetu pomocou vytvoreného TCP spojenia. Na jednom konci tohto prenosu je teda iniciátor a na druhom target. Medzi nimi vytvorená iSCSI úloha je iSCSI požiadavka, pre ktorú je očakávaná odpoveď.
3. Ukončenie iSCSI relácie, zrušenie TCP spojenia nazývané ako iSCSI odhlásenie (logout).

5.3 ISCSI ADRESOVANIE

Pre zabezpečenie komunikácie v iSCSI prostredí je potrebné, aby všetky zariadenia mali pridelené adresy, ktoré musia byť unikátne z dôvodu identifikácie. Ak je target vytvorený prostredníctvom Linuxového systému, jeho adresu je možné špecifikovať podľa vlastných potrieb. Pokiaľ sa ale jedná o samostatné úložné zariadenie, jeho adresa je pridelená automaticky podobne ako adresa pre iniciátora. Tieto pridelené adresy je možné v prípade potreby zmeniť, avšak treba dávať veľký pozor na to, aby boli jedinečné. Sú známe dva rozličné typy adries: [20]

IQN (iSCSI Qualified Name) [21] – tento typ adresy môže byť použitý ľubovoľnou organizáciou, ktorá vlastní registrované doménové meno. Toto meno musí byť rezervované, aby bolo ostatným zabránené vygenerovať iqn adresu s rovnakým doménovým menom, pretože musí byť celosvetovo jedinečná. Príklad:

iqn.1986-03.com.hp:storage.msa2312i.104511842e

Uvedený iqn reťazec pozostáva z nasledujúcich častí:

- Začiatočný reťazec „iqn“ určuje, že sa jedná o iqn adresu (a nie napríklad eui).
- Pole dátumu určuje prvý celý mesiac a rok kedy bolo uvedené doménové meno zaregistrované. Z príkladu je možné vidieť, že to bol marec 1986.
- Za dátumom nasleduje názov rezervovaného doménového mena. Vždy sa zapisuje v opačnom poradí, čiže doména hp.com je zapísaná ako com.hp.
- Znak „:“ oddeľuje úvodný presný formát, ktorý by mal byť dodržiavaný od voliteľnej časti.
- Reťazec za znakom „:“je volený vlastníkom doménového mena, ktorý je zároveň zodpovedný za priradenie celosvetovo unikátneho reťazca. Používa sa napríklad typ produktu, sériové číslo alebo iný reťazec, podľa ktorého je možné dané zariadenie identifikovať.

Eui-64 (Extended Unique Identifier) – typ adresy poskytnutý registračnou autoritou IEEE, ktorá je zodpovedná za priradenie globálne unikátneho identifikátora. Tento typ je tiež často využívaný v ostatných sieťových protokoloch. Formát adresy je nasledovný: [22]

eui.3c970e19a957b14c

Eui adresa sa skladá z dvoch častí:

- Reťazec „eui“ určuje, že sa jedná o eui adresu.
- Druhú časť reťazca tvorí osem hexadecimálnych čísiel, ktoré spolu vytvárajú 64 bitové číslo. Prvých 24 bitov tvorí unikátne číslo organizácie (OUI číslo) priradené IEEE registračnou autoritou a zvyšných 40 bitov je priradených samotnou organizáciou.

5.4 LUN (LOGICAL UNIT NUMBER)

Keďže úložné zariadenie môže poskytovať viacero logických jednotiek, je potrebné každú túto jednotku jednoznačne identifikovať. LUN je unikátne označenie pre konkrétnu logickú jednotku, ktorá predstavuje **úložný priestor** poskytovaný úložným zariadením pripojeným pomocou iSCSI, Fibre Channel alebo iným podobným rozhraním. Je základom pre správu úložísk zdieľaných prostredníctvom siete SAN. [23]

Každé číslo LUN identifikuje špecifickú logickú jednotku, ktorá môže prestavovať buď určitú časť pevného disku, celý pevný disk, alebo viacero pevných diskov spojených pomocou RAID poľa. Pri priraďovaní LUN čísiel logickým jednotkám je všeobecne odporúčané začať až od čísla jeden, pretože nula štandardne predstavuje celú diskovú kapacitu disku alebo diskového poľa.

Priraďovanie konkrétnych logických jednotiek k serverom sa nazýva **zónovanie**. Je to metóda, pomocou ktorej je možné koncovým zariadeniam povoliť prístup iba k zvoleným logickým jednotkám. To znamená, že môžu vidieť iba úložné zariadenia v rámci ich vlastnej zóny, čím je zvýšená bezpečnosť. Zónovanie môže byť použité pri izolácii konkrétneho servera od špecifickej logickej jednotky, prípadne pri priradení špecifickej jednotky skupine zariadení.

6 ÚVOD DO PROBLEMATIKY

Cieľom diplomovej práce bolo vytvorenie a nasadenie riešenia prideľovania úložného priestoru na diskových poliach virtuálnym serverom s ohľadom na flexibilitu jeho správy. Pred začatím samotného riešenia bolo potrebné oboznámiť sa s pridelenými zariadeniami, požiadavkami a s celkovou situáciou v dátovom centre FRI.

6.1 SÚČASNÁ SITUÁCIA V DÁTOVOM CENTRE FRI

Pred zadaním diplomovej práce bola situácia v dátovom centre riešená veľmi jednoducho. Diskové polia so servermi vytvárali SAN sieť, ktorá bola využívaná iba na ukladanie a zálohu bežných užívateľských dát. Nebola využívaná úložná kapacita všetkých diskových polí a taktiež do nej neboli začlenené všetky fyzické servery.

Ako už bolo v úvode spomenuté, súčasný trend smeruje k masívnemu využívaniu virtualizácie a teda k vytváraniu virtuálnych strojov priamo na fyzickom hardvéri z dôvodu jeho lepšieho využitia. Momentálne sú tieto virtuálne stroje ukladané priamo na fyzických strojoch, z čoho plynie rad problémov spojených s výpadkom daného fyzického stroja. V prípade takéhoto výpadku sú ovplyvnené všetky na ňom vytvorené virtuálne stroje tak, že sa stávajú nedostupnými rovnako, ako služby, ktoré poskytujú. Navyše, prístup k obrazom týchto virtuálnych strojov prakticky nie je možný okamžite, pretože ich obsah je uložený na pevných diskoch daného vypnutého fyzického servera.

Staršie servery, podobne ako v súčasnosti využívané v dátovom centre, neumožňujú takú ochranu dát ako napríklad diskové polia, ktoré umožňujú výpadok viacerých diskov súčasne bez straty dát. Preto je potrebné pri týchto serveroch počítať aj s prípadným výpadkom pevných diskov a teda so stratou dát, kedy už nie je možná ich obnova a teda ani obnova virtuálnych strojov. Z tohto dôvodu je odporúčané vykonávať pravidelné zálohovanie dát na iné externé zariadenie.

V prípade, že zlyhaním hardvéru nie sú ovplyvnené pevné disky a server nie je možné zapnúť, je potrebný manuálny zásah a s ním spojená veľmi problematická obnova týchto obrazov. Pri takejto obnove sú k dispozícii dve možnosti:

- Najskôr je pozornosť zameraná na poškodený hardvér a snahu daný problém odstrániť výmenou za nový hardvér. To však nie je vždy jednoduchá záležitosť, pretože dostupnosť niektorých komponentov je dosť obmedzená, pretože objednanie a výmena môže trvať niekoľko dní až týždňov.
- Druhým, vo väčšine prípadov jednoduchším spôsobom, je možnosť získania obrazov z priamo pripojených pevných diskov ich vybratím a následným pripojením k inému serveru, pomocou ktorého dôjde k ich obnove a následnému pripojeniu k inému bežiacemu fyzickému serveru.

V prípade teda, že niektorý z bežiacich serverov plní veľmi dôležitú funkciu, kde je vyžadovaná nepretržitá prevádzka (príkladom môže byť mailový server), obnova takýmito spôsobmi je neprijateľná. Pri takomto kritickom scenári musí byť prevádzka obnovená do niekoľkých minút, ideálne však ihneď a to prevzatím funkcionality iným serverom.

Ďalším negatívom súčasnej situácie je problém so zálohovaním bežiacich virtuálnych strojov spojené s nutnosťou ich vypnutia, čím sú odstavené všetky poskytované služby po dobu prekopírovania obrazu na iné externé zariadenie a tým sa celý proces značne komplikuje.

Z uvedených dôvodov a so súčasným existujúcim nevyužitým vybavením v dátovom centre bolo možné daný problém riešiť lepšie a efektívnejšie.

6.2 POŽIADAVKY NA FUNKČNOSŤ SYSTÉMU

Na základe problémov s ukladaním virtuálnych serverov a uvedených nedostatkov v predchádzajúcej kapitole bolo potrebné situáciu riešiť. S existujúcimi technológiami pre pripojenie diskových polí k serverom sa spája možnosť ukladania obrazov virtuálnych serverov na diskové polia. Tieto obrazy sú chránené bezpečnostnými mechanizmami diskových polí, ktoré sú na vyššej úrovni ako pri serveroch a takisto sú jednoducho prístupné pre všetky servery v SAN sieti, ktorým je umožnený prístup k danému diskovému poľu.

Myšlienka využívania sieťových úložísk pre ukladanie celých virtuálnych serverov umožňuje lepšiu správu, zvýšenie bezpečnosti a v neposlednom rade rapídne znižuje čas pri výpadku obsluhujúceho fyzického servera. V prípade takéhoto hardvérového zlyhania je umožnené jednoduché prevzatie tohto obrazu iným bežiacim fyzickým serverom jednoduchým sprístupnením zväzku, ktorý tento obraz obsahuje. Jeho pripojením a následným spustením je automaticky prevzatá funkcionalita vrátane všetkých poskytovaných služieb, čím je minimalizovaný čas pri výpadku konkrétnych služieb.

V navrhovanom riešení je požadovaná flexibilita pri prideľovaní úložného miesta dostupného pre obrazy virtuálnych strojov na diskových poliach a s tým spojená možnosť jednoduchého zväčšovania v prípade, že by pridelené miesto nepostačovalo. Rovnako je požadovaná možnosť jednoduchého priradenia konkrétneho diskového priestoru ľubovoľnému serveru v SAN sieti a jeho jednoznačná a ľahká identifikácia na základe mena zväzku.

Čo sa týka zálohovania, dôraz sa kladie hlavne na umožnenie zálohovania konkrétnej virtuálnej stanice počas behu. Čo znamená, že nie je potrebné daný server vrátane služieb odstaviť a tým je zabezpečená jeho nepretržitá prevádzka.

Realizácia tejto myšlienky a celkový zamýšľaný stav je možné dosiahnuť viacerými spôsobmi. Preto vznikol problém s výberom toho najvhodnejšieho. Každý spôsob má svoje výhody aj nevýhody, ktoré bolo potrebné preskúmať. Jednotlivé spôsoby, porovnania a s tým spôsobený konkrétny výber je popísaný v kapitole 7.

6.3 PRÍPRAVA PRACOVNÉHO PROSTREDIA

Pre simuláciu a testovanie uvedených požiadaviek bolo potrebné zabezpečiť potrebné hardvérové vybavenie za účelom simulácie SAN siete. Keďže fakulta má dostatok vybavenia, nebol problém poskytnúť vhodné pracovné prostredie. Tvorené bolo modulárnym serverom značky DELL a diskovým poľom značky HP. Ako pracovná stanica bol použitý klasický stolný počítač, ktorý mal k dispozícii dva sieťové adaptéry, čím bolo zabezpečené prepojenie so súkromnou aj verejnou sieťou.

Na úvod celého riešenia bolo potrebné navrhnúť fyzické prepojenie jednotlivých častí takým spôsobom, aby bola zabezpečená komunikácia medzi všetkými zariadeniami a hlavne prístup k manažmentovému rozhraniu, ktoré slúži na správu a konfiguráciu jednotlivých zariadení. Každé zariadenie má niekoľko sieťových rozhraní, kde každé nesie vlastnú jedinečnú IP adresu v sieti. Po zapojení a pridelení IP adries jednotlivým častiam SAN siete, bolo potrebné komunikáciu otestovať.

Po úspešnom otestovaní bol z dôvodu vzájomnej kompatibility, bezpečnosti a jednoduchšieho manažmentu vykonaný upgrade firmvéru najdôležitejších manažmentových častí, ku ktorým patria:

- CMC modul - slúži pre správu celého serverového šasi vrátane všetkých jeho serverov,
- Storage controller – je základom pre správu diskového poľa.

Konfiguráciu je možné vykonať dvomi spôsobmi a to buď v grafickom režime pomocou webového prehliadača alebo pomocou konzoly - zadávaním príkazov do príkazového riadku.

Pre jednoduchšiu a pohodlnejšiu prácu bol vytvorený vzdialený prístup k pracovnej stanici pomocou RDP protokolu, čím bolo umožnené pracovať s celým riešením diaľkovo, bez nutnosti presunu, pretože všetky zariadenia vo vytvorenej SAN sieti boli dostupné pomocou konzoly alebo vzdialeným prístupom.

Ďalším krokom bola príprava jednotlivých fyzických serverov do takej podoby, aby na nich bolo možné vytvárať virtuálne servery. Keďže vzniknuté riešenie malo byť postavené na open-source riešení, ako operačný systém bol zvolený Debian. Tento operačný systém má výhodu v tom, že je to jedna z najstarších distribúcií Linuxu a teda je veľmi stabilný a spoľahlivý. V základnej konfigurácii je veľmi rýchly a taktiež obsahuje celkový potrebný softvér pre prácu s diskovými poľami.

Pre ďalšiu prácu bolo nevyhnutné oboznámiť sa so správou diskového poľa. Medzi jeho základné nastavenia patrí vytvorenie virtuálneho disku, ktorý vzniká spojením viacerých pevných diskov do jedného logického prostredníctvom RAID zapojenia. Pre súčasné potreby bol vytvorený typ RAID 5, ktorý dovoľuje výpadok jedného disku a teda pri menšom počte pevných diskov bez problémov postačuje. Pri použití väčšieho počtu diskov je jednoznačne vhodnejší RAID 6. Ďalším veľmi dôležitým krokom bolo vytvorenie zväzkov a ich pridelenie konkrétnym hostiteľom ako jednotky LUN.

6.3.1 Pridelené zariadenia (hardvér)

Ako už bolo spomenuté, pre simuláciu SAN siete boli zo strany fakulty poskytnuté zariadenia, ktoré sú v skratke popísané nižšie. Súčasťou zadania a taktiež nevyhnutnou podmienkou pre ďalšie riešenie bolo oboznámenie sa s ich správou a základnou konfiguráciou, ktorá ale nebude bližšie popísaná z dôvodu širokého záberu celej problematiky. Ku každému zariadeniu je vydaná niekoľko sto stranová príručka, kde sú popísané všetky nastavenia a funkcie, ktoré jednotlivé zariadenia ponúkajú.

DELL PowerEdge M1000e

Je to modulárny blade (žiletkový) server, čo znamená, že základom je skriňa (šasi), do ktorej sa vkladajú samostatné fyzické servery. Môžu byť polovičnej veľkosti, kedy pri plnej obsadenosti môže obsahovať až šestnásť serverov. Druhým prípadom sú servery plnej veľkosti, ktoré poskytujú viacej priestoru pre hardvér a teda sú spravidla výkonnejšie a samozrejme väčšie, preto je ich možné do skrine uložiť maximálne osem. [24]

Konfigurácii servera nebude venované toľko pozornosti, keďže pre jeho využívanie postačuje nastaviť základnú konfiguráciu, ktorá umožní komunikáciu s ostatnými zariadeniami. Samotná inštalácia serverov prebieha veľmi podobne ako inštalácia osobného počítača. Rozdiel je v tom, že najskôr je potrebné pripojiť sa na konzolu konkrétneho servera a v tejto konzole ponúknuť serveru obraz inštalačného média, z ktorého bude prebiehať inštalácia.

O prepojenie s lokálnou sieťou a s vonkajším svetom sa starajú dva L3 prepínače typu Dell M6220. Obsahujú dvadsať portov, z ktorých je šestnásť vnútorných, zabezpečujúcich prepojenie so všetkými servermi v šasi a zvyšné štyri sú zdieľané externé porty.

Obrázok 10: Dell PowerEdge m1000e
(Podľa: [24])

HP StorageWorks P2000 G2

Je to modulárne diskové pole obsahujúce dva radiče (riadiace jednotky), vďaka ktorým je zabezpečená ich redundancia. V prípade výpadku jedného a samozrejme pri správnej konfigurácii, prevezme riadenie druhý a teda nedochádza k výpadku v komunikácii. Ak bežia oba naraz, prebieha medzi nimi rozkladanie záťaže, čo umožňuje vyššiu priepustnosť dát. Diskové pole je základom pre vznik SAN siete. Poskytuje vysoko-výkonné úložné riešenie, ktoré kombinuje vynikajúci výkon s vysokou dostupnosťou, spoľahlivosťou a taktiež jednoduchou správou. [25]

Pridelený model podporuje pripojenie hostiteľských portov pomocou iSCSI rozhrania a teda sa pripája priamo do ethernetovej siete. Existujú preň aj iné možnosti pripojenia ako napríklad pomocou FC alebo SAS. Šasi je ponúkané v dvoch variantoch a to buď s podporou dvanástich väčších 3.5“ pevných diskov ako možno vidieť v prípade testovaného modelu, alebo potom šasi s podporou dvadsiatich štyroch menších, 2.5“ pevných diskov. K dispozícii bolo osem pevných diskov, kde jeden mal veľkosť 300 GB, čo v skutočnosti predstavuje 2.4 TB celkového úložného priestoru, s ktorým bolo možné ďalej pracovať.

