**Univerzita Palackého v Olomouci**
**Přírodovědecká fakulta**
**Katedra geoinformatiky**

# HODNOCENÍ MECHANISMŮ REPLIKACE VYBRANÝCH DATABÁZOVÝCH SYSTÉMŮ

**Bakalářská práce**

**Lenka TRNOVÁ**

**Vedoucí práce Mgr. Tomáš POHANKA**

**Olomouc 2018**
**Geoinformatika a geografie**

# ANOTACE

Bakalářská práce se zabývá porovnáním replikačních mechanismů ve dvou databázových systémech, a to konkrétně v PostgreSQL a MySQL. Replikace je důležitou součástí práce s daty v databázi, která běží na serveru. K serveru se dotazují další servery, resp. klienti. Teoretická část představuje obecné seznámení se s principem replikace, prací s prostorovými daty v databázích a studium hodnocení databází pomocí předem vytvořených kritérií. Praktická část je zaměřena na vlastní testování, které probíhalo na 2 zařízeních v rámci lokální sítě – stolní PC a notebook. Vstupními daty jsou převážně volně dostupné sady vektorových dat, se kterými student katedry běžně pracuje. Důležitým krokem předcházející samotné testování je vymezení kritérií, která jsou sledována a měřena a slouží k vytvoření výsledného hodnocení. Statistickými metodami dochází k porovnání použitelnosti jednotlivých replikačních mechanismů. Společně s hodnocením je také vytvořeno případové cvičení využití replikace v praxi. Nedílnou součástí práce je step-by-step návod pro studenty, obsahující velmi podrobný postup inicializace replikace právě v těchto dvou systémech na databázových systémech PostgreSQL a MySQL.

# KLÍČOVÁ SLOVA

databáze; replikace; prostorová data; MySQL; PostgreSQL

Počet stran práce: 49
Počet příloh: 4 (z toho 2 vázané a 2 volné)

# ANOTATION

Bachelor thesis is focused on comparing two database systems – PostgreSQL and MySQL. Replication is the essential part of working with data in a database, which is running on a server. Other servers called clients are querying to this server. Theoretical part represents a basic introduction to the rules of replication, working with spatial data in the databases and evaluation of databases based on prepared criteria. The practical part is focused on own testing, which was run on two devices on the local network – personal computer and notebook. Input data are mostly free vector layers which are commonly used by the students. The main step before own testing is to set criteria which will be monitored and measured to make an evaluation. Based on statistical methods both database systems are compared in the practical use of replication. Next, to evaluation, there is also a practical example of replication in the real world. The important part of the work is to prepare a step-by-step manual for students. Manual consists of detailed steps to set and initialise replication on their own of both database systems PostgreSQL and MySQL.

# KEYWORDS

database; replication; spatial data; MySQL; PostgreSQL

Number of pages: 49
Number of appendixes: 4

**Prohlašuji, že**

- bakalářskou práci včetně příloh, jsem vypracovala samostatně a uvedla jsem všechny použité podklady a literaturu. Ve své práci jsem použila návrh replikace v prostředí PostgreSQL pomocí extenze Slony-I vytvořený Markétou Solanskou v rámci její diplomové práce s názvem *Synchronizace a replikace geodat v prostředí ESRI platformy* (2014).

- jsem si vědoma, že na moji bakalářskou/diplomovou práci se plně vztahuje zákon č.121/2000 Sb. - autorský zákon, zejména § 35 – využití díla v rámci občanských a náboženských obřadů, v rámci školních představení a využití díla školního a § 60 – školní dílo,

- beru na vědomí, že Univerzita Palackého v Olomouci (dále UP Olomouc) má právo nevýdělečně, ke své vnitřní potřebě, bakalářskou práci užívat (§ 35 odst. 3),

- souhlasím, aby jeden výtisk bakalářské práce byl uložen v Knihovně UP k prezenčnímu nahlédnutí,

- souhlasím, že údaje o mé bakalářské práci budou zveřejněny ve Studijním informačním systému UP,

- v případě zájmu UP Olomouc uzavřu licenční smlouvu s oprávněním užít výsledky a výstupy mé bakalářské v rozsahu § 12 odst. 4 autorského zákona,

- použít výsledky a výstupy mé bakalářské práce nebo poskytnout licenci k jejímu využití mohu jen se souhlasem UP Olomouc, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly UP Olomouc na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Olomouci dne Lenka Trnová

Děkuji vedoucímu práce Mgr. Tomáši Pohankovi za podněty a připomínky při vypracování práce. Zároveň děkuji rodině, a především příteli za podporu během celého studia.

# OBSAH

# SEZNAM POUŽITÝCH ZKRATEK

| Zkratka | Význam |
|---|---|
| BLOB | Binary large object |
| CPU | Central Processing Unit |
| CSV | Comma-separated values |
| FOSS | Free and open-source |
| Gbps | Gigabits per second |
| HDD | Hard Disk Drive |
| HW | Hardware |
| Mbps | Megabits per second |
| MD5 | Message-Digest algorithm |
| MS | Microsoft |
| NB | Notebook |
| NoSQL | Non relational database |
| OGC | Open Geospatial Consortium |
| PC | Personal Computer |
| PDF | Portable Document File |
| QoS | Quality of Service |
| RAM | Random-access Memory |
| RBR | Row-based replication |
| SBR | Statement-based replication |
| SHP | Shapefile |
| SSB | Single Side Band |
| SSD | Solid-state Drive |
| SW | Software |
| SQL | Structured Query Language |
| TCP | Transmission Control Protocol |
| TPC-H | Decision Support Benchmark |
| WKB | Well-known binary |
| WKT | Well-known text |
| YCSB | Yahoo! Cloud Services Benchmark |

# SEZNAM GRAFŮ

# SEZNAM TABULEK



# SEZNAM OBRÁZKŮ

# ÚVOD

Díky moderním technologiím se prakticky vše odehrává v digitálním světě. Není překvapením, že i prostorová data se uchovávají především v digitální formě. Velkou výhodou je jejich dostupnost, která je prakticky neomezená. Avšak s dostupností přichází i velké riziko, a to ztráta dat. Replikace se na první pohled zdá velmi dobrým způsobem, jak si svá data zálohovat. Avšak replikace umožňuje přístup k datům z několika míst najednou. S přistupováním k datům souvisí také zátěž na serveru, kterým se klienti dotazují. Replikace umožňuje rozložení zátěže mezi více zařízení tím, že vytvoří kopii dat na další server. Následně se klienti dotazují k serveru s kopiemi dat. Tímto způsobem se zamezí přetížení hlavního serveru s primárními daty. Aktuálně existuje nespočet databází, které replikaci umožňují. Pár z nich dokonce i práci s prostorovými daty, které jsou důležitou součástí této práce. V práci budou databázové systémy, základy replikace a důležité pojmy. V praktické části dojde k inicializaci replikace ve dvou replikačních mechanismech. Je zde popsán způsob, jak replikace v daném nástroji nastavit. Po úspěšném nastavení dojde k samotnému otestování na předem vybraných datech. Budou sledována kritéria, která byla navržena na základě rešerše. Díky těmto kritériím dojde k porovnání obou mechanismů a k závěrečnému zhodnocení jejich použitelnosti.

# 1 CÍLE PRÁCE

Cílem bakalářské práce je zhodnocení mechanismů replikace vybraných prostorových databází, konkrétně PostgreSQL a MySQL. Na základě výsledného hodnocení bude stanovena jejich použitelnost.

V teoretické části bude rozbor dostupných druhů replikačních mechanismů v těchto dvou databázových systémech. Dílčí částí je zaměření se na mechanismus replikace, její dostupnost a způsob její inicializace. Dalším dílčím bodem je rešerše prací týkajících se hodnocení databázových replikací a jejích výkonu. Bude následovat výběr nejvhodnějších kritérií, která budou vstupovat do samotného testování. Do teoretické části je potřeba zařadit definování úzkých hrdel systému tzv. „bottleneck“ (šířka pásma, výkon samotných serverů, na kterých se bude testovat). Na základě definování bude snaha minimalizovat jejich vliv.

V praktické části dojde k samotné inicializaci replikace, která vychází z diplomové práce Mgr. Markéty Solanské: *Synchronizace a replikace geodat v prostředí ESRI platformy*. Na jejím základě pak bude probíhat testování replikačních mechanismů a jejich využitelnost při správě prostorových dat. K testování budou využity dostupné prostorové datové sady a vybrané vektorové vrstvy z nich. Výsledkem práce bude podrobná dokumentace z testování společně s vytvořením 2 případových cvičení. Bude se jednat o praktické využití replikačního mechanismu a step-by-step návodů pro studenty, obsahující postup od přípravy importu dat do databáze až po samotnou inicializaci replikace.

# 2 METODY A POSTUPY ZPRACOVÁNÍ

## 2.1 Použité metody

Pro samotné testování byla důležitá rešerše představující základy replikace a replikačního mechanismu. Důležitou částí bylo studium prací týkajících se hodnocení databázových systémů. Ty pomohly nastínit návrh vlastních kritérií vstupujících do testování. Společně s kritérii byly předem dány podmínky, za jakých se bude replikační mechanismus spouštět. Na základě navrženého replikačního modelu Mgr. Markéty Solanské v rámci její diplomové práce byla převzatá část s využitím extenze Slony-I. Od vedoucího práce byl získán postup inicializace replikace v MySQL. Během testování byla sledována předem stanovená kritéria, která vstupovala do srovnání obou replikačních mechanismů. Během testování byla použita předem vybrána vektorová data.

Pro srovnání replikačních mechanismů byly použity statistické metody formou boxplotů, tabulek obsahující další statistické metody (intervalové rozpětí, směrodatná odchylka) a v neposlední řadě grafy porovnávající naměřené hodnoty. Závěrem bylo formou syntézy vytvořené zhodnocení použitelnosti a limitů obou replikačních mechanismů.

## 2.2 Použitá data

Opakovaná inicializace replikace proběhla na již existujících datech vektorového typu. V práci byly použity následující datové sady, které byly zvoleny z důvodu dostupnosti, celkové rozsáhlosti a případné dostupnosti na katedře KGI. Data jsou běžně používána studenty Katedry geoinformatiky.

**Data200**

Topografická databáze České republiky vznikla na základě projektu ERM (EuroRegionalMap). Obsahuje 8 tematických vrstev: administrativní hranice, vodstvo, doprava, sídla, geografická jména, různé objekty, vegetace a povrch, výškopis. Data jsou placená stejně jako ZABAGED, který měl být předmětem testování, avšak pro účely bakalářské a diplomové práce jsou dostupná na katedře právě Data200.

**ArcČR ® 500**

Data poskytovaná firmou ARCDATA PRAHA, s.r.o., která obsahují administrativní členění a socioekonomické údaje o České republice. Data jsou jak vektorová, tak i rastrová – DMR (digitální model reliéfu) a stínovaný reliéf. Data vznikla ve spolupráci s Českým zeměměřičským úřadem a Českým statistickým úřadem. V této práci byla použita data verze 3.3, která jsou volně dostupná ke stažení z webových stránek ARCDATA PRAHA, s.r.o. ArcČR 500 vychází z dat Data200.

