Webhostingový cluster postavený na platformě ARM

Bc. Martin Navrátil

Diplomová práce
2015

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2014/2015

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Martin NAVRÁTIL**
Osobní číslo:	**A12473**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Informační technologie**
Forma studia:	**kombinovaná**
Téma práce:	**Webhostingový cluster postavený na platformě ARM**
Téma anglicky:	**Web Hosting Cluster Built on ARM Platform**

Zásady pro vypracování:

1. **Specifikujte požadavky na vytvoření clusteru pro hostování webových portálů.**
2. **Srovnejte existující řešení pro vytvoření clusteru.**
3. **Implementujte vybraná řešení v testovacím prostředí.**
4. **Proveďte srovnání výkonnosti a spolehlivosti jednotlivých řešení.**
5. **Vytvořte návrh vhodného řešení pro vybranou testovací platformu.**

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **UPTON, Eben. Raspberry Pi: uživatelská příručka. 1. vyd. Brno: Computer Press, 2013, 232 s. ISBN 978-80-251-4116-8.**
2. **MATTHEW, Neil a Richard STONES. Linux: začínáme programovat. Brno: Computer Press, 2008, 829 s. ISBN 978-80-251-1933-4.**
3. **VÁŇA, Vladimír. ARM pro začátečníky. Praha: BEN - technická literatura, 2009, 195 s. ISBN 978-80-7300-246-6.**
4. **KAMENÍK, Pavel. Příkazový řádek v Linuxu: praktická řešení. Vyd. 1. Brno: Computer Press, 2011, 224 s. ISBN 978-80-251-2819-0.**
5. **BÍBR, Ivan. Ubuntu 10.04 CZ: praktická příručka uživatele Linuxu. Vyd. 1. Brno: Computer Press, 2010, 366 s. ISBN 978-80-251-3121-3.**
6. **LASSER, Jon. Rozumíme UNIXu. Vyd. 1. Praha: Computer Press, 2002, 252 s. ISBN 80-722-6706-X.**
7. **PINKER, Jiří. Mikroprocesory a mikropočítače. 1. vyd., 1. dot. Praha: BEN - technická literatura, 2008, 159 s. ISBN 978-80-7300-110-0.**
8. **BORONCZYK, Tim. Beginning PHP6, Apache, MySQL web development. Indianapolis, IN: Wiley Pub., c2009, xxvii, 807 p. Wrox beginning guides. ISBN 0470391146.**

Vedoucí diplomové práce: **Ing. David Malaník, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **6. února 2015**

Termín odevzdání diplomové práce: **15. května 2015**

Ve Zlíně dne 6. února 2015

L.S.

doc. Mgr. Milan Adámek, Ph.D.
děkan

doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně …….

podpis diplomanta

ABSTRAKT

Cílem diplomové práce je návrh clusterového systému poskytující služby pro účely webhostingu za použití energeticky nenáročných zařízení postavených na architektuře ARM. Práce popisuje nejen instalaci a nastavení softwaru na vývojových zařízeních, jako je Raspberry Pi, ale snaží se popsat nebo poskytnout základní informace o tom, že mohou být použita téměř jakákoli zařízení s architekturou ARM. Praktická část práce popisuje situace, kdy ve webhostingovém clusteru selže jedno ze zařízení, a zabývá se i srovnáním výkonu použitých zařízení.

Klíčová slova: cluster, linux, vysoká dostupnost, rozdělování zátěže, web server, apache, nginx, haproxy,gluster, mysql, ARM

ABSTRACT

The aim of the thesis is to design a cluster system, which provides services for webhosting, using energy conserving devices built on ARM architecture. The thesis not only describes the installation and setup of software on developmental devices such as Raspberry Pi, but also describes and provides basic information, which can be used for almost any device with ARM architecture. The practical element of the thesis describes a situation where the web hosting cluster with one of the devices fails, whilst also detailing a performance comparison of all used devices.

Keywords: cluster, linux, high avaibility, load balancing, web server, apache, nginx, haproxy, gluster, ARM

Chci poděkovat své přítelkyni MUDr. Andrei Talašové za podporu ve studiu, za potřebný přísný dohled a za nekonečnou trpělivost. Stejně tak děkuji svému vedoucímu práce, Ing. Davidu Malaníkovi, Ph.D., za odborné vedení a za téma samotné práce.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD

ARM architektura má za sebou již 30 let vývoje a za posledních několik let se dostala do povědomí veřejnosti bezesporu díky Raspberry Pi, všestrannému počítači, který našel využití v IT jako tenký klient, v automatizaci nebo jako domácí multimediální přehrávač. Od uvedení Raspberry Pi byla představena obdobná zařízení jako Banana Pi, Cubieboard, Cubox, Odroid atd., která se předháněla o prvenství ve výkonu, a společným atributem pro tato zařízení se stala malá spotřeba elektrické energie.

Dnes se existence těchto zařízení bere jako samozřejmost a nabízí se možnost nahrazení zastaralých nenasytných serverů novými zařízeními s energeticky úspornou architekturou ARM. Díky operačnímu systému Linux, který je použitelný pro architekturu ARM, se to jeví jako reálné řešení. Pravdou zůstává, že v porovnání s vyspělou architekturou x86 nebo x86-64 se zatím ARM ve výkonu nemůže srovnávat, nicméně pokud by se matematicky sečetl výkon několika ARM zařízení a našlo se ekvivalentně výkonné zařízení s x86, tak v poměru výkonu a spotřebované energie bude jasným vítězem architektura ARM.

Existuje několik možných řešení, jak vytvořit funkční celek z několika zařízení. Takový celek se označuje jako cluster. Pro účely webhostingu jsou vhodné dva typy clusterů – s vysokou dostupností a s rozložením zátěže. Práce se zaměřuje na cluster s rozložením zátěže, protože při ošetření stavu selhání load-balanceru, tedy prvku, který zátěž přerozděluje, a zároveň prvku, který se dá označit jako single point of failure, poskytne vyšší výkon a zaručí vysokou dostupnost.

V clusteru s rozložením zátěže poskytuje několik serverů nejen stejné služby, ale i stejná data. Proto se musí vybrat vhodné techniky, jak zajistit synchronní replikaci dat na úrovni souborového systému a na úrovni databáze. Zajištění těchto vlastností však stojí čas a část výpočetního výkonu, protože se data ukládají podle stupně redundance současně na dva a více serverů. Při návrhu je třeba dobře zvážit, jaký má být stupeň redundance, aby byl zachován požadovaný výkon, nebo aby byl úkon vykonán v přijatelném čase.

Pro webhosting je základním kamenem webový server, který poskytuje uživateli přístup k datům přes webový prohlížeč. V operačním systému Linux se nejčastěji používá webový server s názvem Apache nebo NGINX, ve kterých nechybí podpora skriptovacích jazyků PHP, Perl apod., pomocí kterých si dynamické webové stránky mohou uložit informace do databáze, nejčastěji do MySQL nebo MariaDB.

I. TEORETICKÁ ČÁST

1 ARCHITEKTURA ARM

Počátky architektury ARM sahají už do první poloviny osmdesátých let 20. století. Za zrodem stojí britská firma BBC, která chtěla vytvořit počítač pro vzdělávací účely.

Později se ukázalo, že BBC nemá dostatek prostředků na to, aby dokázala sama počítač navrhnout a aby počítač sama dokázala masově vyrábět a distribuovat. Proto BBC vypisuje tendr, ve kterém vyhrává firma Acorn Computers z Cambridge.

Jako výsledek spolupráce je představen počítač s názvem Acorn BBC Micro ve dvou verzích (model A a vybavenější model B). Obě verze měly levný osmibitový mikroprocesor MOS 6502, který byl dostatečný na jednoduché úkoly naprogramované v BASICu nebo strojovém kódu, nicméně ve snaze vytvořit operační systém s grafickým uživatelským rozhraním se zjistilo, že tento osmi bitový procesor nebude dostatečný.

Firma Acorn Computers se tedy rozhodla, že vyvine vlastní procesor a díky dobře navržené architektuře vycházející z architektury RISC se tyto nové procesory těší velké oblibě, především v Británii. V roce 1983 rovněž vzniká nová společnost Advanced RISC Machines, dnes ARM Holding.

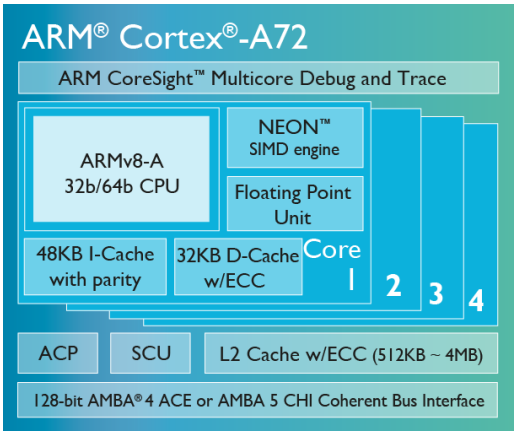
ARM Holding dnes procesory již nevyrábí a drží jen duševní vlastnictví. Výrobci ARM procesorů proto musí platit licenční poplatky při výrobě samotných procesorů.

Firmy, které využívají ARM architekturu ve svých produktech:

Procesory ARM zatím výkonově nemohou konkurovat klasickým procesorům, nicméně úspěšně konkurují ve výkonu, jež se přepočítá na jeden watt spotřebované elektrické energie. To znamená, že jsou ARM procesory energeticky úspornější a efektivnější. Podle tvrzení ARMu energetická úspora činí až 2/3 = 66% při dosažení ekvivalentního výkonu [28].

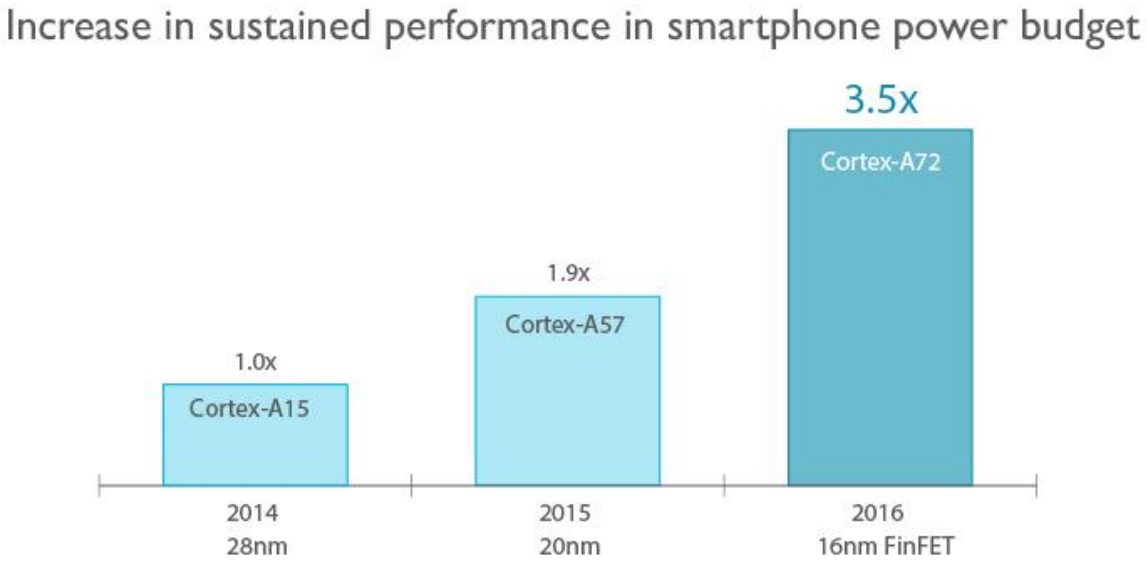
Zatímco Intel nebo AMD se snaží u svých procesorů snížit spotřebu elektrické energie, aby mohli konkurovat na trhu s mobilními zařízeními, ARM jde jinou cestou a snaží se naopak o zvýšení výkonu.

Nejmodernější procesor od společnosti ARM je postavený na architektuře ARMv8-A, má označení Cortex-A72.



Obr. 1 ARM architektura [28]

Z následujícího obrázku je patrné, jakého rychlého vývoje ARM architektura dosahuje. Za dva roky se podařilo výkon procesorů více jak ztrojnásobit.



Obr. 2 Vývoj ARM architektury [28]

2 CLUSTER

Pojem počítačový cluster by se dal popsat jako propojení volně vázaných počítačů. Počítače mezi sebou komunikují a navenek se tváří jako jeden systém. Propojení počítačů v clusteru je řešeno pomocí počítačové sítě [13].

Díky clusterování lze dosáhnout vyššího výpočetního výkonu, zajistit vyšší spolehlivost či vyšší efektivity, než za použití jedné výpočetní stanice.

Clustery se dříve nejčastěji používaly při paralelním zpracování úloh, nyní se také hojně využívají k zajištění vysoké dostupnosti služeb jako je webový server, mail server, databázový server apod.

2.1 Typy Clusterů

Existuje několik typů clusterů. Ty nejpoužívanější jsou popsány následovně:

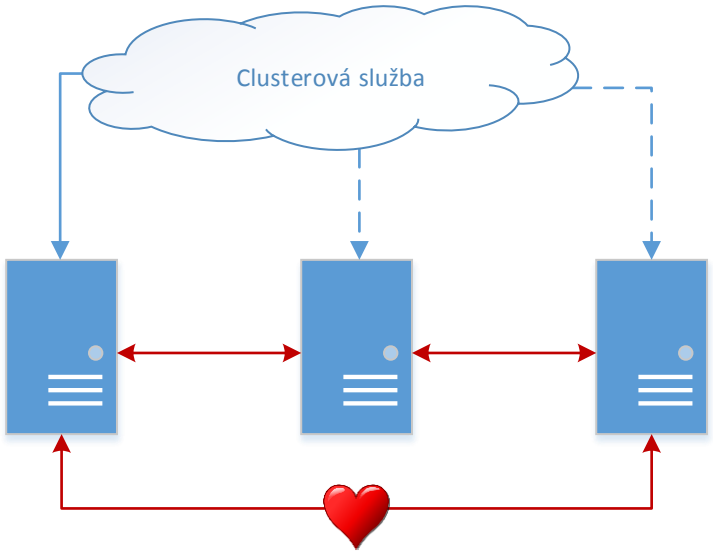
2.1.1 Výpočetní cluster (High Performance)

Je to cluster určený pro řešení náročných počítačových úloh. Výpočet je rozdělen pomocí paralelizace do menších nezávislých úloh. Úlohy běží paralelně na různých nodech v clusteru a zajistí se tím rychlejší výpočet celé úlohy.

Obr. 3 Výpočetní cluster

2.1.2 Cluster s vysokou dostupností (High Avaibility nebo také Failover)

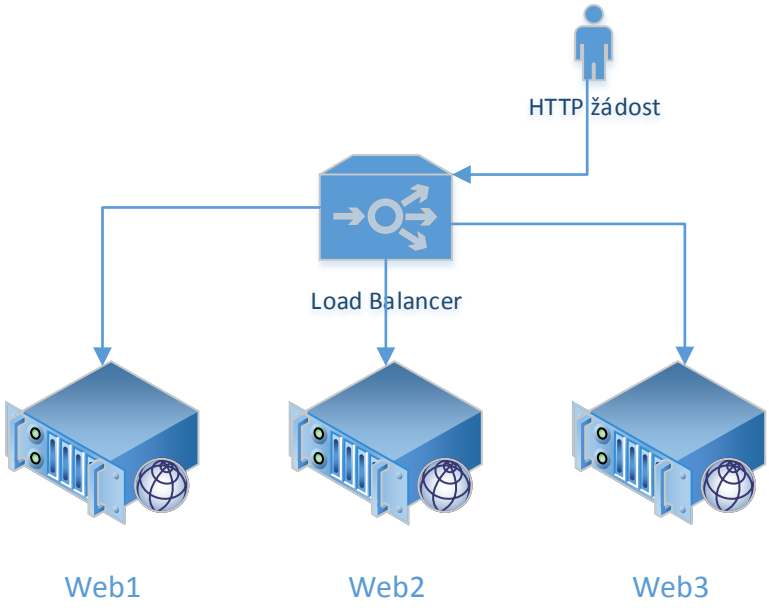
Cílem clusteru s vysokou dostupností je zajistit maximální možnou dostupnost nějaké služby. To je zajištěno tak, že se služba spolu s clusterovým softwarem nainstaluje na několik nodů. Clusterový software pak monitoruje dostupnost jednotlivých nodů a služeb v clusteru. Pokud nějaký nod nebo služba přestane být dostupná, cluster tento stav zaznamená a postará se o to, že je služba poskytnuta z jiného nodu.



Obr. 4 Cluster s vysokou dostupností

2.1.3 Cluster s rozložením zátěže (Load Balancing)

Cílem clusteru s rozložením zátěže je vhodným způsobem rozdělit požadavky na několik nodů, které poskytují stejnou službu. Všechny požadavky směřují nejprve na tzv. load balancer, který pak požadavky dále pouze přesměrovává na jednotlivé nody v clusteru.



Obr. 5 Cluster s rozložením zátěže

2.1.4 Úložný cluster

Poskytuje přístup k diskové kapacitě, jež je umístěna na nodech v clusteru. V tomto typu clusteru je nutné použít speciální souborové systémy. Cílem je zvýšení propustnosti nebo zajištění vyšší dostupnosti.

2.1.5 Gridový cluster

Gridový cluster je soubor počítačů (mohou to být i obyčejné desktopy), které v případě potřeby mohou sdílet svůj výpočetní výkon. Počítače v clusteru jsou do jisté míry na sobě nezávislé.

2.2 Disková pole

Diskové pole souvisí se zkratkou RAID, jež původně znamenala „Redundant Array of Inexpensive Disks“, postupem času se zkratka změnila na „Redundant Array of Independent Disks“.

Jedná se vlastně o propojení několika fyzických disků (počet není stanoven) do jednoho svazku tak, že se celý svazek navenek tváří jako jeden pevný disk. Smyslem tohoto zapojení je například zvýšení výkonu, navýšení kapacity nebo zvýšení bezpečnosti. Zapojení do RAIDu je možné realizovat softwarově, kdy se o spojení stará jádro operačního systému, nebo hardwarově, kdy se k propojení využije fyzický řadič.

2.2.1 RAID 0

Vzniká spojením minimálně dvou disků. V tomto režimu se data zapisují napříč disky střídavě a mělo by docházet k výraznému navýšení rychlosti zápisu. Výsledná teoretická rychlost by měla být n-násobná, kde n je počet disků v poli. Analogicky i čtení z disků by mělo být n-násobně rychlejší.

Výhoda tohoto zapojení je tedy rychlost zápisu a čtení. Nevýhoda je, že data nejsou uložena redundantně a v případě poruchy jednoho z disků následuje nevratná ztráta všech dat.

Obr. 6 Schéma RAID 0

2.2.2 RAID 1

Rovněž vzniká spojením minimálně dvou disků. Data se jednoduše zrcadlí na všechny disky pole. Tím pádem dochází k redundantnímu uložení dat v poli, čímž se zajistí vyšší bezpečnost dat. Výsledná kapacita je rovna kapacitě nejmenšího disku v poli.

Při poruše jednoho z disků nedochází ke ztrátě dat. V okamžiku, kdy se vadný disk vymění za nový, dochází k automatické obnově.

Obr. 7 Schéma RAID 1

2.2.3 RAID 5

Vzniká spojením minimálně tří disků. Celková kapacita je pak rovna n-1 disků. Data jsou ukládána rovnoměrně na všech discích. K datům je dopočítáván a ukládán tzv. kontrolní součet (parita), který v případě poruchy jednoho z disků dokáže chybějící data dopočítat a

úspěšně obnovit. Výpočet kontrolních součtů je samozřejmě o něco náročnější na výpočetní výkon v porovnání s RAID 0 nebo RAID 1.

2.2.4 RAID 6

Vzniká spojením minimálně čtyř disků. Funguje podobně jako RAID 5, ale rozdíl je v tom, že se vypočítávají dva kontrolní součty místo jednoho. Díky tomu odolává proti poruše dvou disků. Navíc pokud u RAID 5 trvala obnova poškozeného disku několik hodin, po tuto dobu RAID 5 neposkytoval žádnou ochranu proti další poruše disků. RAID 6 se používá tam, kde je tedy vyžadována maximální bezpečnost dat.

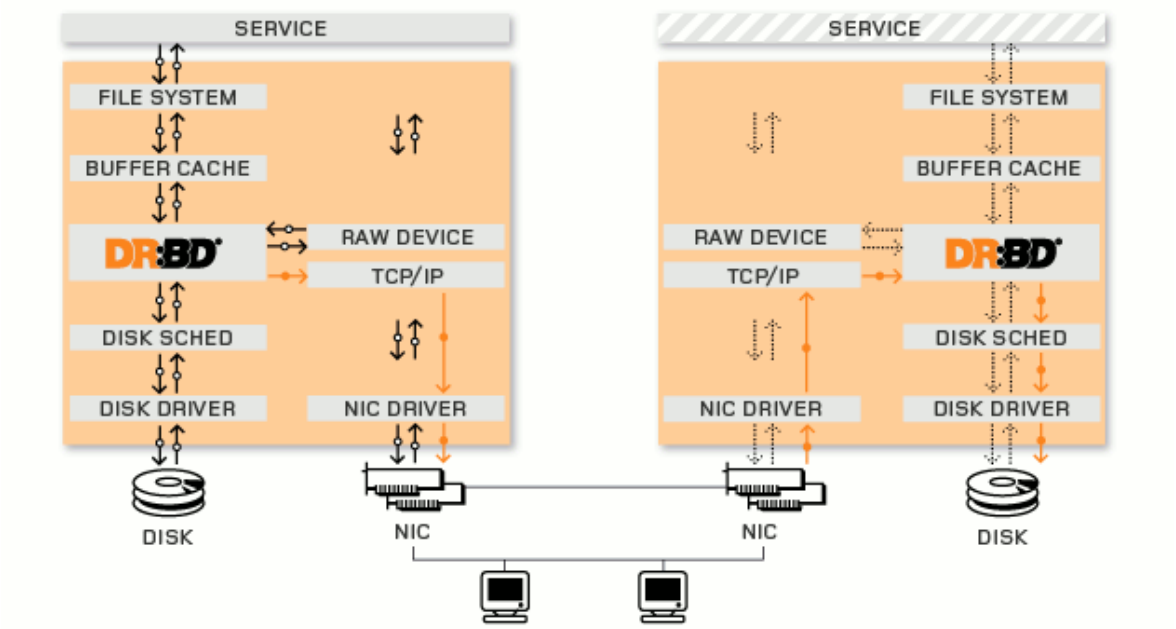
2.2.5 JBOD

„Jen hromada disků“, tak by se dala popsat zkratka JBOD (Just a Bunch of Disks). Data se ukládají na několik disků způsobem, kdy nedochází k redundanci ani zvýšení výkonu. Data se jednoduše řetězí za sebe. Celková kapacita pole je rovna součtu všech kapacit disků v poli.