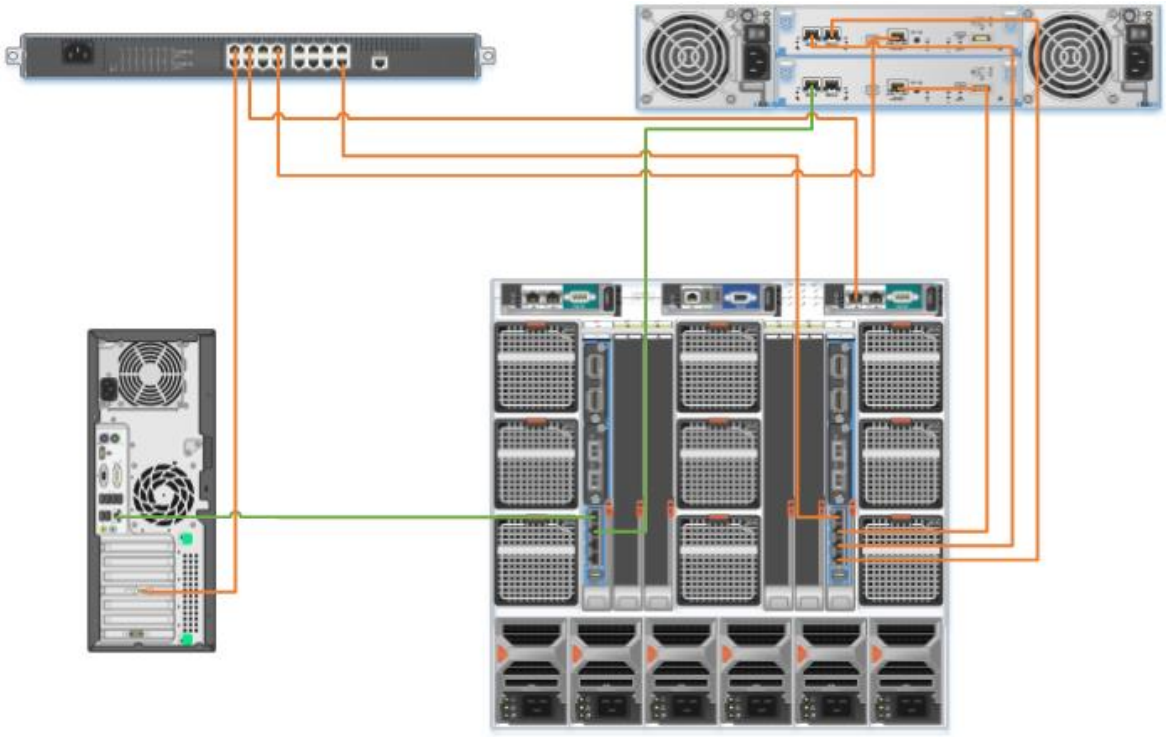
Obrázok 11: HP MSA P2000
(Podľa: [25])

6.3.2 Fyzické zapojenie testovacieho SAN prostredia

Ako už bolo spomenuté, vytvorená SAN sieť obsahuje diskové pole, blade server, osobný počítač a jednoduchý prepínač z dôvodu nedostatku portov pre vnútornú sieť. Samotná výsledná schéma zapojenia je znázornená na Obrázku 12.

Oranžovou farbou je označená vnútorná sieť, ktorá prevažuje, pretože celé riešenie bude nasadené pre použitie na vnútornej sieti. Vonkajšia sieť, zelená farba, v tomto prípade bola zavedená kvôli vzdialenému prístupu.

Ako možno vidieť, diskové pole obsahuje dve riadiace jednotky, kde sú k dispozícii dva sieťové porty pre prenos dát, tretí port je pre manažment diskového poľa. Server obsahuje dva L3 prepínače, ktoré dokážu taktiež pakety smerovať, preto sa dajú použiť aj ako smerovač. Ľavý prepínač je pre vonkajšiu sieť, pravý prepínač pre vnútornú sieť. Taktiež obsahuje CMC modul, ktorý je zapojený do vnútornej siete. Keďže pravý prepínač nemá dostatok portov pre vnútornú sieť, musel byť pridaný prepínač pre rozšírenie portov.



Obrázok 12: schéma fyzického zapojenia

7 ANALÝZA POŽIADAVIEK

Medzi základné požiadavky práce patrí výber vhodného súborového systému porovnávaného z viacerých hľadísk, spôsob ukladania a zálohovania virtuálnych serverov a v neposlednom rade vyriešenie základnej požiadavky, ktorou bol spôsob prideľovania úložného priestoru jednotlivým fyzickým serverom.

7.1 VÝBER SÚBOROVÉHO SYSTÉMU

Keďže je v súčasnosti na výber veľké množstvo rozličných linuxových súborových systémov, nie je možnosť testovania a porovnania všetkých dostupných. Výber bolo potrebné zúžiť na také, ktoré sú v dnešnej dobe najznámejšie, najrozšírenejšie a najstabilnejšie. Preto boli zvolené súborové systémy EXT4, XFS a pre porovnanie k nim bol pridaný jeden z najmladších súborových systémov BtrFS. Z dôvodu ďalšieho nasadenia a používania na fakulte neprichádza do úvahy možnosť experimentovania so súborovými systémami. Je potrebné zvoliť taký, ktorý je overený a ktorý bude plniť celú funkcionalitu potrebnú pre ďalšie používanie aj za cenu nižšieho výkonu. Pričom tento rozdiel nemôže byť priepastný. Posledný menovaný súborový systém je uvedený skôr pre zaujímavosť, pretože v distribúcii Debian je k dispozícii iba jeho experimentálna verzia.

Zvolené súborové systémy bolo potrebné porovnať z viacerých hľadísk. Pozornosť sa zameriavala na vstupno-výstupný výkon, podporu veľkých súborov, schopnosť zväčšovania počas prevádzky a otázkou rýchlosti zotavenia po výpadku systému. Na začiatok je potrebné pripomenúť, že pri práci so súborovými systémami ako napríklad XFS alebo BtrFS je potrebné doinštalovať balík pre ich podporu v systéme.

Vstupno-výstupný výkon súborových systémov

Celé testovanie prebiehalo priamo na fyzickom serveri. Na pevnom disku servera bola vytvorená nová partícia o veľkosti 20 GB, ktorá bola najskôr pripojená k systému s jedným testovaným súborovým systémom. Po vykonaní všetkých testov bola táto partícia pripojená s ďalším súborovým systémom atď. Pre testovanie nebol využitý úložný priestor diskového poľa (iSCSI), pretože výsledky testovania mohli byť ovplyvnené priepustnosťou siete (1 Gb/s) medzi serverom a diskovým poľom. Namerané testy na pripojenom sieťovom úložisku pri čítaní nepresiahli hodnotu 120 MB/s, čo odzrkadľuje približný limit linky. Keďže diskové pole využíva pri jednotlivých operáciách viacero diskov, jeho teoretická priepustnosť by mala byť vyššia ako uvedená nameraná hodnota. Pre objektívnejšie výsledky boli preto použité interné disky servera.

Výkon jednotlivých súborových systémov bol meraný prostredníctvom troch nástrojov – bonnie++, sysbench a dd.

Bonnie++

Je to jednoduchý nástroj, ktorého účelom je porovnanie I/O výkonu súborových systémov. Nachádza sa v linuxových repozitároch a vykonáva komplexnejšie testovanie, ktorého výstupom je rýchlosť zápisu, čítania, vyhľadávania a taktiež zaznamenáva odozvu nameranú pri jednotlivých činnostiach. Samotný test je potrebné vykonať na dátových súboroch väčších ako je množstvo dostupnej operačnej pamäte v systéme, z dôvodu minimalizovania vplyvu ukladania súborov do vyrovnávacej pamäte. Niektoré zdroje uvádzajú, že by to mal byť až desaťnásobok, iné zasa dvojnásobok. V každom prípade musí byť veľkosť rovnaká pri všetkých testoch, aby mohli byť výsledky korektne porovnateľné v rámci viacerých systémov. Bonnie++ poskytuje viacero testov, ktorých vykonanie závisí na

zadaných argumentoch. Znenie príkazu a argumenty, ktoré môže nadobudnúť sú priblížené v nasledovných riadkoch: [26]

```
bonnie++ -d [cesta] -r [veľkosť] -u [používateľ]
d – adresár použitý pre testovanie
r – veľkosť dostupnej operačnej pamäte v systéme
u – užívateľ, pod ktorým sa príkaz vykonáva, nemôže byť ROOT
s – udáva veľkosť vytváraných dát, ak nie je zadaná, použije
sa dvojnásobok operačnej pamäte
n – udáva počet vytvorených súborov pre testovanie
m – pridá popis pre daný výstup
x – udáva počet opakovaní celého testu
```

Po ukončení testov sa zobrazia na obrazovke výsledky, ktorých výpis nie je veľmi prehľadný. Pre lepšie porozumenie stačí prekopírovať spodnú časť výpisu do predpripraveného skriptu, ktorý vygeneruje HTML stránku s prehľadnejším výpisom jednotlivých hodnôt. Slúži na to príkaz:

```
echo "[výsledok]" | bon_csv2html > /tmp/bonnie.html
výsledok – namerané výsledky v teste
```

Pre každý vybraný súborový systém boli vykonané dva testy. Keďže testovaný server mal k dispozícii 2 GB RAM, testy bolo potrebné vykonať s využitím minimálne 4 GB dát. Namerané hodnoty je možné vidieť v nasledovných tabuľkách.

XFS

Tabuľka 1: Výsledky 4GB testu XFS súborového systému

Verzia 1.96		Sekvenčný zápis						Sekvenčné čítanie				Náhodné vyhľadávanie	
	Veľkosť	Po znakoch		Blok dát		Prepísanie		Po znakoch		Blok dát			
		K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	/sec	% CPU
HM3	4G	489	99	87346	10	40968	7	1724	99	104979	7	704.4	11
	odozva	20407us		22293us		267ms		7359us		18449us		71330us	

Tabuľka 2: Výsledky 8GB testu XFS súborového systému

Verzia 1.96		Sekvenčný zápis						Sekvenčné čítanie				Náhodné vyhľadávanie	
	Veľkosť	Po znakoch		Blok dát		Prepísanie		Po znakoch		Blok dát			
		K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	/sec	% CPU
HM3	8G	489	99	85890	10	40116	6	1755	98	94396	6	511.1	10
	odozva	16513us		223ms		253ms		15324us		30929us		266ms	

EXT4

Tabuľka 3: Výsledky 4GB testu EXT4 súborového systému

Verzia 1.96		Sekvenčný zápis						Sekvenčné čítanie				Náhodné vyhľadávanie	
	Veľkosť	Po znakoch		Blok dát		Prepísanie		Po znakoch		Blok dát			
		K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	/sec	% CPU
HM3	4G	308	99	84580	13	41073	6	2053	98	104020	6	966.5	11
	odozva	26390us		197ms		375ms		7066us		128ms		123ms	

Tabuľka 4: Výsledky 8GB testu EXT4 súborového systému

Verzia 1.96		Sekvenčný zápis						Sekvenčné čítanie				Náhodné vyhľadávanie	
	Veľkosť	Po znakoch		Blok dát		Prepísanie		Po znakoch		Blok dát			
		K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	/sec	% CPU
HM3	8G	308	99	82111	13	38992	6	2054	98	91380	6	615.1	8
	odozva	26523us		265ms		390ms		8325us		93630us		132ms	

BtrFS

Tabuľka 5: Výsledky 4GB testu BtrFS súborového systému

Verzia 1.96		Sekvenčný zápis						Sekvenčné čítanie				Náhodné vyhľadávanie	
	Veľkosť	Po znakoch		Blok dát		Prepísanie		Po znakoch		Blok dát			
		K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	/sec	% CPU
HM3	4G	262	99	81161	11	46467	9	1890	95	187431	20	903.5	16
	Odozva	30760us		412ms		369ms		10276us		124ms		116ms	

Tabuľka 6: Výsledky 8GB testu BtrFS súborového systému

Verzia 1.96		Sekvenčný zápis						Sekvenčné čítanie				Náhodné vyhľadávanie	
	Veľkosť	Po znakoch		Blok dát		Prepísanie		Po znakoch		Blok dát			
		K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	/sec	% CPU
HM3	8G	265	99	83031	10	46481	9	1849	95	171981	20	707.6	15
	odozva	30557us		448ms		335ms		12230us		181ms		151ms	

Z výsledkov je možné vidieť veľmi podobný I/O výkon súborových systémov XFS a EXT4, z ktorých má mierne navrch XFS v sekvenčnom čítaní aj zápise. No na druhej strane stráca pri náhodnom vyhľadávaní. Smerodajný údaj, ktorý si treba všímať je zápis a čítanie bloku dát, ktorý je v tabuľke uvádzaný v KB za sekundu. Výkon súborového systému BtrFS je pri zapisovaní podobný zvyšným dvom, ale rýchlosť čítania je v porovnaní s oboma oveľa vyššia. Pokles rýchlosti pri teste na väčšom množstve dát je pravdepodobne spôsobený vyrovnávacou pamäťou. V prípade použitia ešte väčšieho množstva testovaných dát by sa výsledky ustálili v okolí jednej hodnoty. Keďže ale všetky testy prebiehali za rovnakých podmienok, je možné povedať, že približne odzrkadľujú rýchlosť jednotlivých súborových systémov. Pre overenie dosiahnutých výsledkov s testovacím nástrojom bonnie++ bol použitý ďalší nástroj s názvom SysBench.

SysBench

SysBench je modulárny, multiplatformový testovací nástroj pre hodnotenie rôznych dôležitých systémových parametrov. Ponúka viacero testov ako napríklad test CPU, operačnej pamäte, databázový test, test vstupno-výstupných operácií atď. Pre potreby testovania súborového systému bol použitý práve test I/O operácií, ktorého použitie je nasledovné: [27]

- Najskôr je potrebné vytvoriť sadu súborov, na ktorých bude prebiehať testovanie. Aby výsledky testov neboli ovplyvnené čiastkovým ukladaním do vyrovnávacej pamäte, celková veľkosť musí byť väčšia, ako je veľkosť dostupnej operačnej pamäte.

```
sysbench --test=fileio --file-total-size=5G prepare
```

- Po príprave súborov je potrebné vykonať samotný test. Typ testu určuje parameter *file-test-mode*, dĺžku testu v sekundách parameter *max-time.*

```
sysbench --test=fileio --file-total-size=5G --file-test-mode=rndrw --max-time=300 run
rndrw – náhodné čítanie aj zápis
rndrd – náhodné čítanie
rndwr – náhodný zápis
seqrd – sekvenčné čítanie
swqwr – sekvenčný zápis
```

- Po vykonaní potrebných testov je možné všetky vytvorené súbory jednoducho zmazať príkazom.

```
sysbench --test=fileio --file-total-size=5G cleanup
```

Pre každý vybraný súborový systém boli vykonané tri testy – náhodné čítanie a zapisovanie, sekvenčné čítanie a sekvenčné zapisovanie. Vzhľadom na to, že veľkosť dostupnej operačnej pamäte bolo 2 GB, všetky testy boli vykonané na sumárnej veľkosti súborov 5 GB.

Tabuľka 7: SysBench test náhodného čítania a zápisu

	Náhodné čítanie a náhodný zápis				
	čítanie v MB	zápis v MB	celkovo v GB	rýchlosť v MB/s	vykonaných požiadaviek / sek.
XFS	446,25	297,5	0,74375	2,4791	158,66
EXT4	897,19	598,12	1,4603	4,9843	319
BtrFS	93,75	62,5	0,15625	1,3955	89,32

Tabuľka 8: SysBench test sekvenčného čítania a sekvenčného zápisu

	Sekvenčný zápis		**Sekvenčné čítanie**	
	zápis v MB/s	vykonaných požiadaviek / sek.	čítanie v MB/s	vykonaných požiadaviek / sek.
XFS	82,166	5258,61	85,286	5458,29
EXT4	75,087	4805,56	80,733	5166,94
BtrFS	80,016	5121,01	141,93	9083,26

Výsledky testu SysBench ponúkajú mierne neprehľadný výstup, preto boli spísané do dvoch jednoduchých tabuliek. V tabuľke náhodného čítania a náhodného zápisu je možné vidieť, že najrýchlejší súborový systém v tomto hodnotení je EXT4, za ním nasleduje XFS a nakoniec BtrFS. Najrýchlejší EXT4 mal pred XFS dvojnásobný náskok a pred BtrFS dokonca takmer trojnásobný. Pri teste sekvenčného zápisu sú výsledky veľmi podobné, ale nastáva tu opačná situácia, kedy EXT4 je pomalší ako XFS aj BtrFS. Pri sekvenčnom čítaní nastáva podobná situácia ako pri nástroji bonnie++ a teda potvrdzuje namerané výsledky, ktoré ukazujú jednoznačnú prevahu rýchlosti sekvenčného čítania súborového systému BtrFS v porovnaní s XFS aj EXT4. Rozdiel medzi sekvenčným zápisom alebo čítaním je dosť priepastný v porovnaní s náhodným, pretože pri prístupe k náhodným blokom dát je nutné nájdenie daného bloku a až následne je k nemu umožnený prístup, čo trvá určitý čas. Pri sekvenčnom prístupe bloky nasledujú v poradí za sebou, nie je potrebné hľadanie blokov, a preto je tento prístup oveľa rýchlejší (nevzniká tu žiadna odozva).