**BPEJ**

Celostátní databáze BPEJ (bonitovaných půdně ekologických jednotek). V práci je použitá verze aktualizována k 5. 1. 2018. Data jsou volně ke stažení ve formátu shapefile ze stránek Státního pozemkového úřadu, kde jsou průběžně každý měsíc aktualizována. Vrstva slouží k hodnocení úrodnosti půd, kterou rozlišuje na základě pětimístného kódů. 1. číslice značí kód klimatického regionu, 2. a 3. číslice kód hlavní půdní jednotky, 4. číslice sdružený kód sklonitosti a expozice a poslední číslice je sdružený kód skeletovitosti a hloubky půdy. Na základě hodnocení půd lze zjistit, které půdy jsou náchylnější k erozím. Pro testování byla vrstva oříznuta pouze na Olomoucký kraj. I takto oříznutá byla stále nejnáročnější vrstvou celého testování.

**NaturalEarth**

Volně dostupná datová sada, vytvářena dobrovolníky pod správou NACIS (North American Cartographic Information Society). Obsahuje administrativní dělení, fyzicko-geografickou sféru a rastrová data ve formě reliéfu v měřítkách 1 : 10 000 000, 1 : 50 000 000 a 1 : 110 000 000. Data jsou k dispozici pro celý svět. Pro testování byla použita vektorová vrstva světa ve verzi 3.0.1.

## 2.3 Použité programy

**MS Windows**

Celé testování probíhalo v prostředí Microsoft Windows ve verzích 8.1 a 10. Toto prostředí bylo zvoleno pro jeho rozšířenost mezi uživateli. Většina nástrojů, která byla použita, mají grafické prostředí. Všechny nástroje lze odborně používat i v Unix systémech, jako je Linux nebo Mac OS).

**Databáze PostgreSQL**

První část testování byla zaměřena na databázi PostgreSQL ve verzi 9.5, která byla zvolena v návaznosti na KGI server. Pro přívětivější práci byl zvolen grafický program pgAdmin III 1.22.2, který umožňuji i zadávání SQL dotazů. Dále bylo potřeba instalace extenzí: pro práci s prostorovými daty PostGIS 2.3.3 a pro asynchronní replikaci Slony-I 2.2.5.-2.

**Databáze MySQL**

V MySQL probíhala streaming replikace. Pro práci s MySQL bylo použito webového rozhraní phpMyAdmin, které má v sobě podporu prostorových dat již implementovanou. Pro práci na lokálním serveru tzv. *localhostu* bylo použito desktopové aplikace Wampserver verze 1.6.2.37.

**pgAdmin III**

pgAdmin III je volně dostupné grafické uživatelské rozhraní, které je obsaženo přímo v instalačním balíčku PostgreSQL. Podporuje zásuvné moduly (*angl. plugins*), jako právě PostGIS, umožňující práci s prostorovými daty. Práce v tomto prostředí je velmi intuitivní, je dostupná i česká lokalizace. Pro práci v databázi je možno využít SQL konzole. Nejnovější verzí je pgAdmin 4 v aktuální verzi 2.1 k 11. 1. 2018. Vývoj pgAdmin 3 je již zastaven.

**phpMyAdmin**

PhpMyAdmin je volně dostupná webová aplikace pro práci s MySQL či MariaDB. Aplikace je napsaná v jazyku PHP a slouží pro uživatelsky přehlednou práci s databází MySQL přes webový prohlížeč. Pro práci s phpMyAdmin je nutná instalace webového serveru, např. Apache HTTP Server. Důvodem volby právě tohoto nástroje byla znalost již z výuky na katedře a součást balíčku Wampserver. Další možností je např. MySQL Workbench nebo český Adminer.

**Wampserver**

Jedná se o softwarový balíček obsahující Apache HTTP Server, PHP a MySQL databázi. Instalační balíček je dostupný z oficiálních webových stránek Wampserveru. Balíček je pro snazší práci obohacen o webovou aplikaci phphMyAdmin.

**Monitorovací SW**

Pro měření výkonu zařízení bylo použito nástroje HWMonitor Pro. Jeho výhodou je tvorba grafů a export absolutních dat do CSV souboru. Grafy byly upraveny a následně použity ve výsledném porovnání databází. Průměrné hodnoty dat z CSV tabulek byly použity do souhrnných tabulek. Webový nástroj Monyog sloužil pro monitoring celkového času replikování změn v rámci databáze MySQL. Nástroj umožňuje sledovat v reálném čase přenos dat a SQL příkazy.

**Ostatní nástroje**

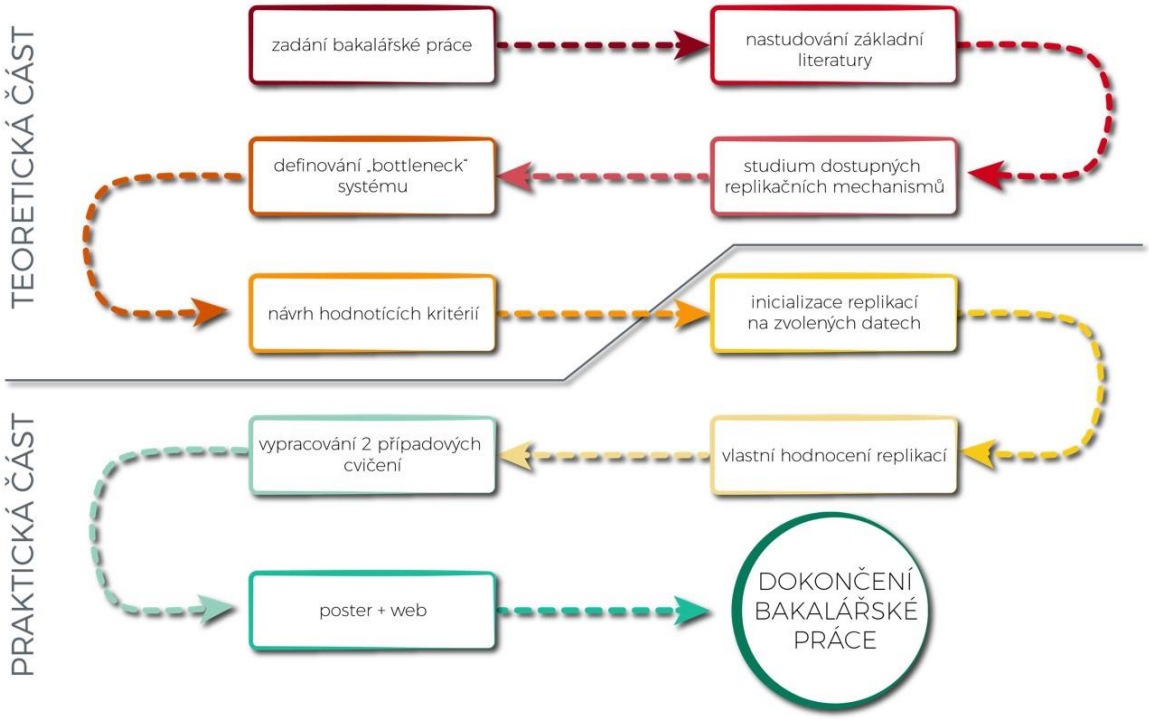
Pro provádění změn v datech uložených v databázích bylo použito desktopového programu QGIS ve verzi 2.18.10. Pro práci s šířkou pásma, resp. pro omezení přenosové rychlosti mezi zařízeními, byl využit program NetLimiter 4. Pro statistické vyhodnocení bylo použito RStudio verze 0.99.892 a MS Excel z balíčku Microsoft Office 2016.

## 2.4 Postup zpracování

Obr. 1 zobrazuje diagram postupu bakalářské práce. V první části byla provedena literární rešerše knižních zdrojů, odborných článků a elektronických studií. Rešerše se zabývá dostupnými replikačními mechanismy v databázových systémech PostgreSQL a MySQL, dostupností replikace a podporou prostorových dat v těchto dvou databázových systémech. Nedílnou součástí rešerše je analýza dostupných prací týkajících se hodnocení databázového systému. Pomocí nich byla zvolena kritéria hodnocení, která vstupovala do testování. Samotný návrh replikace pro dané databázové systémy není předmětem práce, návrh replikace pro databázový systém PostgreSQL pomocí extenze Slony-I je převzatý z diplomové práce Mgr. Markéty Solanské (2014), jen je obdobně použit v rámci lokální sítě. Návrh replikace pro systém MySQL byl poskytnutý vedoucím práce Mgr. Tomášem Pohankou.

V praktické části proběhlo samotné testování, během kterého byla sledována vlastní kritéria, která byla vybrána na základě rešerše. Každý replikační mechanismus v obou databázových systémech byl opakovaně proveden na vybraných testovacích datech, u kterých byly prováděny změny. Na základě stanovených kritérií byla statistickými metodami zhodnocena výkonnost replikace v daném prostředí a na daných datech. Byla nastíněna možnost využití replikačních mechanismů.

Součástí práce je také podrobná dokumentace z testování a dále byla vytvořena případová cvičení, obsahující step-by-step návody pro replikaci v PostgreSQL a MySQL. V neposlední řadě byl vytvořen poster a webová stránka informující o tématu bakalářské práce, která je umístěna na webu katedry.




Obr. 1: Vývojový diagram postupu práce

# 3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

Pro další práci je potřeba definovat pár základních pojmů, které se používají v databázových systémech, se kterými se bude pracovat – PostgreSQL a MySQL. Nedílnou součástí této kapitoly je přiblížení principu replikace. Je nutné podotknout, že každý systém používá jak své vlastní termíny, tak i jiná dělení druhů replikace.

## 3.1 Struktura replikačního modelu

Základní částí modelu replikace je **uzel**. Každý uzel představuje jeden databázový server, který má svojí roli. Uzel může mít buď hlavní, nebo podřízenou roli. **Hlavní server** (*angl. master server*) představuje server, na který se data zapisují, mohou se zde upravovat. Naopak **podřízený server** (*angl. slave server*) slouží pouze pro čtení, data se sem kopírují z master serveru. Vzhledem k tomu, že i přes českou lokalizaci systémů se nadále pro označení serverů v systémech používají anglické termíny, v dalších částech práce se budou používat označení **master** a **slave** z důvodu přehlednosti. Celý systém uzlů (serverů) se nazývá **shluk** (*angl. cluster*). Na základě počtů master uzlů dělíme replikaci na multimaster a master-slave replikaci. Jak již z názvu vyplývá, **multimaster** obsahuje 2 a více master serverů, naopak **master-slave** má pouze jeden master server a následně několik slave serverů.

## 3.2 Replikace a její druhy

Proces **replikace** si lze představit jako vytváření kopií změn v datech z jednoho serveru na druhý. Data se upravují na již zmiňovaném master serveru, následná změna se poté replikuje na slave servery, kde data slouží pouze ke čtení. Replikace se může jevit jako vhodná volba pro zálohování, ale nejedná se o primární funkci. Při samotné záloze se ale musí počítat s tím, že jakákoliv chybná změna se kopíruje na všechny ostatní servery. Možností je nastavení intervalů synchronizace.

Nejdůležitějším účelem replikace je rozložení zátěže na více serverů. Pokud by velké množství dotazování probíhalo pouze na 1 server, tak by server nestíhal a odpovědi ke klientům by byly několikanásobně zpomalené. Odezva serveru by se tak z desítek ms mohla vyšplhat až na jednotky s. Je dvojí možnost škálovatelnosti, resp. rozšiřitelnosti – vertikální a horizontální. **Vertikální škálovatelnost** (*angl. scale up*) znamená, že výkon serveru zvýšíme přidáním daného hardwaru – přidání vnitřní paměti, nahrazení procesoru výkonnějším, naopak **horizontální** (*angl. scale out*) znamená, že zátěž rozložíme přidáním dalších serverů, které naopak nemusí být tak výkonné. Ne každý systém je ale vhodný pro horizontální škálování.