2.3 DRBD

DRBD používá bloková zařízení určená pro clustery s vysokou dostupností. Bloková zařízení jsou zrcadlena po síťové vrstvě, přičemž by se dalo říci, že poskytuje úložiště zapojené v RAID 1 [26].

Následující obrázek ilustruje, jak DRBD funguje. Dva servery (nody) obsahují běžné komponenty v Linuxu, jako je souborový systém, vyrovnávací paměť, disky, TCP/IP stack a síťovou kartu. Černé šipky znázorňují komunikaci mezi těmito komponentami. Oranžové šipky znázorňují, jakým způsobem DRBD zrcadlí data z aktivního nodu na pasivní node.



Obr. 8 Princip DRBD [26]

DRBD dokáže zrcadlit data dvěma způsoby.

2.3.1 Synchronní mirroring

Synchronní replikace zaručuje to, že data budou vždy aktuální. Data se zapisují na obou nodech zároveň a k úspěšnému zápisu dojde jen tehdy, pokud se úspěšně zapíší na oba dva nody. Tento druh zrcadlení se doporučuje pro clustery s vysokou dostupností.

2.3.2 Asynchronní mirroring

Data se nejprve zapíší na lokální disk nodu a poté jsou zrcadlena na node druhý. Asynchronní zrcadlení je vhodné pro zrcadlení na velké vzdálenosti, kdy latence připojení je vyšší než latence zápisu, kterou bychom chtěli dosáhnout.

2.4 GLUSTERFS

GlusterFS je open source distribuovaný souborový systém určený nejen pro velká datová úložiště. GlusterFS dokáže spojit různorodá disková úložiště do obrovského celku dostupného po síti pomocí protokolu RDMA nebo pomocí TCP protokolu, případně kombinací obojího. Výkon, spolehlivost a kapacita tohoto souborového systému závisí nejen na hardwarové konfiguraci strojů, ale také na jejich počtu, propustnosti sítě a propustnosti síťových prvků.

2.4.1 Vlastnosti

Výhody a vlastnosti tohoto systému by se daly shrnout do několika bodů [29].

- Gluster je snadný způsob, jak zajistit vlastní úložný prostor (NAS) na téměř jakémkoli zařízení.
- Gluster je snadno škálovatelný – může se začít s minimálním počtem zařízení a postupem času se mohou přidávat další a další zařízení.
- Failover se dá nakonfigurovat automaticky, bez nutnosti dalšího softwaru – pokud jeden node selže, je automaticky využit node jiný. Po opravení nefunkčního nodu nebo výměnou za nový jsou data automaticky zreplikována.
- Extrémně jednoduchý, takže se běžné scénáře dají nastavit v řádech minut.
- Není třeba žádný extra jaderný modul v jádře Linuxu.
- Gluster může být dostupný pomocí NFS, SMB/CIFS nebo pomocí vlastního nativního souborového systému GlusterFS.
- Enterprise ready – dostupná komerční podpora

2.4.2 Vysvětlení názvosloví u Glusterfs

Vývojáři Glusterfs používají vlastní názvosloví, které je popsáno níže.

Glusterd je daemon/služba, která spravuje svazky a nody v clusteru. Musí běžet na všech nodech v Trusted storage poolu.

FUSE je Filesystem in userspace, což je jaderný modul v Linuxu, který umožňuje neprivilegovaným uživatelům vytvořit vlastní souborový systém bez nutnosti zásahu do samotného jádra. FUSE je tedy jakousi pomyslnou branou do jádra.

Brick je základním stavebním kamenem v GlusterFS. Je reprezentován vyexportovaným adresářem v trusted storage poolu.

Volume (svazek) je logické seskupení bricků.

Block storage jsou bloková zařízení, typicky hard disk.

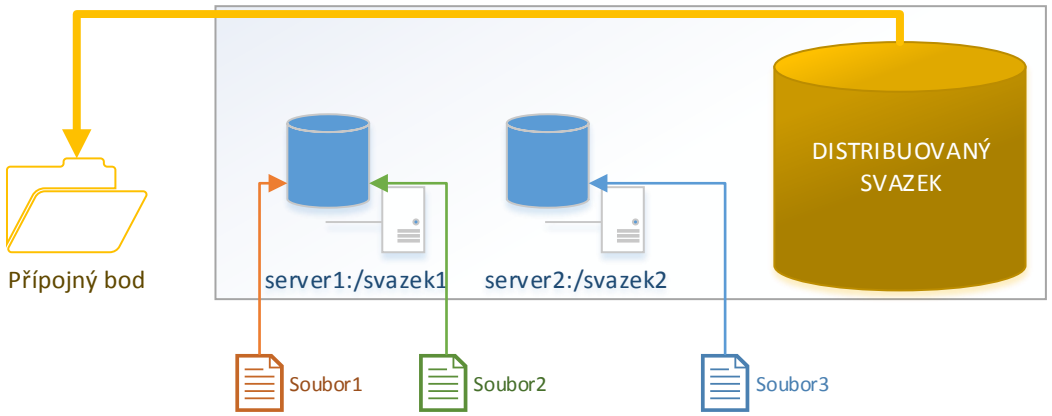
Node je server nebo počítač, který hostuje jeden nebo více bricků.

RDMA je přímý přístup z paměti jednoho počítače do paměti druhého počítače bez využití operačního systému. To zajišťuje vysokou průchodnost a nízkou latenci, což je velice žádoucí pro vlastnosti clusteru.

Trusted Storage Pool je důvěryhodná skupina serverů (nodů).

2.4.3 Typy svazků (volumes)

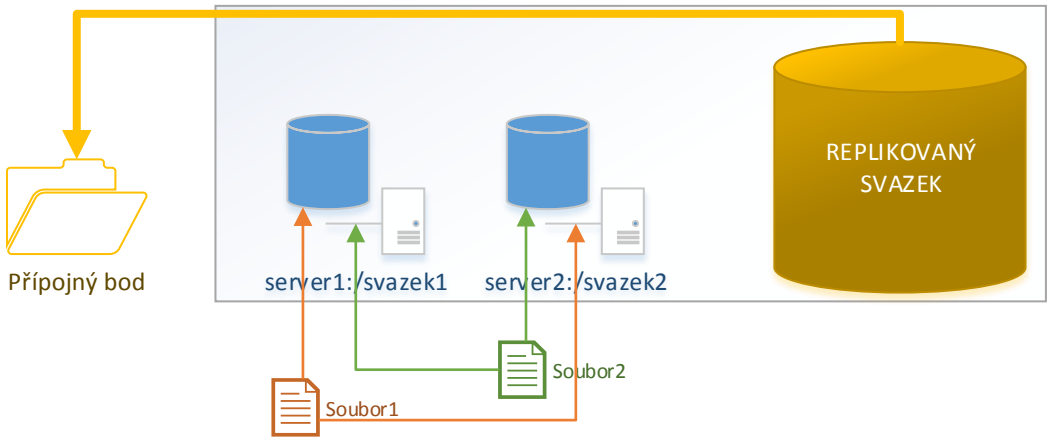
Distribuovaný – soubory jsou distribuovány napříč všemi bricky v trusted storage poolu. Tento typ svazku je vhodný při velkém počtu bricků a efektivní při škálování.



Obr. 9 Distribuovaný svazek

gluster volume create test-volume server1:/exp1 server2:/exp2 server3:/exp3 server4:/exp4

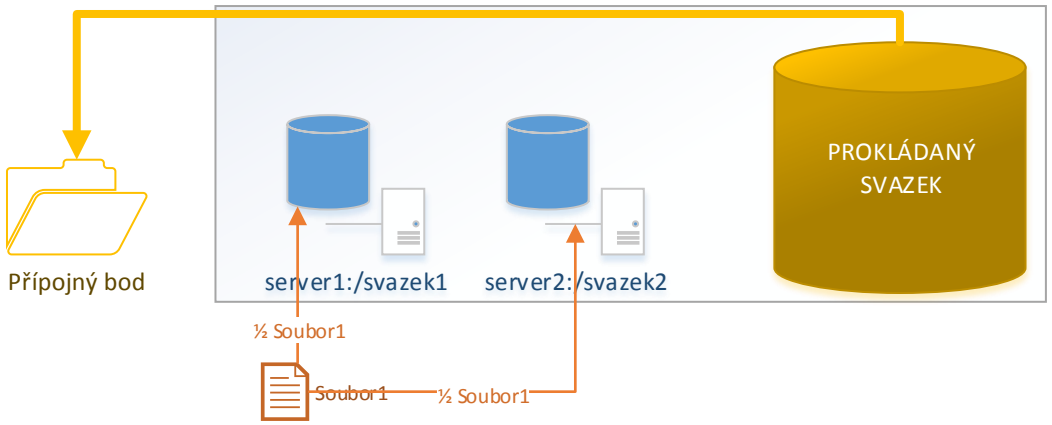
Replikovaný – soubory jsou replikovány napříč všemi bricky v trusted storage poolu. Tento typ je vhodný pro řešení s vysokou dostupností a spolehlivostí.



Obr. 10 Replikovaný svazek

gluster volume create test-volume replica 2 transport tcp server1:/exp1 server2:/exp2

Prokládaný (striped) – soubory jsou rozděleny na menší celky a tyto celky jsou následně uloženy napříč všemi bricky. Tento typ je vhodný v případě, kdy je potřeba uložit větší souborm, než je kapacita jednotlivých disků.



Obr. 11 Prokládaný svazek

gluster volume create test-volume stripe 2 transport tcp server1:/exp1 server2:/exp2

Distribuovaný prokládaný – důraz na výkon a obrovské soubory.

Distribuovaný replikovaný – podobný princip jako RAID 10.

Disperzní – v případě selhání bricku lze data zrekonstruovat pomocí matematických transformací. Disperze udává, kolik funkčních bricků je potřeba k dopočítání chybějících dat.

3 WEBHOSTING

Webhosting je služba, pomocí které můžeme umístit na Internet nějaký obsah. Tento obsah je pak přístupný ostatním pomocí webového serveru.

3.1 Webový server

Webový server je program postavený na modelu klient/server. Vyřizuje požadavky od klientů, kteří očekávají ve svém prohlížeči zobrazení nějaké stránky. Komunikace mezi klientem a serverem probíhá pomocí nezabezpečeného protokolu HTTP nebo s využitím SSL či TLS šifrování se může protokol HTTP zabezpečit. Takto zabezpečený HTTP protokol má označení HTTPS.

Obecně se doporučuje používat zabezpečené spojení pomocí protokolu HTTPS, ovšem pokud se k zabezpečení použijí certifikáty od nedůvěryhodné certifikační autority, webový prohlížeč varuje, že jsou stránky nedůvěryhodné, a uživatel tím získává pocit, že na stránkách je umístěný nebezpečný obsah.

Získání certifikátu pro zabezpečení HTTP protokolu je obvykle zpoplatněno a certifikační autorita si účtuje za každý vydaný certifikát od 1000Kč za rok.

Existuje však jedna certifikační autorita StarComm se sídlem ve Švýcarsku (https://www.startssl.com/), která poskytuje základní certifikáty zcela zdarma. Platnost certifikátu je jeden rok, přičemž po roce je možné zažádat o vydání nového certifikátu. Nový certifikát je opět zdarma.

3.1.1 Apache

Webový server Apache, aktuálně ve verzi 2, je stále nejrozšířenější webový server vůbec. Některé zdroje uvádí, že na serveru Apache bylo hostováno začátkem roku 2015 více jak 50% všech webových stránek na Internetu. V době své největší popularity nebo rozšířenosti na serveru Apache běželo až 70% všech webových stránek [21].

Počátek webového serveru Apache se datuje do roku 1993, kdy se jmenoval ještě jako HTTPd, respektive do roku 1995, kdy byla vydána verze s názvem Apache.

3.1.2 Nginx

Webový server NGINX vychází z modulu pro webový server Apache. Modul měl sloužit jako náhrada za zastaralý modul mod_proxy. Autorem je Igor Sysoev, který později usoudil, že bude lepší, když vytvoří vlastní webový server, jež bude snadno škálovatelný a dobře funkční. Po dvou letech práce v roce 2004 vychází první oficiální verze [27].

Původní cíl nového webového serveru byl, aby mohl bez problémů obsloužit deset tisíc požadavků současně při minimálním využití prostředků, možnost využití cache paměti či reverzní proxy.

Za zmínku stojí některé společnosti, jež Nginx využívají. Z těch nejznámějších: Netflix, Hulu, WordPress.com, GitHub. U nás to jsou například Seznam.cz nebo FTV Prima.

3.1.3 Lighttpd

Webový server Lightttpd je velice podobný serveru Nginx. Vznikl přibližně ve stejnou dobu a pracuje podobně jako Nginx, to znamená, že se soustředí na maximální výkon při užití minimálního množství prostředků. Autorem je Jan Kneschke.

Co se týká technické podpory nebo velikosti komunity kolem tohoto serveru, není tak veliká jako u konkurenčního Nginxu.

3.2 Databázový server

Databázový server je program, který je schopný ukládat velké množství dat a také tato data vracet zpět. Nejčastěji se používají relační databáze. Ve světě MS Windows nejpoužívanější databází je MS SQL SERVER a ve světě Linuxu nejčastěji používanou databází je MySQL. V těchto databázích se používá SQL jako dotazovací jazyk.

3.2.1 MySQL

Jak již bylo zmíněno, MySQL je nejrozšířenější databázový systém na Linuxu. Za jeho zrodem stojí Michal Widenius a David Axmark, kteří tento software uvolnili pod licencí GNU General Public License. V roce 2009 kupuje MySQL firma Sun Microsystems (Oracle), jež mění původní licenční podmínky. Proto mnozí zastánci otevřeného softwaru vyslovili obavy z budoucího vývoje MySQL.

3.2.2 MariaDB

MariaDB vzniká právě kvůli tomu, že MySQL převzal Oracle. Michael Widenius opět stojí za zrodem nového databázového serveru, který se od původního MySQL serveru příliš funkčně neliší, ale dává možnost podílet se na jeho vývoji prakticky každému, protože na rozdíl od převzaté MySQL databáze je pod licencí GPL.

3.3 Zajištění vysoké dostupnosti

Pro zajištění vysoké dostupnosti je třeba mít na paměti, že nestačí zajistit replikované služby, ale tyto služby musí být dostupné na jednom místě. Na příklad za adresou www.mujcluster.cz se může skrývat load-balancer, sám žádné služby neposkytuje, ale došlé požadavky jen rozděluje a přeposílá na jednotlivé servery v clusteru. Tím vzniká úzké místo, potenciálně nebezpečné, v clusterovém systému a musí se ošetřit i situace, kdy load-balancer selže. Ve spolehlivém clusteru proto bývá záložní load-balancer, který do hry vstupuje právě při selhání hlavního load-balanceru převzetím jeho role.

Implementací v linuxovém prostředí je několik, mezi nejznámější a často používané patří Heartbeat, Pacemaker, CoroSync nebo moderní Keepalived.

3.4 Keepalived

Keepalived je směrovací software, který technicky řeší failover a failback situace. Obsahuje v sobě VRRP protokol, který je základním kamenem směrování při těchto situacích, a pomocí něhož se nastavují vlastnosti IP protokolu na monitorovaných zařízeních. Funguje na čtvrté vrstvě OSI modelu [20].

II. PRAKTICKÁ ČÁST

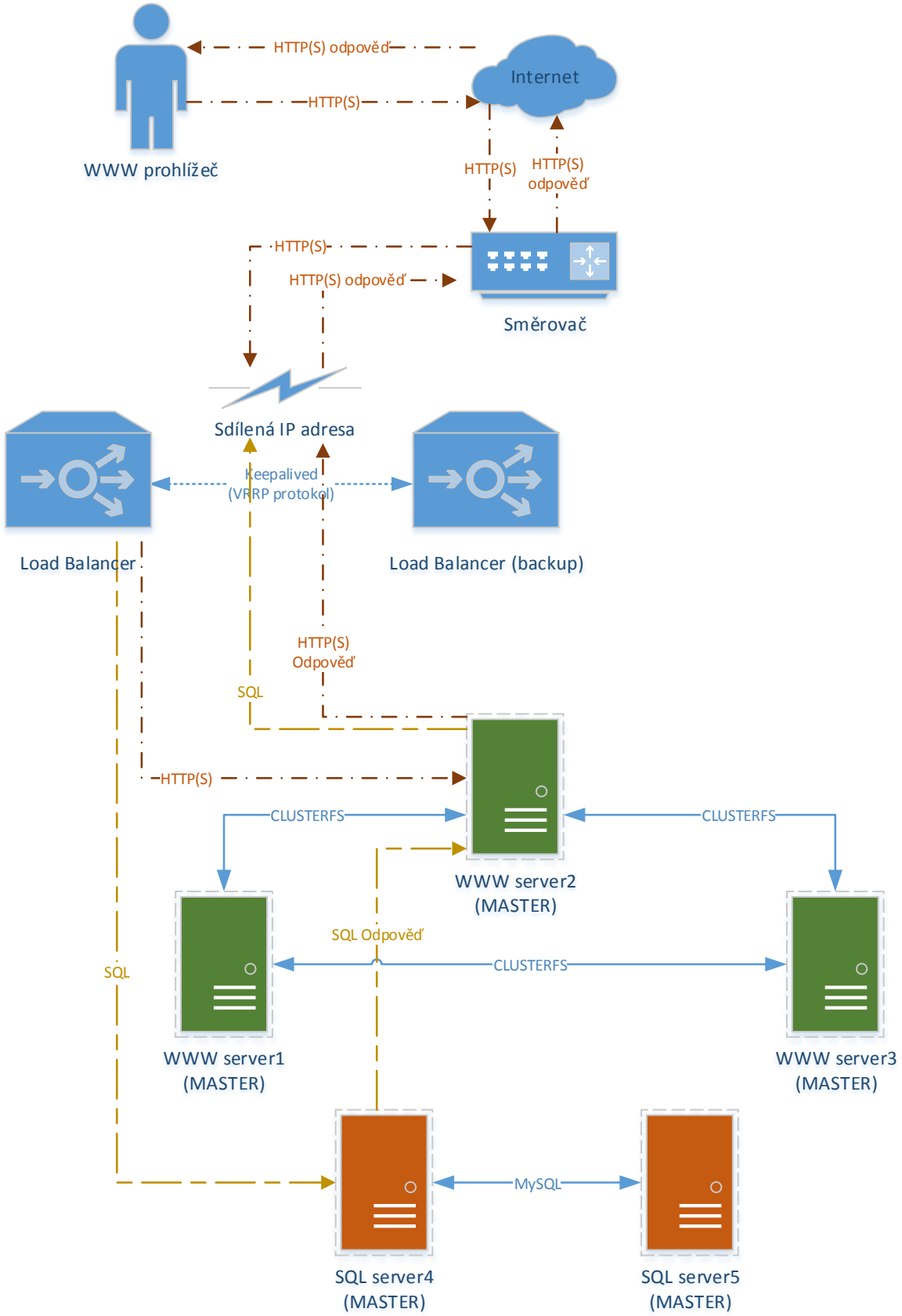
4 NÁVRH CLUSTERU

Pro maximální dostupnost služeb webhostingového clusteru je potřeba zajistit chod služeb po maximální možnou dobu, v ideálním případě by všechny služby měly fungovat nepřetržitě a to bez jakéhokoli výpadku. V reálném světě tento stav však není možný a musí se počítat s tím, že nějaká služba nebo systém přestane jednou, alespoň na chvíli, fungovat. Každý, kdo má počítač ví, že se jednou za čas musí provést jeho údržba, provést upgrade, nainstalovat softwarové aktualizace, bezpečnostní záplaty apod., kdy je často potřeba počítač restartovat. Když se restartuje domácí počítač nebo pracovní stanice, nikdo si toho ani nevšimne. Co se ale stane v případě, když se restartují servery, které poskytují nějaké služby? Ano, služby přestanou být dostupné. A když server přestane fungovat úplně, nastává tragédie.

V navrženém clusteru jsou služby poskytovány redundantně, to znamená, že každá služba běží na minimálně dvou zařízeních. Jako možné clusterové varianty se nabízely failover cluster a cluster s rozdělováním zátěže.

Vzhledem k tomu, že ARM architektura nedisponuje přebytkem výkonu, vhodnější typ cluster je cluster s rozdělováním zátěže. Tento typ cluster je vhodnější proto, že podobně jako failover cluster dokáže ošetřit situace, když je nějaký server v clusteru nedostupný, a zároveň umí využít výpočetní výkon všech serverů, když naopak všechny servery jsou dostupné.

Aby servery poskytující webovou nebo databázovou službu měly mezi sebou konzistentní informace, musí se zajistit synchronní replikace dat, o kterou se stará Gluster a MySQL.



Obr. 12 Navržený cluster

5 POUŽITÉ ZAŘÍZENÍ V CLUSTERU

Díky tomu, že během psaní diplomové práce bylo k dispozici hned několik zařízení s architekturou ARM, naskytla se možnost webhostingový cluster nejen navrhnout, ale i realizovat.

Původní záměr směřoval ke clusteru s operačním systémem Arch Linux ARM a s operačním systémem Debian Wheezy. Bohužel první komplikace nastala po instalaci Glusteru, kdy se novější verze tohoto distribuovaného souborového systému nerozuměla se starší verzí na Debianu. Jelikož byl Arch Linux ARM preferovanou distribucí, pokračoval návrh cluster a jeho následná implementace právě na něm. To se nakonec ukázalo jako špatná volba, protože webové servery NGINX i Apache někdy hlásily, že nelze nalézt soubor, který byl umístěn na glusterfs. Po těchto komplikacích se přešlo na linuxovou distribuci Debian, ve které všechno funguje.

5.1 Vhodné operační systémy

Existuje několik operačních systémů, které jsou vhodné jako operační systémy pro servery v clusteru. Všechny to jsou linuxové distribuce a včechny jsou poskytovány zdarma.

5.1.1 ArchLinux ARM

Domovská stránka: http://archlinuxarm.org/

Jedná se o distribuci, která je určená výhradně pro ARM zařízení. Podporuje platformy ARMv5, ARMv6 i ARMv7. Vychází z klasického Arch Linuxu, díky němuž poskytuje vyladěnou platformu, nejnovější aplikace a dobrou podporu. Charakterizoval bych jej třemi slovy:

- Jednoduchý
- Optimalizovaný
- Inovativní

5.1.2 Debian (Wheezy)

Domovská stránka: https://www.debian.org/

Debian je ve srovnání s Arch Linuxem velice konzervativní. Přesto se však jedná o jednu z nejrozšířenějších serverových linuxových distribucí. Má tři hlavní větve (stable, testing, unstable) lišící se podle funkčnosti. Aktuální stabilní verze nese označení Wheezy.