Namerané rozdiely v hodnotách medzi nástrojom bonnie++ a SysBench vznikajú, pretože testovanie pomocou SysBench trvalo dlhšiu dobu na rovnakých dátach, čím bola menej zapojená vyrovnávacia pamäť, a preto sú výsledky presnejšie. Pri testovaní väčšieho množstva dát by nástroj bonnie++ dosiahol podobné hodnoty.

dd

Vykonáva základný test, ktorý neponúka veľa možností. Po spustení vytvára súbor o zadanej veľkosti v čo možno najkratšom čase. Výstupom je rýchlosť a čas trvania zápisu. Podobne ako pri nástroji bonnie++ je potrebné zabezpečiť, aby veľkosť vytváraného súboru bola väčšia ako množstvo dostupnej pamäte v systéme. V opačnom prípade budú výsledné rýchlosti reprezentovať rýchlosť I/O operácií operačnej pamäte. Najväčšiu výhodu predstavuje to, že je súčasťou väčšiny linuxových distribúcií. Preto nie je potreba žiadnej dodatočnej inštalácie, čo je veľmi výhodné pri niektorých uzavretých distribúciách. Želaná veľkosť výsledného súboru sa dosiahne vynásobením veľkosti bloku s počtom opakovaní. Test je možné spustiť nasledujúcim príkazom:

```
dd if=/dev/zero of=/mnt/HM bs=1M count=5000
if – predstavuje vstupný súbor
of – predstavuje výstupný súbor
```

Podobne ako v predchádzajúcich testoch bolo potrebné príkazom vytvoriť taký veľký súbor, ktorý bude niekoľkokrát väčší ako veľkosť dostupnej operačnej pamäte. Preto boli pre každý súborový systém vykonané tri testy 4 GB, 8 GB a 16 GB. Výsledky boli nasledovné:

Tabuľka 9: Príkaz dd pre porovnanie rýchlosti zápisu súborových systémov

	4GB		**8GB**		**16GB**	
	čas v sek.	rýchlosť v MB/s	čas v sek.	rýchlosť v MB/s	čas v sek.	rýchlosť v MB/s
XFS	43,4865	96,5	92,0944	91,1	192,422	87,2
EXT4	44,7989	93,6	95,35	88	198,441	84,5
BtrFS	45,7794	91,6	92,5064	90,7	193,889	86,5

Ako možno z tabuľky vidieť, výsledky tretieho testu na sekvenčný zápis iba potvrdzujú predchádzajúce namerané výsledky. Na základe nich je možné konštatovať, že rýchlosť zápisu jednotlivých súborových systémov je veľmi porovnateľná a rozdiely sú minimálne. Z toho dôvodu sekvenčný zápis nebude rozhodujúcim kritériom pre výber vhodného súborového systému.

Z výsledkov všetkých troch testov boli zistené nasledovné skutočnosti. Výsledky náhodného čítania a zápisu naznačujú, že súborový systém EXT4 sa najlepšie hodí na prácu s malými súbormi. Pri práci s veľkými súbormi je naopak spomedzi všetkých troch súborových systémov najpomalší aj keď v porovnaní s XFS sú výsledky čítania aj zápisu veľmi podobné a ich rozdiel je veľmi malý. Najmenšie rozdiely sú medzi sekvenčným zápisom jednotlivých súborových systémov, kde najlepší výsledok dosiahol XFS súborový systém. V čítaní bol XFS na druhom mieste a najrýchlejší bol posledný testovaný BtrFS. Nárast oproti ďalším dvom je takmer 80%, čo už nie je zanedbateľný rozdiel. V prípade častejšej práce s malými súbormi je možné jednoznačne odporučiť EXT4 súborový systém, ktorý na prijateľnej úrovni zvláda aj prácu s veľkými súbormi. V prípade využívania súborového systému prevažne na ukladanie veľkých súborov, sa ako stabilný a výkonný javí

XFS súborový systém, ktorý dosahoval vyrovnané výsledky. Pokiaľ by sa ale jednalo o rýchlosť čítania veľkých súborov, na túto činnosť by bol najvhodnejší BtrFS.

Kritérium výkonu však nie je pre nasadenie na fakulte najpodstatnejšie, preto je potrebné preskúmať aj ďalšie ponúkané vlastnosti porovnávaných súborových systémov.

Ďalšie požiadavky pre výber súborového systému

- **Podpora veľkých súborov** - všetky testované súborové systémy umožňujú uloženie súboru o veľkosti 16 TB (XFS a BtrFS dokonca až niekoľko EB). Bez ďalšieho dokazovania je možné konštatovať, že je to postačujúca veľkosť. Pre porovnanie je možné uviesť, že celé diskové pole, na ktorom prebiehali testy malo kapacitu 2.4TB. To znamená, že by bolo potrebné takmer sedem takýchto polí na uloženie jedného súboru. Požiadavka veľkosti súborov pri porovnávaní jednotlivých súborových systémov vôbec nezaváži.
- **Schopnosť zmeny veľkosti** – veľkosť súborových systémov EXT4 a BtrFS je možné ľubovoľne meniť oboma smermi. Čo znamená, že použité zväzky je možné zväčšiť a neskôr v prípade potreby aj zmenšiť o požadovanú veľkosť. Súborový systém XFS má v tomto veľkú nevýhodu, pretože ho možno iba zväčšiť. Tento hendikep je dosť závažný, pretože so zväzkami sa bude pravdepodobne veľmi často manipulovať oboma smermi.
- **Rýchlosť zotavenia po výpadku** – hodnotenie tohto kritéria je dosť náročné, pretože pri pokuse o nasimulovanie rovnakého poškodenia dát u všetkých troch súborových systémov nastáva problém. Je možné povedať, že všetky tieto súborové systémy si vedia poradiť s bežnými výpadkami. Pre každý z nich existuje nástroj pre kontrolu a opravu konzistencie súborového systému.

  XFS - `xfs_check, xfs_repair`
  EXT4 - `fsck.ext4`
  BtrFS - `btrfsck`

  EXT4 a XFS majú miernu výhodu, pretože na rozdiel od BtrFS používajú **žurnál**, ktorého prínosy sú priblížené v kapitole 2.1.

Zhrnutie

Z porovnávaných kritérií je možné konštatovať, že EXT4 je vyladený súborový systém, ktorý sa hodí na prácu s menšími súbormi. Rovnako nezaostáva ani v spracovávaní veľkých súborov. Jeho veľkosť je možné meniť oboma smermi z čoho môže ťažiť v porovnaní s XFS. Súborový systém XFS je rýchlejší v sekvenčnom čítaní aj zápise a pre nasadenie v serverových riešeniach je vhodnejší. Pokiaľ je ale požadované kritérium zmenšovania súborového systému, vhodnejší je EXT4. Súborový systém BtrFS prekvapil hlavne v rýchlosti sekvenčného čítania, pri ktorom mal veľký náskok v porovnaní s XFS aj EXT4. Rovnako ako pri EXT4 je možné meniť jeho veľkosť oboma smermi, ale keďže je stále používaný v experimentálnej verzii, momentálne nasadenie nie je odporúčané.

Každý súborový systém má svoje výhody aj nevýhody, a preto jeho výber závisí hlavne na konkrétnom využití daného servera. Pre použitie v dátovom centre na fakulte sú vhodné súborové systémy XFS a EXT4. V prípade potreby zmenšovania zväzkov je odporúčané využitie EXT4, v opačnom prípade je vhodnejší XFS.

7.2 VÝBER ULOŽENIA VIRTUÁLNEHO STROJA

Spôsob uloženia virtuálneho stroja je taktiež veľmi dôležitý z hľadiska ďalšieho riešenia. Po pridelení hardvérových prostriedkov virtuálnemu stroju je potrebné vybrať kam

a akým spôsobom sa bude daný stroj ukladať. Ako už bolo viackrát spomenuté, pre ukladanie sa využíva priestor poskytnutý diskovým poľom. Otázkou ale zostáva, akým spôsobom bude tento virtuálny stroj na pridelený zväzok uložený. Do úvahy je možné brať dva spôsoby:

- Uloženie do jedného súboru a následné uloženie na pridelený LV. Vytvorený súbor nadobúda príponu IMG.
- Obsah virtuálneho stroja vrátane všetkých jeho súborov, sa ukladá priamo na logický zväzok, na ktorom je priamo po inštalácii systému automaticky vytvorená tabuľka partícií.

Od tohto výberu závisí neskoršie nasadenie. Medzi hlavné kritériá pri výbere vhodnejšieho spôsobu patrí rýchlosť I/O operácií, náročnosť administrácie a zálohovania jednotlivých virtuálnych strojov.

Rýchlosť vstupno-výstupných operácií

Je prvým porovnávacím kritériom. Na testovanie rýchlosti vykonaných operácií existuje množstvo dostupných metód. V ďalších riadkoch sú priblížené dve známe metódy: [26]

Bonnie++ - testy boli vykonané na dvoch virtuálnych strojoch, kde každý používal iný spôsob uloženia. Keďže oba mali k dispozícii 2 GB RAM, testy bolo potrebné vykonať s využitím minimálne 4 GB dát. Pre objektívnejšie porovnanie boli vykonané v každom systéme dva testy – o veľkosti 4 a 8 GB. Namerané hodnoty je možné vidieť v nasledovných tabuľkách:

Tabuľka 10: Výsledky testu o veľkosti 4GB na VM uloženej v súbore

Verzia 1.96		Sekvenčný zápis						Sekvenčné čítanie				Náhodné vyhľadávanie	
	Veľkosť	Po znakoch		Blok dát		Prepísanie		Po znakoch		Blok dát			
		K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	/sec	% CPU
debianIMG_LV1	4G	627	99	96174	20	45609	8	2440	96	120312	11	905.1	22
	Odozva	31552us		693ms		422ms		30633us		69948us		179ms	

Tabuľka 11: Výsledky testu o veľkosti 8GB na VM uloženej v súbore

Verzia 1.96		Sekvenčný zápis						Sekvenčné čítanie				Náhodné vyhľadávanie	
	Veľkosť	Po znakoch		Blok dát		Prepísanie		Po znakoch		Blok dát			
		K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	/sec	% CPU
debianIMG_LV1	8G	629	99	93814	13	45532	8	2319	94	114748	11	801.9	22
	Odozva	12794us		781ms		416ms		70852us		84389us		155ms	

Tabuľka 12: Výsledky testu o veľkosti 4GB na VM uloženej priamo na LV

Verzia 1.96		Sekvenčný zápis						Sekvenčné čítanie				Náhodné vyhľadávanie	
	Veľkosť	Po znakoch		Blok dát		Prepísanie		Po znakoch		Blok dát			
		K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	/sec	% CPU
debianLV_VG2	4G	634	99	98541	14	45640	8	2503	96	126545	12	994.3	24
	Odozva	12682us		406ms		450ms		29098us		74315us		605ms	

Tabuľka 13: Výsledky testu o veľkosti 8GB na VM uloženej priamo na LV

Verzia 1.96		Sekvenčný zápis						Sekvenčné čítanie				Náhodné vyhľadávanie	
	Veľkosť	Po znakoch		Blok dát		Prepísanie		Po znakoch		Blok dát			
		K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	/sec	% CPU
debianLV_VG2	8G	627	97	93553	13	44132	8	2440	94	116879	11	717.1	19
	Odozva	25379us		622ms		792ms		44337us		112ms		1231ms	

Z výsledkov testov je možné pozorovať, že rozdiel medzi čítaním a zápisom na jednotlivých virtuálnych strojoch je veľmi malý. Pre overenie dosiahnutých výsledkov bola použitá ďalšia metóda.

Príkaz dd - pre testovacie účely bol vytvorený súbor o veľkosti 8 GB priamo v súborovom systéme, ktorý bol 4x väčší ako veľkosť operačnej pamäte. Výsledky sú zhrnuté do nasledovnej tabuľky:

Tabuľka 14: Porovnanie rýchlostí testov pomocou príkazu DD

	Čas vytvorenia	**Rýchlosť zápisu**
IMG súbor	83,5772 s	98 MB/s
Log. zväzok	82,8784 s	98,8 MB/s

Ako možno vidieť, odchýlka je minimálna a teda iba potvrdzuje výsledky namerané pomocou komplexnejšieho nástroja bonnie++. Výsledky sú veľmi podobné, z čoho možno konštatovať, že vzniknutý rozdiel je zanedbateľný a teda kritérium rýchlosti nebude rozhodujúce.

Keďže virtuálne stroje, na ktorých prebiehali testy sa nachádzajú na sieťovom diskovom poli, výsledky veľmi záležia od jeho aktuálneho zaťaženia a od počtu požiadaviek, ktoré v danom čase spracováva. Preto tieto výsledky slúžia ako približný údaj, ktorý sa nedá považovať za štatisticky úplne presný a navyše časy a rýchlosti zápisu/čítania sa môžu pri neskorších testoch mierne líšiť. Uvedené testy prebiehali za rovnakých podmienok, preto približne odzrkadľujú rýchlosť jednotlivých porovnávaných spôsobov.

Náročnosť administrácie a zálohovania

Ďalším kritériom pri porovnávaní je otázka **administrácie a zálohovania**. Z hľadiska ďalšieho nasadenia sú obe tieto požiadavky veľmi dôležité. Slovom administrácia sa rozumie všetko od vytvorenia, zotavenia po výpadku, možnosť migrácie až po zrušenie virtuálneho stroja.

Čo sa týka vytvárania, pri uložení virtuálneho stroja do súboru sa postupuje bežným spôsobom. Najskôr musí byť pripravený adresár, do ktorého bude vytváraný obraz uložený. Je jedno, či tento adresár bude na lokálnom disku alebo to bude sieťový logický zväzok pripojený do stromovej štruktúry hostiteľského systému. Jednoducho sa vyberie cieľ uloženia pre virtuálny stroj. Po pridelení prostriedkov a nainštalovaní systému môžeme overiť existenciu vzniku tohto súboru vo zvolenom adresári.

Pri ukladaní virtuálneho stroja priamo na predpripravený logický zväzok je potrebné daný zväzok pripojiť do LVM manažéra ako ďalší úložný priestor. Bez tohto kroku nebude možné, ako cieľ vybrať kompletný obsah vybraného logického zväzku. Výhodou je, že tento zväzok nie je potrebné pripájať k súborovému systému hostiteľa, čo má ale za následok ťažší prístup k jeho súborom. S tým je ďalej spojená nevýhoda pri kopírovaní virtuálneho stroja

na iný zväzok alebo iný fyzický server, kde je súbor a teda celý virtuálny stroj možné preniesť jednoduchým prekopírovaním. Pri prenášaní celého zväzku vrátane jeho partícii nastáva problém a je možné konštatovať, že v tomto je súbor oveľa praktickejší.

V prípade prerušenia spojenia s úložiskom je nutné virtuálny stroj spustiť nanovo v oboch testovaných prípadoch. Pokiaľ sa nedá jeho práca ukončiť korektne, bude nutné virtuálny stroj vypnúť natvrdo. Tomuto scenáru sa bohužiaľ nedá vyhnúť, pretože pri strate spojenia s úložiskom sa vyskytnú chyby čítania a funkcionalita je zachovaná len na úrovni operačnej pamäte. Po opätovnom naviazaní spojenia je v prvom rade potrebné zbaviť sa LVM vstupno-výstupných chýb, ktoré sa priebežne zobrazujú na obrazovke. Potom v prípade súboru znovu pripojiť zväzok, na ktorom bol uložený a až následne je umožnené spustenie virtuálneho stroja. V druhom prípade nie je pripojenie zväzku nutné.

Rovnako na konkrétnom LV sa môže nachádzať iba jeden virtuálny stroj, ale pri ukladaní do súboru môže byť na logickom zväzku týchto súborov až do vyčerpania jeho kapacity. Ako je v neskoršej kapitole spomenuté, migrácia za behu počíta s uložením virtuálneho stroja do súboru, pretože pred začatím samotného procesu je potrebné vytvoriť na cieľovom hostiteľovi prázdny súbor o rovnakej veľkosti a s rovnakou cestou. Z tohto zistenia vyplýva, virtuálny stroj, ktorý nie je uložený v súbore túto migráciu neumožňuje. Je to veľká nevýhoda vzhľadom na to, že táto funkcionalita bude pravdepodobne často využívaná.

Pri vykonávaní **zálohy** vypnutého virtuálneho stroja je v prípade súboru, ako už bolo spomínané, možné jeho jednoduché prekopírovanie na iný zväzok. Čo sa týka vykonania zálohy počas behu systému, znovu má navrch súbor, pretože pri vytvorení snímky (viď kapitola 8.3) daného zväzku je súbor znovu možné jednoducho prekopírovať na iný zväzok. V prípade zväzku sú obe možnosti značne zložitejšie. V prípade prevzatia virtuálneho stroja iným fyzickým serverom vychádza o niečo lepšie zväzok, pretože po jeho sprístupnení nie je potreba pripojenia k systému, stačí ho rovno vybrať pri vytváraní nového virtuálneho stroja.

Ďalšou požiadavkou bola možnosť zväčšenia dostupného úložiska virtuálneho stroja. Pri oboch prípadoch sa dá aplikovať rovnako zložitá metóda, ktorá je bližšie popísaná v kapitole 8.4.