Bell a kol. (2010) uvádí, že dalším možným využitím replikace může být technika zvaná **dual-master setup**. Ta slouží k zachování master serveru za každé okolnosti tím, že dochází ke zrcadlení dvou serverů mezi sebou. Jedná se o záložní (*angl. standby*) server, který se spouští pouze v případě pádu hlavního serveru, aby jej mohl zastoupit.

Replikace také může sloužit ke zkopírování změn na velké vzdálenosti na další server (georeplikace). Pro zachování nízké odezvy (latence) pro databázové dotazy, je vhodné umístit databázový server co nejblíže klientům. Jedná se především o mezikontinentální komunikaci, aby dotazy nesměřovaly na primární master server vzdálený tisíce kilometrů. Takto budou všechny dotazy směřovat na sever v lokaci klienta a odpovědi ze strany serveru budou mnohem rychlejší.

### 3.2.1 Dělení replikací

Máme několik typů replikace. Jak již výše zmiňované multimaster a master-slave, tak i mnoho dalších dělení. Jedním z nich je dělení na synchronní a asynchronní. **Synchronní** replikace zajišťuje, že provedená změna na master serveru se zapíše současně na všechny slave servery. Pokud by nebylo možné synchronizaci v celé databázové síti provést, celá transakce bude zrušena. Je to obdoba bankovního převodu – nemůže se stát, že by se peníze odečetly, aniž by se někde nepřičetly. **Asynchronní** replikace se liší tím, že se změna nejprve zapíše na master serveru a buď v určitém kroku, nebo za vhodné situace (vytíženost serveru) se následně zapíše na slave servery. Tím vzniká nekonzistentnost mezi master serverem a slave serverem po určitou dobu. Doba záleží na nastavení samotné replikace a může být v řádech ms až po dny. Böszörmenyi a Schönig (2013) dále uvádějí, že také může docházet ke zpoždění (tzv. *lag*), jehož délka závisí na velikosti dat, ale převážně se jedná o ms až s.

Dalším typem dělení může být fyzická a logická replikace. **Fyzická** replikace pracuje s binárním kódem, ve kterém jsou uložené informace o provedené změně v datech. Data se vždy ukládají na stejná místa na disku. Je jednodušší na vytvoření, její využití je převážně pro zálohování nebo ke zvýšení výkonu clusteru přidáním dalšího serveru. Všechny servery musí být na stejné verzi. Fyzická replikace navíc umožňuje pouze master-slave replikace. **Logická** replikace funguje tak, že zadaný SQL dotaz se posílá v replikaci dál. Je mnohem flexibilnější než fyzická replikace, nezáleží na verzích, které nemusí být jednotné pro celý cluster. Navíc ani obsah nemusí být uložený na stejném místě na disku, jak je tomu u fyzické replikace. Naopak, logická replikace je náročnější na vytvoření, ale díky její flexibilnosti se počáteční náročnost vyplatí.

### 3.2.2 Replikace v jiných databázových systémech

Replikace je možné provádět i v jiných databázových systémech, ve kterých je dělení replikace a označení serverů odlišné. V rešerši jsou níže rozepsány způsob replikace v Oracle Database a MS SQL Server.

#### Oracle Database

Je jedním z nejznámějších komerčních databázových systémů. Systém nabízí replikace dvojího typu. První je **základní replikace** (*angl. basic replication*). Dochází k replikaci tabulky, ke které mají uživatelé přístup, ale pouze ke čtení, všechny změny probíhají na straně master serveru. Tento typ replikace je vhodný především pro distribuci informací. Oficiální dokumentace Oracle uvádí příklad, kdy obchodní řetězec potřebuje mít vždy aktuální ceny jednotlivého zboží v každé své prodejně, čehož lze díky jednoduché replikaci docílit.

Druhou možností je **symetrická replikace** (*angl. symmetric replication*). Hlavním rozdílem od základní replikace je možnost úpravy dat bez omezení, lze je aktualizovat z jakéhokoliv uzlu v systému. Vhodným využitím je při posílání dat na velké vzdálenosti, kdy zároveň slouží i k rozložení zátěži mezi více bodů.

**MS SQL Server**

Další významným komerčním systémem je SQL Server od firmy Microsoft. Označení pro master serveru se zde uvádí termín **vydavatel** (*angl. Publisher*), termín slave je zde nahrazen označením **odběratel** (*angl. subscriber*). Pro přehlednost se dále bude používat termínů master a slave. Replikace je zde trojího typu: snímkovací, slučovací a transakční.

Základním a nejjednodušším typem je **snímkovací** (*angl. snapshot)* replikace, ze které vychází další dva druhy. Principem je periodické posílání kopií dat do adresáře, do kterého mají přístup slave pouze s právy pro čtení. Data nemusí být vždy aktuální, je zde jistá latence. Mazilu (2010) uvádí vhodnost této metody při použití statických dat, která se nemění nebo je přípustné mít neaktuální data po určitou dobu.

Druhým typem je **slučovací** *(angl. merge)* replikace, která vychází ze snapshot, avšak všechny změny provedené na straně master serveru jsou sledovány pomocí spouštěčů (*angl. trigger*). Zároveň se replikace účastní ještě distributor. Po opětovném spojení master a slave serveru dojde k synchronizaci všech změn. Rozdílem je, že změny se mohou provádět jak na master, tak i ze strany slave serveru. Může tím však dojít ke konfliktu v datech (např. změna stejného řádku na straně master i slave serveru). Problém se ale vyřeší pomocí mechanismů, které určí, jaká data se přijmou.

Posledním typem je **transakční** replikace, která zasílá změny okamžitě při jejím vzniku ke všem slave serverům. Je možnost odeslání přímo v čase vzniku změny nebo v daných časových intervalech. Metoda je vhodná v případě, kdy je potřeba velmi nízké latence v datech. Požadavek na správnou funkčnost je spolehlivé internetové připojení všech uzlů. Důvodem je transakční log, který by bez připojení rychle narůstal na velikosti z důvodu hromadění změn.

## 3.3 PostgreSQL

PostgreSQL je open source objektově-relační databázový systém, jež byl vytvořený na Kalifornské univerzitě v Berkeley. Nativní řešení replikačního mechanismu v PostgreSQL od verze 9.0 je streaming replikace. Další volbou pro replikaci jsou dostupné extenze jako je například Slony-I, pgpool II, Londiste, Bucardo aj. Extenze se liší mezi sebou dostupným programovacím jazykem, mechanismem replikace (synchronní, asynchronní) a také platformou, na které běží. Většina z nich je kompatibilní pouze s Linuxovým prostředím.

**Slony-I**

Slony je jednou z mnoha replikačních extenzí pro PostgreSQL. Je napsaná v programovacím jazyku C Janem Wickem. Jedná se o trigger-based master-slave replikaci (Marcotte, 2005). Webová stránka *EssentialSQL* definuje trigger jako spouštěč, který je spuštěn společně s konkrétní akcí v databázi. Většina triggerů je nastavena tak, aby byla spouštěna v případě provedení změny v datech v tabulkách. Změny se ukládají do logovacího souboru. Na pozadí běží program **slon**, který logovací soubory prochází. Pokud nalezne změnu, replikuje ji na slave server a poté logovací soubor smaže. Obrovskou výhodou Slony-I je, že není potřeba stejné verze PostgreSQL u všech uzlů replikace. Důsledkem toho se provádí změny pouze u daných objektů, nemění se zbytečné celá databáze. Vhodný nástroj pro inicializaci replikace je **slonik**. Jedná se o skript, který komunikuje se Slony-I a informuje jej o uzlech – master a slave serveru.

### 3.3.1 Inicializace replikace v PostgreSQL

Jak již bylo výše zmíněno, v prostředí PostgreSQL lze replikaci inicializovat více způsoby. Jedním ze způsobů je pomocí extenze **Slony-I**, kterou uvádí ve své diplomové práci Markéta Solanská a jejíž postup byl pro tuto inicializaci převzat a upraven. Celý replikační model probíhá v clusteru, který se skládá z uzlů (v našem případě stolní počítač a notebook). **Replikační sadou** nazýváme tabulky, které jsou replikovány mezi jednotlivými uzly clusteru.

Prvním důležitým krokem je nastavení povolení replikace v rámci brány firewall. Do příchozích pravidel firewall povolit port 5432 a komunikační protokol TCP. Dále je nutná úprava textového souboru *pg_hba.conf*, který se nachází ve složce *data* v adresáři PostgreSQL. Do tohoto souboru je nutné na jeho konec dopsat IP adresy master (např. 10.0.0.35) a slave serveru (např. 10.0.0.36). Pokud je potřeba přistupovat k serveru přes síť internet, je možné zadat adresu 0.0.0.0/0. Lomítko za IP (32bit) adresou určuje kolik bitů z 32bitové IPv4 adresy zleva je fixních. Komunikaci lze ponechat s šifrováním MD5.

```
# master server
Host     all   all   10.0.0.35/32      md5
# slave server
Host     all   all   10.0.0.36/32      md5
```

Dále je nutné vytvořit 2 nové textové soubory – pro master a slave server. Každý soubor je vhodné v prostředí Windows uložit do složky *Bin* v adresáři PostgreSQL, zajistí se tak snazší aktivace. Podrobnější postup je uveden v step-by-step návodu. Po úspěšném napojení je již replikace funkční a jakákoliv úprava provedená v replikační sadě na master serveru se promítne na slave serveru.

## 3.4 MySQL

Oficiální stránky MySQL uvádí, že databázový systém poskytuje jak synchronní, tak asynchronní replikaci. Na základě konfigurace je možné replikovat všechny databáze, vybrané databáze nebo pouze vybrané tabulky databáze.

Výchozím nastavením je aktivní asynchronní mechanismus, proto slave servery nemusí být trvale připojeny k master serveru pro získávání aktualizací změn v datech. Aktualizace se mohou provádět přes vzdálená připojení, dočasná, a dokonce i přes přerušovaná připojení. Využívání replikace v prostředí MySQL má několik výhod. Je zde možnost horizontální škálovatelnosti, čímž se zátěž z jednoho serveru rozloží na více serverů. Zápis a aktualizace dat se provádí na master serveru, čtení pouze na slave serverech. Tímto modelem se zajišťuje rychlý zápis dat a zároveň se zvyšuje rychlost čtení. Výhodou je distribuce dat na velké vzdálenosti. Z hlavních dat se vytvoří místní kopie, která slouží k následnému dotazování se, aniž by bylo potřeba přístupu ke vzdálenému master serveru.