5.1.3 Gentoo

Domovská stránka: https://www.gentoo.org/

Gentoo je linuxová distribuce založená kompletně na zdrojových kódech, respektive balíčky v této distribuci jsou jen tzv. meta balíčky, které se musí ze zdrojových kódů nejprve zkompilovat a poté nainstalovat. O balíčky se stará balíčkovací systém portage.

Výhody této distribuce jsou bezesporu v tom, že se do systému může jednoduše nainstalovat jakákoli součást systému nebo program. Kompilace programů ze zdrojových kódů může přispět ke stabilnějšímu operačnímu systému.

Nevýhoda je snad jen jedna: časová náročnost. Aktualizace systému může trvat i několik dnů, než se všechny balíčky zkompilují.

5.2 Raspberry Pi, model B

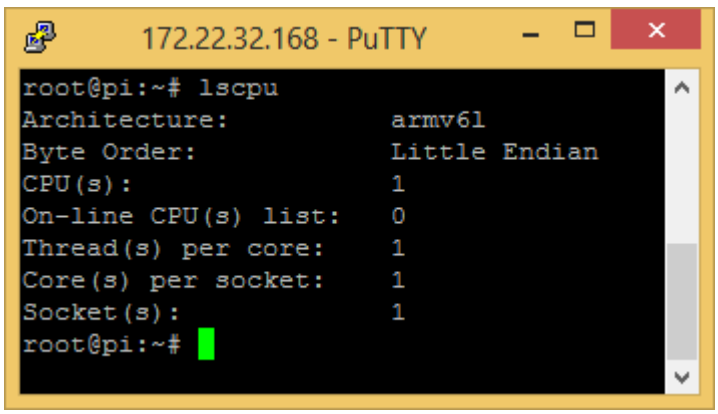
Raspberry Pi je snad první ARM zařízení, které v roce 2012 odstartovalo masové rozšíření ARMu pro podobná využití jako běžný PC, přičemž původní záměr měl úplně jiné ambice. Někdy v roce 2006 pan Eben Upton začal stále více pociťovat znepokojení nad tím, že se k němu nebo jeho kolegům, kteří taktéž vyučovali na univerzitě v Cambridge počítačovou vědu, hlásí čím dál tím méně studentů. Proto přišel s nápadem vyrobit jednoduchý počítač, pomocí kterého by se studenti mohli připojit k Internetu, případně se učit programovat. To vše za minimální cenu. Díky rychlému vývoji procesorů pro mobilní zařízení se mohl po roce 2008 nápad zrealizovat, respektive mohly začít vznikat první prototypy dnešního Raspberry Pi.

Obr. 13 Raspberry Pi, model B [10]

5.2.1 Technické parametry

Tab. 1 Technické parametry Rapberry Pi, model B

SoC	Broadcom BCM2835
CPU	1x 700MHz ARMv6, Cortex A6
RAM	512MB
NETWORK	10/100Mbit Ethernet



Obr. 14 Raspbery Pi, model B – výpis lscpu

5.2.2 Instalace operačního systému (Arch Linux)

Instalace operačního systému na Raspberry se většinou provádí tak, že se použije obraz paměťové karty, který je obvykle ke stažení na stránkách buď samotného Raspberry Pi nebo na stránkách linuxové distribuce. V případě Arch Linuxu toto neplatí, instalace se provádí v několika krocích a je třeba mít k dispozici počítač s Linuxem.

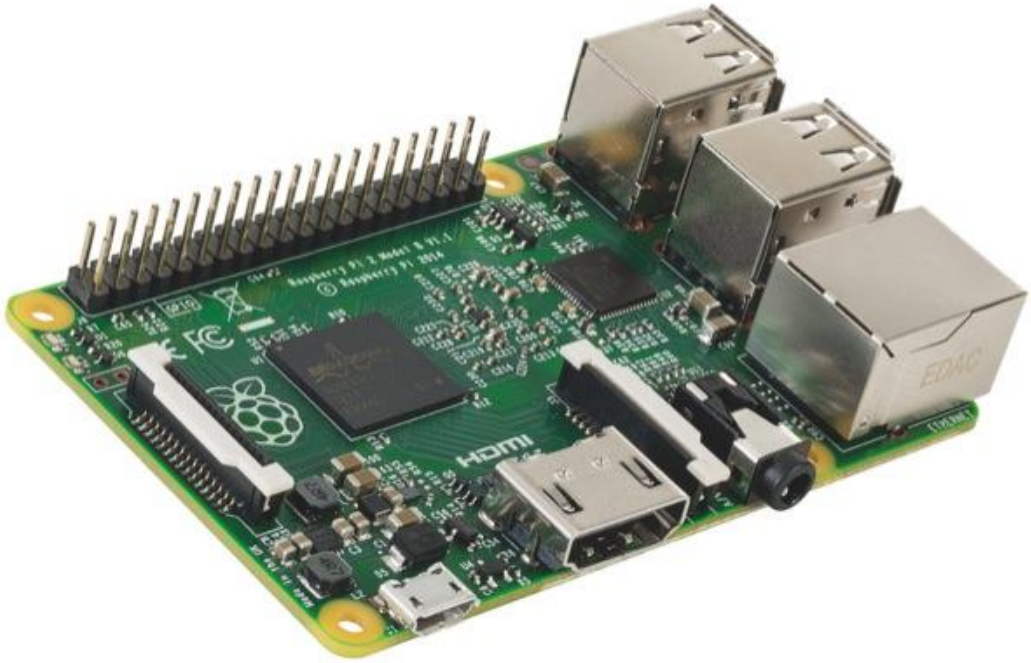
1. Příprava paměťové karty pomocí fdisku
    - sudo fdisk /dev/sdx
2. Vymazání starých oddílů (pokud existují) a vytvoření nových
    - Stiskem klávesy o se smažou staré oddíly
    - Stiskem klávesy p se zobrazí seznam oddílů (v tuto chvíli by měl být prázdný)
    - Stiskem klávesy n se vytvoří nový oddíl, stiskem p se bude jednat o primární, stiskem 1 označíme oddíl jako první, stiskem ENTERu souhlasíme, že se začátek oddílu bude nacházet na defaultním prvním sektoru, napsáním +100M vytvoříme diskový oddíl o velikost 100MB. Potvrdíme ENTERem
    - Stiskem klávesy t a pak c označíme oddíl jako W95 FAT 32 (LBA)
    - Stiskem klávesy n vytvoříme druhý primární oddíl (p) a dvakrát stiskneme ENTER – vytvoří se druhý oddíl na zbytku karty
    - Stiskem klávesy w zapíšeme změny na kartu
3. Vytvoření souborového systému na prvním oddílu
    - sudo mkfs.vfat /dev/sdx1
4. Vytvoření souborového systému na druhém oddílu
    - sudo mkfs.ext4 /dev/sdx2
5. Připojení oddílů
    - mkdir root
    - sudo mount /dev/sdx2 root
    - mkdir root/boot
    - sudo mount /dev/sdx1 root/boot
6. Stažení aktuálního souborového systému a následné rozbalení
    - wget http://archlinuxarm.org/os/ArchLinuxARM-rpi-latest.tar.gz
    - sudo bsdtar -xpf ArchLinuxARM-rpi-latest.tar.gz -C root
    - sync
7. Odpojení karty ze systému

- sudo umount root/boot root

Tímto máme připravený základní systém pro Raspberry Pi. První přihlášení do systému je možné skrze připojený monitor nebo pomocí SSH příkazové řádky. Login a heslo je: root/root.

5.3 Raspberry Pi2, model B

Po velkém úspěchu první verze Raspberry Pi se počátkem tohoto roku objevila na trhu druhá verze. Při zachování stejných rozměrů se u tohoto modelu zvýšila paměť RAM na 1GB a také procesor běží na vyšší frekvenci (900MHz), přičemž disponuje čtyřmi jádry s architekturou ARM v7. Kromě rozměrů se výrobci podařilo zachovat i cenu, která činí 35 USD.

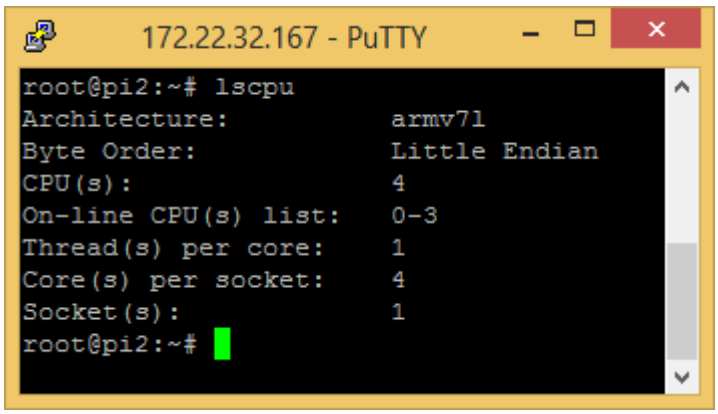


Obr. 15 Raspberry Pi 2 [10]

5.3.1 Technické parametry

Tab. 2 Technické parametry Raspberry Pi 2

SoC	Broadcom BCM2836
CPU	4 x 900 MHz ARMv7, Cortex A7
RAM	1 GB
NETWORK	10/100 Mbit Ethernet



Obr. 16 Raspberry Pi 2 – výpis lscpu

5.3.2 Instalace operačního systému (Arch Linux ARM)

Operační systém se instaluje naprosto shodným způsobem jako u předchozího modelu. Díky novější architektuře procesoru je ale u Arch Linuxu k dispozici optimalizovaná verze souborového systému. Proto se doporučuje její použití, místo kompatibilní předchozí verze. Ke stažení je na adrese: http://archlinuxarm.org/os/ArchLinuxARM-rpi-2-latest.tar.gz

5.4 Android TV box (CS968)

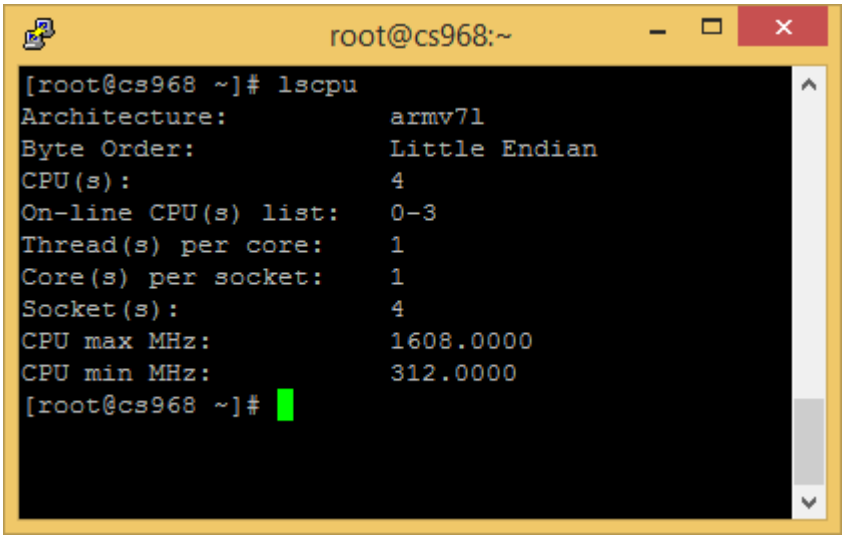
Jedná se o jeden z prvních TV boxů vybavených ethernetovým rozhraním a poměrně výkonným čtyř jádrovým procesorem Rockchip RK3188 pracujícím s taktem 1,6GHz. Je dodáván s operačním systémem Android a výrobce alternativní operační systém jako je Linux nepodporuje. Na webu jsou však návody, jak do zařízení nainstalovat Ubuntu 12.04 LTS. Díky panu Ianu Morrisovi, který stojí za projektem Linuxium, stačí dnes stáhnout předpřipravený image a nahrát jej na micro SD kartu. Pomocí reverzního inženýrství totiž zjistil, jakým způsobem funguje proprietární zavaděč (bootloader) u zařízení, která používají ARM procesory firmy Rockchip. Předpřipravený image však obsahuje jádro, ve kterém chybí mnoho jaderných modulů (např. podpora NFS, podpora málo používaných souborových systémů apod.), proto je potřeba sestavit jádro vlastní, které bude obsahovat moduly, které jsou nutné pro potřeby clusteru.

Obr. 17 CS968 [1]

5.4.1 Technické parametry

Tab. 3 Technické parametry CS968

SoC	Rockchip RK3188
CPU	4x 1,6 GHz ARMv7, Cortex A9
RAM	2 GB
NETWORK	10/100 Mbit Ethernet, WIFI 802.11 b/g/n



Obr. 18 CS968 – výpis lscpu

5.4.2 Příprava linuxového jádra

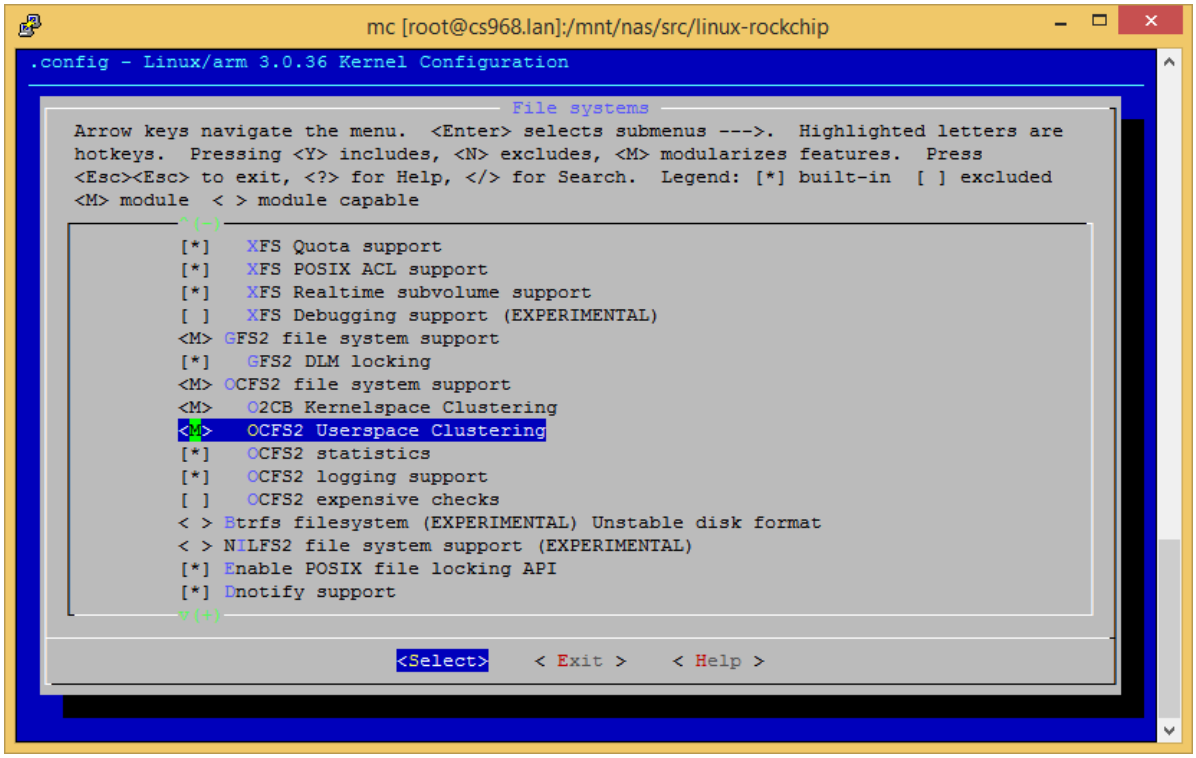
Ke kompilaci jsou potřeba zdrojové kódy linuxového jádra. Výrobci do standardního linuxového jádra často přidávají své zdrojové kódy, aby zajistili správné fungování svých výrobků. Pro procesor RK3188 je možné použít jádro určené pro vývojovou desku Radxa Rock. Stažení jádra:

git clone -b radxa-stable-3.0 https://github.com/radxa/linux-rockchip.git

Po stažení nejnovější verze je třeba spustit konfiguraci jádra, ve které se zvolí, jaké moduly má obsahovat nebo co má obsahovat samotné jádro.

Konfigurace jádra:

make menuconfig



Obr. 19 Konfigurace linuxového jádra

Když je jádro nakonfigurované, kompilace se spustí příkazem „make“, případně se zvolí počet použitých vláken při kompilaci přepínačem „j“.

make –j4

make modules

Jádro se sestaví a nainstalují se moduly.

make modules_install

Specifikum pro Rockchip je, že se jádro musí podepsat pomocí nástroje rkcrc, který je obsažen ve zdrojových kódech. Podepsané jádro se pak nahraje do vnitřní NAND paměti nebo na SD kartu.

5.5 Ubiquiti EdgeMax Lite (ERLite-3)

I když tento router nepatří mezi produkty, které pohání ARM architektura, stojí za zmínku. Procesor je postavený na konkurenční architektuře MIPS64, jež používá zredukovanou instrukční sadu (RISC), stejně jako architektura ARM. Mezi společné vlastnosti patří nízká spotřeba a spolehlivost. Disponuje třemi gigabitovými ethernetovými porty a hlavně hardwareovou akcelerací přepínání paketů, díky čemuž zvládne přepnout milion paketů za sekundu. Spolu s operačním systémem, který vychází z operačního systému Vyatta, potažmo

Debianu, a cenou 99USD z něj dělá velkého konkurenta zavedených značek jako Cisco, Juniper nebo MikroTik.

Standardně balíčky pro Debian nejsou v původním firmwaru obsaženy, respektive chybí definice repositářů. Pomocí následujících příkazů se repositář Debianu Wheezy přidá (v CLI konzoli):

configure

set system package repository wheezy components 'main contrib non-free'

set system package repository wheezy distribution wheezy

set system package repository wheezy url http://http.us.debian.org/debian

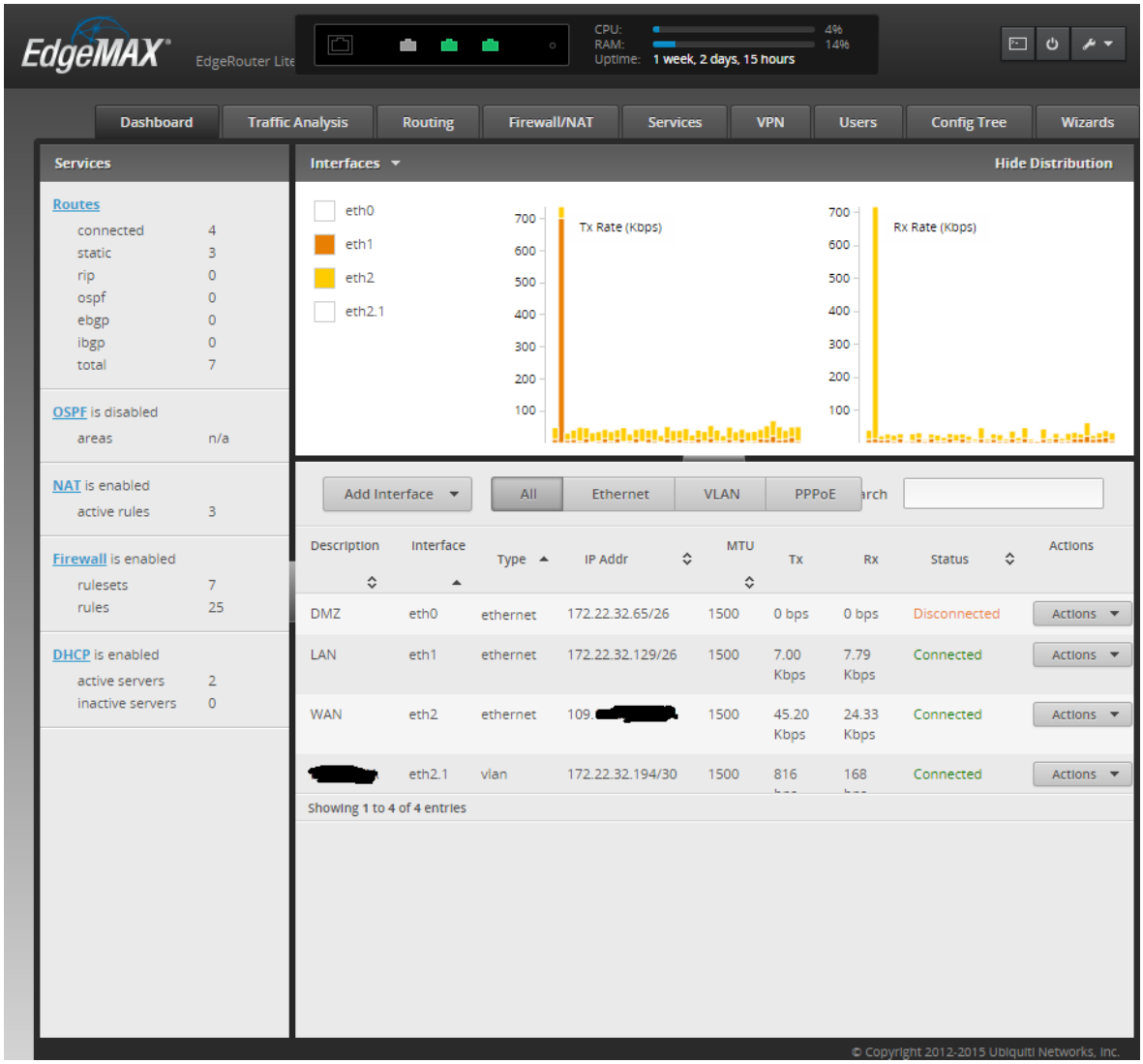
set system package repository wheezy-security components main

set system package repository wheezy-security distribution wheezy/updates

set system package repository wheezy-security url http://security.debian.org

commit;save;exit

Po této úpravě je možné používat balíčkovací systém apt-get. Pozor, není však vhodné aktualizovat celý systém pomocí příkazu apt-get upgrade z toho důvodu, že Ubiquiti používá své systémové balíčky – zoptimalizované pro tento produkt.