Po analýze uvedených kritérií z hľadiska nasadenia a využívania na fakulte je možné konštatovať, že práca so samostatným súborom, ktorý obsahuje celý virtuálny stroj je vo väčšine hľadísk jednoduchšia a rýchlejšia. Hoci práca so zväzkom ako úložiskom ponúka tiež niekoľko výhod, na čele s rýchlejšími vstupno-výstupnými operáciami, je možné aj napriek tomu odporučiť ukladanie virtuálnych strojov do súboru. V ďalšom riešení budú všetky virtuálne stroje vytvárané týmto spôsobom.

7.3 VÝBER SPÔSOBU PRIDEĽOVANIA ÚLOŽNÉHO MIESTA

Navrhovaným riešením je využitie úložného priestoru poskytovaného diskovým poľom. Diskové pole pracuje na princípe vytvárania a prideľovania zväzkov konkrétnym hostiteľom. Tu sa hneď vynára otázka, akým spôsobom tieto zväzky prideľovať. Riešenie prideľovania úložného miesta ponúka viacero možností.

Prvou možnosťou, ktorú bolo treba preskúmať, bol **prístup viacerých hostiteľov k jednému sieťovému zväzku**. Riešenie by mohlo vyzerať tak, že by bol poskytnutý jeden veľký zväzok všetkým hostiteľom, ktorí by tento priestor zdieľali. Po skúmaní je možné povedať, že tento zväzok si môžu pripojiť viacerí hostitelia. Avšak pri zápise nastáva problém, pretože protokol iSCSI neposkytuje uzamykanie súborov, čo by mohlo spôsobiť

vážne poškodenie dát. Problém s prístupom viacerých hostiteľov k jednému iSCSI zväzku riešia klastrové súborové systémy, ktoré umožňujú takýto zápis. Medzi najznámejšie patria napr. GFS a OCFS2. Predstava fungovania s jedným zväzkom bola taká, že na tomto zväzku by mal každý hostiteľ vlastný pracovný adresár. Nevýhodou ale je, že adresáre nie sú od seba oddelené, takže v prípade problému s ukladaním dát môže nastať situácia, kedy virtuálny stroj začne pridelený adresár nekontrolovateľne zapĺňať. Tým, že adresáre nie sú oddelené, bude hostiteľ brať úložný priestor všetkým ostatným. To môže mať za následok spotrebovanie celej úložnej kapacity, čím by nastal závažný problém, pokiaľ by táto skutočnosť nebola zaznamenaná včas. Rovnako by bol problém s organizovaním celého priestoru pre viacerých hostiteľov, ktorý by bolo lepšie oddeliť z dôvodu bezpečnosti dát.

Klastrové súborové systémy neboli reálne nasadené, pretože na základe uvedených nedostatkov je možné konštatovať, že idea spoločného zdieľaného priestoru pre všetkých hostiteľov nebude vhodným spôsobom pre nasadenie do prevádzky.

Druhou možnosťou je **pridelenie jedného zväzku každému hostiteľovi**. V porovnaní s predchádzajúcim spôsobom odstraňuje niekoľko nedostatkov, ale taktiež sa s týmto princípom vynára aj niekoľko problémov. Keďže vytváranie a prideľovanie zväzkov sa celé vykonáva cez manažment diskového poľa, nie je žiadúce, aby sa každá prípadná zmena na zväzku diala práve cez diskové pole. Naopak, túto manipuláciu so zväzkami je potrebné zabezpečiť práve z hostiteľského operačného systému, pretože sa počíta s vytvorením skriptov pre konkrétne činnosti. Príkladom môže byť zväčšenie zväzku alebo pridelenie zväzku inému hostiteľovi. Každý hostiteľ má k dispozícii v systéme iba svoj zväzok, a preto v prípade pridelenia inému hostiteľovi je potrebné pristúpiť k diskovému poľu. Čiastočne by sa tento problém dal riešiť pridelením všetkých zväzkov každému hostiteľovi, ale v prípade pripájania do systému a pri názvoch jednotlivých diskov by nastala veľmi neprehľadná situácia, ktorá by veľmi komplikovala prácu. Taktiež po pripojení úložiska do systému tento zväzok nadobúda určitý názov vo formáte sdXY, kde X označuje poradie disku a Y na ňom vytvorenú partíciu. Tento názov je automaticky pridelený systémom, čo má za následok nemožnosť vhodného pomenovania zväzku pre identifikáciu virtuálneho servera, ktorému patrí. Dokonca, ak nastane výpadok poľa alebo reštart iSCSI služby, tento zväzok môže dostať iný názov, čo spôsobí značné problémy. Zväzok bude potrebné pripojiť nanovo pod iným názvom a ak nejaká iná služba pracovala so zväzkom pod starým názvom, preruší sa a bude hlásiť chyby. Ďalšou nevýhodou je obmedzený počet vytváraných zväzkov, z čoho rovnako vyplýva obmedzený počet hostiteľov, ktorým môžu byť zväzky pridelené.

Je veľmi ťažko predstaviteľné nasadenie tohto spôsobu z hľadiska administrácie a celkového používania, preto bolo potrebné nájsť nový spôsob prideľovania úložného miesta.

Treťou možnosťou je umožnenie správy úložného priestoru poskytnutého diskovým poľom **pomocou LVM manažéra**, ktorého základné vlastnosti a prínosy sú bližšie popísané v kapitole 3. Správca logických zväzkov (LVM manažér) odstraňuje všetky nedostatky predchádzajúcich dvoch možností a javí sa ako najlepšie riešenie pre celkové nasadenie a ďalšie využívanie na fakulte.

LVM (Logical Volume Management) [28] predstavuje abstraktnú vrstvu medzi operačným systémom a úložným priestorom. Pre jeho správu sú dostupné rovnako ako grafické nástroje, tak aj príkazový riadok. Keďže v budúcnosti sa na niektorých fakultných serveroch nepočíta s grafickým prostredím, všetky nastavenia boli vykonané prostredníctvom príkazového riadku.

Veľkou výhodou tohto prístupu je minimálna potreba ďalšej konfigurácie z pohľadu diskového poľa. Stačí prideliť všetkým hostiteľom jeden veľký zväzok, ktorý bude následne pomocou LVM manažéra rozdelený na menšie časti, nazývané tiež logické zväzky, ktoré si budú môcť jednotliví hostitelia pripojiť do svojho systému. Pristúpiť k diskovému poľu bude potrebné iba v prípade nedostatku priestoru na poskytnutom zväzku. Problém s miestom môže byť vyriešený buď zväčšením daného zväzku alebo poskytnutím ďalšieho zväzku, ktorý sa pridá do existujúcej skupiny zväzkov. Ak však bude poskytnutý dostatočne veľký priestor pre uloženie všetkých virtuálnych serverov a dát, k diskovému poľu sa bude pristupovať iba v prípade potreby pridelenia zväzku novému hostiteľovi. Všetky konfiguračné nastavenia vrátane zmeny veľkosti logických zväzkov, ich sprístupnenie iným hostiteľom v prípade vzniknutého problému, vymazanie zväzkov a mnohé ďalšie nastavenia budú vykonávané prostredníctvom LVM manažéra hostiteľského systému. V prípade, že jeden hostiteľ vykoná zmeny v štruktúre LVM, ostatní hostitelia túto zmenu spozorujú hneď, ako po tejto zmene nanovo prehľadajú štruktúru LVM. Viditeľnosť zmien na ostatných hostiteľoch je zabezpečená, pretože LVM metadáta sú uložené rovnako ako v systéme, tak aj priamo na diskoch, čím je umožnená jednoduchá obnova LVM štruktúry v prípade novej inštalácie systému. Taktiež odstraňuje problém s názvami pridelených logických zväzkov, pretože pri ich vytváraní je potrebné zadať parameter s názvom vytváraného zväzku. Ten je možné zadať podľa servera, ku ktorému patrí. Rovnako má tieto názvy pod svojou správou LVM manažér, ktorý fyzické zväzky skladá do skupiny zväzkov na základe ich UUID čísla. Preto v prípade výpadku iSCSI služby a možnému následnému premenovaniu týchto fyzických zväzkov nedochádza k rozpadu vytvorenej skupiny a teda ani k žiadnemu výpadku z pohľadu užívateľov alebo aplikácií.

Pre ľahšiu predstavu a pochopenie princípu vytvárania logických zväzkov v rámci LVM pre jednotlivé hostiteľské servery slúži nasledujúci obrázok:

Obrázok 13: Princíp vytvárania zväzkov pomocou LVM
(Podľa: [28])

Na obrázku sú znázornené dva sieťové zväzky poskytnuté diskovým poľom z dôvodu lepšej predstavy spájania rôznych pridelených zväzkov do jednej skupiny. Na začiatku vytvárania je poskytnutý jeden zväzok, ktorý je pridelený všetkým dostupným hostiteľom a druhý je možné kedykoľvek doplniť do existujúcej skupiny v prípade potreby. Veľkosť

sieťového disku by mala byť vhodne zvolená, pokiaľ má byť minimalizovaný zásah do konfigurácie diskového poľa. Z tohto zväzku je možné vytvoriť LVM fyzický zväzok. Rovnako je možné sieťový zväzok rozdeliť na viacero partícií a z každej potom vytvoriť fyzické zväzky, ktoré budú základom pre vytvorenie jednej, prípadne viacerých skupín zväzkov. Po vytvorení skupiny zväzkov je z nej možné vytvárať LVM logické zväzky s vlastným názvom, ktoré si môžu pripájať konkrétni hostitelia. V prípade potreby rozšírenia logického zväzku je potrebné mať v jeho skupine zväzkov dostatok voľného miesta. V opačnom prípade je nutné zväčšiť priestor skupiny zväzkov o ďalší LVM fyzický zväzok. S LVM manažérom je potrebné zaobchádzať opatrne a uvážene, pretože pri strate dát je ich obnova veľmi náročná.

7.4 RIEŠENIE ZÁLOHOVANIA

Pravidelné zálohovanie virtuálnych serverov a dát je veľmi dôležité z dôvodu ich zachovania v prípade rôznych katastrofických scenárov, pri ktorých je možné o tieto dôležité dáta prísť. Záloha by mala byť riešená na iné diskové pole, ideálne umiestnenom na inom mieste (v inej serverovni) ako zdrojové pole. Možnosť migrácie virtuálnych serverov je tiež veľmi užitočná napríklad pri vyvažovaní výkonu fyzických serverov. V nasledujúcich riadkoch budú priblížené zistené skutočnosti v tejto problematike.

Zálohu zväzkov, ktoré nesú obrazy konkrétnych virtuálnych serverov je možné vykonať viacerými spôsobmi. Pri prvom spôsobe je nutné ukončiť prácu s bežiacim virtuálnym serverom, pretože pri súčasnom zapisovaní a čítaní tých istých dát by mohlo dôjsť k ich strate. Po vypnutí je umožnené jeho jednoduché prekopírovanie na iný logický zväzok. Táto realizácia je jednoduchá, ale za cenu odstavenia všetkých služieb, čo nie je vždy umožnené. Druhým, efektívnejším spôsobom, je vytvorenie snímky daného zväzku. Toto riešenie sa ponúka v prípade, že bežiaci systém uložený na danom zväzku nie je možné pozastaviť ani ukončiť.

Snímka zväzku je špeciálny typ zväzku, ktorý predstavuje všetky dáta, ktoré obsahoval zdrojový zväzok v čase vykonania snímky. Je to v podstate „zmrazený“ logický zväzok v konkrétnom čase. Snímka na začiatku neobsahuje žiadne dáta, iba pevné odkazy na existujúce dáta v zdrojovom zväzku. Zapĺňa sa postupným vykonávaním zmien na zdrojovom zväzku, kedy LVM automaticky klonuje dáta, na ktorých boli vykonané zmeny. Pôvodné dáta ukladá na zväzok snímky a zmenené zostávajú na zdrojovom zväzku. Z tohto dôvodu nie je potrebné snímke priradiť rovnakú veľkosť akú má zdrojový zväzok, ale stačí iba takú časť, ktorá s dostatočnou rezervou pokryje vykonávané zmeny. Po vytvorení snímky a jej pripojenia k systému sa otvára možnosť bezpečného kopírovania na iný logický zväzok. Nevýhodou je, že počas existencie snímky je možné badať značné spomalenie systému v prípade jeho zaťaženia. Vytvoreniu snímky predchádza zmrazenie súborového systému a ukončenie prebiehajúcich zápisov. Snímka by preto nemala byť vytváraná v čase prebiehajúcej inštalácie dôležitého softvéru, prípadne v čase iného zásahu do systému, pretože by pri prerušení tejto činnosti mohla byť narušená systémová integrita. Odporúčaný čas vytvorenia je v čase najmenšieho zaťaženia virtuálneho servera alebo najlepšie pri vypnutom serveri, kedy sa urobí snímka zväzku a následne je možné server znovu zapnúť – výpadok služieb by bol v takomto prípade minimálny. Je to najbezpečnejší spôsob.

Najdôležitejšie fakty o LVM snímkach, ktoré je potrebné vedieť pri ich vytváraní je možné zhrnúť do nasledovných bodov:

- Veľkosť zväzku, na ktorom je vytváraná snímka nemusí byt rovnako veľký ako zdrojový zväzok.

- LVM snímky používajú technológiu copy-on-write, ktorej princíp spočíva vo vytváraní kópii rovnakých dát iba v prípade, keď je vydaný pokyn pre zápis dát. V opačnom prípade sa využíva iba odkaz na už existujúce dáta.
- LVM snímka potrebuje mať takú veľkosť, aby dokázala zastrešiť všetky zmeny vykonané na zdrojovom zväzku počas jeho existencie.
- Snímka je v podstate presná kopia stromu „inode“ zdrojového zväzku v čase keď bola vytvorená.

8 IMPLEMENTÁCIA RIEŠENIA

Na úvod je potrebné oboznámiť sa s prvým krokom, ktorým je pripojenie hostiteľov k úložnému priestoru diskového poľa. Bez tohto kroku nebude možné pokračovať v ďalšom riešení. Ďalej je potrebné priblížiť prácu s LVM, vytvorenie snímky, migráciu virtuálnych serverov, princíp iSCSI Multipathingu a v závere prípadné riešenie havarijných situácií.

8.1 PRIPOJENIE K DISKOVÉMU POĽU

ISCSI spojenie s diskovým poľom je zabezpečené prostredníctvom linuxového iniciátora **open-iscsi**. Je to vysokovýkonná, prenosovo nezávislá, multiplatformová implementácia iSCSI protokolu. [29] Open-iscsi iniciátor je dostupný pod rovnakým názvom v linuxových repozitároch. Inštaluje sa pomocou príkazu:

```
apt-get install open-iscsi
```

V rámci inštalačného balíka sa nainštaluje aj konfiguračná utilita **iscsiadm**, pomocou ktorej je možné vykonávať príkazy pre správu celého spojenia vrátane jeho vytvorenia. Pre nadviazanie spojenia je najskôr potrebné „objavenie“ úložiska konkrétnym hostiteľom. Slúži na to príkaz:

```
iscsiadm -m discovery -t st -p portal
m – mód príkazu, v tomto prípade sa jedná o objavenie
t – typ, musí byť st=sendtargets
p – požadovaný portal, pokiaľ sa nezadá port, berie sa
štandardný 3260
```

Tento príkaz slúži na vzájomné objavenie, pretože po jeho zadaní bude daný hostiteľ viditeľný na diskovom poli. Bez zadania tohto príkazu nebude diskové pole o hostiteľovi vedieť, a preto mu nebude možné prideliť žiadny zväzok. Ďalší krok spočíva v pridelení zväzku danému hostiteľovi. Aby hostiteľ videl pridelený zväzok, je jeho povinnosťou pripojiť sa na toto diskové pole a vytvoriť s ním reláciu pomocou príkazu:

```
iscsiadm -m node -T iqn_pola -p portal --login
```

V prípade vykonaných zmien na iných hostiteľoch napr. pri vytvorení nových logických zväzkov z existujúcej skupiny alebo zmien na diskovom poli, napr. pridelenie ďalšieho zväzku konkrétnemu serveru, je potrebné na danom hostiteľovi obnoviť spojenie a tým umožniť zobrazenie vytvorených zmien pomocou jedného z príkazov:

```
iscsiadm -m session -R
iscsiadm -m node -R
```

Ďalšie užitočné príkazy, s ktorými je potrebné sa oboznámiť pri ďalšom používaní príkazu iscsiadm:

- výpis pripojených targetov:

```
iscsiadm -m session list
```

- výpis informácii o pripojenom targete:

```
iscsiadm -m session -P1
P – určuje dĺžku výpisu, môže byť 1-3
```

- odpojenie od diskového poľa:

```
iscsiadm -m node -T iqn_pola -p portal --logout
```

Veľmi dôležitý poznatok, na ktorý treba upozorniť je, že pri základnom nastavení je potrebné pripájať diskové pole k systému ručne. Pre **automatické pripojenie k diskovému poľu** a vytvorenie konkrétnych relácií hneď po štarte systému je potrebné vykonať nasledujúce úpravy:

- Upraviť konfiguračný súbor */etc/iscsi/iscsid.conf/* takým spôsobom, že v sekcii *Startup settings* je nutné odkomentovať riadok *node.startup* = *automatic* a riadok manuálneho pripojenia naopak zakomentovať.
- Rovnako v adresári */etc/iscsi/nodes/* je potrebné upraviť súbor *default* pre každú reláciu, ktorá sa má spúšťať po štarte systému prepísaním riadku *node.startup* z *manual* na *automatic.*

Po vytvorení relácie je možné na pridelenom zväzku vytvoriť súborový systém a následne ho pripojiť k hostiteľskému systému. Po tomto kroku s ním bude možné pracovať ako s lokálnym úložiskom. Postup je nasledovný:

1. V hostiteľskom systéme je potrebné preskúmať dostupné zväzky. Je to možné vykonať pomocou príkazu:

   ```
   fdisk -l
   ```

2. Po identifikácii prideleného zväzku diskového poľa je na ňom nutné vytvoriť súborový systém a následne ho pripojiť k hostiteľskému systému. Pre vytvorenie súborového systému existuje viacero príkazov, výber konkrétneho závisí od jeho typu. Napríklad:

   ```
   mkfs.xfs /dev/sdX
   mkfs.ext4 /dev/sdX
   X – dosadiť pridelené písmeno zväzku
   ```

3. Po vytvorení súborového systému je možné pripojiť tento zväzok k systému pomocou nasledovného príkazu:

   ```
   mount /dev/sdX /adresar
   ```

Keďže ručné pripájanie by značne komplikovalo prácu, je možné upraviť súbor */etc/fstab* tak, aby sa daný zväzok pripájal automaticky pri štarte systému. Je potrebné do nového riadku pridať názov zväzku, jeho prípojný bod v systéme, použitý súborový systém a z parametrov sú najdôležitejšie:

- auto – znamená, že zariadenie bude automaticky pripojené k systému po štarte alebo tiež po zadaní príkazu *mount –a,*
- _netdev – parameter, ktorý hovorí, že je to sieťové zariadenie a teda bude pripojené až po nabehnutí siete.