Vondra (2011) naopak uvádí, že v prostředí MySQL jsou k dispozici 3 druhy replikací – statement-based, row-based a mixed-based. **Statement-based** (SBR) loguje celé SQL příkazy obsahující změnu dat. Tato replikace je nastavená jako výchozí v MySQL 5.5. Výhodou je použití mnohem méně dat k zápisu do logů, tzn., že celá záloha zabírá méně místa a v případě obnovení ze zálohy je celkový čas mnohem kratší. Velkou nevýhodou jsou příkazy, které nejsou pro tento mechanismus bezpečné. Mezi ně spadají např. LOAD_FILE(), UUID(), UUID_SHORT(), USER(), FOUND_ROWS() atd. V **Row-based** (RBR) replikaci master server ukládá změny do binárního logu,

kde indikuje, které řádky byly pozměněny. Jedná se o nejbezpečnější typ replikace, který je navíc používaný i v jiných databázových systémech. Nevýhodou je velikost dat, jelikož RBR ukládá více dat, než je potřeba do logu uložit. Poslední možností je **mixed-based** (MBR) replikace. Ta kombinuje předešlé způsoby dohromady. Podle potřeby buď použije RBR nebo SBR mechanismus.

Oficiální dokumentace MySQL dělí replikaci podle jejího využití. Jedním z nich je využití replikace pro **zálohu**. Principem je replikace dat na slave server formou zálohy, kdy se slave server pozastaví a nebude ovlivněn procesem na master serveru. Vhodným nástrojem je zde *mysqldump*, který umožní logické zálohování pomocí sady SQL příkazů, které lze spustit pro obnovení původních dat. Nástroj je vhodný pro databáze menších velikostí, při velkém objemu dat je tento nástroj nevýhodný, je mnohem lepší replikovat primární data. Další možností využití replikace je horizontální škálovatelnost, tedy **rozložení zátěže** mezi více servery. Princip je vhodný především pro systém, kde dochází k velmi častému čtení a zároveň k málo častým zápisům. Příkladem využití je webová stránka, ke které se klient připojí, čte si stránky a k aktualizacím dochází výjimečně při správě. Dalším využitím může být **zvýšení výkonu replikace** přidáním druhého master serveru nebo **záloha master serveru** v případě jeho pádu.

### 3.4.1 Inicializace replikace v MySQL

V prostředí MySQL lze replikaci realizovat pomocí nástroje phpMyAdmin. Po přihlášení do phpMyAdmin a po připojení k databázi se v záložce provede nastavení master serveru. V záložce *Replikace* se zvolí nastavení nadřízeného serveru. Je potřeba zvolit, jaká databáze se bude nebo nebude replikovat. Následně se vygenerují příkazy, které je potřeba zkopírovat do konfiguračního souboru *my.ini*, který při použití Wampserveru lze najít v lokaci \wamp64\bin\mysql\mysql5.7.19. Vytvoří se nový uživatel, kterému se přiřadí funkce *Replication client* a *Replication slave*. Dalším krokem je vytvoření stejné databáze na slaver serveru. V phphMyAdmin se následně zvolí možnost *Nastavení podřízeného serveru*, kde se vygeneruje ID serveru, které je potřeba zkopírovat do souboru *my.ini*. Předposledním krokem před spuštěním replikace je zadání údajů o uživateli, který byl vytvořen na master serveru a jeho IP adresa.

## 3.5 Prostorová data

Solanská (2014) uvádí, že prostorová data se ukládají do databází na základě standartu OGC *Simple Features for SQL 1.2.1*. Definuje formáty dvojího typu – ve formě textu (WKT) nebo binárního kódu (WKB). Dle standardu jsou funkce pro získání geometrie z databáze *ST_AsBinary (geometry)* pro bitový zápis WKB a *ST_AsText (geometry)* pro textovou podobu WKT.

MySQL má podporu prostorových dat v sobě již nativně implementovanou. Naopak PostgreSQL potřebuje nainstalovat extenzi PostGIS umožňující práci s prostorovými daty. Extenze umožňuje i práci s rastrovými daty. MySQL sice umožňuje import rastru, ale neumí s ním dále pracovat. Pouze ho zobrazí.

### 3.5.1 Import dat v prostředí PostgreSQL

Do prostředí databáze PostgreSQL lze data importovat díky extenzi PostGIS, která se instaluje přes Stack Builder. Import je možné provést buď přímo v aplikaci *PostGIS Shapefile 2.0 and DBF Loader Exporter*, nebo pomocí příkazové řádky.

**PostGIS**

Postgis je volně dostupná nadstavba s otevřeným kódem pro PostreSQL, která umožňuje práci s prostorovými daty v databázi. Umožňuje ukládat prostorová data do dvojího formátu – geometrická (*angl. geometry*) a geografická (*angl. geography*). Data uložená v geografickém formátu musí být vždy v souřadnicovém systému WGS 84, všechna měření jsou v metrech. Naopak při geometrickém formátu mohou být data uložena v jakémkoliv souřadnicovém systému, následné měření je v daných jednotkách daného systému. Existuje také PostGIS Raster, který umožňuje import rastrových dat a následnou práci s nimi.

### 3.5.2 V prostředí MySQL

Prostředí MySQL má v sobě již nativně implementovanou podporu prostorových dat. Podporovanými formáty jsou: Geometry, point, linestring, polygon, multipoint, multilinestring, multipolygon, geometrycollection. Import se provede v prostředí phpMyAdmin, kde se vybere záložka *Import*. Shapefile (SHP) lze importovat nekomprimovaný, nebo komprimovaný ve formátu ZIP. Důležitým krokem při importu je změnění formátu souboru na *Soubor ESRI*. Je zde podpora rastru ve formátu BLOB (*angl. Binary large object*), avšak nenacházejí se zde prostorové funkce jako je tomu u PostGIS.

## 3.6 Hodnotící kritéria replikace

Hlavním cílem práce je objektivní porovnání 2 databázových replikací na základě předem stanovených kritérií. Proto je důležitou součástí rešerše již vydaných prací týkajících se hodnocení databázových systémů na základě určitých podmínek. V pracích se testovala výkonnost databázových systémů obecně. Nejednalo se přímo o testování replikačního mechanismu, avšak lze zde nalézt kritéria, která lze aplikovat i na testování replikace.

### 3.6.1 Hodnocení databázových systémů

Jeden z mnoha přístupů srovnání databázových systémů použil Tiwari (2011). Na základě daných kritérií srovnal NoSQL databáze. Použil 5 kritérií, z nichž první je **škálovatelnost**, o které bylo již mluveno v kapitole 3.2. Dalším kritériem je **transakční integrita a konzistence**, které má význam pouze tehdy, pokud se data upravují, mění se jejich hodnota, vytvářejí nebo mažou. **Datové modelování**, které slouží pro definování struktury dat v dané databázi (hierarchický, relační, síťový, objektově-relační model). **Podpora dotazování** je nedílným požadavkem pro téměř každou databázi. Posledním kritériem je **přístup a dostupnost rozhraní**. Mimo tato kritéria Tiwari také testoval výkon daných databází pomocí **Yahoo! Cloud Services Benchmark** (YCSB). Jedná se o soubor nástrojů, který obsahuje generátor pracovního zatížení již s vytvořeným příkladem pracovního zatížení. Nástroj je volně dostupný na GitHub[1],

[1] https://github.com/brianfrankcooper/YCSB

je napsán v JAVA a provádí testy na mnoha NoSQL databázích, ale i na MySQL. Dále Tiwari popisuje 2 populární testovací případy. Jedním z nich je **50/50 read and update** test, který je považován za update-heavy test, ze kterého zjistil, že MySQL má porovnatelnou latenci s ostatními systémy do 4 000 operací za s, poté má latence velmi rychlý růst. Naopak druhý **95/5 read and update** test je označován za read-heavy test, při které bylo vyvozeno, že MySQL má lepší výkonnost a škálování při zvýšené pracovní zátěži.

Boicea a kol. (2012) provedli srovnání výkonů databázových systémů MongoDB (NoSQL) a Oracle (SQL) pomocí trojice experimentů, ve kterých sledovali uplynulý čas pro vložení dat, uplynulý čas pro změnu hodnoty dat a uplynulý čas pro vymazání dat. Všechny tři experimenty byly prováděny na odlišných datech v rozpětí od 10 až po milion záznamů. Výsledkem zjistili, že s velkým počtem záznamů roste celkový čas vložení, změny dat a mazání záznamů především v databázovém systému Oracle. Ve výsledku má NoSQL lepší výkon než SQL.

Kabakus a Kara (2016) srovnávali NoSQL a SQL databáze pomocí 4 kritérií, při kterých sledovali uplynulý čas a celkové využití paměti. Kritériem bylo zapsání dvojice klíč–hodnota, přečtení hodnoty odpovídající danému klíči, smazání dané dvojice odpovídající danému klíči a získání všech dat. Pro každý experiment data exponenciálně rostla, aby se ověřilo, jak velikost dat ovlivňuje výkonnost každé databáze. Výsledkem bylo zhodnocení, že každá databáze (NoSQL a SQL) nabízí jiná řešení a nemůže být nahrazena tou druhou. Pokud není potřeba větší konzistence v systému, pak je lepší volbou NoSQL systém, který nabízí větší dostupnost, škálovatelnost a vysoký výkon.

Abramova a kol. (2014) testovali NoSQL databáze (Cassandra, HBase, MongoDB, OrientDB a Redis) na základě času potřebného pro provedení daného požadavku. Testování probíhalo v nástroji YCSB, který obsahuje set výchozích zatížení, ze kterých byly vybrány tři. 50/50, 100/0 a 0/100 read and update, jež jsou nejpoužívanějšími operacemi. Zároveň generovali 600 000 náhodných záznamů, kdy každý z nich obsahoval 10 polí o celkové velikosti 1 kb pro jeden řádek. Zatížení bylo nasimulováno 1000 operacemi, jež představují 1000 žádostí poslaných do databáze, přičemž byla měněna operace a počet uložených záznamů. Existují i další zátěžové testy (*angl. benchmarks*) jako je např. TPC-H nebo SSB, které jsou ale určeny spíše pro SQL databáze.

Dolgikh (2014) ve své diplomové práci využila nástroje *pgbench*, který opakovaně spouští SQL příkazy a sleduje počet transakcí za s. Testováním se sledoval vliv velikosti databáze na počet transakcí za s. Velikost databáze byla nastavena pomocí parametru *scale*. Výsledkem testování je, že při parametru velikosti databáze (*pgbench scale factor*) větším jak 1250 intenzivně klesá výkon celé databáze a provedení SQL příkazů je náročnější a trvá delší čas. Pro sledování zatížení hardwarové části bylo použito nástroje SAR, který monitoruje jednotlivé prvky HW jako je aktivita procesorů a využití operační paměti. K vykreslení grafů údajů naměřených programem SAR bylo použito programu KSAR. Pomocí tohoto programu bylo také sledováno využití sítě.

Bylo zjištěno, že nejdůležitějším kritériem u testování je čas potřebný pro replikaci změn. Proto byl čas jako kritérium zvolen i pro vlastní testování. Důležitým testovacím prvkem, který ovlivňuje dobu replikace je množství záznamů. Proto byla vybrána data tak, aby se mezi sebou lišila právě počtem záznamů. Vzhledem k tomu, že se navíc jedná o prostorová data, byl brán ohled také na počet lomových bodů. Dalšími přístupy bylo využití testovacích nástrojů, které však vstupní data generovala na základě nastavení zátěžového testu a testování neprobíhalo na reálných datech.