Obr. 20 Webové rozhraní Ubiquiti EdgeMax Lite

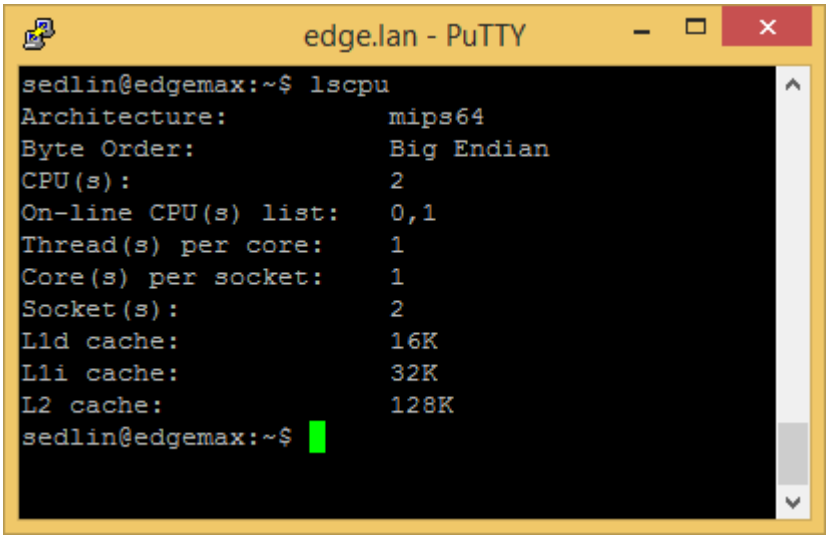
V povedeném grafického rozhraní EdgeMax nabízí i přímý přístup k příkazovému řádku.

Obr. 21 Ubiquiti EdgeMax Lite [9]

5.5.1 Technické parametry

Tab. 4 Technické parametry Ubiquiti EdgeMax Lite

CPU	2x 500 MHz MIPS64
RAM	512 MB
NETWORK	3x 10/100/1000 Mbit Ethernet



Obr. 22 Ubiquiti EdgeMax Lite – výpis lscpu

5.6 ZyXEL NSA325 v2

Výrobce píše o tomto zařízení, že je to multimediální server pro domácí použití a pro použití v menších firmách. Důležité však je, že z technického pohledu se jedná o ARM zařízení a velice dobře poslouží jako NAS server, který bude v clusteru použit jen jako zálohovací zařízení. Výrobce sice k tomuto produktu nenabízí žádnou linuxovou distribuci, nicméně možnost nahradit originální firmware operačním systémem Linux existuje.

Díky tomu, že všechny počítače fungují na stejných principech. Důležité je vědět, že v počítačích existuje tzv. zavaděč systému, který se po spuštění počítače (zařízení) snaží z nějakého úložiště zavést operační systém (firmware) do paměti RAM. Stačí tedy jen nastavit, aby zavedl operační systém z USB flash disku.

Na Internetu [30] existuje i návod, jak to udělat, ale je k tomu potřeba i TTL převodník pro sériovou linku. Pomocí převodníku se dá připojit toto zařízení k počítači a za použití konzole (putty na Windows nebo screen na Linuxu) je možné zavaděč (bootloader) nakonfigurovat.

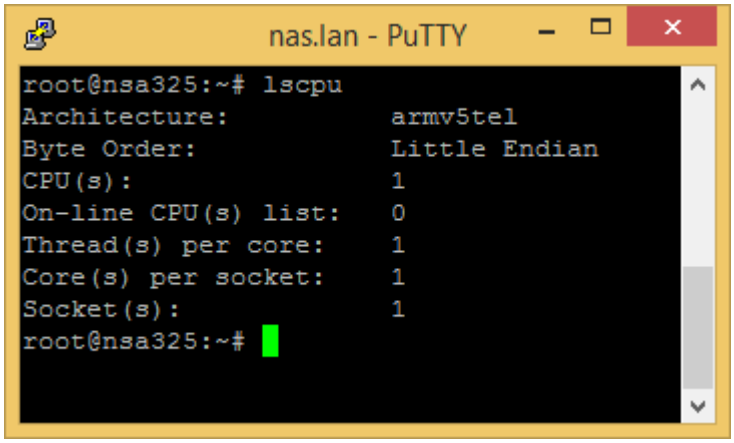
Pro linuxovou distribuci Debian existuje webová nadstavba, která umožňuje i méně zdatným administrátorům jednoduchou správu celého systému pomocí webového rozhraní. Nadstavba se jmenuje OpenMediaVault [31] a je určena především pro správu NAS serverů.

Obr. 23 NSA325 v2 [11]

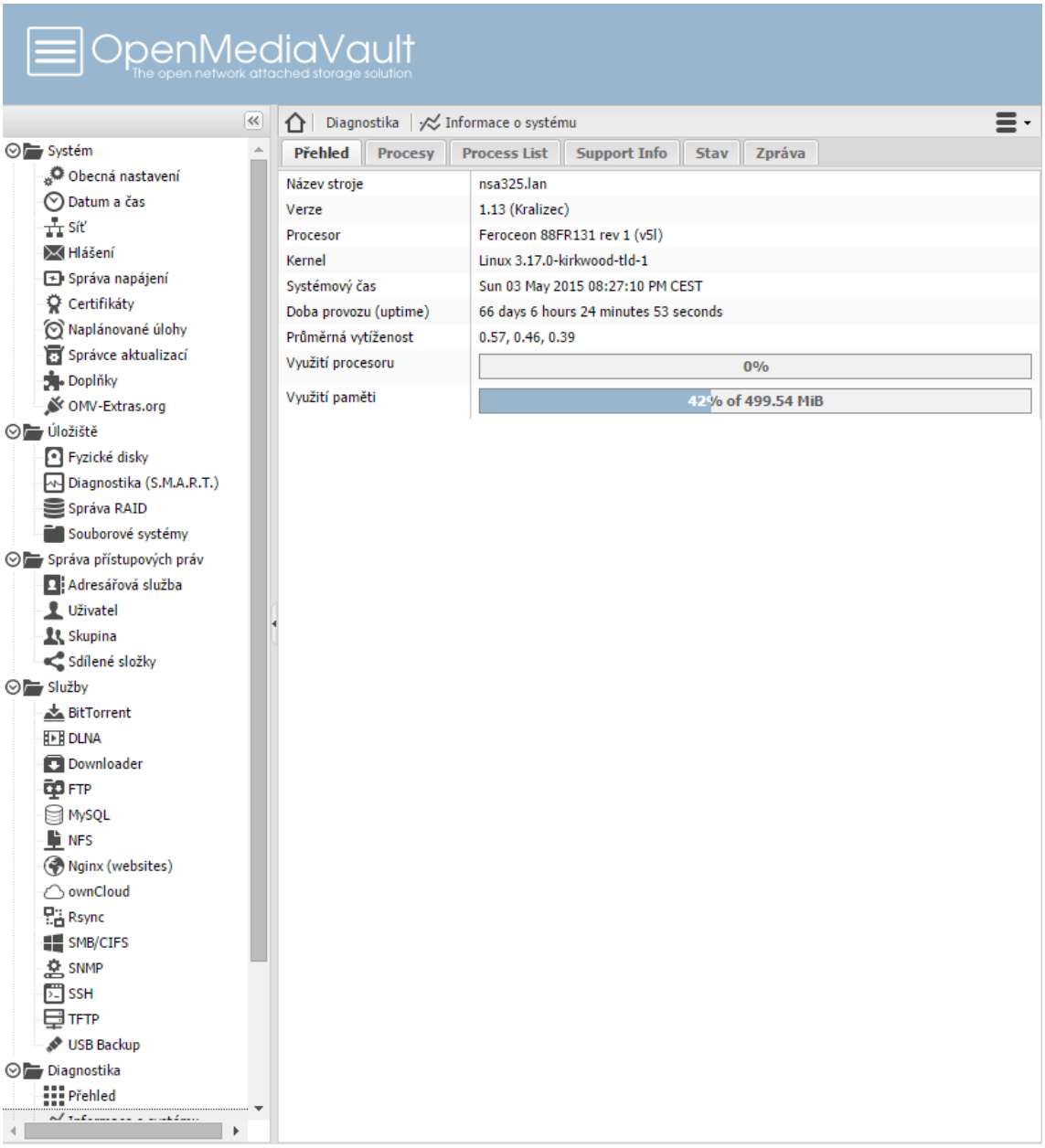
5.6.1 Technické parametry

Tab. 5 Technické parametry NSA325 v2

SoC	Marvell 88F6282 Kirkwood
CPU	1x 1,6 GHz ARMv5TE
RAM	512 MB
NETWORK	3x 10/100/1000 Mbit Ethernet
HDD	2x SATA



Obr. 24 NSA325 v2 – výpis lscpu



Obr. 25 Webové rozhraní OpenMediaVault

Jak lze vidět z obrázku, OpenMediaVault umožňuje nastavit systémový čas, který lze synchronizovat s NTP serverem, lze také konfigurovat síťové rozhraní a je možní si nechat zasílat oznamování událostí pomocí emailů skrze SMTP protokol.

V OpenMediaVaultu je také pokročilá správa disků. Zařízení NSA 325 disponuje dvěma SATA rozhraními, ke kterému se dají připojit klasické SATA disky (2,5 nebo 3,5 palcové). Tyto disky lze také zapojit do diskového pole RAID1, nebo RAID0. K zajištění vyšší spolehlivosti jsou disky zapojeny do RAID1, kde jsou data uloženy na oba dva disky zároveň.

Toto zařízení je osazeno 2x 2TB Western Digital Green disky s celkovou kapacitou 2TB a toto zařízení bude výhradně sloužit jako zálohovací, to znamená, že nebude přímo sloužit potřebám clusteru, ale poslouží velice dobře pro uchování všech dat použitých v clusteru.

Pro představu průměrná rychlost kopírování velkého souboru (4GB) do tohoto zařízení dosahuje reálně 40MB/s.

6 INSTALACE A NASTAVENÍ SERVEROVÝCH APLIKACÍ

V této fázi jsou všechna zařízená vybavena operačním systémem Debian Wheezy a následuje fáze, kdy je potřeba nainstalovat potřebný software, který bude zajišťovat služby v clusteru, což znamená: redundantní úložiště dat, webový server, databázový server a load balancer (rozdělování zátěže).

6.1 Glusterfs

O redundantní úložiště dat se postará Glusterfs, který je potřeba nainstalovat na všechny nody, které budou použity jako webový server.

Pro názornost bude popsána instalace a nastavení Glusterfs jen na dvou nodech:

172.22.32.167 (pi2.lan)

172.22.32.168 (pi.lan)

6.1.1 Instalace

Instalace se provede pomocí standardního instalačního nástroje v Debianu.

sudo apt-get install glusterfs-server

Po nainstalování na Debianu se služba (glusters-server) spustí automaticky a zároveň se přidá i do seznamu služeb, které se mají spustit automaticky i po startu. Pro případné ruční nastartování nebo restartování lze použít příkaz:

/etc/init.d/glusterfs-server start|stop|restart

Když služba běží, je potřeba nody navzájem provázat. Provázání nodů se provede tak, že se vytvoří tzv. Trusted Storage Pool, respektive že se na jednom z nodů přidá do tohoto seznamu node druhý.

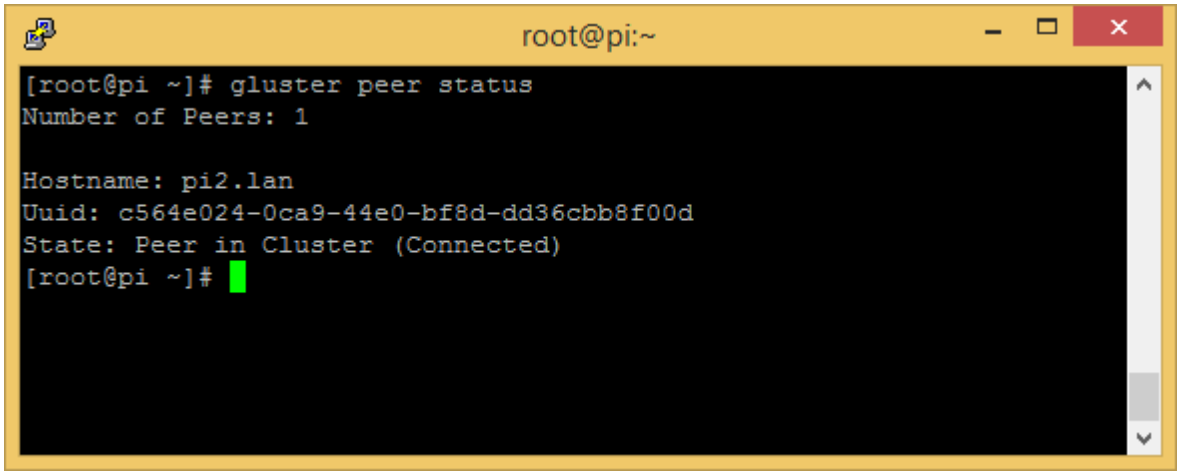
Na nodu pi.lan se spuštěním následujícího příkazu přidá druhý nod pi2.lan do Trusted Storage Poolu:

gluster peer probe pi2.lan

Pro ověření, že jsou oba nody provázané a navzájem o sobě vědí, slouží příkaz:

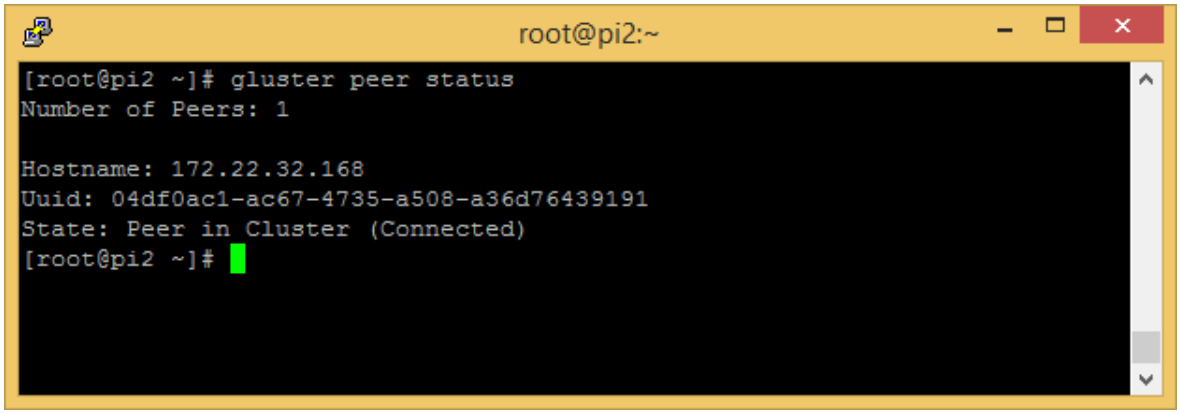
gluster peer status

Na prvním nodu pi.lan příkaz zobrazí:



Obr. 26 Nastavení Glusterfs na prvním nodu

Na druhém nodu pi2.lan příkaz zobrazí:



Obr. 27 Nastavení Glusterfs na druhém nodu

Je tedy vidět, že na druhém nodu pi2.lan se do Trusted Storage Poolu první nod pi.lan (v podobě IP adresy) přidá zcela automaticky.

Když jsou nody provázané, je možné na nich vytvořit libovolný typ Glusterfs svazku. O možných typech svazků pojednává teoretická část práce. V navrženém clusteru bude použit pro zvýšení dostupnosti replikovaný svazek, kde se data budou replikovat na nody v clusteru, na kterých poběží webový server.

6.1.2 Vytvoření replikovaného svazku

Pro vytvoření replikovaného svazku se provede na libovolném nodu v Trusted Storage Poolu příkazem:

gluster volume create wwwgluster replica 2 transport tcp pi.lan:/wwwgluster pi2.lan:/wwwgluster

Není třeba vytvářet žádné adresáře ručně – vše potřebné udělá Glusterfs. Příkazem se vytvořil svazek s názvem wwwgluster, který je fyzicky uložen na nodech (bricks) pi.lan a pi2.lan v adresáři /wwwgluster.

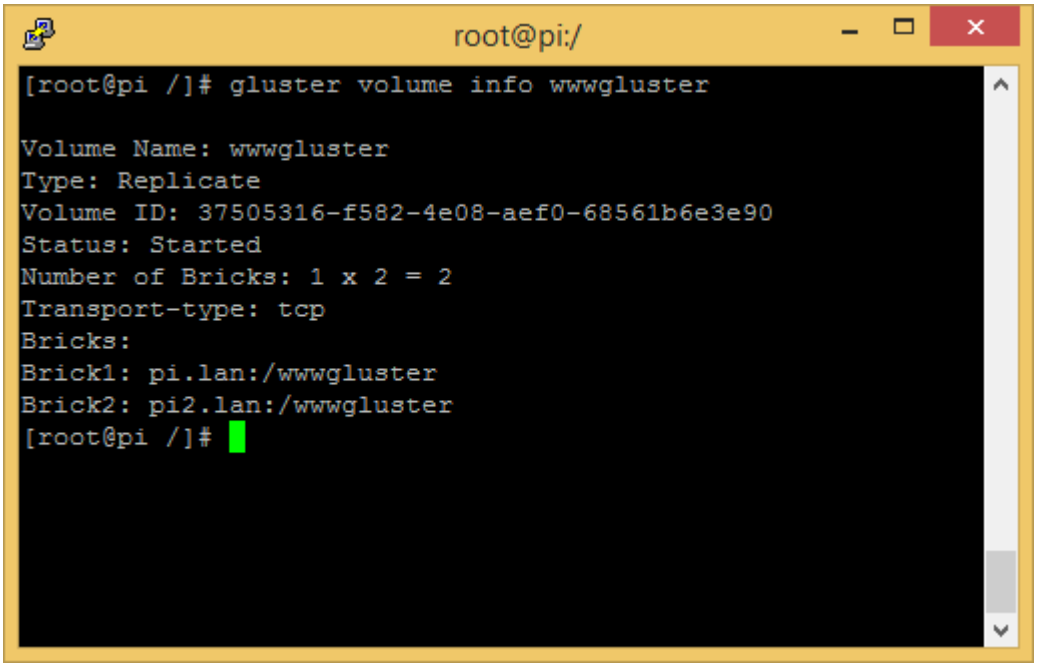
Zajímavé je, že pro Glusterfs dokáže pracovat s existujícím souborovým systémem a že nevyžaduje pro sebe dedikovat celé blokové zařízení.

Po vytvoření se svazek ještě musí nastartovat příkazem:

gluster volume start wwwgluster

Informace o svazku:

gluster volume info wwwgluster



Obr. 28 Glusterfs informace o svazku

Svazek wwwgluster bude využívat webový server, který bude ze svazku číst a možná i zapisovat. Vhodné je i nastavení správného vlastníka svazku. V případě uživatelského účtu, pod kterým je webový server, se jedná o UID a GID 33:

gluster volume set wwwgluster storage.owner-uid 33

gluster volume set wwwgluster storage.owner-gid 33

6.1.3 Přístup ke svazkům

Ačkoli data uložená ve svazku jsou přístupná a dostupná na každém nodu v adresáři wwwgluster, přímo se k nim přistupovat nesmí, protože by správně nefungovala replikace.

6.1.3.1 Nativní klient

Nejsnazší a nejspolehlivější způsob, jak přistupovat ke Glusterfs svazkům, je s použitím nativního klienta. Na nodech, kde je již nainstalovaný server, je automaticky nainstalovaný i klient. Pokud je však potřeba nainstalovat Glusterfs klient i na jiném nodu nebo na klientském počítači, lze jej nainstalovat:

sudo apt-get install glusterfs-client

Pro automatické připojení svazku lze v souboru /etc/fstab přidat řádek:

localhost:/wwwgluster /srv/http/www glusterfs defaults,_netdev 0 0

Pro ruční připojení svazku:

mount -t glusterfs localhost:/wwwgluster /srv/http/www/

Výhoda nativního klienta spočívá také v tom, že dokáže komunikovat s více nody zároveň. V případě svazku wwwgluster se jedná o replikovaný svazek na dvou nodech. Bylo by tedy dobré, kdyby klient měl informaci o obou nodech. Pokud jeden z nodů selže, nativní klient okamžitě naváže komunikaci s druhým nodem.

Nevýhodou ale je, že nativní klient neexistuje pro platformu Windows.

6.1.3.2 NFS protokol

Přístup pomocí NFS protokolu je také poměrně snadný. Stačí nainstalovat balíček:

sudo apt-get install nfs-common

Tím se nainstalují potřebné knihovny pro práci s NFS serverem. Server se nemusí nijak konfigurovat, o všechno se postará služba glusterfs-server.

A i když je soubor /etc/exports prázdný (v Linuxu jsou v tomto souboru definovány složky dostupné přes NFS), příkaz

showmount -e localhost

zobrazí:

Export list for localhost:

/wwwgluster *

Tím pádem svazek wwwgluster je dostupný po síti odkudkoli i přes NFS.

6.1.3.3 CIFS protokol

Protokol CIFS je ve Windows světě známý jako sdílení souborů a složek. Na Linuxu je tento protokol implementovaný v podobě Samby.

Instalace se provede:

sudo apt-get install samba

U Samby je třeba nastavit sdílenou složku pomocí konfiguračního souboru /etc/samba/smb.conf.