Pokiaľ daný zväzok nie je vytvorený cez LVM, je potrebné do */etc/fstab* uviesť UUID číslo zväzku namiesto blokového názvu, pretože po reštartovaní iSCSI komunikácie sa môže blokový názov zväzku zmeniť. UUID číslo zostáva rovnaké.

8.1.1 Autentifikácia iSCSI komunikácie

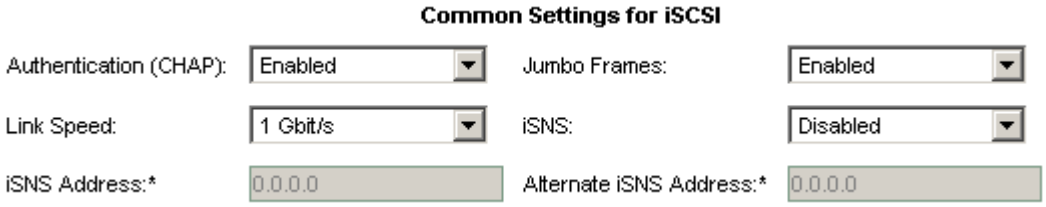
Pre zabezpečenie pripojenia konkrétnych iniciátorov k targetu je možné použiť autentifikáciu prostredníctvom CHAP protokolu, ktorý sa postará o overenie ešte pred začatím samotnej komunikácie. Je to základná úroveň zabezpečenia, založená na zdieľaní kľúča medzi oboma účastníkmi. Sú známe dva typy CHAP autentifikácie:

- **jednosmerná** – pri tejto úrovni je overovaný iba iniciátor voči targetu,

- **obojsmerná** – pri tejto úrovni je iniciátor aj target overovaný vzájomne medzi sebou.

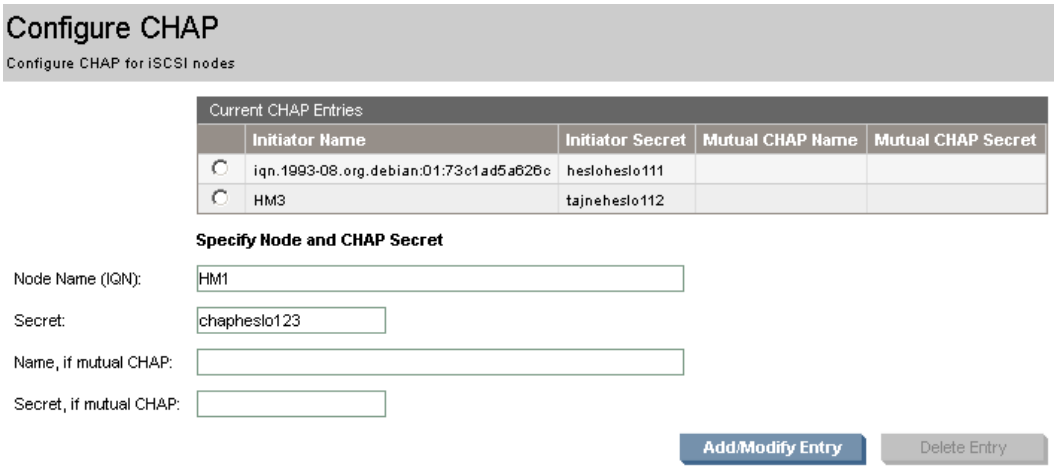
Základný postup pri konfigurácii jednosmernej CHAP autentifikácie:

1. Najskôr je potrebné povoliť CHAP autentifikáciu na diskovom poli v nastaveniach hostiteľských rozhraní.



Obrázok 14: Povolenie CHAP autentifikácie na diskovom poli

2. Pre každého hostiteľa nastaviť kľúč, ktorého veľkosť musí byť minimálne 12 znakov.



Obrázok 15: Konfigurácia CHAP autentifikácie pre konkrétneho hostiteľa

3. U konkrétneho hostiteľa (iniciátora) v konfiguračnom súbore */etc/iscsi/iscsid.conf* v sekcii CHAP povoliť autentifikáciu a objavenie targetu odkomentovaním nasledovných riadkov. Namiesto parametrov *username* a *password* je potrebné dosadiť IQN názov a kľúč rovnaký ako bol zadaný na diskovom poli.

```
node.session.auth.authmethod = CHAP
node.session.auth.username = username
node.session.auth.password = password

discovery.sendtargets.auth.authmethod = CHAP
discovery.sendtargets.auth.username = username
discovery.sendtargets.auth.password = password
```

4. Opätovné spustenie fázy objavenia diskového poľa.

```
iscsiadm -m discovery -t st -p IPadresaPola
```

5. Reštart služby *open-iscsi.*

```
/etc/init.d/open-iscsi restart
```

8.2 PRÁCA S LVM

Predtým ako bude možné vytvárať LVM objekty je potrebné vytvoriť jeden alebo viacero sieťových zväzkov na diskovom poli. Z testovania bolo zistené, že najlepší systém prideľovania spočíva vo vytvorení jedného veľkého zväzku, ktorý sa pridelí všetkým hostiteľom. V prípade potreby rozšírenia je stále k dispozícii možnosť vytvorenia zväzku z ďalšieho diskového poľa. Tento priestor bude poskytnutý LVM manažérovi. [30]

8.2.1 Vytvorenie LVM objektov

Podľa potreby je najskôr možné pridelený zväzok rozdeliť na menšie časti - partície. Z každej tejto časti bude ďalej vytvorený LVM fyzický zväzok. Rovnako je možné ponechať zväzok bez rozdelenia a z celého tohto priestoru vytvoriť jeden veľký LVM fyzický zväzok.

1. Vytvorenie LVM fyzických zväzkov – z prideleného sieťového zväzku alebo z jeho časti sa vytvárajú LVM fyzické zväzky pomocou príkazu:

   ```
   pvcreate /dev/sdXY
   X - je konkrétny pevný disk a Y na ňom vytvorená partícia
   ```

2. Vytvorenie skupiny zväzkov – vytvára sa spojením požadovaného počtu vytvorených fyzických zväzkov. Pre vytvorenie skupiny s názvom „pole01“, ktorá bude obsahovať fyz. zväzky „/dev/sdX1“ a „/dev/sdX2“ je určený príkaz:

   ```
   vgcreate pole01 /dev/sdX1 /dev/sdX2
   ```

3. Vytvorenie logických zväzkov – z vytvorenej skupiny je možné vytvárať požadované logické zväzky. Pre vytvorenie zväzku o veľkosti 20 GB s názvom „mail“ zo skupiny „pole01“ je určený príkaz:

   ```
   lvcreate -L 20G -n mail pole01
   L – veľkosť vytváraného zväzku
   n – názov vytváraného zväzku
   ```

4. Vytvorenie súborového systému na konkrétnom LV – ešte predtým ako je pripojený vytvorený LV k operačnému systému, je na ňom potrebné vytvoriť súborový systém. Príkazy, ktoré na to slúžia sa líšia na základe zvoleného súborového systému:

   ```
   mkfs.xfs /dev/pole01/mail
   mkfs.ext4 /dev/pole01/mail
   mkfs.btrfs /dev/pole01/mail
   ```

5. Pripojenie vytvoreného LV k systému – pripájanie v Linuxe sa realizuje pomocou pripojenia zväzku do určitého adresára a to buď do existujúceho alebo novovytvoreného. Základný príkaz pre pripojenie vyzerá takto:

   ```
   mount /dev/pole01/mail /mnt/VM
   ```

8.2.2 Aktivácia LVM objektov

Keďže štruktúra LVM je uložená aj priamo na diskoch, vytvorené LVM objekty vidia všetci hostitelia, ktorí majú povolený prístup k danému zväzku diskového poľa. Preto si môže ľubovoľný fyzický server aktivovať zväzok vytvorený iným serverom. Aktivácia konkrétnych LVM objektov sa deje pomocou príkazov:

```
vgchange -ay /dev/pole01
lvchange -ay /dev/pole01/mail
a – parameter určujúci aktiváciu, môže mať dve hodnoty –
y(yes)/n(no)
```

Zväzky je možné aktivovať v systéme podľa potreby, avšak pokiaľ je potrebný reštart fyzického servera alebo nastane jeho výpadok, vybrané zväzky sa neaktivujú.

Automatická aktivácia skupín zväzkov alebo konkrétnych logických zväzkov pri štarte systému neprebieha automaticky, pretože inicializačný skript pre iSCSI sa spúšťa neskôr ako skript pre LVM, v ktorom je štandardne nastavená aktivácia všetkých existujúcich skupín. Keďže sa tento LVM skript spúšťa skôr, služba iSCSI a teda ani spojenie s diskovým poľom ešte nebolo nadviazané, nedochádza k aktivácii žiadneho sieťového LVM zväzku. Keďže oba skripty majú pre svoje spustenie nadväznosti na iné skripty, nie je možné do tejto nadväznosti zasahovať, pretože by mohlo dôjsť k slučke. Rovnako nie je možné zmeniť poradie spúšťania, pretože pri aktualizácii by mohlo dôjsť k opätovnej zmene v poradí, preto by to bolo len dočasným riešením.

Situáciu rieši skript /etc/*rc.local*, ktorý je spustený po nabehnutí všetkých systémových služieb a teda až po pripojení k diskovému poľu. Používa sa pre spustenie vlastných služieb. Do tohto skriptu boli pridané dva riadky:

```
/etc/init.d/lvm2 start
mount -a
```

Keďže LVM aktivuje pri spúšťaní všetky skupiny zväzkov vrátane ich logických zväzkov, aktiváciu konkrétnych zväzkov je možné zabezpečiť nastavením filtra v konfiguračnom súbore */etc/lvm/lvm.conf*. Aby sa daný zväzok následne pripojil k systému je potrebné upraviť súbor */etc/fstab.*

Vypisovanie jednotlivých zväzkov pre aktiváciu nie je veľmi praktické. LVM ponúka možnosť vytvoriť LVM tag [31], ktorý slúži pre zoskupenie LVM objektov rovnakého typu. Je možné zoskupiť niekoľko logických zväzkov a tento vytvorený tag priradiť konkrétnemu hostiteľovi. Vytvoriť jednotlivé tagy je možné na ľubovoľnom hostiteľovi, pretože tag je pridaný do metadát logického zväzku:

```
lvchange --addtag @HM1 /dev/pole01/mail
@HM1 – názov vytváraného tagu
```

Názov tagu sa prideľuje podľa názvu hostiteľského servera. Vytvorený požadovaný tag je možné vložiť do filtra zväzkov konfiguračného súboru */etc/lvm/lvm.conf* konkrétneho hostiteľa. Po reštarte systému budú aktivované iba zväzky zoskupené vo vytvorenom tagu.

```
volume_list = [ "@HM1" ]
```

Akonáhle je aktivovaný tento filter, novovytvorené zväzky je potrebné zakaždým priradiť do tagu povoleného na danom serveri, v opačnom prípade nebude daný zväzok vytvorený. Príkaz:

```
lvcreate --addtag @$(uname –n) –L20G –n mail pole01
```

Odobratie konkrétneho zväzku z vytvoreného tagu je možné vykonať pomocou nasledovného príkazu:

```
lvchange --deltag @HM1 /dev/pole01/mail
```

Pre výpis LVM objektov konkrétneho tagu je možné použiť príkaz:

```
lvs @HM1
```

Pre výpis všetkých existujúcich tagov je používaný príkaz:

```
lvs –o lv_name,lv_tags
```

8.2.3 Výpis a odstránenie LVM objektov

Pre výpis konfigurácie vytvorených súčastí sa používa príkaz *display*. Základ príkazu zostáva pri všetkých rovnaký, mení sa iba jeho začiatok, ktorý určuje o aký typ výpisu sa jedná. Preto sa rozlišujú tieto príkazy:

```
pvdisplay, vgdisplay, lvdisplay
```

Tieto príkazy vypíšu všetky vytvorené súčasti daného typu. Pokiaľ sa požaduje výpis jedného konkrétneho, pridáva sa za príkaz jeho názov:

```
lvdisplay /dev/pole01/mail
```

Tiež je možné použiť jednoduchý skrátený výpis pomocou príkazov:

```
pvs, vgs, lvs
```

Jednoduchý súhrnný výpis pre všetky súčasti vytvorené pomocou LVM manažéra je možné vyvolať pomocou príkazu:

```
lvmdiskscan
```

Podobne ako pri pridávaní alebo rozširovaní je možné danú súčasť aj odstrániť. Slúži na to príkaz *remove*. V prípade odstraňovania logického zväzku je v prvom rade potrebné brať na vedomie, že odstránením zväzku sa tiež odstránia všetky na ňom uložené dáta. Preto je potrebné najskôr vykonať zálohu dát a taktiež je potrebné daný zväzok odpojiť. Poznáme tieto príkazy pre odstránenie:

```
pvremove /dev/sdbX
vgremove /dev/pole01
lvremove /dev/pole01/mail
```

8.2.4 Zmena veľkosti LVM objektov

Veľkou výhodou LVM manažéra je možnosť kedykoľvek zväčšiť skupinu zväzkov alebo konkrétny logický zväzok, ktorému dochádza voľné miesto. Rovnako sa otvára možnosť pre zmenšenie zväzkov. Pri rozširovaní konkrétnej súčasti sa využíva príkaz *extend*, pri zmenšení príkaz *reduce* Podobne ako pri výpise, začiatok príkazu určuje o aký typ rozširovania alebo zmenšenia sa jedná.

VGextend - v prípade potreby rozšírenia vytvorenej skupiny (jej aktuálna veľkosť nie je postačujúca) musí byť k dispozícii minimálne jeden voľný fyzický zväzok, o ktorého veľkosť sa bude daná skupina rozširovať. Parametre tohto príkazu sú meno skupiny, ktorá bude rozšírená a fyzický zväzok potrebný na rozšírenie.

```
vgextend pole01 /dev/sdX3
vgreduce pole01 /dev/sdX3
```

LVextend - pri rozširovaní logického zväzku je potrebné mať dostatok voľného miesta v skupine, z ktorej sa ide tento priestor odobrať. Parametre príkazu sú konkrétny zväzok a veľkosť, o ktorú bude zväzok zväčšený:

```
lvextend -L +10G /dev/pole01/mail
```

Podobne funguje príkaz pre zmenšenie. Zmenšovať zväzok je najlepšie pri odpojenom logickom zväzku. Rovnako je nutné, aby bol dostatočný voľný priestor pre zmenšenie. V opačnom prípade by mohlo dôjsť k strate dát.

```
lvreduce -L -10G /dev/pole01/mail
```

Uvedené príkazy však **nezmenia veľkosť použitého súborového systému**. Preto je potrebné zadať podľa použitého súborového systému jeden z nasledujúcich príkazov pre zväčšenie a využitie voľnej kapacity logického zväzku:

Tabuľka 15: Rozšírenie niektorých súborových systémov

EXT3,4	**umount** */mnt/VM* **resize2fs** */dev/pole01/mail* **mount** */dev/pole01/mail /mnt/VM*
XFS	**xfs_growfs** */mnt/VM*
BtrFS	**btrfsctl** -r +[velkost] */mnt/VM*

Pri zmenšovaní súborového systému je potrebné postupovať opatrnejšie. Ako už bolo spomenuté, XFS súborový systém nie je možné zmenšiť. Z vybraných súborových systémov je možné zmenšiť iba EXT4 a BtrFS. Pri zmenšovaní zväzku, na ktorom je EXT4 súborový systém, je potrebné najskôr zmenšiť súborový systém a až potom veľkosť zväzku. V opačnom prípade bude zmenšenie súborového systému hlásiť chyby rozdielnej veľkosti. Najskôr je však potrebné zväzok zo systému odpojiť, skontrolovať konzistenciu súborového systému a až potom je možné zmeniť veľkosť. Súborový systém BtrFS má výhodu v tom, že daný zväzok nie je potrebné odpojiť, zmenšovanie je možné počas pripojenia v systéme. Postup pri zmenšovaní oboch vybraných súborových systémoch je znázornený v nasledujúcej tabuľke.