### 3.6.2 Úzká hrdla systému

Stránka managementmania.cz definuje **úzké hrdlo** (*angl. Bottleneck*) jako v určitém ohledu limitující a rizikový prvek systému. Ve výrobním či procesním pojetí pravidlo říká, že výrobní či jiný proces (řetězec) je tak rychlý, jak rychlá je jeho nejpomalejší část (článek řetězu). Úzká hrdla systému mohou v extrémních případech způsobit zpomalování dotazování se v databází. Ve výsledku může být dotaz přerušen z důvodu překročení časového limitu na dotaz. Úzkým hrdlem systému může být hardwarová část systému, např. procesor (CPU), operační paměť (RAM), pevný disk (HDD, SSD) nebo síťová karta limitující rychlost přenosu mezi zařízeními. Úzkým hrdlem také může selhání serveru. Vliv má cesta, přes kterou musí replikovaná změna jít.

Mimo HW systému ovlivňuje výkon databáze i např. komunikační linka (šířka pásma, propustnost sítě). Peterka (1991) říká, že čím větší je šířka pásma přenosového kanálu, tím větší je přenosová rychlost, kterou lze na něm dosáhnout. Rozumí se, že čím větší přenosová rychlost je, tím rychleji se změna může replikovat.

V kapitole došlo k vysvětlení důležitých termínů, které se budou nadále v této práci používat. Nejpoužívanějšími termíny bude označení serverů – master a slave. Tyto dva termíny bude možné během celého testování nespočetně zaregistrovat. Pro shrnutí – master server bude zařízení, ze kterého se změna bude odesílat a slave server ji bude přijímat. Byly zde nastíněny limitující prvky replikace, kterými jsou hardwarové prvky počítače, které budou dílčím prvkem pro samotné testování. Jejich parametry budou sledovány a následně vyhodnocen jejich vliv na samotný replikační mechanismus.

# 4 TESTOVÁNÍ REPLIKAČNÍCH MECHANISMŮ

Na základě rešerše a získaných poznatků bylo provedeno testování v databázových systémech PostgreSQL a MySQL. Pro objektivní porovnání výkonů replikací v jednotlivých systémech bylo potřeba stanovit kritéria, která budou během testování sledována a na základě nichž bude v závěru vyhotoveno zhodnocení vhodnosti jednotlivých replikačních mechanismů. Bude testováno Slony-I, u kterého se jedná o asynchronní logickou replikaci a u MySQL se jedná o streaming logickou replikaci. U Slony-I lze vybrat jaká data se z dané databáze budou replikovat, u MySQL se replikuje celá databáze se všemi tabulkami.

## 4.1 Kritéria testování

Pro vlastní testování byla navržena kritéria vycházející z již vydaných prací, viz kapitola 3.6. Nejdůležitějším prvkem celého testování je celkový čas provedení dané operace. Na základě rešerše, kritérium úspěšnosti přenosu není objektivním měřítkem výkonnosti replikace, jelikož ve většině případech je vždy přenos úspěšný v celém rozsahu, neboť nestane se, aby byla přenesena pouze část.

### 4.1.1 Výkon serveru

Nedílným faktorem, který velmi ovlivňuje výkon replikace je samotný server, resp. zařízení, na kterém replikace probíhá. Pro porovnání byla použita dvě výkonem odlišná zařízení – osobní počítač (*angl. personal computer)* a notebook. Dále se již budou používat pouze označení PC a NB. Konkrétní parametry jsou uvedeny v tabulce níže. Zařízení byla připojena k síti pomocí síťové kabelu OEM CAT5E UTP, který vedl do routeru. V dalším testování bylo využito přímého napojení zařízení pomocí stejného kabelu.

Pro otestování samotného vlivu procesoru serveru bylo provedeno testování na obou zařízeních v opačném směru – PC nyní představoval slave server a NB naopak master server. Během samotného testování bylo jasné, že pro méně výkonné zařízení bude náročnější i samotná změna geometrie. Zatímco na stolním PC trvalo přesunutí vrstvy *BPEJ* cca 20 min, na NB se čas vyšplhal na více jak 40 min.

Tab. 1: Parametry zařízení

| Parametry | stolní PC | notebook |
|---|---|---|
| **operační systém** | Windows 8.1 | Windows 10 Home |
| **RAM** | 8 GB | 4 GB |
| **CPU** | Intel® Core™ i5-4590, 3.30 GHz | Intel® Core™ i3-2330M, 2.20 GHz |
| **Grafická karta** | NVIDIA GeForce GTX 960 | NVIDIA GeForce GT-540M, 1 GB VRAM |
| **Síťová karta** | Realtek® RTL8111G Gigabit | Broadcom NetLink (TM) |
| **Síťový kabel** | OEM CAT5E UTP | OEM CAT5E UTP |

### 4.1.2 Využití procesoru

Aby bylo možné replikační mechanismy porovnat, bylo monitorováno využití procesoru. Pro monitorování bylo použito programu HWMonitor Pro, který umožňuje sledovaná data zapisovat do logu, který se po konci nahrávání vytvoří. V logu jsou obsaženy vygenerované grafy pro jednotlivé prvky znázorňující, jak se jejich hodnoty mění v čase. Bylo sledováno využití procesoru a také využití na jednotlivých jádrech.

### 4.1.3 Přenosová rychlost mezi zařízeními

Pokud replikace probíhá v rámci jednoho serveru, není zde vliv šířky pásma ani rychlosti připojení. Pokud replikace neprobíhá na 1 zařízení, musí se s těmito vlivy počítat. Omezením přenosové rychlosti mezi zařízeními se zajistí sledování vlivu rychlosti přenosu při replikaci změn. Přenosová rychlost byla omezena programem NetLimiter 4, který umožňuje limitovat jak lokální síť, tak i připojení k internetu. V rámci testování byla limitována pouze lokální síť, protože zařízení pracují na lokálním připojení. Rychlost připojení je dána síťovou kartou, kterou obě zařízení disponují, viz Tab. 1. Došlo k omezení stahování a nahrávání na jednotných 10 Mbps (Megabits per second) na obou zařízeních. Hodnota je dána na základě rychlostí, kterých bylo dosaženo bez omezení pomocí MySQL mechanismu. U MySQL hodnoty nepřekročily 10 Mbps, proto byla tato hodnota nastavena, aby bylo otestováno chování Slony-I při takto nízké přenosové rychlosti. Je pravda, že tato hodnota je velmi nízká, ovšem jedná se o simulaci, kdy šířku pásma může využívat i jiný proces. Uživatel může každému procesu přiřadit konkrétní šířku pásma, aby proces neomezoval další procesy. Tato hodnota byla nastavena, aby bylo reálné porovnat časy u obou replikačních mechanismů za stejných podmínek, resp. stejné přenosové rychlosti. Vyšší omezení by v případě MySQL znamenalo prodloužení již takto časově náročné operace.

Aby se omezil vliv routeru, byla zařízení spojena přímo. Rychlost a duplexní režim na stolním PC lze nastavit maximální hodnota až 1 Gbps (Gigabits per second), avšak NB měl limitní hodnotu 100 Mbps. Rozdílnost maximálních hodnot je důsledkem síťových karet, kterými obě zařízení disponují. I když by stolní PC měl vyšší přenosovou rychlost, tak by touto rychlostí NB nebyl schopen data přijímat, proto byla hodnota na obou zařízeních nastavena na 100 Mbps.

### 4.1.4 Čas přenosu

Důležitou proměnnou pro sledování výkonu replikace je čas. Časem je zde myšlena doba, za kterou se změna replikuje na slave server. Čas byl sledován pomocí příkazové řádky v prostředí PostgreSQL. V prostředí MySQL bylo použito programu Monyog. Jednotlivé naměřené časy byly zapsány v s do tabulky v prostředí MS Excel, které byly následně statisticky vyhodnoceny pomocí grafu, tabulky, popř. vloženy do R Studia pro tvorbu boxplotů.

### 4.1.5 Formát a množství dat

Při testování byla zohledněna variabilita prostorových dat. Z každé datové sady byly k testování použity pouze vybrané vrstvy, které se od ostatních liší strukturou a objemem. Vektorová data byla zvolena tak, aby v testování byly zastoupeny jak bodové, liniové tak i polygonové vrstvy. Důležitým parametrem pro testování je množství lomových bodů, které liniové a polygonové vrstvy obsahují. Z datové sady ArČR500 byla vybrána bodová vrstva *Castiobcebody*. Z datové sady Data200 byly vybrány dvě vrstvy – polygonová vrstva obcí a liniová vrstva řek ČR. Další vrstvou je polygonová vrstva *BPEJ*, která ale pro svou náročnost při původní rozloze na celou ČR, byla ořezána na území Olomouckého kraje. I tak je stále vrstva největší, co se počtu lomových bodů týče. Polygonová vrstva *svět* představující kontinenty světa z datové sady Natural Earth obsahuje sice pouze 1 polygon, ale svým počtem lomových bodů převyšuje vrstvy s větším množstvím záznamů. V Tab. 2 jsou získané údaje o počtu lomových bodů, počtu záznamů a velikosti tabulky v prostředí pgAdmin III.

Tab. 2: Shrnující údaje o použitých vektorových vrstvách

| název vrstvy | zdroj/datová sada | typ | počet lomových bodů | počet záznamů | velikost tabulky |
|---|---|---|---|---|---|
| BPEJ | Státní pozemkový úřad | polygonová | 3 725 023 | 31 280 | 228 MB |
| Castiobcebody | ArcČR ® 500 | bodová | – | 15 092 | 12 MB |
| Data200_reky | Data200 | liniová | 338 959 | 14 606 | 21 MB |
| Data200_obce | Data200 | polygonová | 672 299 | 6 353 | 27 MB |
| svět | NaturalEarth | polygonová | 411132 | 1 | 4,39 MB |

## 4.2 Postup testování

Testování probíhalo ve dvou databázových systémech – PostgreSQL a MySQL. Postupy se od sebe téměř neliší, jen je rozdíl v použitém softwaru. Výhodou obou systémů je podpora práce s daty v programu QGIS. Během testování byly provedeny následující operace:

1) Změna geometrie všech prvků vektorové vrstvy v prostředí QGIS
2) Atributová změna záznamů pomocí SQL příkazu u vrstvy *BPEJ*

Během testování budou sledována tato kritéria:

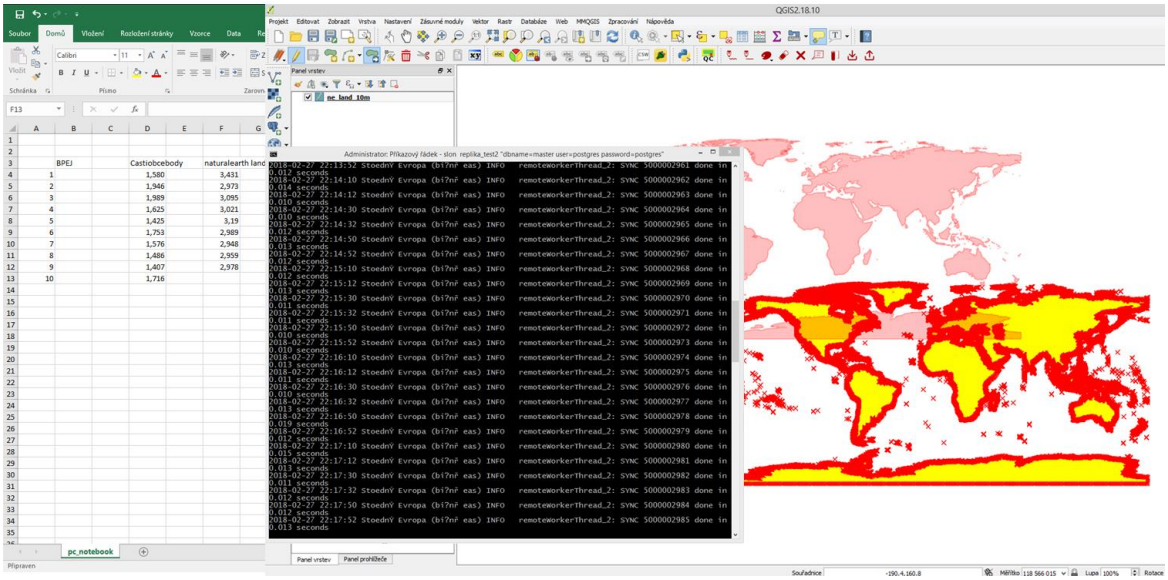
- Zatížení procesoru master/slave serveru
- Přenosová rychlost odesílání/přijímání dat
- Čas potřebný pro replikování změny

### 4.2.1 Testování v prostředí PostgreSQL

Od PostgreSQL verze 9.0 disponuje systém nativní streaming replikací. Existuje 18 extenzí, které umožňují další typy replikace. Zvolenou extenzí pro replikaci byla Slony-I z důvodu návaznosti na práci Mgr. Markéty Solanské. Výhodou extenze je její snadnost inicializace replikace, a především funkčnost v prostředí MS Windows. Testování pro každou vrstvu bylo zopakováno desetkrát pro následné statistické vyhodnocení. Po deseti měřeních, již hodnoty nevykazovaly žádné odchylky, proto je tento počet dostačující. Replikace probíhala v rámci PC a NB. Během testování byly role master a slave serveru uděleny jak PC, tak i NB. Pro většinu testování byla role master přiřazena právě PC.