Do tohoto souboru je třeba přidat instrukci, že se má zpřístupnit určitá složka. Složka je s tímto nastavením přístupná komukoli. Přístup je zajištěný pod identitou www-data, což je vlastně účet, pod kterým je spuštěn webový server:

[wwwgluster]

comment = Glusterfs www volume

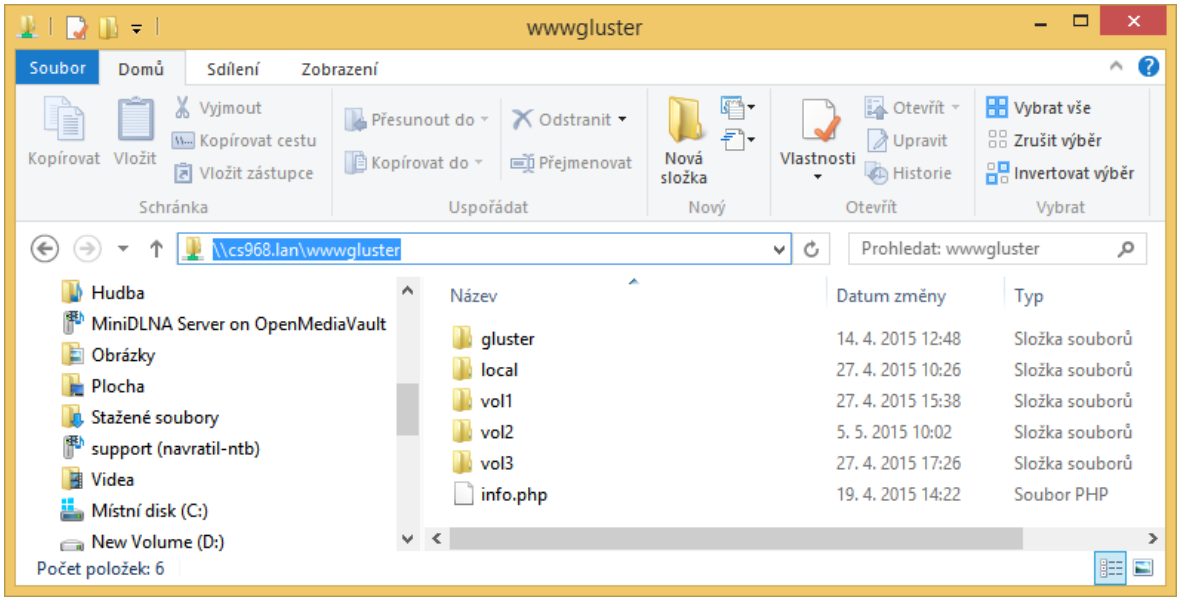
CIFS path = /srv/http/www

read only = no

guest ok = yes

force user = www-data

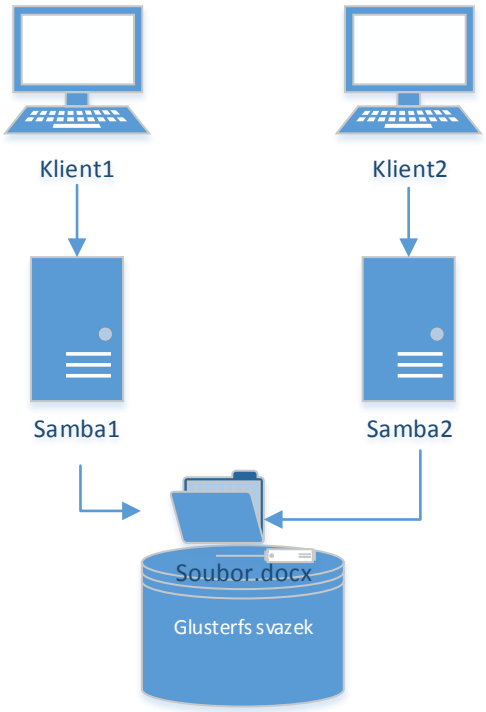
force group = www-data



Obr. 29 Přístup ke Glusterfs svazku z Windows

6.1.3.4 NFS nebo CIFS v clusteru

Speciálním případem zpřístupnění svazku pomocí NFS nebo CIFS protokolů může být zapojení v clusteru, kdy NFS server nebo Samba server mohou být nainstalovány na několik stanic, které zpřístupňují jeden Glusterfs svazek.



Obr. 30 Samba v clusteru

To může být problém, protože jeden server neví o tom, že existuje jiný server, který sdílí stejný svazek. V momentě, kdy jeden klient otevře například dokument ve Wordu, soubor se neuzamkne a nastane kolizní situace.

Proto se musí zajistit určitá vzájemná komunikace mezi Samba nebo NFS servery. U NFS serveru komunikace probíhá, protože je implementovaná v interním NFS serveru samotného Glusterfs. U Samby se to ale musí řešit pomocí CTDB, což je clusterové rozšíření TDB (Trivial Database), která se používá pro uchování informací o uživatelích, zámcích souborů apod.

CTDB navíc zajišťuje i virtuální IP adresy, pomocí kterých řeší fail-over stavy.

Přestože v navrženém clusteru pro webhosting tento způsob zpřístupnění svazku použitý není, je natolik zajímavý, že si zaslouží popsat nastavení – alespoň Samba.

Předpokládejme, že Samba je nainstalovaná.

Zbývá tedy doinstalovat CTDB:

sudo apt-get install ctdb

CTDB databáze musí být uložena na sdíleném Glusterfs svazku. Je možné, aby byla uložena v adresáři, který se přímo sdílí, ale vhodnější je vytvořit si nový svazek. Tento svazek se může použít i pro konfigurační soubory Samby a CTDB.

Pro jistotu se provede záloha původních konfiguračních souborů.

Do souboru public_addresses se napíše seznam IP adres, přes které bude Samba cluster dostupný. Tyto IP adresy jsou plovoucí, to znamená, že CTDB zajišťuje i kontrolu zdraví nodů. Pokud jeden z nodů selže, jeho IP adresa se přiřadí k jinému nodu.

/gluster/lock/public_addresses
192.168.1.101/24 eth0

192.168.1.102/24 eth0
V souboru nodes se definuje seznam skutečných IP adres. Tyto adresy budou sloužit k vzájemné komunikaci nodů.

/gluster/lock/nodes
172.22.32.167

172.22.32.169

Konfigurační soubor samotného CTDB. V něm je mimo jiné uvedeno, že CTDB ovládá Sambu. Rozumí se tím zapínání a vypínání služby Samba. Proto je potřeba vypnout automatické spuštění Samby po startu.

/gluster/lock/ctdb

CTDB_PUBLIC_ADDRESSES=/gluster/lock/public_addresses

CTDB_NODES=/etc/ctdb/nodes

CTDB_MANAGES_SAMBA=yes

CTDB_SET_DeterministicIPs=1

CTDB_SET_RecoveryBanPeriod=120

CTDB_SET_KeepaliveInterval=5

CTDB_SET_KeepaliveLimit=5

CTDB_SET_MonitorInterval=15

Když jsou veškeré konfigurační soubory připraveny, mohou se vytvořit symbolické odkazy – fyzicky jsou konfigurační soubory uloženy na Glusterfs svazku. Tento svazek musí být automaticky připojen po každém startu systému.

mv /etc/defaults/ctdb /etc/defaults/ctdb.orig

mv /etc/samba/smb.conf /etc/samba/smb.conf.orig

ln -s /gluster/lock/ctdb /etc/defaults/ctdb

ln -s /gluster/lock/nodes /etc/ctdb/nodes

ln -s /gluster/lock/public_addresses /etc/ctdb/public_addresses

ln -s /gluster/lock/smb.conf /etc/samba/smb.conf

Po nastartování služby

/etc/init.d/ctdb start

bude Samba dostupná na IP adrese 192.168.1.101 a 192.168.1.102.

6.2 DRBD

DRBD je alternativní způsob, jak zajistit redundantní uložení dat v clusteru. Ačkoli Glusterfs se zdá být jako ideální řešení, DRBD je léty prověřené řešení, které stojí za vyzkoušení.

DRBD, na rozdíl od Glusterfs, pracuje s celým blokovým zařízením, což může být nevýhoda, protože použitá zařízení disponují většinou jen jednou paměťovou kartou s jedním oddílem. Proto se musely použít externí USB flash disky.

Po nainstalování pomoci

sudo apt-get install drbd8-utils

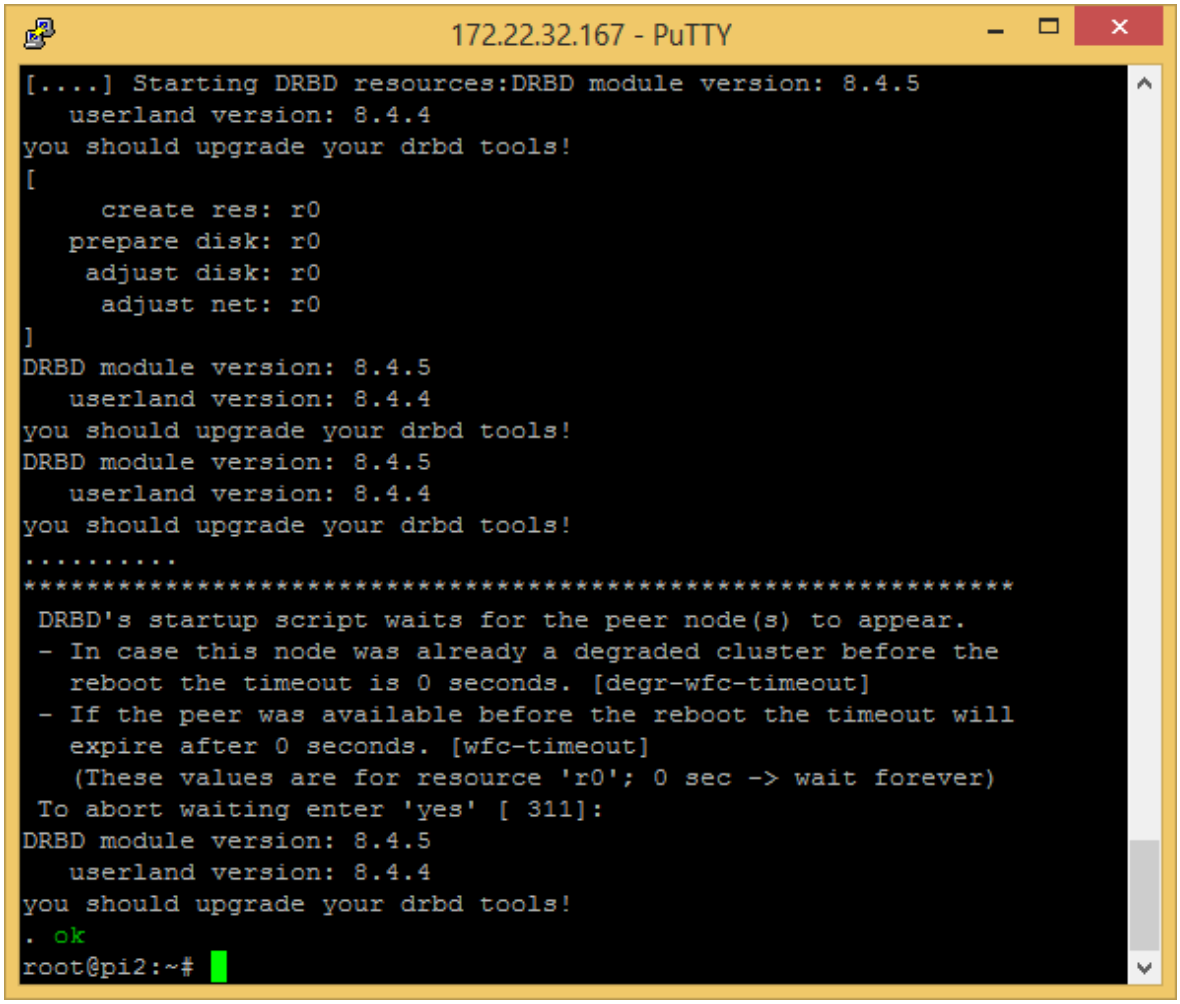
se musí podobně jako u Glusterfs vytvořit replikovaný svazek. Nastavení svazku se konfiguruje pomocí konfiguračního souboru v /etc/drbd.d/. Musí se vytvořit textový soubor, nechť se jmenuje r0.res. Do něj se vloží:

```
resource r0 {
   on pi.lan {
   device    /dev/drbd1;
   disk      /dev/sda1;
   address   172.22.32.168:7789;
   meta-disk internal;
  }
  on pi2.lan {
   device    /dev/drbd1;
   disk      /dev/sda1;
   address   172.22.32.167:7789;
   meta-disk internal;
   }
}
```

Služba DRBD se nastartuje příkazem

/etc/init.d/drbd start

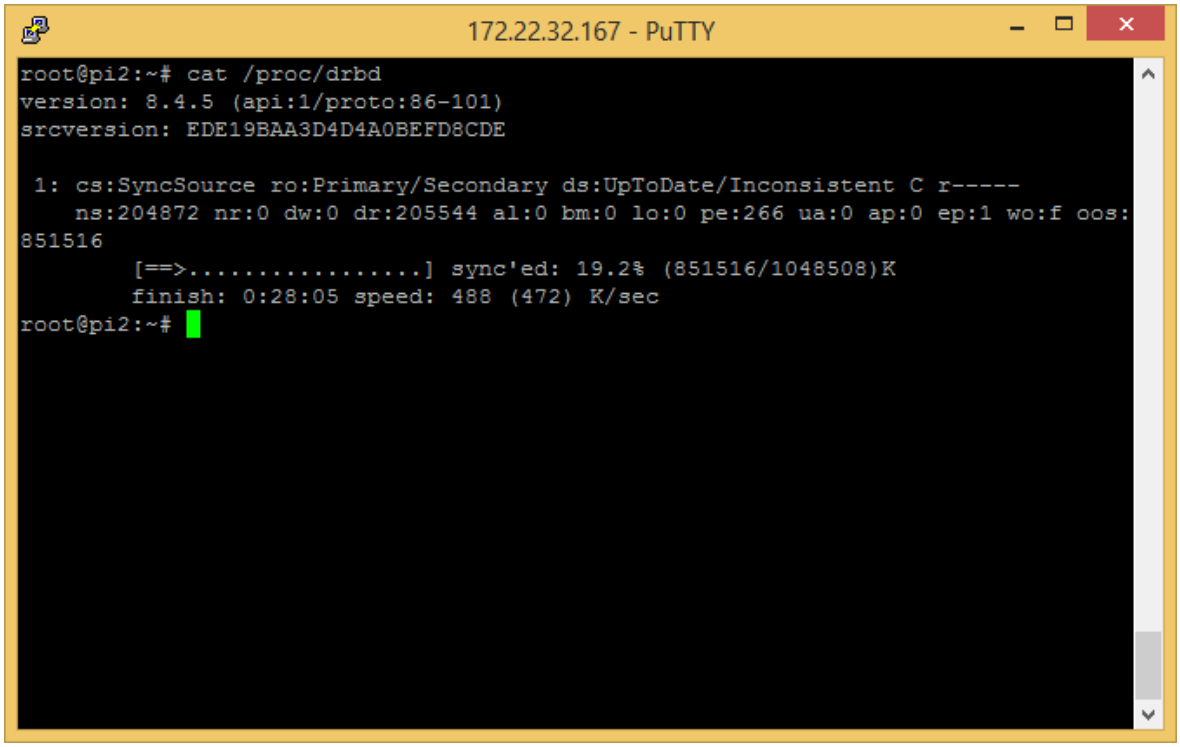
Bohužel po prvním nastavení DRBD služba nefungovala tak, jak se očekávalo. Přestože byla služba spuštěna na obou nodech, DRBD stále čekala na navázání komunikace se svým protějškem. Navíc se po startu DRBD objevovala zpráva, že DRBD modul je novější, než DRBD tools.



Obr. 31 Komplikace s DRBD

Až po reinstalaci drbd8-tools balíčku a několika restartech DRBD začala pracovat.

Nemilým překvapením však byla rychlost synchronizace dat mezi nody. Rychlost dosahovala pouhých 742K/sec.



Obr. 32 Synchronizace pomocí DRBD

6.3 Webový server Apache

Jako jedno z řešení webového serveru bude použit http server Apache. Instalace se provede příkazem:

sudo apt-get install apache2

Všechny konfigurační soubory jsou umístěny v /etc/apache2. Nejdůležitější nastavení se provádí v souboru /etc/apache2/apache2.conf.

Apache je ve výchozím nastavení nastaven tak, že podporuje tzv. virtuální servery, což znamená, že na jednom zařízení může běžet několik webových serverů na různých portech nebo dostupných pod odlišnými jmennými záznamy. Například jeden server může běžet na adrese http://pi.lan, jiný server na adrese http://pi.lan:8080 nebo www.mujcluster.cz.

V Debianu je připravený adresář /etc/apache2/sites-available, kde jsou umístěny konfigurační soubory pro dostupné virtuální servery, a adresář /etc/apache2/sites-enabled, kde jsou umístěny symbolické odkazy na dostupné virtuální servery. Je to poměrně jednoduchá technika, která administrátorům umožňuje snadno a rychle virtuální server vypnout nebo zapnout.

Pro navržený cluster je třeba vytvořit alespoň jeden virtuální server. Ve složce /etc/apache2/sites-available se vytvoří textový soubor, např. cluster, a do něj se vloží následující obsah:

```
<VirtualHost *:10081>
    ServerAdmin webmaster@localhost
    DocumentRoot /srv/http/wwwgluster
    ErrorLog ${APACHE_LOG_DIR}/error.log
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Tím se vytvoří virtuální server, který bude dostupný na portu 10081 a dokumenty – webové stránky budou umístěny v adresáři /srv/http/wwwgluster. Připomeňme si, že tento adresář je přípojným bodem Glusterfs, který je synchronně replikován.

Když je vytvořená konfigurace virtuálního serveru, musí se nalinkovat do povolených virtuálních serverů:

```
ln -s /etc/apache2/sites-available/cluster /etc/apache2/sites-enabled/cluster
```

V Debianu u Apache se ještě musí explicitně povolit nově používaný port 10081 v konfiguračním souboru /etc/apache2/ports.conf. Do souboru se přidají dva řádky:

```
NameVirtualHost *:10081
Listen 10081
```

Po restart serveru Apache příkazem

```
/etc/init.d/apache2 restart
```

je obsah adresáře /srv/http/wwwgluster dostupný pomocí HTTP protokolu.

6.3.1 Nativní podpora PHP

Standardně není podpora PHP ani jiných skriptovacích jazyků spolu s Apache nainstalována. Pokud je potřeba podporu PHP mít, musí se doinstalovat. Apache umí PHP skripty interpretovat nativně a stačí nainstalovat potřebné knihovny:

```
sudo apt-get install libapache2-mod-php5
```

Tím se nainstaluje i celý PHP balík. Konfigurační soubory pro PHP jsou umístěny v adresáři /etc/php5.

Nyní, když se apache restartuje, bude umět interpretovat PHP skripty.

Pokud se tedy v adresáři /srv/http/wwwgluster vytvoří jednoduchý skript s názvem info.php a do něj se vloží:

<?php phpinfo(); ?>

Webový server Apache skript bude interpretovat na adrese http://pi.lan:10081/info.php tak, jak je zobrazuje následující obrázek vypsáním informací o PHP. Informací o PHP je na několik stran, nicméně pro demonstraci postačí jen část.

System	Linux pi.lan 3.18.7+ #755 PREEMPT Thu Feb 12 17:14:31 GMT 2015 armv6l
Build Date	Mar 29 2015 15:13:44
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-curl.ini, /etc/php5/apache2/conf.d/20-gd.ini, /etc/php5/apache2/conf.d/20-mcrypt.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini
PHP API	20100412
PHP Extension	20100525

Obr. 33 PHP info s Apache

6.4 Webový server NGINX

Pro srovnání bude použit i webový server NGINX, který, jak je popsáno v teoretické části, principiálně funguju trochu jinak a měl by být méně náročný v konzumaci operační paměti a v určitých případech i rychlejší.

Instalace se provede příkazem:

sudo apt-get install nginx

Konfigurační soubory jsou umístěny v adresáři /etc/nginx a i tady jsou v Debianu připravené adresáře /etc/nginx/sites-available a /etc/nginx/sites-enabled.

Opět vytvořením konfiguračního souboru v /etc/nginx/sites-available s názvem cluster definujeme nový virtuální server:

```
server {
    listen   10082;
    root /srv/http/wwwgluster;
    index index.html index.htm;
    server_name localhost;
    location / {.
        try_files $uri $uri/ /index.html;
    }
    location ~ /\.ht {
        deny all;
    }
}
```

Virtuální server bude naslouchat na jiném portu, než na kterém naslouchá virtuální server v Apache, tím vzniká možnost paralelního běhu – na nodu se stane dostupným jak webový server Apache, tak i webový server NGINX. Všimněme si, že direktiva root odkazuje na stejný adresář, který je použitý s webovým serverem Apache.

Vytvořením symbolického odkazu v /etc/nginx/sites-enabled a restartováním webového serveru se virtuální server povolí:

ln -s /etc/apache2/sites-available/cluster /etc/apache2/sites-enabled/cluster

/etc/init.d/nginx restart

6.4.1 Podpora PHP s PHP-FPM

NGINX nativně neumí interpretovat PHP jazyk, proto při zpracování PHP skriptů musí volat PHP-FPM (FastCGI Process Manager), což je nezávislý doplněk k webovým serverům,

které interpretaci neumějí. Může se však použít i jako doplněk k Apache serveru, který nativní interpretaci sice umí, ale s použitím PHP-FPM může PHP skripty interpretovat rychleji. Rychlejší interpretace se může dosáhnout např. uložením dotazů, které se často za sebou opakují, do paměti.

Instalace PHP-FPM:

sudo apt-get install php-fpm

Konfigurační soubory jsou umístěny v /etc/php5/fpm. PHP-FPM může naslouchat na unix socketu (standardně je socket umístěn ve /var/run/php5-fpm.sock), nebo na TCP/IP sockete, čímž vzniká možnost nechat si interpretovat PHP skripty na jiném serveru. Rychlejší je ale klasický unix socket a pro potřebu clusteru je to i vhodnější způsob.

Když je služba php-fpm nainstalována, je potřeba ji spustit:

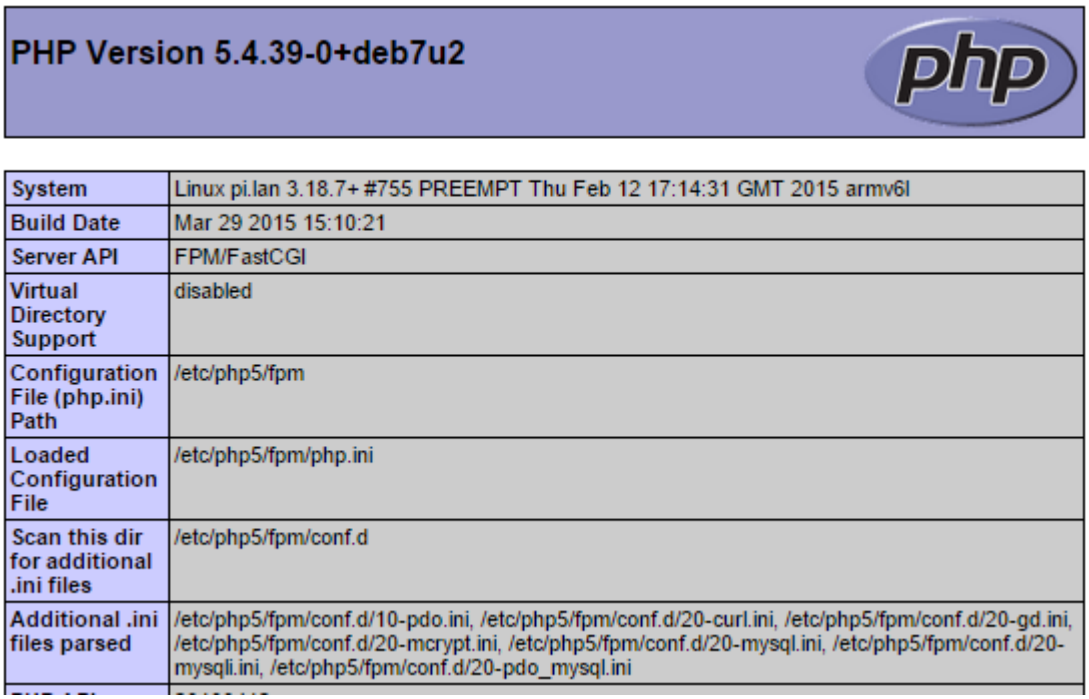
/etc/init.d/php-fpm start

Nyní se musí modifikovat konfigurace virtuálního serveru, kde je potřeba uvést, že NGINX může interpretovat PHP skripty pomocí PHP-FPM.

Mezi direktivu server{} se musí přidat direktiva location{}, která říká, že všechny soubory s příponou php se mají interpretovat zavoláním PHP-FPM. Konfigurace NGINX se může zdát pro někoho poměrně nepřehledná. Mohou za to snad regulární výrazy, které jsou v konfiguracích hojně využívány.