Tabuľka 16: Zmenšenie niektorých súborových systémov

EXT3,4	**umount** */mnt/VM* **e2fsck** –f */dev/pole01/mail* **resize2fs** */dev/pole01/mail [celkovavelkost]* **lvreduce** *–L -[velkost] /dev/pole/mail* **mount** */dev/pole01/mail /mnt/VM*
BtrFS	**btrfsctl** -r [velkost] */mnt/VM*

8.2.5 Konfiguračný súbor LVM

Základným konfiguračným súborom LVM manažéra je súbor *lvm.*conf, ktorý sa nachádza v adresári */etc/lvm/*. Obsahuje informácie a nastavenia pre konkrétne zariadenia ako napríklad výber adresára pre vytváranie skupín zväzkov, umiestnenie logovacieho súboru, nastavenie zálohovania a umiestnenia metadát, nastavenia týkajúce sa aktivácie konkrétnych logických zväzkov a mnoho ďalších. V aktivačnej časti je umožnené povolenie pripojenia iba konkrétnych logických zväzkov. Po zrušení komentára riadka *volume_list = ["pole01/mail"]* je aktivovaný filter, ktorý umožní systémovú aktiváciu iba tých zväzkov alebo skupín, ktoré sú uvedené v hranatých zátvorkách. Pri pokuse o aktiváciu zväzku, ktorý nie je povolený, sa na obrazovke zobrazí chybové hlásenie. Vďaka tomuto nastaveniu je zachovaná bezpečnosť zväzkov a ich dát, pretože pokiaľ je daný zväzok povolený iba

v jednom systéme, ostatní si ho nemôžu pripojiť. Nevýhodou je nutnosť editácie konfiguračného súboru zakaždým, keď je potrebné povoliť zväzok iného hostiteľa.

8.3 VYTVORENIE SNÍMKY ZVÄZKU

Pre vytvorenie snímky konkrétneho logického zväzku je potrebné vytvoriť ďalší logický zväzok pomocou príkazu:

```
lvcreate -L 2G -s -n nazov_snimky /dev/pole01/mail
s - parameter, ktorý určuje, že daný LV bude snímkou
n - názov vytváranej snímky
```

Ešte pred samotným vytvorením je potrebné premyslieť koľko priestoru daný snímok potrebuje. Je kritické, aby bolo pridelené dostatok miesta pre vykonávané zmeny na danom zväzku až pokiaľ nie je snímok odpojený.

Veľkosť alokovaného priestoru a teda v podstate veľkosť snímky tvoria vykonané zmeny na zdrojovom logickom zväzku počas existencie snímky. Aktuálnu veľkosť je možné vidieť v riadku "*Allocated to snapshot*" po zadaní príkazu:

```
lvdisplay /dev/pole01/mail
```

Vytvorený snímok je potrebné pripojiť k súborovému systému. Je potrebné, aby na zdrojovom zväzku bol predvytvorený súborový systém. V opačnom prípade vzniká problém a snímku nebude možné pripojiť k systému a teda ani zazálohovať jej obsah.

Ak je na ňom vytvorený XFS súborový systém, tak pri pripájaní vzniká problém - každý pripájaný LV s XFS súborovým systémom ma unikátny identifikátor označovaný ako UUID. Ak je vytváraný snímok určitého LV, tak tento snímok dostáva presne rovnaké číslo UUID ako jeho zdroj (originál). Preto vzniká problém, kedy systém hlási ich zhodu a neumožní pripojenie ďalšieho LV s rovnakým UUID na jednom serveri. Tento problém je možné vyriešiť dvomi spôsobmi:

- Použitie parametra(option) *nouuid* v príkaze:

  ```
  mount -o nouuid /dev/pole01/backup /mnt/backup
  ```

- Ďalšou možnosťou je zmena identifikátora UUID pre konkrétny XFS súborový systém vytvorenej snímky, kedy je pomocou nasledujúceho príkazu vygenerované nové UUID:

  ```
  xfs_admin -U generate /dev/pole01/snapshot
  ```

Po pripojení snímky k stromovej štruktúre systému môžeme bezpečne vykonať zálohu virtuálneho servera, prípadne celého logického zväzku jednoduchým prekopírovaním na iný zväzok, prípadne na úplne iné zariadenie.

Po vykonanej zálohe je potrebné tento snímok odpojiť a príslušný LV zmazať pomocou príkazu:

```
lvremove /dev/pole01/backup
```

V prípade potreby vrátenia originálneho obrazu do stavu v momente vykonania snímky, napr. pri testovaní nejakého nového softvérového balíka, kedy výsledky neboli uspokojivé, sa otvára možnosť vrátenia týchto zmien pomocou príkazu:

```
lvconvert --merge /dev/pole01/snapshot
```

Pred vykonaním príkazu je potrebné odpojenie oboch zväzkov (originál aj snímku). Ak sa na origináli nachádza koreňový súborový systém, je tiež potrebné po zadaní príkazu, deaktivovať a znovu aktivovať zdrojový logický zväzok, aby sa zmeny uplatnili.

8.4 ZMENA VEĽKOSTI OBRAZU VIRTUÁLNEHO STROJA

Každému virtuálnemu stroju je na začiatku pridelený úložný priestor, ktorého potrebnú veľkosť nie je ľahké odhadnúť. Preto môže počas jeho existencie dôjsť k situácii, kedy je potrebné tento priestor zväčšiť. Zmenu veľkosti obrazu a teda jemu dostupného úložiska môžeme vykonať viacerými spôsobmi.

Prvou možnosťou je zväčšenie a následné narábanie s konkrétnym obrazom, bez nutnosti vytvárania ďalšieho, prázdneho obrazu. Pri tomto spôsobe je ale potrebné použiť externý nástroj na úpravu partícií, pretože manipuláciou s partíciami na bežiacom virtuálnom stroji môže dôjsť k strate dát. Postup je nasledovný:

1. Zväčšenie veľkosti obrazu súboru virtuálneho stroja.

   ```
   Qemu-img resize mail.img +10G
   ```

2. Zväčšenie súborového systému virtuálneho stroja pomocou vybraného externého nástroja. V tomto prípade bol použitý Gparted. Po spustení Gparted treba postupovať týmto spôsobom - vymazať swap partíciu, rozšíriť želaný zväzok a vytvoriť novú swap partíciu. Pre vykonanie zmien je potrebné tento proces uložiť a následne vykonať.
3. Spustenie virtuálneho stroja a následná aktivácia novovytvorenej swap partície.

   ```
   Swapon /dev/swap
   ```

Druhou možnosťou je použitie balíka libguestfs-tools, ktorý je dostupný v linuxových repozitároch. Je to súbor nástrojov, ktorý umožňuje prístup a modifikáciu obrazu disku konkrétneho virtuálneho stroja. Ešte predtým je potrebné vytvoriť prázdny obraz o požadovanej novej veľkosti, čím vzniká problém. Pretože je potrebné mať na danom logickom zväzku dostatok miesta pre pôvodný, ale aj nový obraz. Následne je možné prekopírovanie obsahu pôvodného obrazu do nového s pridanými parametrami pre zväčšenie alebo úpravu konkrétnej partície. Postup:

1. inštalácia potrebného balíka:

   ```
   apt-get install libguestfs-tools
   ```

2. v prípade výskytu chyby pri inštalácii je potrebné zadať nasledujúci príkaz:

   ```
   update-guestfs-appliance
   ```

3. vytvorenie nového prázdneho súboru o potrebnej novej veľkosti, napr. pomocou príkazu:

   ```
   truncate -s 30G novy.img
   ```

4. použitie nástroja virt-resize, ktorý na základe použitých parametrov pôvodný obraz prekopíruje do nového:

   ```
   virt-resize mail.img novy.img --resize
   /dev/sda2=+200M --expand /dev/sda1
   resize – zmena veľkosti o požadovanú hodnotu
   expand – rozšírenie do prípustnej veľkosti
   ```

Ako už bolo spomenuté, nevýhoda tohto spôsobu je, že na zväčšenie konkrétneho obrazu je potrebné vytvoriť nový súbor o potrebnej veľkosti. V prípade, že na konkrétnom zväzku nie je dostatok voľného miesta, je potrebné tento zväzok zväčšiť alebo zväčšovaný obraz musí byť prekopírovaný na iný zväzok, kde bude túto zmenu možné vykonať. Toto presúvanie značne komplikuje celý postup zväčšovania.

Z uvedených spôsobov je možné konštatovať, že ak je problém s voľným miestom na zväzku daného zväčšovaného virtuálneho stroja, vychádza lepšie prvá možnosť. V opačnom prípade je lepšie použiť druhú možnosť.

8.5 MIGRÁCIA BEŽIACICH VIRTUÁLNYCH SERVEROV

Pre migráciu je odporúčané použiť zdieľané sieťové úložisko, ktoré urýchľuje tento proces, pretože nie je potreba presúvania žiadnych dát. Keďže v súčasnom riešení má každý virtuálny stroj k dispozícii svoj vlastný úložný priestor na diskovom poli, je potrebné prenos riešiť mimo tohto spoločného úložiska. To spôsobí, že daný virtuálny stroj musí byť kompletne presunutý na iný logický zväzok.

Live migrácia je spojená s viacerými problémami. V prvom rade je nutné aby na cieľovom logickom zväzku daného servera bolo dostatok úložného priestoru, v opačnom prípade je potrebné požadovaný priestor vytvoriť. Po alokovaní potrebného priestoru na cieľovom hostiteľovi je potrebné vytvoriť nový prázdny diskový obraz s rovnakým názvom, s rovnakou veľkosťou a s rovnakým umiestnením (zhodnou cestou k súboru) aké má obraz zdrojového virtuálneho stroja. Bez tejto prípravy nebude možné daný obraz migrovať a úloha skončí s chybovým hlásením. Prázdny diskový obraz je možné vytvoriť pomocou príkazu:

```
qemu-img create [-f fmt] nazovsuboru [veľkosť]
fmt - je formát súboru, môže byť napr. raw, qcow2 atď.
```

Tiež je potrebné v nastaveniach virtuálneho disku prenášaného virtuálneho stroja nastaviť v záložke „nastavenia výkonu“ mód vyrovnávajúcej pamäte na „žiadny“. V opačnom prípade migrácia nebude umožnená a skončí s chybovým hlásením, v ktorom upozorňuje na možné poškodenie prenášaných dát uložených vo vyrovnávajúcej pamäti.

Proces migrácie je možné spustiť nasledovným príkazom. Najdôležitejšie parametre sú *–live*, ktorý hovorí o tom, že pôjde o live migráciu a parameter *–copy-storage-all*, ktorý musí byť zadaný, keď sa jedná o nezdieľané úložisko. Uvedenú IP adresu je potrebné nahradiť IP adresou cieľového hostiteľa.

```
migrate --live VM1 qemu+ssh://IPadresa/system --
copy-storage-all --verbose --persistent --
undefinesource tcp://IPadresa
IPadresa – IP adresa cieľového fyzického servera
tcp – parameter, ktorý je potrebné zadať v prípade, že proces
migrácie zlyhá s chybou prekladu adresy
```

8.6 ISCSI MULTIPATHING

Multipathing je technika, ktorá umožňuje využitie viacerých fyzických ciest pre prenos dát medzi hostiteľom a externým úložiskom (diskovým poľom). V prípade výpadku jednej cesty je nepretržite k dispozícii niekoľko ďalších, čo znamená, že spojenie je redundantné. Pre vytvorenie redundantných ciest je potrebné, aby diskové pole obsahovalo dve aktívne iSCSI riadiace jednotky, z ktorých každá musí mať inú IP adresu a ideálne aj vlastný prepínač. Týmto je zabezpečená redundancia riadiacich jednotiek a zároveň aj

prepínačov. Rovnako je potrebné, aby server obsahoval minimálne dva sieťové adaptéry. V prípade, že riadiaca jednotka má aktívne obe dátové rozhrania, celková dátová prevádzka sa rozkladá medzi obe cesty - dochádza k vyvažovaniu výkonu. Typ vyvažovania závisí od konkrétnych nastavení. Ak celá riadiaca jednotka vypadne, okamžite bude nahradená.

Pri využívaní LVM je nutná konfigurácia multipathingu pre dané zariadenie ešte pred implementáciu samotného LVM. Pre rozpoznanie multipathing zariadení, ako fyzických zariadení je potrebná úprava konfiguračného súboru */etc/lvm/lvm.conf* takým spôsobom, aby bol prístup k úložisku zabezpečený prostredníctvom multipathing vrstvy. [32]

Základom pre využívanie redundantných ciest v operačnom systéme hostiteľa je inštalácia balíka *multipath-tools*, ktorý je dostupný v linuxových repozitároch pomocou príkazu:

```
apt-get install multipath-tools
```

V prípade potreby zavedenia systému z diskového poľa je tiež potrebná inštalácia balíka *multipath-tools-boot* pomocou príkazu:

```
apt-get install multipath-tools-boot
```

Bez inštalácie základného balíka by každá vytvorená relácia k tomu istému úložisku pre každý pridelený zväzok predstavovala nové blokové zariadenie. To by v prípade dvoch ciest a troch pridelených zväzkov znamenalo šesť blokových zariadení na danom hostiteľovi. Inštalovaný balík zabezpečí vytvorenie virtuálneho blokového zariadenia, ktoré pokrýva všetky cesty k danému zväzku a má pridelený unikátny WWID identifikátor. Pre aktiváciu všetkých ciest je potrebné pripojenie tohto virtuálneho zväzku k súborovému systému (nestačí pripojiť iba jeden konkrétny zväzok).

Po inštalácii balíka a pripojení zväzku bude základná redundancia automaticky zabezpečená. Situáciu je možné overiť príkazom pre vypísanie redundantných ciest:

```
multipath -ll
```

Vo výpise možno vidieť, že pre každý virtuálny zväzok je vypísaná skupina existujúcich ciest, ktorých počet sa rovná počtu bežiacich relácií. Pri troch vytvorených reláciách môže výpis vyzerať podobne ako na Obrázku 16. Dlhý názov v zátvorke predstavuje WWID identifikátor a názov pred zátvorkou je možné dosiahnuť zmenou parametra *user_friendly_names* v konfiguračnom súbore. Rovnako je možné vidieť, že virtuálny zväzok *mpathb* zastrešuje dve skupiny ciest. Jedna skupina predstavuje spojenie s jednou riadiacou jednotkou pomocou dvoch ciest - jedna k zväzku *sdh* a druhá k zväzku *sdd.* Druhá skupina predstavuje spojenie s druhou riadiacou jednotkou po jednej ceste – k zväzku *sdc*. Všetky tri zväzky predstavujú jeden a ten istý zväzok reprezentovaný tromi reláciami.

```
mpathb (3600c0ff0001117c97efba85201000000) dm-3
HP,MSA2312i
size=112G features='1 queue_if_no_path'
hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=65 status=active
| |- 4:0:0:7 sdh 8:112 active ready  running
| `- 9:0:0:7 sdd 8:48  active ready  running
`-+- policy='round-robin 0' prio=10 status=enabled
  `- 6:0:0:7 sdc 8:32  active ready  running
```

Obrázok 16: Výpis redundantných ciest pre virtuálny zväzok „mpathb“

Pre ďalšie nastavenia, vylepšenia a celkovú optimalizáciu je potrebné vytvoriť súbor multipath.conf v adresári /etc. Príklad vytvoreného konfiguračného súboru viď Príloha C.

```
touch /etc/multipath.conf
```

V tomto súbore je možné okrem iného nastaviť priradenie zrozumiteľnejších názvov namiesto WWID čísiel pre virtuálne zväzky, spôsob vyvažovania výkonu, zrušenie redundantných ciest pre niektoré zariadenia (zabezpečuje to tzv. čierna listina) a takisto je možné nastaviť vlastné aliasy pre jednotlivé virtuálne zväzky.

Nastavenia je možné vykonať všeobecne, pre všetky zariadenia v sekcii *defaults* alebo pre každé zariadenia samostatne v sekcii *multipath.*

Po editácii konfiguračného súboru je potrebné znovu načítanie novej konfigurácie pomocou príkazu:

```
/etc/init.d/multipath-tools reload
```

Reštart celej služby zabezpečí príkaz:

```
/etc/init.d/multipath-tools restart
```

8.7 RIEŠENIE HAVARIJNÝCH SITUÁCIÍ

Pod pojmom havarijná situácia je možné rozumieť výpadok niektorého zariadenia alebo viacerých zariadení súčasne. Riešenie tejto situácie závisí od konkrétneho výpadku.

Pri výpadku celého diskového poľa je automaticky ukončené poskytovanie služieb všetkých virtuálnych serverov, ktoré mali na ňom uložený svoj obraz. V hosťujúcom systéme budú jednotlivé virtuálne servery hlásiť chyby čítania rovnako ako LVM v hostiteľskom systéme. Pre opätovne správny chod je potrebné ukončiť prácu s týmito servermi a spustiť ich nanovo. Rovnako je potrebné pripojený zväzok znovu pripojiť k systému a danej skupine zväzkov alebo konkrétnemu logickému zväzku obnoviť stav z dôvodu odstránenia chýb na LVM jedným z príkazov:

```
vgchange --refresh /dev/pole01
lvchange --refresh /dev/pole01/mail
```

V prípade objavenia sa takejto situácie je na jej vyriešenie poskytnutý jednoduchý skript, ktorý zariadi spomínané úkony. Nemusí byť vhodný na každú situáciu, preto je možné na jeho základe napísať ďalšie skripty.