V rámci celého testování byl sledován čas v příkazové řádce a na pozadí běžel program HWMonitor Pro, který snímal další údaje. Čas potřebný pro replikování změny na NB byl vždy zapsán do tabulky v MS Excel. Během testování na obou zařízeních běžela pouze samotná replikace a SW potřebný k testování, aby se omezilo vlivu ostatních programů na využití procesoru. Součástí testování je také omezení přenosové rychlosti mezi zařízeními. Testování probíhalo stejně jako bez omezení, pouze navíc v pozadí běžel program NetLimiter 4 s nastavenou omezenou přenosovou rychlostí.

Na obr. 2 lze vidět průběh testování. Prostředí QGIS, ve kterém se provádí změna geometrie posunutím celé vrstvy. V příkazové řádce se vypisují informace o chodu replikace. Během replikování změny se v řádce zobrazí celkový čas potřebný pro provedení operace. Čas se následně zapisuje do tabulky MS Excel, která se po testování vyhodnotí.



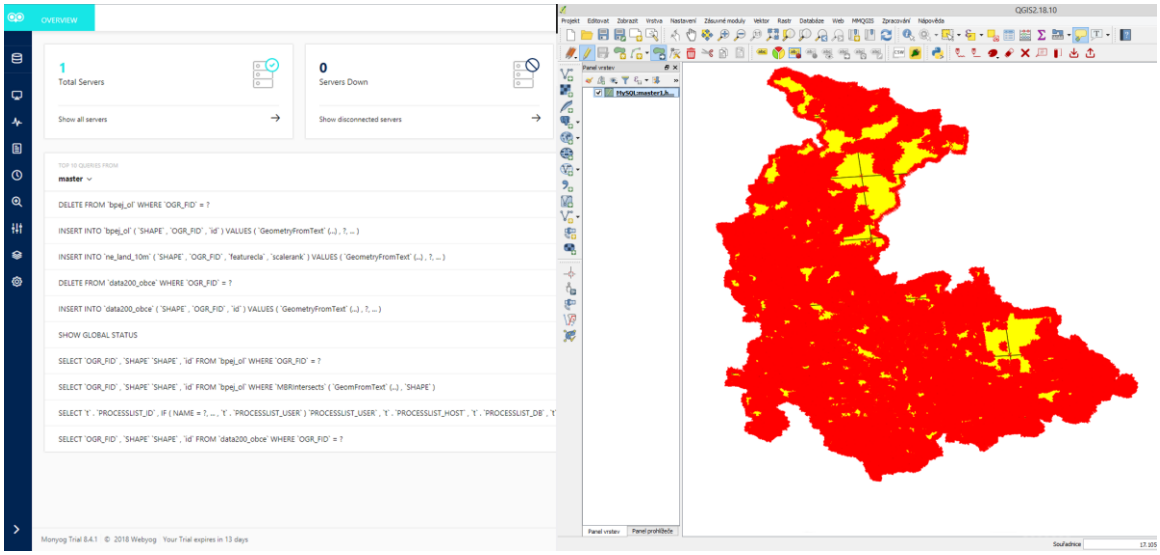

Obr. 2: Ukázka z testování v rámci PostgreSQL pomocí QGIS a příkazové řádky

### 4.2.2 Testování v prostředí MySQL

Testování v prostředí MySQL bylo obdobné PostgreSQL. Změny v datech byly provedeny stejným způsobem. Změny v geometrii byly provedeny v programu QGIS. Atributové změny byly provedeny v phpMyAdmin. Testování bylo pro každou vrstvu opět provedeno desetkrát, obzvlášť u MySQL je měření potřeba více. Čas zde totiž velmi kolísal a během deseti měření nebylo zjištěno dalších větších odchylek.

Pro měření času trvání replikování změny bylo využito programu Monyog, který běží jako webová aplikace. Zde je možnost sledovat dotazy, které běžely na obou serverech. Je zde také funkce monitoringu v reálném čase, kde je možné mimo čas sledovat i množství odeslaných a přijatých dat. Během monitoringu běží časová osa, na které je vidět okamžik, ve kterém se data začaly odesílat. Jednoduše se z ní odečetl čas potřebný pro replikaci změn.

Na obr. 3 lze vidět prostředí QGIS, ve kterém se provádí změna geometrie vrstvy *BPEJ*. Po uložení a úspěšné replikaci změny se v prostředí Monyog vyčte celkový čas potřebný pro tuto operaci, resp. čas potřebný pro provedení daných dotazů.




Obr. 3: Testování v rámci MySQL pomocí QGIS a Monyog

### 4.2.3 Provedené operace

Jak již bylo výše zmíněno, kromě změny geometrie celé vrstvy se testovala atributová změna ve vrstvě *BPEJ*. Pomocí příkazů byla provedena změna u 30 000 řádků. Nižší počet řádků byl příliš rychlý ke změření a případné tvorbě grafů. To samé platí pro ostatní vrstvy. Proto pro porovnání slouží pouze časově nejnáročnější vrstva – *BPEJ*. U této vrstvy bylo možné z naměřených hodnot vytvořit validní grafy.

Atributová změna byla v obou systémech provedena pomocí SQL příkazu UPDATE. Příkaz UPDATE se celkově řadí mezi náročnější operace. Nebylo proto nutné testovat vkládání nových záznamů pomocí příkazu INSERT. Čas pro vložení 30 000 záznamů by byl nižší než pro úpravu již stávajících záznamů. Příkaz UPDATE byl v rámci replikace pomocí Slony-I zadán v prostředí pgAdmin v SQL konzoli. Příkaz vypadal následovně: `UPDATE bpej SET b5 = 00100 WHERE gid < 30001;`. Příkazem SET byla nastavena změna hodnoty atributu *b5* na hodnotu 00100, kdy příkazem WHERE se změna limitovala pro prvních 30 000 řádků pomocí ID sloupce s názvem *gid*. Příkaz v databázovém systému MySQL byl prakticky totožný s prostředím PostgreSQL. Jediný rozdíl byl v názvu atributu s ID, který se zde označuje *OGR_FID* (u PostgreSQL se jednalo o *gid*).

# 5 POROVNÁNÍ DATABÁZÍ

Pro testování byly v obou systémech použity stejné vrstvy a stejné metody provedení změn v datech. Lze je tedy mezi sebou porovnat. V rámci porovnání byla použita kritéria navržená před samotným testováním, viz kapitola 4.1.

Pro tvorbu grafů byla použita vrstva *BPEJ*, u které se měnila geometrie všech prvků. Svým počtem lomových bodů převyšuje ostatní vrstvy. Náročnost vrstvy se projevila nejvyššími hodnotami u všech sledovaných kritérií. U menších vrstev byl problém, že replikace změny proběhla velmi rychle. Ostatní vrstvy jsou použity u dalších kritérií, resp. nastavených podmínek, ve kterých se měří pouze čas. Podmínky, které byly nastaveny během testování jsou následující:

1) **PC jako slave server** – prohození rolí zařízení, PC představuje slave server, NB je tedy jako master server
2) **PC jako master server** – výchozí nastavení během testování
3) **Přímé napojení** – PC jako master server spojený s NB síťovým kabelem
4) **Omezení na 10 Mbps** – PC jako master server s omezenou přenosovou rychlostí na 10 Mbps

Cílem porovnání databází mezi sebou je zjištění, jakým způsobem oba replikační mechanismy fungují. Oba replikační mechanismy fungují odlišně, což se projevilo u všech kritérií. Pomocí Slony-I se změna v datech replikovala tak, že se pouze měnil daný atribut, resp. geometrie, ne celá vrstva. Jako první krok se změna provedla na master serveru, až poté se informace o změně odeslala na slave server.

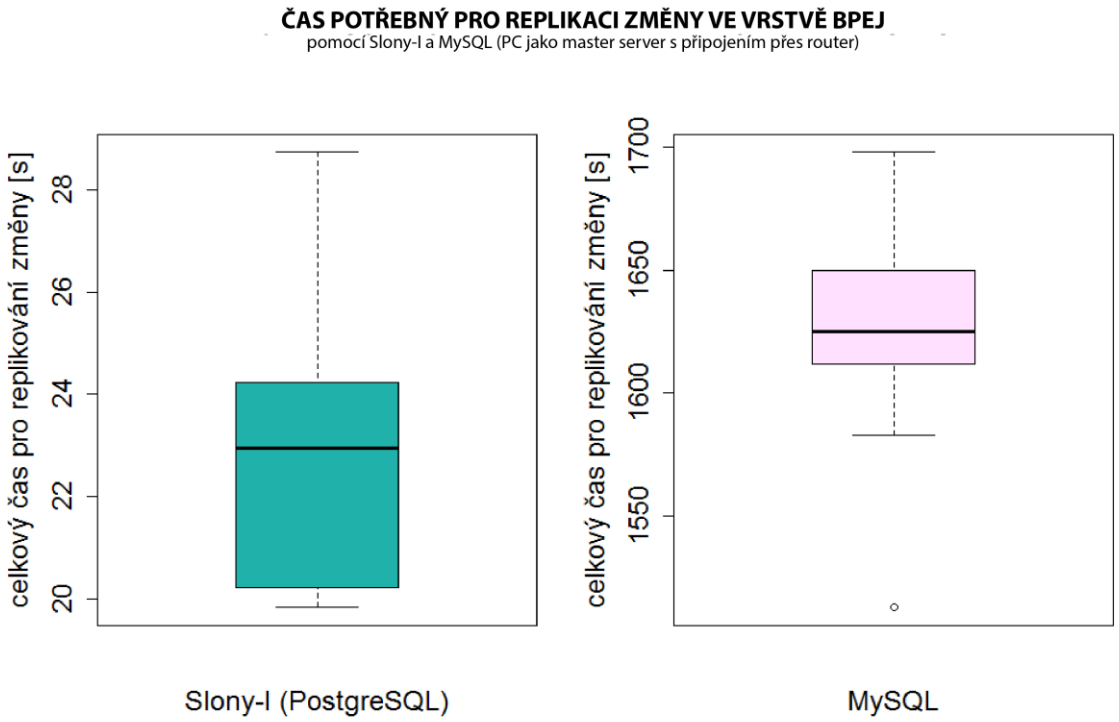
Testování v prostředí MySQL bylo časově náročnější, pohybovalo se v řádech desítek min, proto lze v grafech vidět pouze část měření, konkrétně prvních 10 min od zahájení replikace. Důvodem delšího času je způsob replikování změn. Ze sledování průběhu replikace a příkazů v programu Monyog bylo zjištěno, že změny v geometrii jsou provedeny následovně viz Obr. 4. Jako první se celá vrstva, resp. daný prvek smaže a následně se znova vkládá s upravenými hodnotami. Zatímco se změny provádí na master serveru, tak se již odesílají na slave server. Je tedy logické, že tímto způsobem čas pro replikování změn mnohonásobně nabyde oproti Slony-I.