```
location ~ \.php$ {
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_index index.php;
        include fastcgi_params;
    }
```

Po opětovném restartu server NGINX začne fungovat podpora PHP. Podobně jako u serveru Apache, zobrazením adresy http://pi.lan:10082/info.php se zobrazí podobná stránka.



PHP Version 5.4.39-0+deb7u2

System	Linux pi.lan 3.18.7+ #755 PREEMPT Thu Feb 12 17:14:31 GMT 2015 armv6l
Build Date	Mar 29 2015 15:10:21
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/fpm
Loaded Configuration File	/etc/php5/fpm/php.ini
Scan this dir for additional .ini files	/etc/php5/fpm/conf.d
Additional .ini files parsed	/etc/php5/fpm/conf.d/10-pdo.ini, /etc/php5/fpm/conf.d/20-curl.ini, /etc/php5/fpm/conf.d/20-gd.ini, /etc/php5/fpm/conf.d/20-mcrypt.ini, /etc/php5/fpm/conf.d/20-mysql.ini, /etc/php5/fpm/conf.d/20-mysqli.ini, /etc/php5/fpm/conf.d/20-pdo_mysql.ini

Obr. 34 PHP info s PHP-FPM

V kolonce Server API není uveden NGINX, ale FPM/FastCGI.

6.4.2 NGINX jako HTTP load balancer

Webový server NGINX může sloužit i jako load blancer v navrženém clusteru. Teoreticky by mohl dobře posloužit i Apache, ale díky tomu, že je NGINX skromný na paměť, je pro ARM architekturu, respektive pro použitá zařízení, vhodnější.

Konfigurační soubory jsou stejné jako u klasického nastavení. Virtuální server (load balancer) bude naslouchat na portu 10082 a bude přeposílat požadavky na webové servery v clusteru. V direktivě upstream se definuje seznam webových serverů, případně se zvolí, jakým způsobem se bude přerozdělování zátěže provádět. V ukázkovém konfiguračním souboru je uveden „ip_hash“, což zaručuje, že si load balancer bude pamatovat hashe (otisky) IP adres, ze kterých budou přicházet požadavky, a bude všechny další požadavky ze stejných IP adres směrovat na jeden server v clusteru. Tím se vytvoří na určitou dobu persistentní spojení mezi klientem a webovým serverem v clusteru. Kdyby load balancer směroval požadavky pokaždé na jiný server v clusteru, nepřežily by session proměnné.

Záleží však na povaze webové aplikace nebo obsahu, který má být dostupný pomocí webového serveru, jestli jsou důležité session proměnné, a podle toho se může zvolit vhodný algoritmus rozdělování zátěže.

Ukázková konfigurace pro loadbalancer /etc/nginx/sites-available/lb:

```
upstream cluster_nginx {
   ip_hash;
   server cs968.lan:10082;
   server pi2.lan:10082;
   server pi.lan:10082 backup;
}
server {
   listen 10082;
   location / {
      proxy_pass http://cluster_nginx;
      proxy_set_header Host $host:10082;
   }
}
```

NGINX nabízí tři algoritmy rozdělování zátěže:

1. Round-Robin – je vyhrazena určitá doba směrování na první server, druhý a další server.
2. Least-Conencted – požadavek je přesměrován na server s nejmenší aktivitou.
3. IP-Hash – požadavky směrovány na servery podle příchozí IP adresy.

Kromě možných algoritmů NGINX umožňuje vážené rozdělování zátěže pomocí definice vah u jednotlivých webových serverů, monitoring stavů serverů, definici backup serverů, které se mají využít jen v případě, že všechny hlavní servery selžou apod.

6.5 Databáze MySQL

Webhosting je místo přístupné přes Internet, kde si uživatel může uložit své soubory a zpřístupnit je ostatním. Někdy kromě tohoto místa je potřeba i databáze, ve které jsou data uložena podle určitého řádu. K těmto datům musí být rychlý přístup.

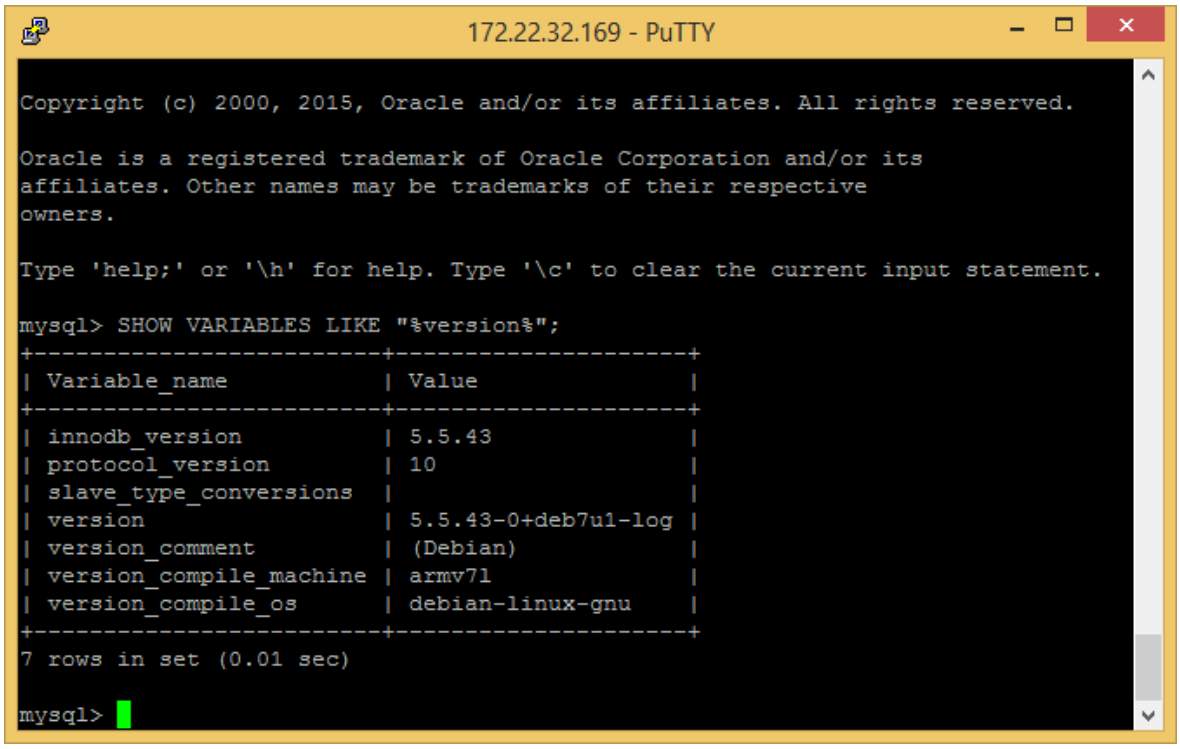
Instalace se provede příkazem:

sudo apt-get install mysql-server

Jelikož PHP skripty budou chtít k databázi také přistupovat, musí se doinstalovat knihovna:

sudo apt-get install php5-mysql

V Debian Wheezy je databázový server MySQL distribuovaný ve verzi 5.5, který standardně neumožňuje automatické přepínání režimu MASTER/SLAVE [22]. V této verzi se mohou data jen replikovat na jeden nebo více SLAVE databázových serverů. Pokud by přesto bylo zapotřebí servery přepínat, musí se použít vlastní skripty, které budou zajišťovat monitoring serverů a v případě výpadku SLAVE povýší na MASTER. Je také možné použít řešení pomocí Hearbeat – to ale není tak elegantní jako vestavěná funkce v nových verzích MySQL.



Obr. 35 MySQL v Debianu Wheezy

Nicméně už v následující verzi 5.6 je možné MASTER/SLAVE servery přepínat pomocí podpůrného nástroje „mysqlfailover“ poměrně elegantně bez nutnosti užití zastaralých řešení nebo vlastních skriptů.

Jinou alternativou se může zdát použití databázového serveru MariaDB, ta ale není v Debianu standardně obsažena a je potřeba nejprve přidat repositáře. V současné době MariaDB nabízí k použití dvě verze: 5.5 a 10.0.

Nutno dodat, že MySQL a MariaDB se v něčem liší. Jedna z věcí, v čem se právě liší, je i způsob replikování. V MariaDB se synchronní replikací zabývá projekt Galera, která v klasickém MySQL obsažena není.

V Arch linuxu databázový server MariaDB již nahradil MySQL, proto se dá předpokládat, že i Debian nabízí alespoň možnost zvolit si MariaDB. Nikoli. V Debianu Wheezy nejsou připravené balíčky s MariaDB. Samozřejmě díky tomu, že MariaDB je open source, jsou k dispozici zdrojové kódy a teoreticky není problém si binární balíčky vytvořit jejich kompilací. Zkušenost však říká, že kompilace někdy může selhat vinou chybějících závislostí, nebo ještě hůře – závislostí, které jsou nainstalovány, ale nejsou ve správné verzi. Pokud se kompilace přesto povede, případné budoucí bezpečnostní aktualizace nebudou nainstalovány.

Vzhledem k situaci ohledně databázového serveru se bude muset použít nastavení MASTER/SLAVE. To ovšem nevadí, protože k hostování statických souborů (například zdrojové kódy, obrázky, videa) databáze není potřeba.

Pokud by webhostingový cluster přeci jen vyžadoval databázový server v nastavení MASTER/MASTER , může být použitý jiný operační systém – novější verze Debianu (Jessie) nebo již zmiňovaný ArchArm linux.

6.5.1 Nastavení Master serveru

V implicitní konfiguraci je MySQL server nastaven tak, že naslouchá pouze na loopback adrese (127.0.0.1), což znamená, že server není přístupný ze sítě. Proto je tento řádek v konfiguraci /etc/mysql/my.cnf třeba zakomentovat a umožnit MySQL serveru, aby byl dostupný i po síti LAN, potažmo aby naslouchal na všech adresách.

[mysqld]

#bind-address = 127.0.0.1

Aby MySQL server věděl, že má fungovat v Master nastavení, je potřeba v konfiguračním souboru přidat parametry:

log-bin

server-id=1

Změna nastavení se projeví až po restartu MySQL serveru příkazem:

/etc/init.d/mysql restart

Po instalaci je třeba databázi inicializovat – musí se vytvořit prázdná databáze. Spuštěním příkazu:

mysql_secure_installation

se v konzoli spustí průvodce, který spustí inicializaci.

Nyní je server připraven pro replikaci. Před replikací se musí SLAVE server připojit a oznámit MASTER serveru, že je připraven. SLAVE server se připojí a přihlásí pomocí uživatele - identity, která musí existovat na MASTER serveru.

Pokud identita neexistuje, dá se vytvořit pomocí SQL příkazu. Nejsnazší způsob, jak vykonat SQL příkazy nabízí konzole. Pomocí příkazu:

mysql -uroot –p

se spustí SQL klient, který pod identitou root může vykonávat jakékoli SQL dotazy.

Zadáním SQL dotazu:

CREATE USER ‚uzivatel‘@‘%‘ IDENTIFIED BY ‚heslo‘;

se vytvoří identita s uživatelským jménem „uživatel“, heslem „heslo“ a možností připojit se k SQL serveru odkudkoli – to značí znak „%“.

Vytvořená identita zatím nemá žádná oprávnění, musí se tedy nastavit dalším SQL dotazem:

*GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'uzivatel'@'%' IDENTIFIED BY 'heslo';*

Pro aktualizaci uživatelských oprávnění je třeba ještě spustit SQL dotaz:

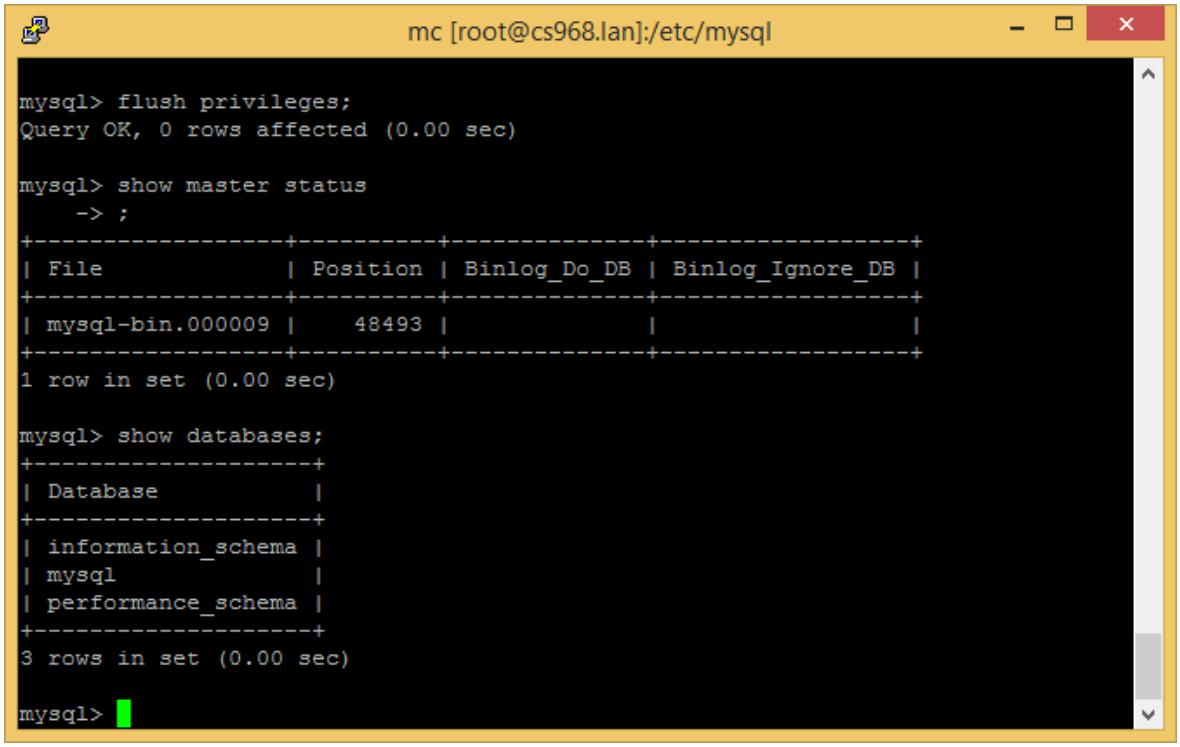
FLUSH PRIVILEGES;

Tímto se nastavilo vše potřebné pro replikaci na MASTER serveru.

Pro ověření nastavení poslouží SQL dotaz:

SHOW MASTER STATUS;

který zobrazí následující tabulku:



Obr. 36 Nastavení replikace v MySQL - MASTER

6.5.2 Nastavení Slave serveru

Nastavení SLAVE serveru je značně jednodušší. Po nainstalování MySQL serveru je třeba se připojit pomocí mysql klienta, podobně jako na MASTERu a vykonat následující SQL dotaz:

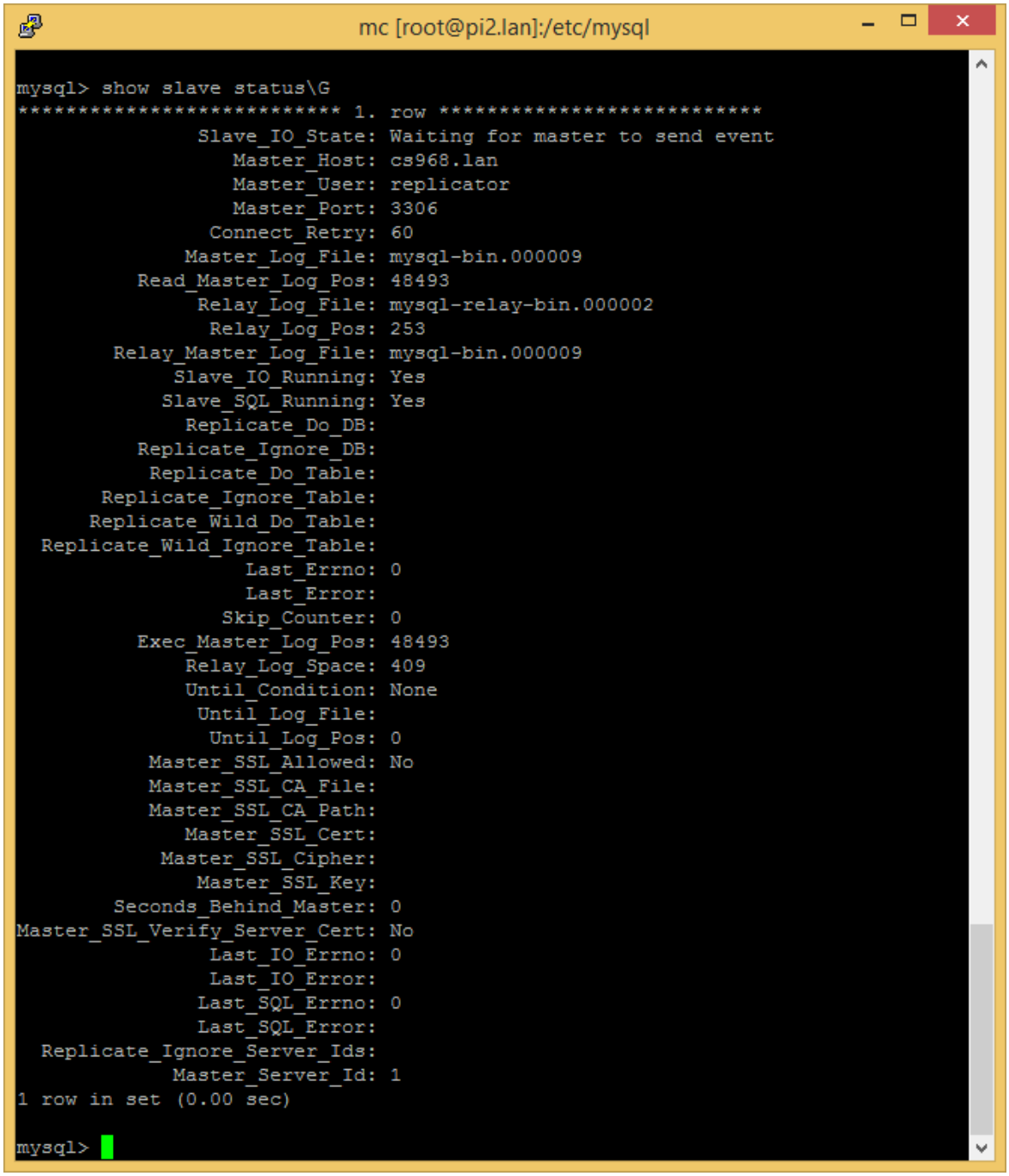
SLAVE STOP;

CHANGE MASTER TO MASTER_HOST='172.22.32.169', MASTER_USER='uzivatel', MASTER_PASSWORD='heslo', MASTER_LOG_FILE='mysql-bin.000009', MASTER_LOG_POS=49483;

SLAVE START;

SHOW SLAVE STATUS\G

V dotazu se použije identita, která je vytvořena na MASTER serveru, vepíše se správná IP adresa nebo jméno MASTER serveru a důležité je také definovat parametry MASTER_LOG_FILE a MASTER_LOG_POS, které určují počáteční fázi replikace.



Obr. 37 Nastavení replikace v MySQL - SLAVE

Tímto se nastavila MASTER/SLAVE replikace na MySQL serveru. Pokud se například vytvoří databáze, tabulka nebo se vloží nějaká data na MASTER serveru, za malou chvíli se vše zreplikuje na SLAVE serveru.

6.6 HAProxy

HAProxy je další implementace aplikace, která provádí rozdělování zátěže, a další alternativa do navrženého clusteru. Kromě rozdělování zátěže HTTP protokolu, který funguje na sedmé vrstvě OSI modelu, si poradí i s rozdělováním zátěže na čtvrté vrstvě OSI modelu. Práce se zmiňuje v předchozích kapitolách o Samba clusteru, u nějž by se rozdělování zátěže mohl postarat DNS round-robin algoritmus, nicméně pomocí HAProxy by se mohlo zajistit efektivnější rozdělování zátěže.

HAProxy disponuje kromě klasických algoritmů round-robin, leastconn nebo source (i NGINX se nazývá ip-hash).

Pro HTTP rozdělování zátěže to jsou:

- uri – levá část URI je hešována a podělena celkovou vahou dostupných serverů.
- url-param – specifický parametr je vyhledán v každém GET požadavku
- hdr(name) – je vyhledán specifický parametr v hlavičce HTTP požadavku

Instalace se provede příkazem:

sudo apt-get haproxy

Konfigurační soubor jsou umístěny v /etc/haproxy. Pro nastavení clusteru

frontend cluster

bind 172.22.32.129:10081

mode http

default_backend cluster

backend cluster

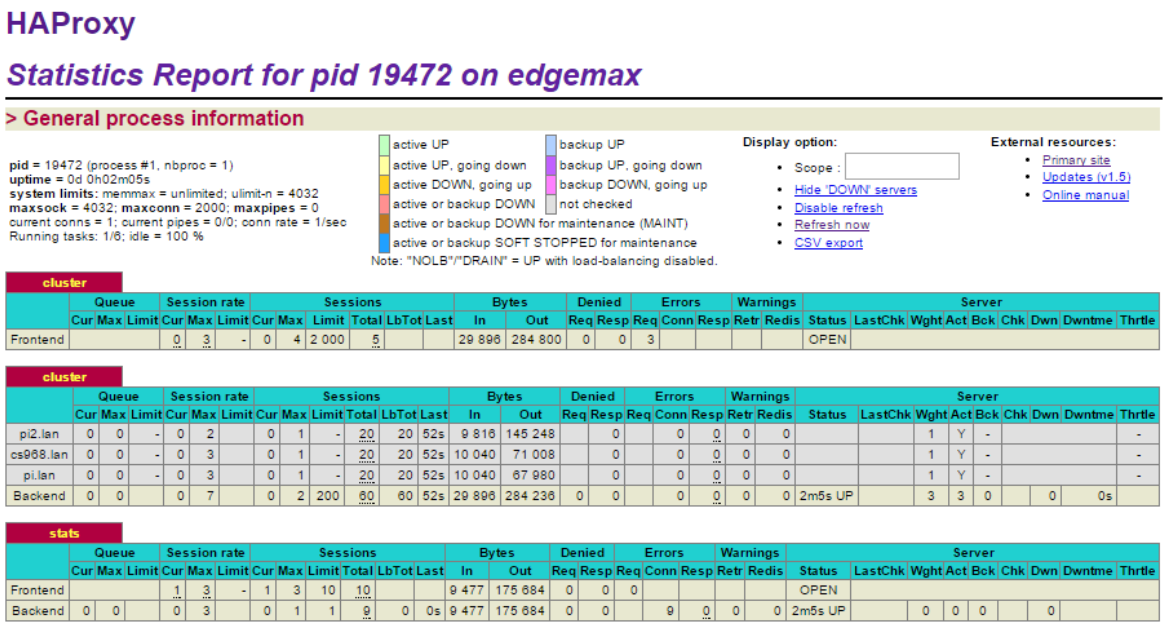
mode http

balance roundrobin

server pi2.lan 172.22.32.167:10082

server cs968.lan 172.22.32.169:10082

server pi.lan 172.22.32.168:10082



HAProxy

Statistics Report for pid 19472 on edgemax

> General process information

pid = 19472 (process #1, nbproc = 1)
uptime = 0d 0h02m05s
system limits: memmax = unlimited; ulimit-n = 4032
maxsock = 4032; **maxconn** = 2000; **maxpipes** = 0
current conns = 1; current pipes = 0/0; conn rate = 1/sec
Running tasks: 1/6; idle = 100 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance
backup UP
backup UP, going down
backup DOWN, going up
not checked
Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:
- Scope :
- Hide 'DOWN' servers
- Disable refresh
- Refresh now
- CSV export

External resources:
- Primary site
- Updates (v1.5)
- Online manual

cluster

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	3	-	0	4	2 000	5			29 896	284 800	0	0	3					OPEN								

cluster

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
pi2.lan	0	0	-	0	2		0	1	-	20	20	52s	9 816	145 248		0		0	0	0	0			1	Y	-				-
cs968.lan	0	0	-	0	3		0	1	-	20	20	52s	10 040	71 008		0		0	0	0	0			1	Y	-				-
pi.lan	0	0	-	0	3		0	1	-	20	20	52s	10 040	67 980		0		0	0	0	0			1	Y	-				-
Backend	0	0		0	7		0	2	200	60	60	52s	29 896	284 236	0	0		0	0	0	0	2m5s UP		3	3	0		0	0s	

stats

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				1	3	-	1	3	10	10			9 477	175 684	0	0	0					OPEN								
Backend	0	0		0	3		0	1	1	9	0	0s	9 477	175 684	0	0		9	0	0	0	2m5s UP		0	0	0		0		

Obr. 38 Statistiky HAProxy

7 VÝKONOVÉ TESTY

Test propustnosti síťových rozhraní na zařízení

7.1 Iperf

Program iperf se používá na měření maximální propustnosti TCP a UDP protokolu. Používá se také k optimalizaci nastavení těchto protokolů.

Instalace:

sudo apt-get install iperf

Spuštění iperf v režimu server:

sudo iperf –s

Zobrazí se informace, že server naslouchá na portu 5001 protokolu TCP:

Server listening on TCP port 5001

TCP window size: 85.3 KByte (default)

Spuštění iperf v režimu klient:

sudo iperf –c IP_ADRESA_SERVERU

po spuštění příkazu se zahájí měření. Výsledek může vypadat následovně:

Client connecting to edge.lan, TCP port 5001

TCP window size: 43.8 KByte (default)

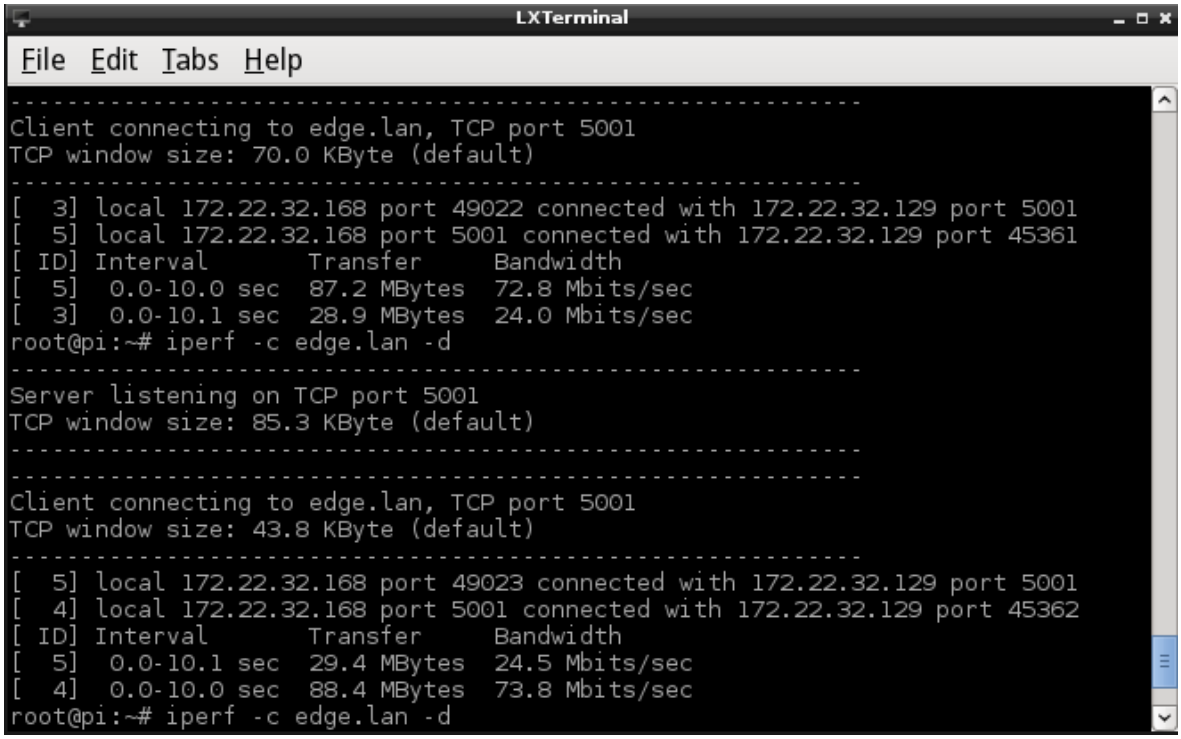
---

[3] local 172.22.32.167 port 43460 connected with 172.22.32.129 port 5001

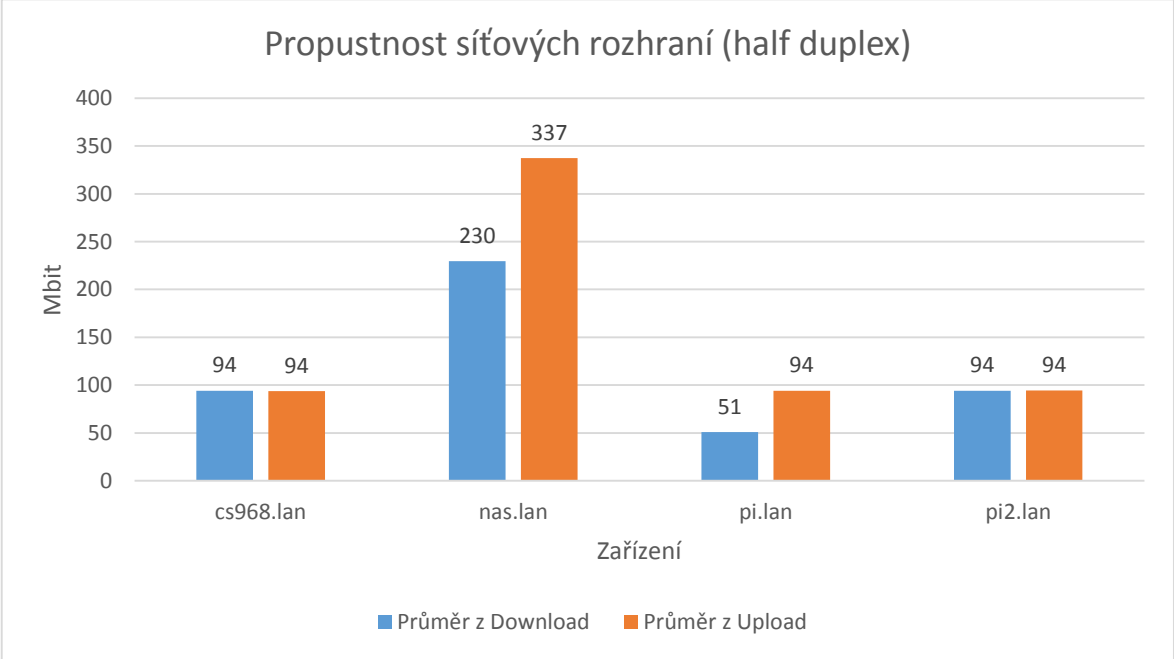
[ID] Interval Transfer Bandwidth

[3] 0.0-10.0 sec 113 MBytes 94.3 Mbits/sec

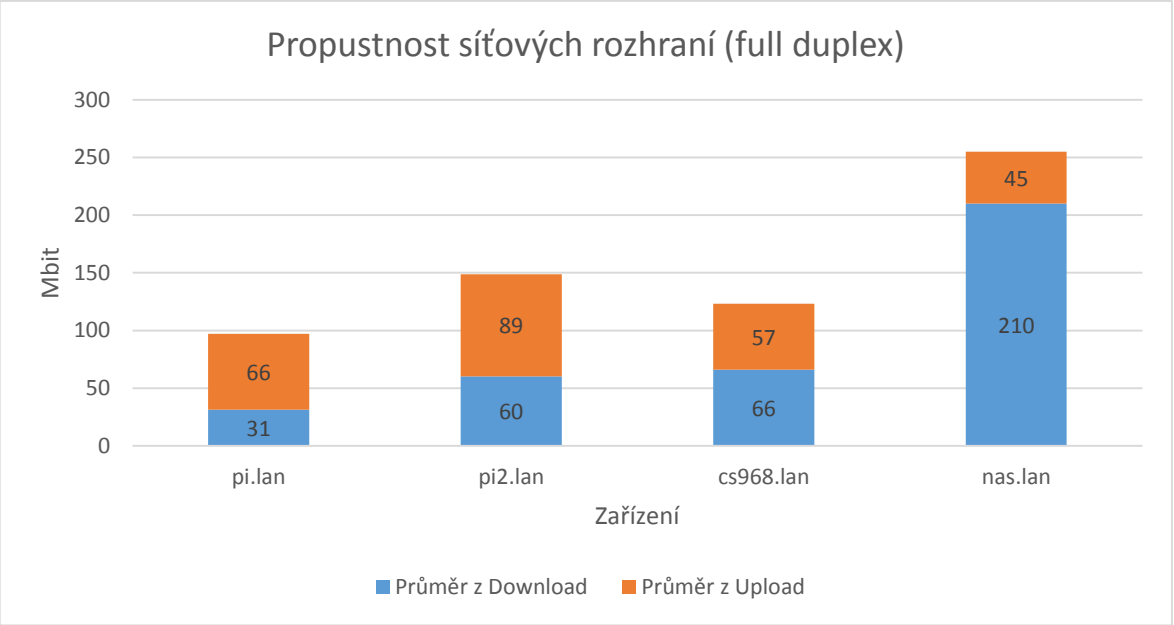
Naměřila se rychlost 94,3 Mbits, přičemž se přeneslo 113MB. A to v čase do deseti sekund.



Obr. 39 Ukázka měření pomocí iperf



Obr. 40 Výsledky měření iperf – half duplex

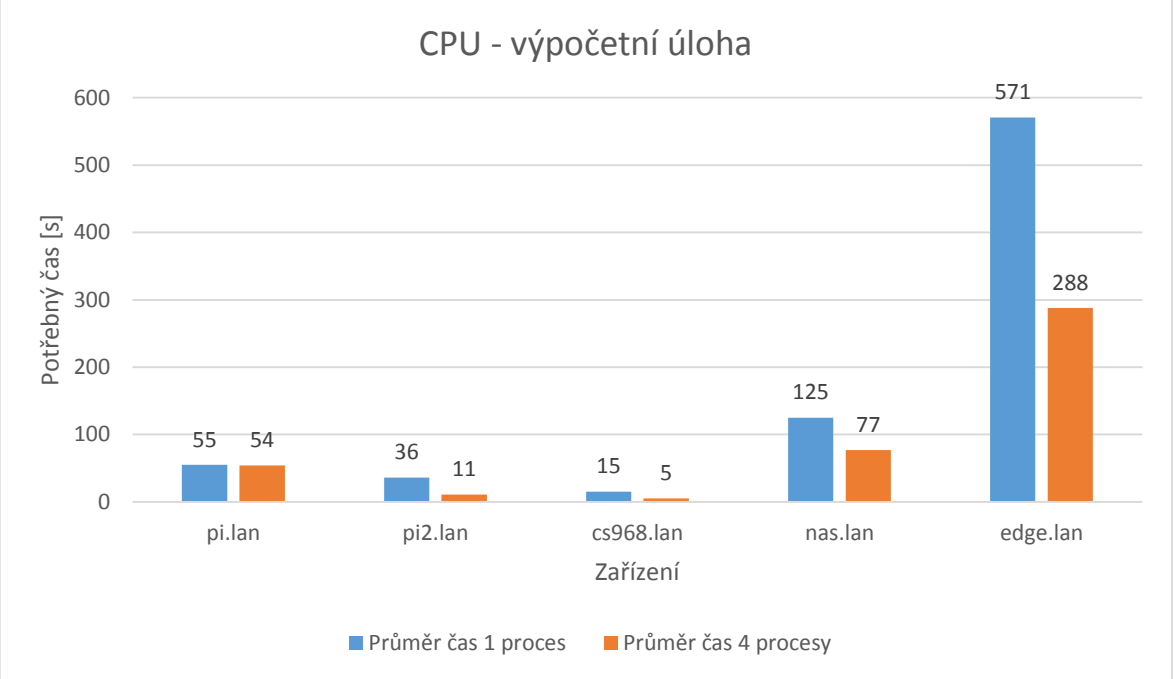


Obr. 41 Výsledek měření iperf – full duplex

7.2 CPU

Pro testování výkonu CPU se použil výkonový test sysbench. Bylo provedeno 20 měření na každém testovaném zařízení.

Test byl spuštěn s parametry: *sysbench --test=cpu --cpu-max-prime=2000 --num-threads=4 run*, kde parametr num-threads udává počet paralelních vláken.



Obr. 42 CPU benchmark

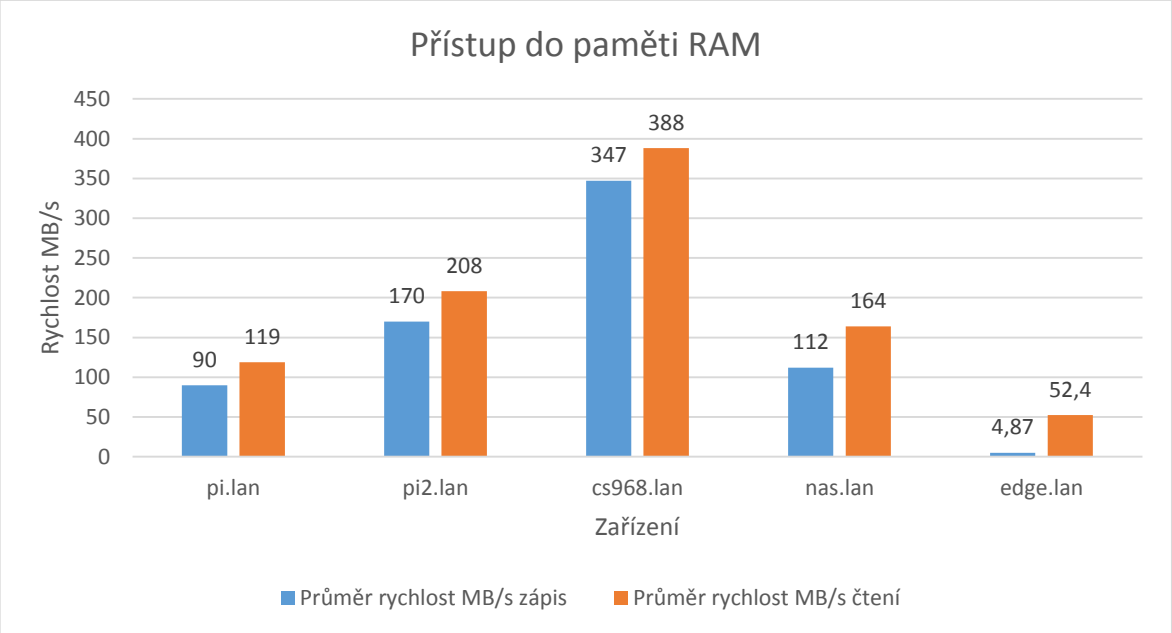
Při porovnání výsledků měření je dobře vidět, že všechna čtyři ARM zařízení dosahují lepšího výkonu, než zařízení s procesorem MIPS. Směrovač edge.lan sice pracuje na nejnižší frekvenci (2x500Mhz) ze všech srovnávaných zařízeních, nicméně se očekávalo, že dosáhne alespoň polovičního výkonu než pi.lan, které pracuje na frekvenci 1x700Mhz. Očekávání se nenaplnilo.

Dále je z grafu čitelné, že u jedno jádrového pi.lan se výpočet úlohy neurychlí použitím paralelního zpracování, naopak u čtyř jádrového pi2lan nebo cs968.lan se výpočet urychlí cca 3x.

7.3 Operační paměť RAM

Test rychlosti čtení a zápisu z a do paměti RAM byl proveden také pomocí výkonového testu sysbench. Nemilým překvapením byly nízké hodnoty u směrovače edge.lan, u cs968.lan byly naměřeny nejvyšší hodnoty.

Test byl spuštěn s parametry: *sysbench --test=memory --memory-total-size=256M --memory-oper=read run*, kde parametr memory-oper definuje čtení, nebo zápis.