Pokiaľ ale vypadne iba jedna riadiaca jednotka diskového poľa, takúto situáciu je možné veľmi efektívne riešiť. V prvom rade je potrebné mať pripojené obe riadiace jednotky a potom nakonfigurovať multipathing, ktorý je bližšie popísaný v kapitole 8.6. Po výpadku jednej riadiacej jednotky, bude automaticky použitá ďalšia cesta cez druhú riadiacu jednotku.

V prípade výpadku fyzického servera je potrebné jeho virtuálne servery prevziať inými dostupnými fyzickými servermi sprístupnením zväzkov, na ktorých boli umiestnené. Po pripojení daného zväzku do systému je potrebné načítať obraz virtuálneho servera správcom virtuálnych serverov a následne virtuálny server znovu spustiť, čím bude zabezpečený opätovný beh všetkých jeho služieb.

Ak však vypadne prúd a dôjde k reštartovaniu všetkých zariadení, ich nábeh by mal byť bezproblémový, pokiaľ pri náhlom vypnutí neprebiehal zápis do dôležitých systémových súborov. Pri simulovaní výpadku sa nepodarilo nasimulovať takúto situáciu, nábeh bol vo všetkých prípadoch bezproblémový. Každopádne nie je možné takúto situáciu

vylúčiť, preto je potrebné zálohovanie v pravidelných intervaloch. Pri simulácii výpadku, kedy prebiehal zápis používateľských súborov, došlo iba k strate práve zapisovaných dát. S podobnou situáciou je potrebné počítať aj pri výpadku samotného diskového poľa.

8.8 MOŽNOSTI ĎALŠIEHO RIEŠENIA

Súčasťou navrhovaného riešenia bolo vytvorenie jednoduchých skriptov pre základné činnosti ako napríklad pripojenie alebo odpojenie diskového poľa, vytvorenie zväzku s priradením konkrétnemu hostiteľovi, migrácia virtuálneho servera atď. Skripty poskytujú značné zjednodušenie práce, pretože celé riešenie používa veľké množstvo príkazov a bude nasadené pre používanie na fakulte. Vytvorené skripty ponúkajú základnú funkcionalitu, v ktorej nie sú zahrnuté všetky možné prípady. Z toho dôvodu je potrebné niektoré existujúce skripty upraviť a rovnako pre niektoré činnosti napísať nové skripty. Po vyladení a spracovaní všetkých potrebných skriptov bude v budúcnosti vytvorené webové grafické rozhranie, cez ktoré budú jednotlivé skripty spúšťané. Obsluha tohto riešenia bude jednoduchšia a hlavne prehľadnejšia pre používateľov.

Zálohovanie virtuálnych serverov je momentálne riešené prostredníctvom vytvárania snímok logických zväzkov. Ako bolo v práci spomenuté, nie vždy je zálohovanie zväzkov s bežiacimi virtuálnymi servermi bezpečné. Preto je zálohovanie odporúčané riešiť pri minimálnom zaťažení alebo ideálne pri vypnutom virtuálnom serveri. V budúcnosti by táto problematika mala byť lepšie preskúmaná a zameraná na vytváranie snímok konkrétnych virtuálnych serverov.

Možnosti multipathingu sú v práci orientované smerom redundancie ciest. Pri výpadku jednej cesty je táto cesta nahradená ďalšou. Pokiaľ je aktívnych viacero ciest súčasne, malo by medzi nimi dochádzať k vyvažovaniu výkonu. V práci bolo vyvažovanie nastavené, ale z vykonaných testov nebolo viditeľné reálne zlepšenie. Preto je odporúčané preskúmanie a otestovanie všetkých možností vyvažovania výkonu.

ZÁVER

Napredovanie technológie virtualizácie serverov sa prejavilo aj v dátovom centre fakulty FRI ŽUŽ, kde je možné nájsť množstvo virtuálnych serverov. Ich ukladanie na lokálne pevné disky fyzických serverov malo byť nahradené lepším riešením, pri ktorom by boli tieto servery ukladané na doposiaľ veľmi málo využívané úložisko diskových polí. Toto sieťové úložisko ponúka viacero výhod v porovnaní s lokálnym úložiskom hlavne na úrovni bezpečnosti dát, veľkosti poskytovaného diskového priestoru a v neposlednom rade jednoduchým prístupom k vytvoreným zväzkom vďaka využitiu rýchlej SAN siete. Vo fakultnom dátovom centre sa nachádza množstvo zariadení, takže pre účel práce bolo možné vytvoriť jednoduché vlastné testovacie prostredie SAN siete.

Cieľom práce bolo riešenie dátových úložísk pre ukladanie virtuálnych serverov. Ešte pred samotným riešením bolo potrebné vyriešiť požadované čiastkové ciele. Najskôr bolo nutné oboznámiť sa s aktuálnou situáciou v dátovom centre, vlastnosťami a konfiguráciou diskových polí a serverov, ich fyzickým prepojením a vytvorením SAN siete. Nasledujúcim krokom bolo zvládnutie protokolu iSCSI pre prístup k sieťovým úložiskám a rovnako aj správcu logických zväzkov (LVMv2).

Ďalším cieľom bolo porovnanie zvolených súborových systémov. Tieto súborové systémy boli porovnávané z viacerých hľadísk. Najväčšia pozornosť sa kládla na vstupno-výstupný výkon, schopnosť zmeny veľkosti, podporu veľkých súborov a rovnako na schopnosť zotavenia po možnom výpadku. Na základe uvedených kritérií najvyrovnanejšie výsledky podal súborový systém EXT4, ktorý je možné odporučiť v prípade potreby zmenšovania zväzkov, v opačnom prípade je odporúčaný súborový systém XFS.

S hlavným cieľom bola spojená analýza možných riešení prideľovania úložného miesta. Z uvedených spôsobov bol z hľadiska administrácie, zálohovania a celkovej funkčnosti ako najvhodnejší zvolený spôsob pridelenia jedného veľkého zväzku z diskového poľa všetkým fyzickým serverom, ktorý bol ďalej delený pomocou LVM manažéra na menšie časti – logické zväzky. Úložný priestor jednotlivých častí bol následne sprístupnený podľa potreby jednotlivým serverom.

V praktickej časti bola pozornosť ďalej venovaná pripojeniu serverov k diskovým poliam s možnou autentifikáciou, obsluhe LVM manažéra, vytváraniu záloh serverov prostredníctvom snímok zväzkov a narábaniu s obrazmi virtuálnych serverov. Rovnako boli priblížené možnosti a s nimi spojené problémy s migráciou bežiacich virtuálnych serverov na iný fyzický server z dôvodu vyvažovania výkonu alebo potreby odstávky daného servera. Nakoniec bola práca zameraná smerom aktivácie redundantných ciest k diskovému poľu, takže v prípade výpadku jednej, automaticky preberá funkciu ďalšia aktívna cesta.

Pre zautomatizovanie niektorých hlavných činností boli vytvorené jednoduché skripty prostredníctvom interpretera BASH, ktoré sú priložené v prílohe. Jednotlivé skripty je možné spúšťať v príkazovom riadku. V budúcnosti sa však počíta s rozšírením vytvorených skriptov a vytvorením webového rozhrania kvôli zjednodušeniu práce s celým vytvoreným riešením.

ZOZNAM BIBLIOGRAFICKÝCH ODKAZOV

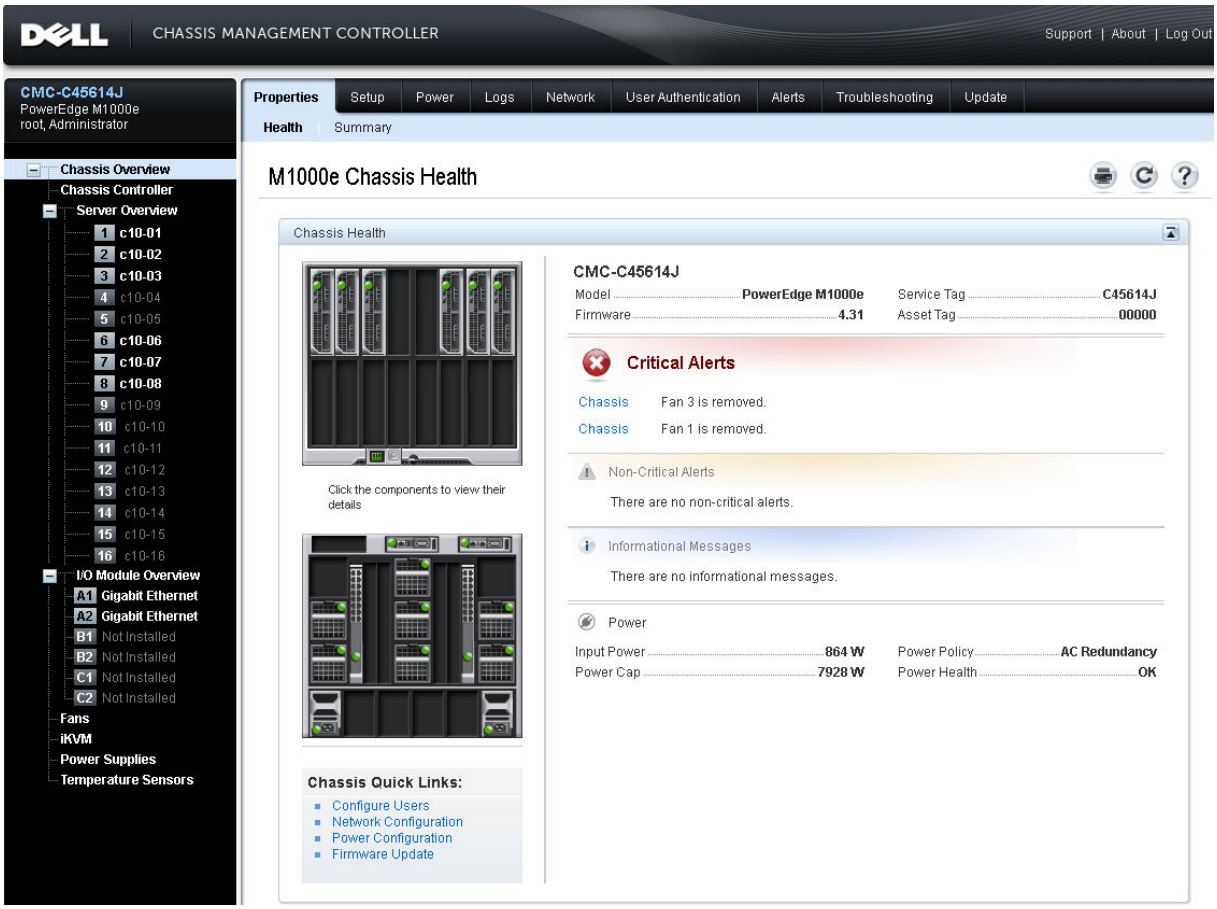
1. Arstechnica: *Virtualization with VMware.* [online]. Dostupné na internete: http://arstechnica.com/business/2011/02/virtualization-in-the-trenches-with-vmware-part-1-basics-and-benefits/
2. EMM: *Virtualizácia serverov*. [online]. Dostupné na internete: http://www.emm.sk/sk/produkty-a-sluzby/IT-riesenia/virtualizacia-serverov
3. Enterpriseefficiency: *Virtualization.* [online]. Dostupné na internete: http://www.enterpriseefficiency.com/author.asp?section_id=1830&doc_id=244104
4. Linux-KVM: *General KVM information.* [online]. Dostupné na internete: http://www.linux-kvm.org/page/FAQ
5. IBM: *Anatomy of a Linux hypervisor.* [online]. Dostupné na internete: https://www.ibm.com/developerworks/library/l-hypervisor/
6. Linuxnix: *What is KVM virtualization in Linux?* [online]. Dostupné na internete: http://www.linuxnix.com/2013/02/what-is-kvm-virtualization-in-linux.html
7. Wiki.xen: *Xen Project Software Overview.* [online]. Dostupné na internete: http://wiki.xen.org/wiki/Xen_Overview
8. Linuxexpres: *Souborové systémy v Linuxu.* [online]. Dostupné na internete: http://www.linuxexpres.cz/blog/souborove-systemy-v-linuxu
9. Techtarget: *Metadata.* [online]. Dostupné na internete: http://whatis.techtarget.com/definition/metadata
10. Cyberciti: *Understanding Linux filesystem Inodes*. [online]. Dostupné na internete: http://www.cyberciti.biz/tips/understanding-unixlinux-filesystem-inodes.html
11. Linfo: *Journaling Filesystem Definition.* [online]. Dostupné na internete: http://www.linfo.org/journaling_filesystem.html
12. Oss.sgi: *Open Source XFS™ for Linux*. [online]. Dostupné na internete: http://oss.sgi.com/projects/xfs/datasheet.pdf
13. Kernelnewbies: *Ext4.* [online]. Dostupné na internete: http://kernelnewbies.org/Ext4
14. Btrfs.wiki.kernel: *Main Page.* [online]. Dostupné na internete: https://btrfs.wiki.kernel.org/index.php/Main_Page
15. Tldp: *LVM HOWTO.* [online]. Dostupné na internete: http://tldp.org/HOWTO/LVM-HOWTO/
16. Searchstorage.techtarget: *Network-attached storage.* [online]. Dostupné na internete: http://searchstorage.techtarget.com/definition/network-attached-storage
17. Snia: *What Is a Storage Area Network.* [online]. Dostupné na internete: http://www.snia.org/education/storage_networking_primer/san/what_san
18. Techopedia: *Storage Area Network (SAN).* [online]. Dostupné na internete: http://www.techopedia.com/definition/1116/storage-area-network-san
19. POELKER, CH., NIKITIN, A.: *Storage Area Networking for dummies.* 2009. Hoboken. Wiley Publishing. ISBN: 978-0-470-38513-5
20. Cuddletech: *Quick Guide to iSCSI on Linux.* [online]. Dostupné na internete: http://www.cuddletech.com/articles/iscsi/
21. Ietf: *Internet Small Computer Systems Interface.* [online]. Dostupné na internete: https://www.ietf.org/rfc/rfc3720.txt

22. Standards.ieee: *EUI-64TM.* [online]. Dostupné na internete: http://standards.ieee.org/develop/regauth/tut/eui64.pdf
23. Searchstorage.techtarget: *Logical unit number.* [online]. Dostupné na internete: http://searchstorage.techtarget.com/definition/logical-unit-number
24. Dell: *Dell PowerEdge M1000e.* [online]. Dostupné na internete: http://www.dell.com/learn/us/en/uscorp1/corp-comm/image-gallery-poweredge-servers
25. HP: *HP StorageWorks MSA2000.* [online]. Dostupné na internete: http://www.hp.com/hpinfo/newsroom/press_kits/2010/JustRightITopk/MSA2000_G1_White_Paper.pdf
26. Jamescoyle: *Benchmark disk IO Bonnie++.* [online]. Dostupné na internete: http://www.jamescoyle.net/how-to/599-benchmark-disk-io-with-dd-and-bonnie
27. Sysbench.sourceforge: *SysBench manual.* [online]. Dostupné na internete: http://sysbench.sourceforge.net/docs/#fileio_mode
28. Linuxconfig: *Linux LVM.* [online]. Dostupné na internete: http://linuxconfig.org/linux-lvm-logical-volume-manager
29. Open-iscsi: *Linux* Open-iSCSI.* [online]. Dostupné na internete: http://www.open-iscsi.org/docs/README
30. Redhat: *Logical Volume Manager Administration.* [online]. Dostupné na internete: https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/5/html-single/Logical_Volume_Manager_Administration
31. Centos: *LVM Object Tags.* [online]. Dostupné na internete: http://www.centos.org/docs/5/html/5.2/Cluster_Logical_Volume_Manager/lvm_tags.html
32. Softpanorama: *Linux Multipath.* [online]. Dostupné na internete: http://www.softpanorama.org/Commercial_linuxes/Devices/multipath.shtml#n2011_0806_linux_san_multipathing_using_device_mapper