TOP 10 QUERIES FROM
**master**

| | COUNT | TOTAL TIME | AVERAGE LATENCY |
|---|---|---|---|
| DELETE FROM `bpej_ol` WHERE `OGR_FID` = ? | 31K | 13:36.000 | 00.026 |
| INSERT INTO `bpej_ol` ( `SHAPE` , `OGR_FID` , `id` ) VALUES ( `GeometryFromText` (...) , ?, ... ) | 31K | 13:34.000 | 00.026 |

Obr. 4: Příkazy potřebné pro replikaci změny geometrie v prostředí MySQL

Výsledkem jsou naprosto odlišné časy potřebné pro replikaci, které jsou zvizualizovány pomocí boxplotů viz Graf 1. Boxploty jsou sice situovány vedle sebe, avšak je nelze mezi sebou porovnat na první pohled. Oba systémy mají naprosto odlišné časy, proto každý boxplot má svou vlastní stupnici. Zatímco u Slony-I se čas pohybuje v rozmezí 20–30 s, u MySQL se čas vyšplhal až na 1700 s, což je v přepočtu více jak 28 min. U jiné vrstvy takto vysoký čas nebyl již naměřen.

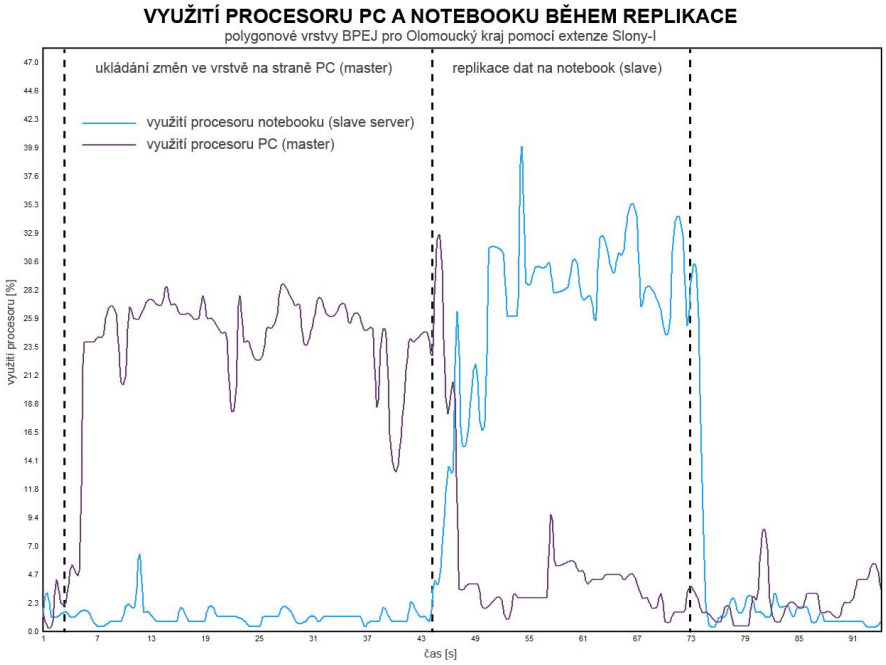



Graf 1: Boxploty časů potřebných pro replikaci změn ve vrstvě BPEJ

## 5.1 Využití procesoru

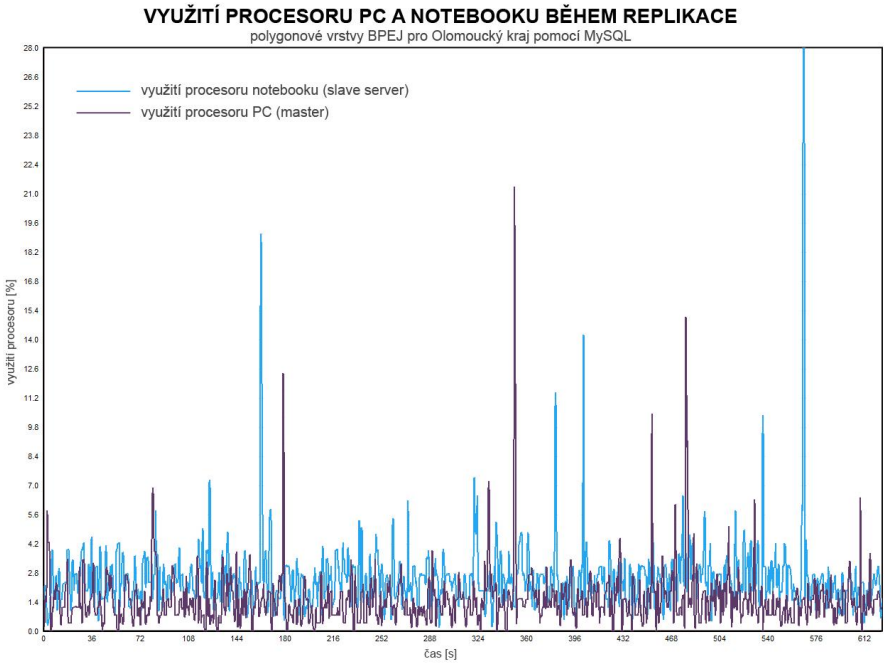
Jako první proměnná, která byla během testování měřena, je využití procesoru. Pro srovnání obou databází bylo zvoleno měření při nastavení stolního PC jako master serveru a připojení přes router.

Graf 2 porovnává využití procesoru master a slave serveru během replikace změny ve vrstvě *BPEJ* pomocí Slony-I. V grafu je pro větší přehlednost pomocí přerušovaných čar vymezen úsek samotné replikace. Barevně je odlišeno zatížení procesoru stolního PC a NB. Na první pohled je patrný moment, ve kterém se změna ve vrstvě ukládá a začíná replikovat na slave server. Zatížení procesoru na straně master serveru je pouze po dobu uložení změn do samotné vrstvy. Vrchol dosahuje zatížení až 33 %, poté zatížení klesá k hodnotám nepřekračující 10 % využití procesoru. U slave serveru tvořeného méně výkonným NB je viditelné vyšší zatížení procesoru, dosahujících hodnot až 40 %. Po úspěšné replikaci změn se hodnoty vrací do normálu kolem 5 % zatížení. Využití procesoru nikdy neklesne na nulové hodnoty. Vždy běží jisté služby a programy na pozadí. Avšak byla snaha vliv těchto programů minimalizovat. Dá se říct, že pod 5 % již proces replikace neběžel ale hodnoty zatížení nikdy nebudou nulové, protože procesor pracuje neustále.




Graf 2: Využití procesoru PC (master) a NB (slave) v rámci replikace vrstvy BPEJ pro Olomoucký kraj pomocí Slony-I

Na Grafu 3 lze vidět, že využití procesoru se v průběhu replikace ve výkyvech vyšplhá až na hodnoty nepřesahující využití procesoru nad 30 %. Výkyvy by se daly připsat náročnějšímu SQL dotazu, resp. záznamu, ve kterém je uloženo více lomových bodů. Mezi výkyvy procesory běží téměř na hodnotách, které jsou typické pro klidový režim. V grafu lze vidět, že zatížení slave serveru je vyšší než master serveru. Důvodem je výkon zařízení. NB v roli master server je méně výkonným zařízením, proto stejná operace zatíží procesor více než je tomu právě u PC.

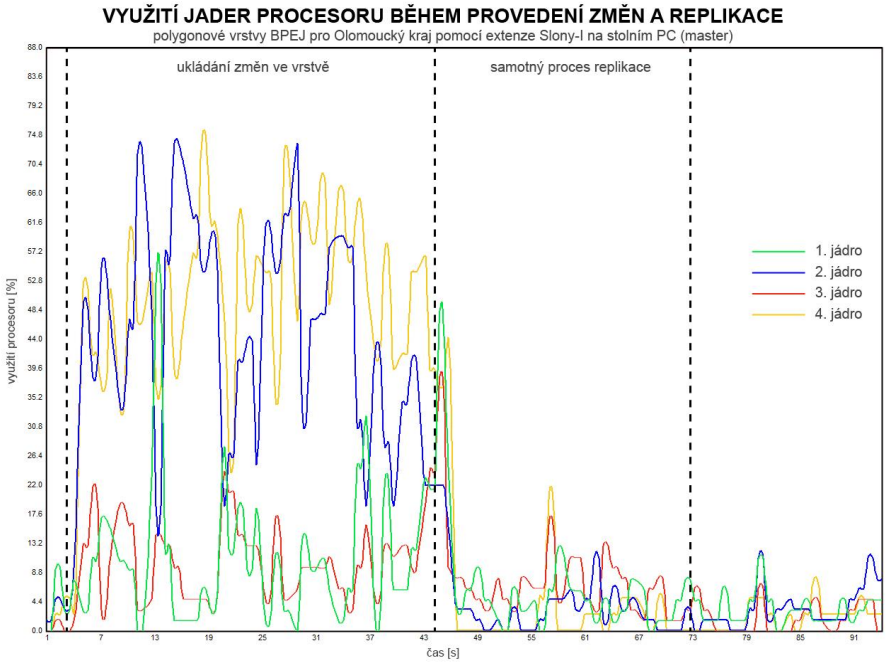



Graf 3: Využití procesoru PC (master) a NB (slave) v rámci replikace vrstvy BPEJ pro Olomoucký kraj pomocí MySQL

Při srovnání obou grafů je ihned patrný rozdíl. Zatímco u Slony-I během replikace pracuje pouze slave server přijímající změnu v datech, u MySQL pracují oba servery po celou dobu replikace. Jiným způsobem replikování dat je v důsledku i jiné % využití procesoru. U Slony-I využití procesoru dosahuje až 40 %, zatímco u MySQL hodnoty v průměru nepřekročí 10 %.