Obr. 43 RAM benchmark

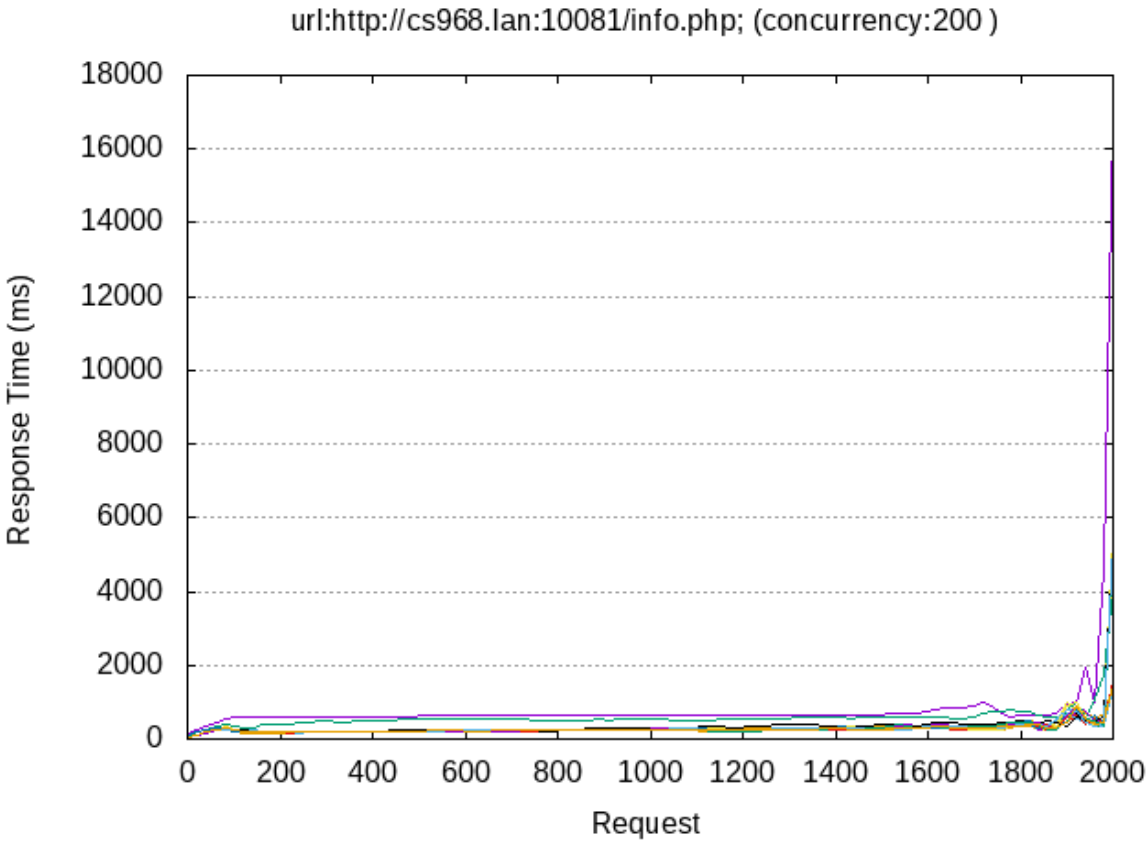
7.4 Webový server – Apache vs. NGINX

Pro srovnání výkonu webových serverů byl použit nástroj Apache Benchmark. Je to jednoduchý nástroj, který se používá k zjištění, kolik žádosti za sekundu dokáže webový server

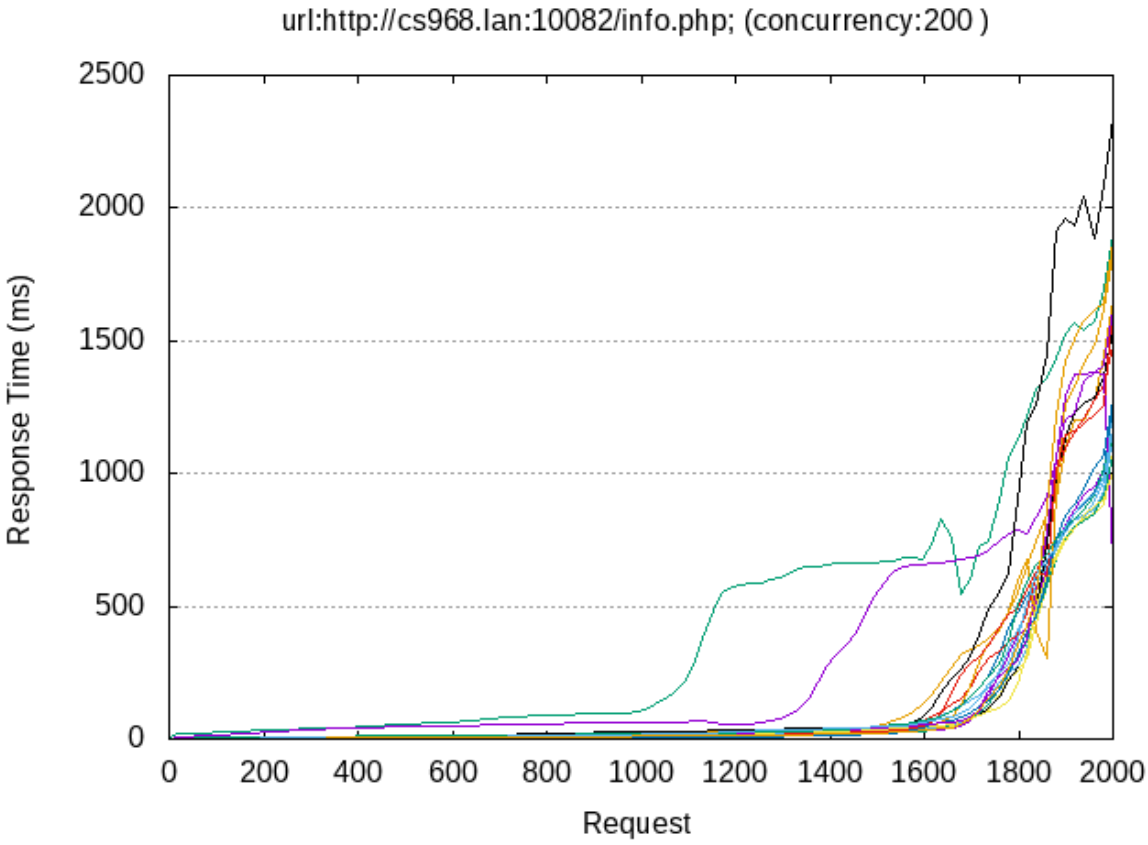
obsloužit. Bylo provedeno celkem dvacet měření na webovém serveru Apache a dvacet měření na webovém serveru NGINX. K měření se použila dynamická stránka, která zobrazuje informace o verzi PHP.

Výsledky měření byly zpracovány programem gnuplot, který se spolu s Apache Benchmarkem dá použít k automatizovaným testům na pravidelné bázi.

Na ose x je vyneseno počet žádostí na server a na ose y čas potřebný k obsloužení těchto žádostí. Měření probíhalo při 200 konkurenčních spojení.



Obr. 44 Webový server Apache



Obr. 45 Webový server NGINX

Z měření se dá usoudit, že oba servery dosahují podobných výsledků, dalo by se říct, že NGINX odpovídá o několik procent rychleji, což se znatelně projeví při velkém množství žádostí.

Rozhodně se ale nedá konstatovat, že NGINX je vždy rychlejší než Apache. Skutečně záleží na povaze webového obsahu.

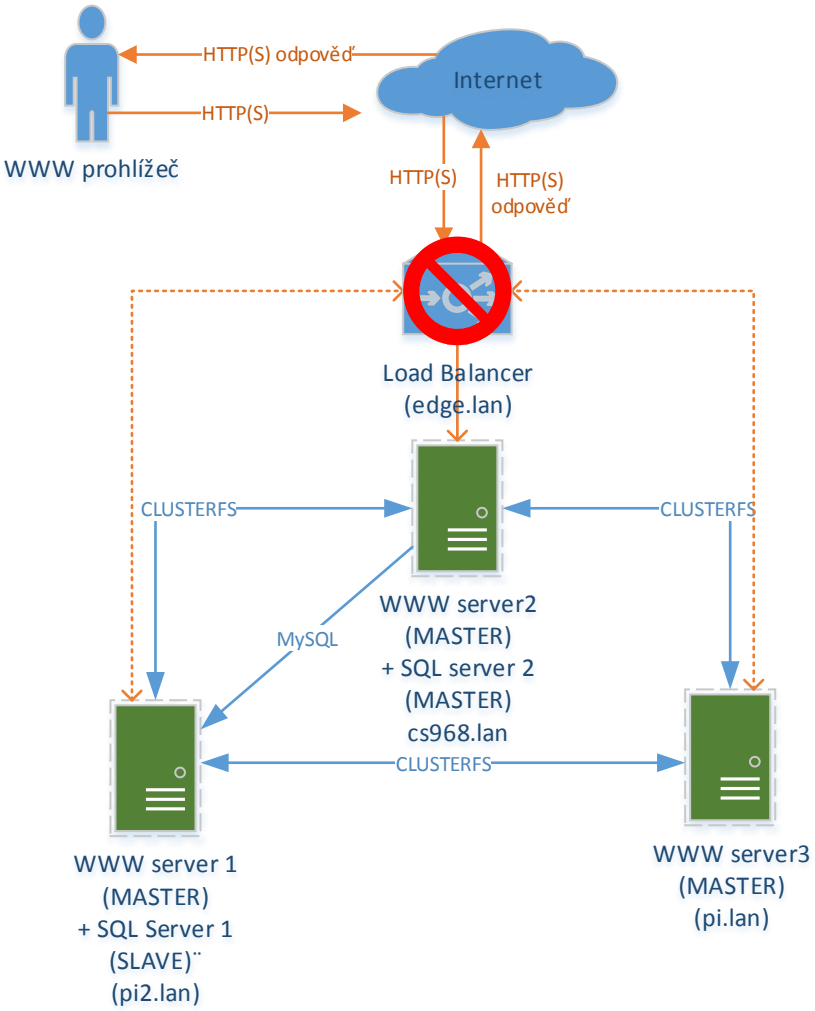
8 CHOVÁNÍ NAVRŽENÉHO CLUSTERU PŘI SELHÁNÍ

V každé systému, který je složen z několika částí, se může jedna jeho malá část pokazit. Webhostingový cluster byl původně navržený tak, že odolá při selhání každé služby nebo zařízení, na kterém služba běží. Při stavbě reálného clusteru muselo dojít ke kompromisům, které spolehlivost celého clusteru snižují, na druhou stranu přináší možné scénáře selhání, jež mohou být popsány.

8.1 Selhání Load Balanceru

Load balancer je vstupní branou do clusteru a v případě, že existuje jen jedna brána, která se při selhání uzavře, znamená to znepřístupnění clusteru. Odborně se tento jev popisuje anglickým výrazem **single point of failure**.

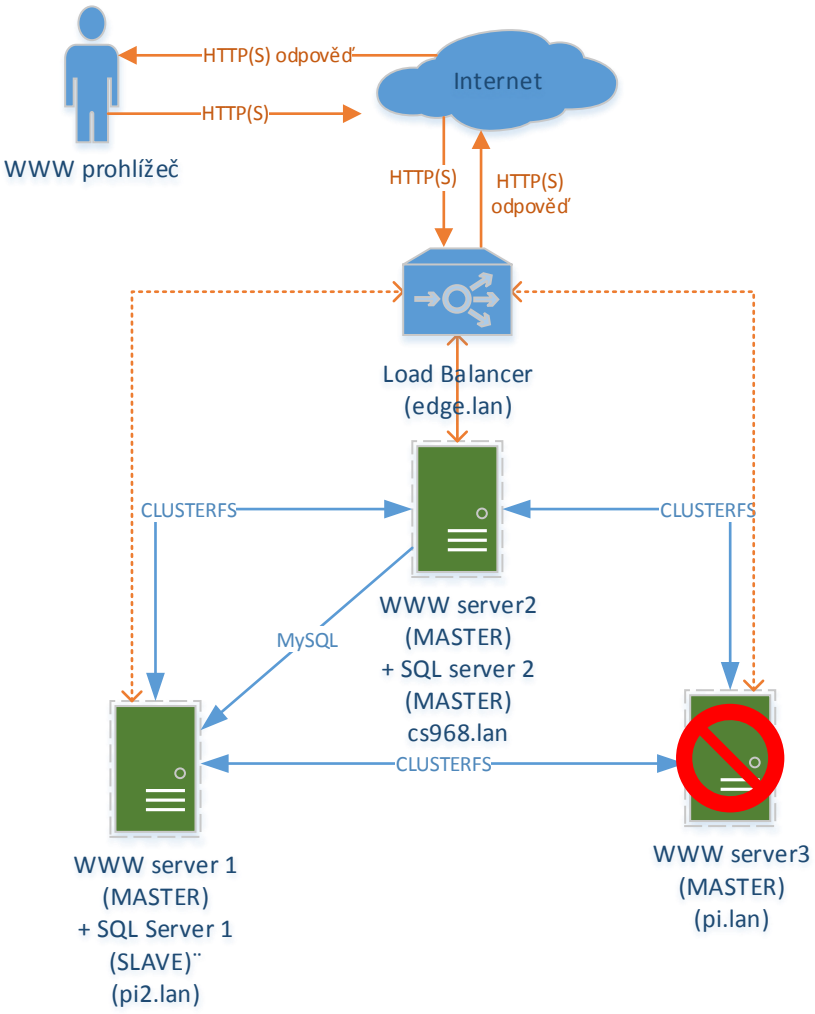
Řešením je použití záložního load balanceru, který při selhání prvního převezme činnost.



Obr. 46 Selhání load balanceru

8.2 Selhání webového serveru

V clusteru jsou použity celkem tři servery, které poskytují webové služby. Pokud služba nebo server, na kterém běží, selže, uživatel bude automaticky pomocí load balanceru přesměrován na jiný server. V případě, že uživatel prohlíží statické stránky, nemusí změnu směrování vůbec zaregistrovat. Pokud by uživatel např. stahoval soubor nebo si prohlížel video, dojde k přerušení stahování.



Obr. 47 Selhání webového serveru

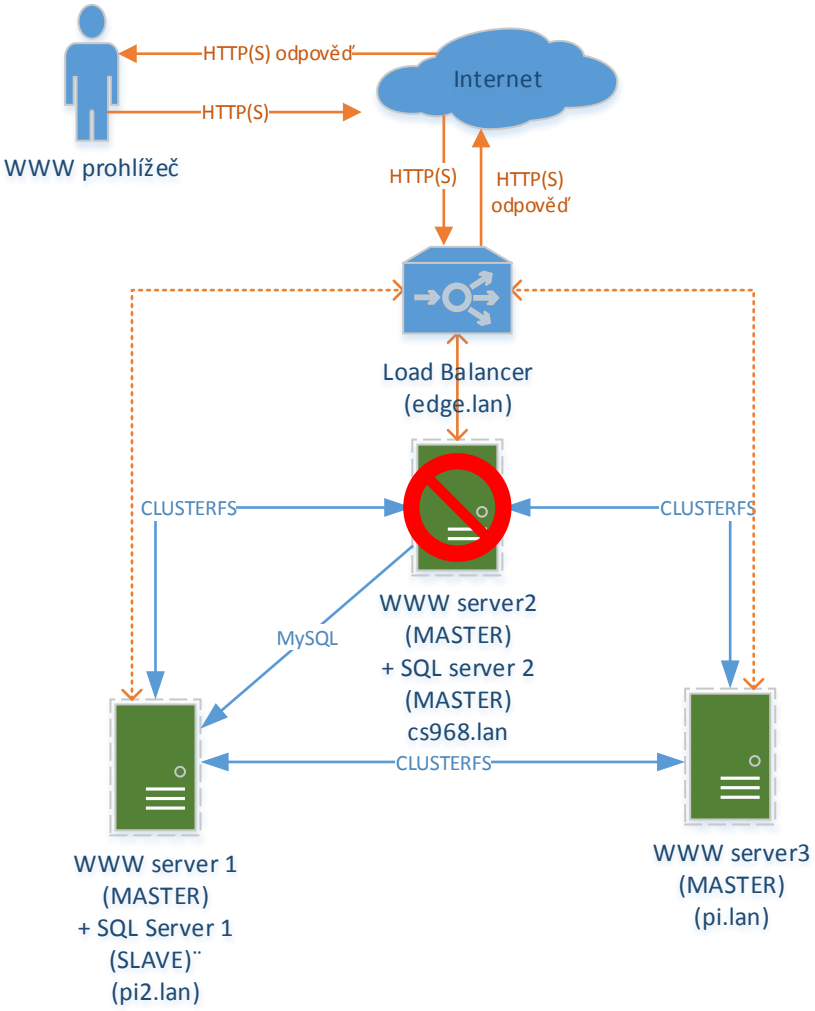
8.3 Selhání databázového serveru

Pokud selže databázový server, cluster bude fungovat dál s většími či menšími obtížemi. Jak velké obtíže budou, záleží na tom, jestli selhal databázový server MASTER, nebo databázový server SLAVE. Návrh clusteru popisuje MASTER-MASTER databázové servery, takže když jeden MASTER selže, zastoupí ho ihned druhý MASTER.

Při stavbě reálného clusteru se však zjistilo, že použitý operační systém Debian Wheezy nabízí pouze MySQL databázi ve verzi 5.5, která nastavení MASTER-MASTER neumí. Pro úplnost – aktuální verze stabilní MySQL databáze je 5.6.

Pokud je tedy v clusteru databáze v nastavení MASTER-SLAVE a selže MASTER, pak nastávají větší obtíže - služba v clusteru není dostupná. Existuje možnost pasovat fungující SLAVE server do role MASTERa, nebo pokažený MASTER opravit. Jako záložní zdroj dat k tomuto účelu poslouží SLAVE.

Pokud je tedy v clusteru databáze v nastavení MASTER-SLAVE a selže SLAVE, pak nastávají jen menší obtíže – neprobíhají pouze replikace z MASTERa na SLAVE.



Obr. 48 Selhání databázového serveru

ZÁVĚR

Ideálním řešením webhostingového clusteru postaveném na architektuře ARM je cluster s rozložením zátěže. Tento typ clusteru dokáže optimálně využít výpočetní výkon všech zařízení v clusteru a v kombinaci s redundantním load balancerem zajistí i vysokou dostupnost.

Díky dispozici pěti různých zařízení, z toho čtyř s architekturou ARM a jednoho s architekturou MIPS, bylo možné navržený cluster sestavit, ovšem bez možnosti redundantního load balanceru. Absence redundantního load balanceru znamená, že v je v clusteru slabé místo označované jako single point of failure a v případě, že load balancer vypoví službu, celý cluster přestává být dostupný. Toto chování může být u profesionálního webhostingu nežádoucí, nicméně pro menší firmy s omezeným rozpočtem přijatelné řešení. Výhoda hostování serverů na architektuře ARM spočívá v nízké spotřebě elektrické energie, potažmo v levném provozu.

V clusteru je použit operační systém Linux v distribuci Debian Wheezy. Tato distribuce se ukázala jako vhodný operační systém, ve které se používají otestované a funkční verze linuxových aplikací. Na druhou stranu při řešení synchronní replikace databáze MySQL distribuce nabídla pouze verzi databázového serveru, která synchronní replikaci nativně nepodporuje. Z toho důvodu je v clusteru implementována asynchronní replikace, která pro cluster s rozdělováním zátěže není vhodná. Nicméně práce se soustředila na rozdělování zátěže webových serverů v clusteru. O synchronní replikaci soborů na webových serverech se stará distribuovaný souborový systém Glusterfs, který v Debianu Wheezy funguje bez problémů. Tento souborový systém se ukázal jako dostatečně robustní a spolehlivé řešení. S ohledem na velikost navrženého clusteru byly použity replikované svazky.

Přístup k souborům pomocí http protokolu byl zajištěn pomocí webových serverů Nginx a Apache. Nginx navíc posloužil jako load balancer využívající implementaci reverzního proxy serveru.

Pro zjištění výkonu byly provedeny výkonové testy, které porovnávají výkon samotných zařízení. Rovněž byly provedeny testy výkonu webových serverů Apache a Nginx.

V současné době klasické servery s procesory ARM zatím příliš nejsou dostupné, dá se ale předpokládat, že se v časovém horizontu několika let stanou nedílnou součástí zelených datacenter.

SEZNAM POUŽITÉ LITERATURY

[1] CS968. 2015. *Aliexpress.com* [online]. [cit. 2015-05-13]. Dostupné z: http://www.aliexpress.com/item/the-newest-android-4-4-2-Quad-Core-Android-TV-Box-CS968-Mic-RK3188-2G-RAM/1867619486.html?spm=2114.32010308.4.52.xaRPgX

[2] *Getting started with MariaDB*. [2013]. Birmingham: Packt Publishing, 1 online zdroj (100 pages). ISBN 978-1-78216-810-2.

[3] JELÍNEK, Lukáš. 2010. *Vytváříme vlastní distribuci Linuxu: od návrhu po fungující systém*. Vyd. 1. Brno: Computer Press, 304 s. ISBN 978-80-251-2433-8.

[4] SOSINSKY, Barrie A. 2010. *Mistrovství – počítačové sítě*. Vyd. 1. Brno: Computer Press, 840 s. Mistrovství (Computer Press). ISBN 978-80-251-3363-7.

[5] HORÁK, Jaroslav a Milan KERŠLÁGER. 2011. *Počítačové sítě pro začínající správce*. 5., aktualiz. vyd. Brno: Computer Press, 303 s. ISBN 978-80-251-3176-3.

[6] SARKAR, Dipankar. 2011. *Nginx 1 web server implementation cookbook: over 100 recipes to master using the Nginx HTTP server and reverse proxy*. Birmingham, U.K.: Packt Open Source Pub., iv, 220 p. ISBN 978-1-84951-496-5.

[7] KABIR, M. 2004. *Apache server 2*. Vyd. 1. Brno: Computer Press, 724 s. ISBN 80-251-0319-6.

[8] NEDELCU, Clément. 2010. *Nginx HTTP server: adopt Nginx for your web applications to make the most of your infrastructure and serve pages faster than ever* [online]. Birmingham, U.K.: Packt Pub., ix, 327 p. [cit. 2015-05-13]. ISBN 978-1-84951-086-8. Dostupné z: http://www.amazon.com/dp/1782162321/?tag=ebooks0056-20

[9] *Edgerouter-lite*. 2015. Ubiquiti Networks [online]. [cit. 2015-05-13]. Dostupné z: https://www.ubnt.com/edgemax/edgerouter-lite/

[10] *Raspberry Pi* [online]. 2015. [cit. 2015-05-13]. Dostupné z: https://www.raspberrypi.org/

[11] *NSA325*. 2015. ZyXEL [online]. [cit. 2015-05-13]. Dostupné z: http://www.zyxel.com/cz/cs/products_services/nsa325.shtml?t=p

[12] *Debian* [online]. 2015. [cit. 2015-05-13]. Dostupné z: https://www.debian.org/

[13] KOPPER, Karl. 2005. *The Linux enterprise cluster: build a highly available cluster with commodity hardware and free software*. San Francisco: No Starch Press, 430 s. ISBN 15-932-7036-4.

[14] FORTA, Ben. 2012. *MariaDB crash course*. Upper Saddle River, NJ: Addison-Wesley, xi, 287 p. ISBN 03-217-9994-1.

[15] ROSENBROCK, Eric a Eric FILSON. 2005. *Linux, Apache, MySQL a PHP: instalace a konfigurace prostředí pro pokročilé webové aplikace*. 1 vyd. Překlad Karel Voráček. Praha: Grada, 344 s. ISBN 80-247-1260-1.

[16] BEIGHLEY, Lynn a Michael MORRISON. 2009. *Head First PHP and MySQL*. Sebastopol: O'Reilly, xxxviii, 774 s. Head first series. ISBN 978-0-596-00630-3.

[17] *MySQL profesionálně: optimalizace pro vysoký výkon*. 2009. Vyd. 1. Brno: Zoner Press, 712 s. Encyklopedie webdesignera. ISBN 978-80-7413-035-9.

[18] DYER, Russell J. T. 2015. *Learning MySQL and MariaDB: Heading in the Right Direction with MySQL and MariaDB*. Sebastopol, CA 95472: O'Reilly Media. ISBN 978-1449362904.

[19] *Mikroprocesory s architekturou ARM*. 2012. Root.cz [online]. [cit. 2015-05-13]. Dostupné z: http://www.root.cz/clanky/mikroprocesory-s-architekturou-arm

[20] *Keepalived: loadbalancing and high-availability* [online]. 2015. [cit. 2015-05-13]. Dostupné z: http://www.keepalived.org/

[21] *Apache: HTTP server project* [online]. 2015. [cit. 2015-05-13]. Dostupné z: http://httpd.apache.org/

[22] *MySQL: The world's most popular open source database* [online]. 2015. [cit. 2015-05-13]. Dostupné z: http://www.mysql.com/

[23] *BBC and ARM to equip UK kids with Raspberry Pi-style Micro Bit coding device*. 2015. www.theinquirer.net [online]. [cit. 2015-05-13]. Dostupné z: http://www.theinquirer.net/inquirer/news/2399360/bbc-and-arm-to-provide-1m-uk-kids-with-raspberry-pi-style-micro-bit-coding-device

[24] *HAProxy: Configuration Manual*. 2015. HAProxy [online]. [cit. 2015-04-21]. Dostupné z: http://www.haproxy.org/download/1.3/doc/configuration.txt

[25] *Archlinuxarm.org* [online]. 2015. [cit. 2015-05-13]. Dostupné z: http://archlinuxarm.org/

[26] *The DRBD User's Guide* [online]. 2015. [cit. 2015-05-13]. Dostupné z: https://drbd.linbit.com/users-guide/drbd-users-guide.html

[27] *Nginx*. 2015. Ngx_http_upstream_module [online]. [cit. 2015-05-13]. Dostupné z: http://nginx.org/en/docs/http/ngx_http_upstream_module.html

[28] *ARM: The Architecture for the Digital World* [online]. 2015. [cit. 2015-05-13]. Dostupné z: http://www.arm.com/

[29] *Gluster* [online]. 2015. [cit. 2015-05-13]. Dostupné z: http://www.gluster.org/

[30] *ZyXEL NSA325 v2*. 2015. NAS-Central [online]. [cit. 2015-05-14]. Dostupné z: http://zyxel.nas-central.org/

[31] *OpenMediaVault: The open network attached storage solution* [online]. 2015. [cit. 2015-05-14]. Dostupné z: http://www.openmediavault.org/

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

RISC	Reduced instruction set computing.
HTTP	Hypertext transfer protocol.
CIFS	Common internet file systém.
WEB	World wide web.
ARM	Advanced RISC machines.
RDMA	Remote direct memory access.
NAS	Network access serve.
SMB	Server message block protocol.
SSL	Secure sockets layer.
TLS	Transport layer security.
SQL	Structured query language.
GPL	General public license.
SoC	System-on-a-chip.
CPU	Central processing unit.
RAM	Random access memory.
SSH	Secure shell.
SD	Secure Digital card.
TTL	Time-to-live.
HDD	Hard disk drive.
SATA	Serial advanced technology attachment.
TB	Terabyte.
IP	Internet protocol address.
UID	Unique identifier.
GID	Group ID.
CTDB	Cluster trivial database.

TDB	Trivial database.
OSI	Open systems interconnection.
TCP	Transmission control protocol.
UDP	User datagram protocol.

SEZNAM OBRÁZKŮ

SEZNAM TABULEK