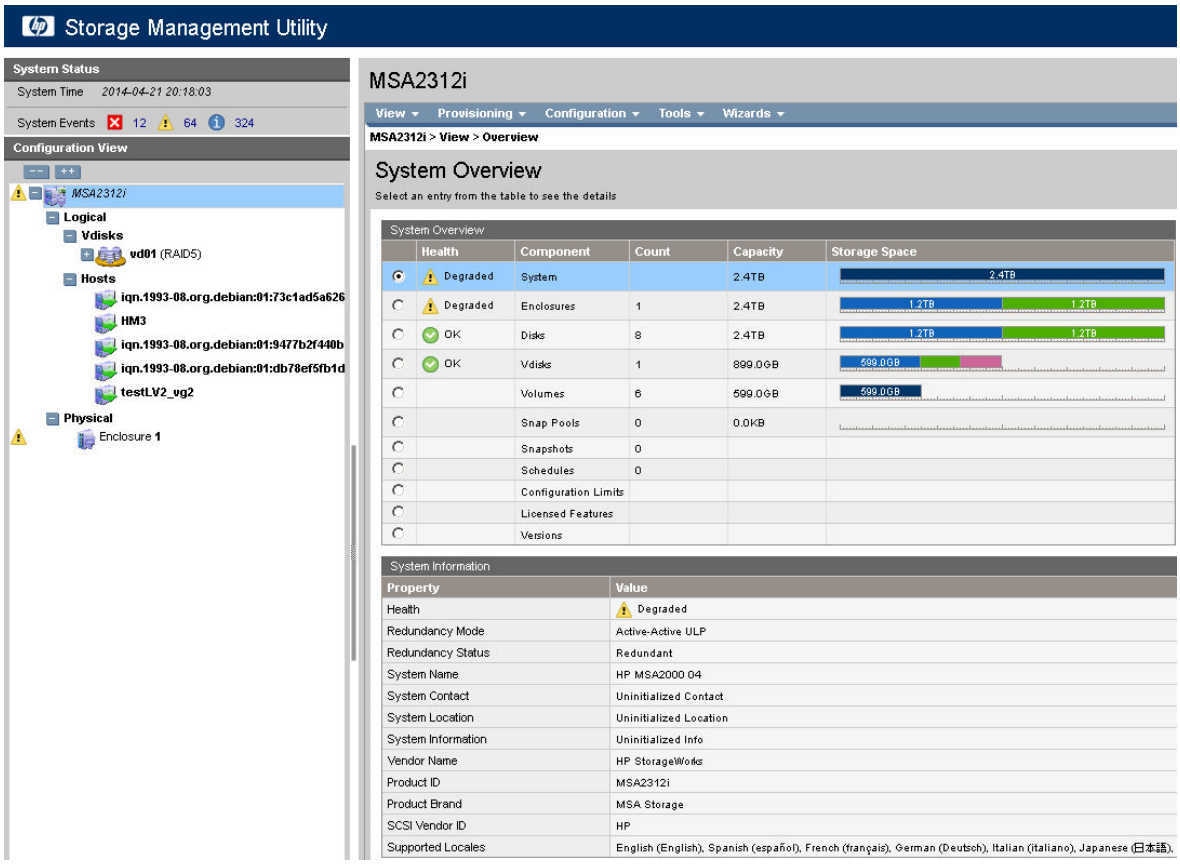
ZOZNAM PRÍLOH

PRÍLOHY

Príloha A: Úvodná obrazovka riadiacej jednotky servera



Príloha B: Úvodná obrazovka správcu úložiska HP StorageWorks P2000



Príloha C: Príklad vytvoreného konfiguračného súboru pre Multipathing

```
defaults {
        user_friendly_name      yes
        path_selector           "round-robin 0"
        path_grouping_policy    multibus
        path_checker            directio
        prio                    const
        rr_min_io_rq            "1"
        rr_weight               priorities
        failback                immediate
        no_path_retry           "5"
        polling_interval        "5"
}
blacklist {
        devnode "^(ram|sda|fd|md|dm-|)[0-9]*"
        devnode "^hd[a-z][[0-9]*]"
        devnode "^vd[a-z]"
}
multipaths {
        multipath {
                wwid 3600c0ff00035b1648e2d4a51720000
                alias backup
        }
        multipath {
                wwid 3600c0ff00035b1648e2d4a51720000
                alias mail
        }
}
```

Príloha D: Skript pre pripojenie k diskovému poľu

```
#!/bin/bash
# skript pre uvodne pripojenie(prihlasenie) k iSCSI targetu
s1=$1
s2="help"
if [ $s1 = $s2 ]
      then
      echo "je potrebne zadat 1 parameter - IP hladaneho iSCSI
      targetu"
      exit
fi

if [ $# -ne 1 ]
      then
      echo "je zadanych $# parametrov"
      echo "je potrebne zadat 1 parameter - IP hladaneho iSCSI
      targetu"
      exit
      else
      #kontrola IP adresy
      if !(ping -w 1 $1 >/dev/null)
            then
            echo "nespravna IP!"
            exit
      fi
echo "pripajam ku zadanemu iSCSI targetu\n"
#ulozenie IQN nazvu diskoveho pola
output=$(iscsiadm -m discovery -t st -p $1 | cut -d " " -f2 | head
-1)
# output=$(iscsiadm -m discovery -t st -p 10.2.0.41 | awk -F " "
'{print $2)' | sed -n '1p'
sleep 1
#pripojenie k diskovemu polu
iscsiadm --mode node --targetname "$output" --portal "$1:3260" --
login
fi
```

Príloha E: Skript pre odpojenie z diskového poľa

```
#!/bin/bash
# skript pre odpojenie pripojeneho iSCSI targetu
s1=$1
s2="help"
if [ $s1 = $s2 ]
     then
     echo "je potrebne zadat 1 parameter - IP hladaneho iSCSI
     targetu"
     exit
fi
if [ $# -ne 1 ]
     then
     echo "je zadanych $# parametrov"
     echo "je potrebne zadat 1 parameter - IP hladaneho iSCSI
     targetu"
     else
     #kontrola IP adresy
     if !(ping -w 1 $1 >/dev/null)
          then
          echo "nespravna IP!"
          exit
     fi
echo "odpajam sa od zadaneho iSCSI targetu\n"
#odpojenie od diskoveho pola
iscsiadm --mode node --targetname "$output" --portal "$1:3260" --
logout
fi
```

Príloha F: Skript pre vytvorenie nového logického zväzku

```
#!/bin/bash
# skript pre vytvorenie noveho logickeho zvazku

p1=$1
p2=$2
p3=$3
p4="ext4"
h="help"
if [ $p1 = $h ]
      then
      echo "je potrebne zadat 3 parametre - velkost, nazov a nazov
      VG"
      echo "velkost zvazku sa udava v MB(M) alebo GB(G)"
      echo "priklad: 20G/20000M"
      exit
fi
if [ $# -ne 3 ]
      then
      echo "je zadanych $# parametrov"
      echo "je potrebne zadat 3 parametre - velkost a nazov a nazov
      VG"
      else
      #vytvorenie LV a pridanie do tagu
      lvcreate --addtag @$(uname -n) -L $p1 -n $p2 $p3
      #vytvorenie suboroveho systemu
      mkfs.$p4 /dev/$p3/$p2
      #vytvorenie adresara pre pripojenie vytvoreneho LV
      mkdir -m 775 /mnt/VMs/$p2
      mount /dev/$p3/$p2 /mnt/VMs/$p2
      lvscan
fi
```

Príloha G: Skript pre odstránenie logického zväzku

```
#!/bin/bash
# skript pre odstranenie existujuceho logickeho zvazku

p1=$1
p2=$2
h="help"
if [ $p1 = $h ]
      then
      echo "je potrebne zadat 2 parametre - nazov LV a VG"
      exit
fi
if [ $# -ne 2 ]
      then
      echo "je zadanych $# parametrov"
      echo "je potrebne zadat 2 parametre - nazov LV a VG"
      else
      umount /mnt/VMs/$p1
      #deaktivacia odstranovaneho LV
      lvchange -an /dev/$p2/$p1
      #odstranenie LV
      lvremove /dev/$p2/$p1
      lvscan
      #odstranenie adresara pre pripojenie
      rmdir /mnt/VMs/$p1
fi
```

Príloha H: Skript pre rozšírenie EXT4 logického zväzku

```
#!/bin/bash
# skript pre rozsirenie existujuceho EXT4 logickeho zvazku

p1=$1
p2=$2
p3=$3
p4="ext4"
h="help"
if [ $p1 = $h ]
      then
      echo "je potrebne zadat 3 parametre - nazov LV a VG,
      rozsirovana velkost"
      echo "rozsirovana velkost sa udava v MB alebo GB"
      echo "priklad: mail pole01 20G"
      exit
fi
if [ $# -ne 3 ]
      then
      echo "je zadanych $# parametrov"
      echo "je potrebne zadat 3 parametre - nazov LV a VG,
      rozsirovana velkost"
      echo "rozsirovana velkost sa udava v MB alebo GB"
      echo "priklad: mail pole01 20G"
      else
      #ak je na danom LV spustena VM, treba ju ukoncit a jej
      velkost zvacsit pomocou jednej z uvedenych metod
      #virsh shutdown $p1
      umount /mnt/VMs/$p1
      #zvacsenie LV o pozadovanu velkost
      lvextend -L +$p3 /dev/$p2/$p1
      #kontrola suboroveho systemu EXT4 pred vykonanim zmien
      e2fsck -f /dev/$p2/$p1
      #zvacsenie suboroveho systemu do velkosti LV
      resize2fs /dev/$p2/$p1
      #pripojenie k systemu
      if mount /dev/$p2/$p1 /mnt/VMs/$p1
            then
            echo "pripojenie uspesne"
            else
            echo "pripojenie zlyhalo"
      fi
fi
```

Príloha I: Skript pre rozšírenie XFS logického zväzku

```
#!/bin/bash
# skript pre rozsirenie XFS logickeho zvazku

p1=$1
p2=$2
p3=$3
h="help"
if [ $p1 = $h ]
      then
      echo "je potrebne zadat 3 parametre - nazov LV a VG,
      rozsirovana velkost"
      echo "rozsirovana velkost sa udava v MB alebo GB"
      echo "priklad: mail pole01 20G"
      exit
fi
if [ $# -ne 3 ]
      then
      echo "je zadanych $# parametrov"
      echo "je potrebne zadat 3 parametre - nazov LV a VG,
      rozsirovana velkost"
      echo "rozsirovana velkost sa udava v MB alebo GB"
      echo "priklad: mail pole01 20G"
      else
      #ak je na danom LV spustena VM, treba ju ukoncit a jej
      velkost zvacsit pomocou jednej z uvedenych metod
      #virsh shutdown $p1
      #zvacsenie LV o pozadovanu velkost
      lvextend -L +$p3 /dev/$p2/$p1
      #zvacsenie suboroveho systemu do velkosti LV
      xfs_growfs /mnt/VMs/$p1
      #pripojenie k systemu
      if mount /dev/$p2/$p1 /mnt/VMs/$p1
            then
            echo "pripojenie uspesne"
            else
            echo "pripojenie zlyhalo"
      fi
fi
```

Príloha J: Skript pre zmenšenie EXT4 logického zväzku

```
#!/bin/bash
# skript pre zmensenie EXT4 logickeho zvazku

p1=$1
p2=$2
p3=$3
h="help"
if [ $p1 = $h ]
      then
      echo "je potrebne zadat 3 parametre - nazov LV a VG,
      zmensovana velkost"
      echo "zmensovana velkost sa udava v celych GB"
      echo "priklad: mail pole01 20G"
      exit
fi
if [ $# -ne 3 ]
      then
      echo "je zadanych $# parametrov"
      echo "je potrebne zadat 3 parametre - nazov LV a VG,
      zmensovana velkost"
      echo "zmensovana velkost sa udava v celych GB"
      echo "priklad: mail pole01 20G"
      else
      #ak je na danom LV spustena VM, treba ju ukoncit a jej
      velkost zmensit pomocou jednej z uvedenych metod
      #virsh shutdown $p1
      #ulozenie adresara VM
      DIR=/mnt/VMs/$p1
      #zistenie aktualnej velkosti LV
      SIZE=$(df -h|grep $DIR|awk '{print $2}'|cut -d "." -f1)
      umount /mnt/VMs/$p1
      #zmensovana velkost
      RED_SIZE=$(echo "$p3"|cut -d "G" -f1)
      #vypocet novej velkosti
      NEW_SIZE=$(($SIZE-$RED_SIZE))G
      #kontrola suboroveho systemu EXT4 pred vykonanim zmien
      e2fsck -f /dev/$p2/$p1
      #zmensenie suboroveho systemu na pozadovanu velkost
      if resize2fs /dev/$p2/$p1 $NEW_SIZE
            then
            #zmensenie LV o pozadovanu velkost
            if lvreduce -L -$p3 /dev/$p2/$p1
                  then
                  mount /dev/$p2/$p1 /mnt/VMs/$p1
                  echo "operacia uspesna"
                  else
                  #vratenie zmien ak zlyha zmensenie LV
                  e2fsck -f /dev/$p2/$p1
                  resize2fs /dev/$p2/$p1
                  mount /dev/$p2/$p1 /mnt/VMs/$p1
                  echo "operacia zlyhala"
            fi
      fi
fi
```

Príloha K: Skript pre sprístupnenie konkrétneho logického zväzku

```
#!/bin/bash
# skript pre spristupnenie konkretneho logickeho zvazku

p1=$1
p2=$2
#nastavenie adresara
DIR=/mnt/VMs/$p1

h="help"
if [ $p1 = $h ]
      then
      echo "je potrebne zadat 2 parametre - nazov LV a nazov VG"
      exit
fi
if [ $# -ne 2 ]
      then
      echo "je zadanych $# parametrov"
      echo "je potrebne zadat 2 parametre - nazov LV a nazov VG"
      else
      #pridanie daneho LV do svojho tagu kvoli moznosti aktivacie
      lvchange --addtag @$(uname -n) /dev/$p2/$p1
      #spristupnenie potrebneho zvazku
      lvchange -ay /dev/$p2/$p1
      #kontrola ci uz je dany subor vytvoreny
      if [ ! -d "$DIR" ]
            then
            #vytvorenie suboru
            mkdir -m 775 /mnt/VMs/$p1
      fi
      #pripojenie k systemu
      mount /dev/$p2/$p1 /mnt/VMs/$p1
      lvscan
fi
```

Príloha L: Skript pre znovu načítanie konkrétneho logického zväzku

```
#!/bin/bash
# skript pre znovu nacitanie konkretneho logickeho zvazku
v pripade vyskytu chyb
s1=$1
s2=$2
s3=$3
s4="help"
if [ $s1 = $s4 ]
      then
      echo "je potrebne zadat 2 parametre - nazov LV, nazov VG"
      echo "ak je spustena VM, treba zadat treti parameter -  nazov
      VM"
      exit
fi
#ak VM na zvazku nebezi
if [ $# -eq 2 ]
      then
      #znovunacitanie celej skupiny, do ktorej patri dany zvazok
      vgchange --refresh /dev/$s2
      echo "znovu pripajam LV s nazvom $s1"
      umount /dev/$s2/$s1
      mount -a
      else
      #ak je na danom zvazku spustena VM
      if [ $# -eq 3 ]
            then
            vgchange --refresh /dev/$s2
            echo "ukoncujem VM s nazvom $s3"
            virsh destroy $s3
            sleep 3
            echo "znovu pripajam LV s nazvom $s1"
            umount /dev/$s2/$s1
            mount -a
            else
            echo "je zadanych $# parametrov"
            echo "je potrebne zadat 2 al 3  parametre - nazov LV,
            VG a VM"
fi
fi
```

Príloha M: Skript pre zálohu konkrétneho virtuálneho stroja

```
#!/bin/bash
# skript pre zalohu konkretneho virtualneho stroja
p1=$1
p2=$2
h="help"
DIR="/mnt/VMs/backup/$p1"
DIR_SNAP="/mnt/VMs/snapshot/$p1"
if [ $p1 = $h ]
      then
      echo "je potrebne zadat 3 parametre - nazov VM, VG a velkost
      snapshotu"
      exit
fi
if [ $# -ne 3 ]
      then
      echo "je zadanych $# parametrov"
      echo "je potrebne zadat 3 parametre - nazov VM, VG a velkost
      snapshotu"
      else
      #kontrola existencie daneho suboru
      if [ ! -d "$DIR" ]
            then
            lvchange --addtag @$(uname -n) /dev/vg2/backup
            mkdir -m 775 /mnt/VMs/backup/$p1
            echo "adresar $p1 vytvoreny"
      fi
      #spristupnenie zvazku pre zalohovanie - potrebne prisposobit
      lvchange -ay /dev/vg2/backup
      mount /dev/vg2/backup /mnt/VMs/backup
      #vytvorenie, spristupnenie a pripojenie snapshotu daneho
      zvazku
      lvcreate --addtag @$(uname -n) -L $3 -s /dev/$p2/$p1 -n
      snap_$p1
      #kontrola existencie daneho suboru
      if [ ! -d "$DIR_SNAP" ]
            then
            mkdir -m 775 /mnt/VMs/snapshot/$p1
            echo "adresar $p1 vytvoreny"
      fi
      #pripojenie vytvoreneho snapshotu k systemu
      mount /dev/$p2/snap_$p1 /mnt/VMs/snapshot/$p1
      echo "Kopirujem..."
      #prekopirovanie obrazu VM
      if cp /mnt/VMs/snapshot/$p1/$p1.img /mnt/VMs/backup/$p1/
            then
            #odpojenie a vymazanie snapshotu
            umount /mnt/VMs/snapshot/$p1
            lvremove /dev/$p2/snap_$p1
            #odpojenie a deaktivacia backup adresara
            umount /mnt/VMs/backup
            lvchange -an /dev/vg2/backup
            echo "Hotovo"
      fi
fi
```

Príloha N: Skript pre live migráciu na iný fyzický server

```
#!/bin/bash
# skript pre live migraciu VM na iny fyzicky server
s1=$1
s2=$2
s3="help"
if [ $s1 = $s3 ]
      then
      echo "je potrebne zadat 2 parametre - IP noveho servera a
      nazov VM"
      exit
fi

if [ $# -ne 2 ]
      then
      echo "je zadanych $# parametrov"
      echo "je potrebne zadat 2 parametre - IP noveho servera a
      nazov VM"
      else
      #kontrola IP adresy
      if !(ping -w 1 $1 >/dev/null)
            then
            echo "nespravna IP!"
            exit
      fi
      echo "priprava na migraciu..."
      #zistenie umiestenia VM
      output=$(virsh dumpxml $2 | grep "source file" | cut -d "'" -
      f2)
      #zistenie velkosti VM
      size=$(du -k -h $output | awk -F " " '{print $1}')
      #vytvorenie prazdneho suboru na novom fyzickom serveri
      v potrebnom adresari a o potrebnej velkosti
      rsh $1 qemu-img create $output $size
      #spustenie procesu migracie
      virsh migrate --live $2 qemu+ssh://$1/system --copy-storage-
      all --verbose --persistent --undefinesource tcp://$1
fi
```