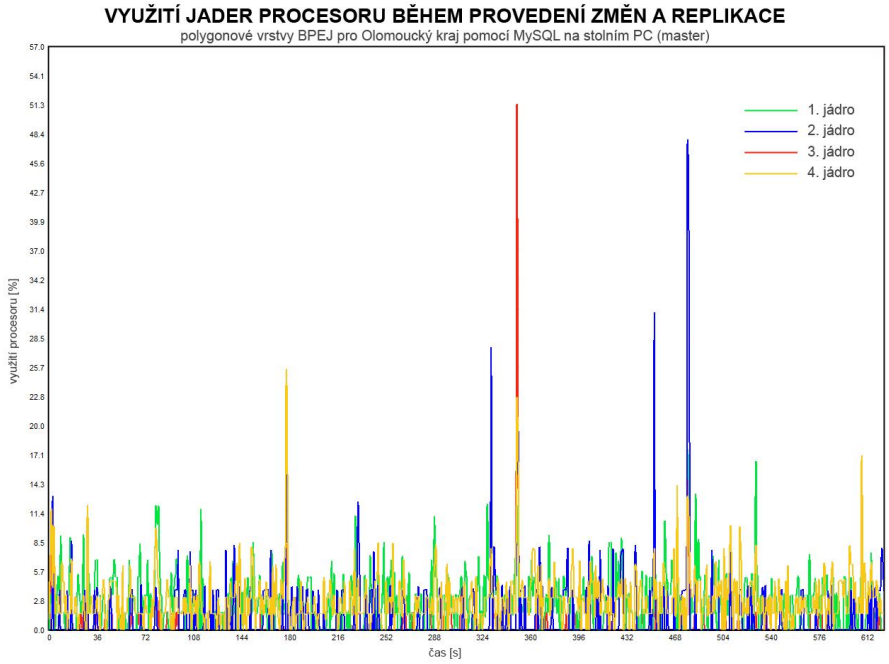
### 5.1.1 Využití jednotlivých jader procesoru

Mimo měření průměrného využití procesoru bylo také sledováno využití na jednotlivých jádrech procesoru. Pro zobrazení využití jednotlivých jader během replikace byl zvolen pouze PC ve funkci master serveru, který disponuje 4 jádry. Graf 4 zobrazuje využití jednotlivých jader během provedení změn v datech a následné replikace. Graf je rozdělený na dvě hlavní části – první část představuje čas, během kterého se změny v datech ukládaly do vrstvy v rámci QGIS a druhá část, ve které se změny již posílají na slave server. Je patrné, že druhá část již není na procesor nikterak náročná, využití procesoru se pohybuje v hodnotách, ve kterých běží zařízení v klidu. Naopak ukládání změn pro takto velkou vrstvu je náročné a využití procesoru se dostalo až do necelých 75 %. Z grafu lze také vyčíst, že nejvíce zátěže proběhlo na 2. a 4. jádru procesoru. Byla snaha nalézt vysvětlení, proč právě tato jádra mají největší zátěž, ale žádná zmínka v dokumentacích, ani v diskuzních fórech nebyla.




Graf 4: Využití jednotlivých jader procesoru master serveru během ukládání a replikace změn ve vrstvě BPEJ

Graf 5 zobrazuje využití jednotlivých jader naopak během replikace pomocí MySQL. Využití jednotlivých jader bez výjimečných výkyvů nepřesahuje ani na jednom serveru hodnotu 20 %. U výkyvů v zátěži procesoru lze vidět, že se vždy jedná o jiné jádro. Ani u dokumentace MySQL nebyla nalezena zmínka o využití jednotlivých jádrech. Pouze zmínka v diskuzi o tom, že MySQL automaticky využívá více jader.

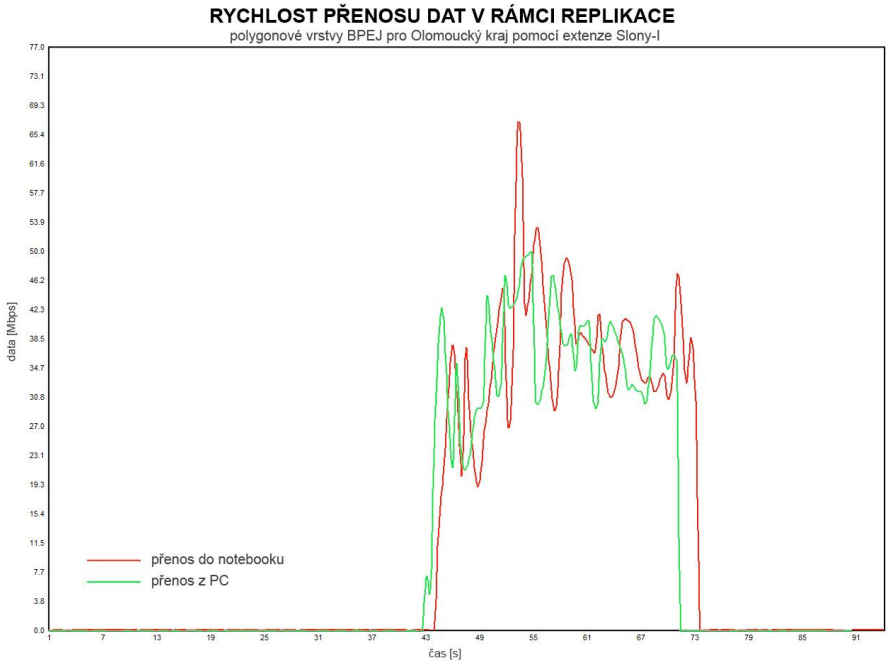



Graf 5: Využití jednotlivých jader procesoru master serveru během ukládání a replikace změn ve vrstvě BPEJ pomocí MySQL

## 5.2 Přenosová rychlost dat

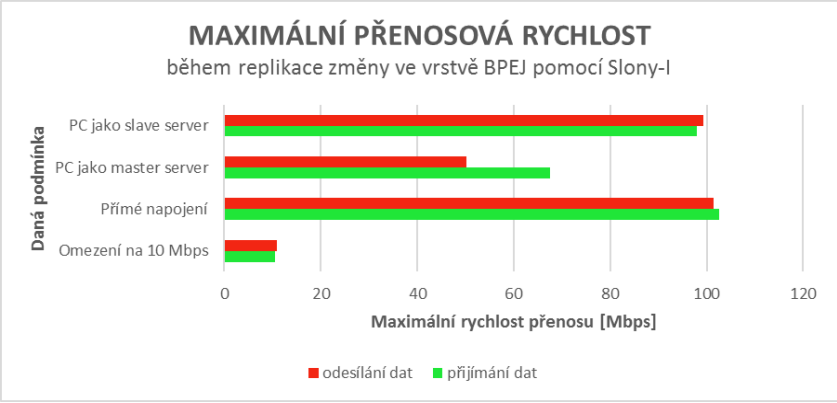
Dalším měřeným kritériem byla dosahující přenosová rychlost. Byly vytvořeny grafy, které shlukují naměřené hodnoty pro oba servery. Jak master serveru, který daná data odesílá, tak pro slave server, který změny přijímá.

Graf 6 zobrazuje přenosovou rychlost dat ze strany PC (master) na NB (slave) během prvního měření. Z grafu lze velmi dobře vyčíst, v jakém momentu bylo spuštěno odesílání změn na slave server. Přenos dat probíhal mezi 44. a 74. s, samotný přenos tedy trval přibližně 30 s. Rychlost přenosu dat ani na jednom ze zařízení nepřekročila hodnotu 70 Mbps. Po úspěšném přenesení změn na slave server přenosová rychlost samozřejmě klesne téměř na nulovou hodnotu. Neklesne na nulovou hodnotu z důvodu komunikace mezi master slave a kvůli některým službám MS Windows a dotazům Slony-I.



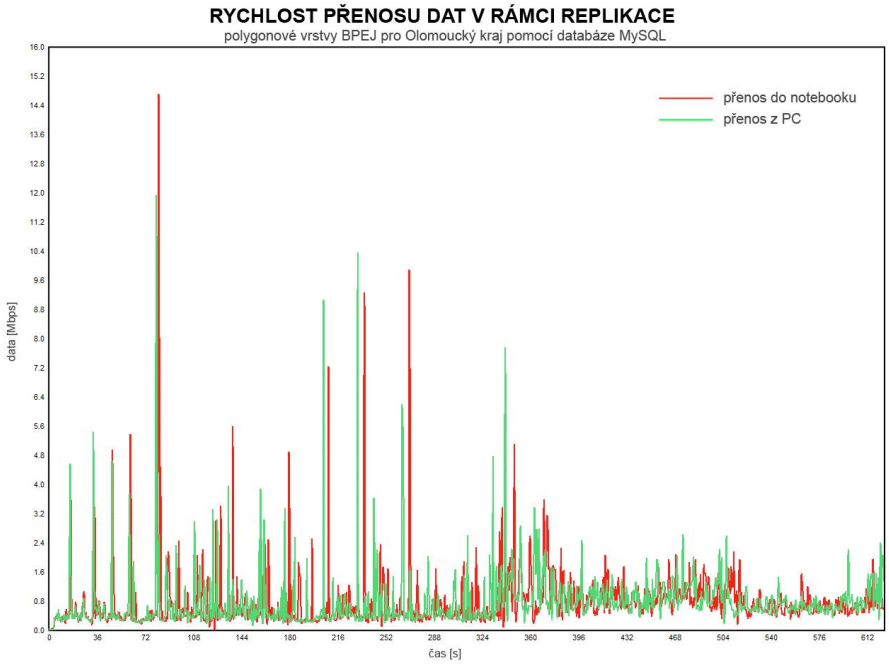

Graf 6: Rychlosti přenosu dat v rámci replikace vrstvy BPEJ pomocí Slony-I

Graf 7 porovnává maximální dosažené přenosové rychlosti za daných podmínek. Je logické, že nejnižších rychlostí je dosaženo při omezení na 10 Mbps. Hodnota není přímo rovna 10 Mbps, vždy měření kolísalo řádově do 10 % a může zde být malá nepřesnost. Z grafu je patrné, že při přímém napojení obou zařízení bylo dosaženo nejvyšších přenosových rychlostí. Zatímco při PC jako master serveru byla přenosová rychlost nižší, při prohození rolí se hodnoty zvýšily. Je to způsobeno tím, že dotazující server je výkonnější zařízení. Výkonnější je z pohledu rychlosti zapisování změn k sobě na disk.



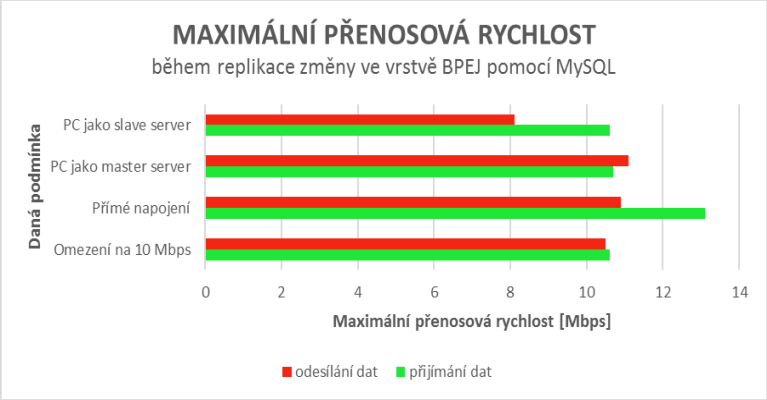

Graf 7: Maximální přenosové rychlosti během replikace změn v BPEJ za daných podmínek pomocí Slony-I

Graf 8 představuje rychlosti přenosu dat v prostředí MySQL během replikace vrstvy *BPEJ*. Maximální rychlost přenosu dat byla kolem 15 Mbps, ale jedná se o jeden z nejvyšších výkyvů. Jinak rychlost přenosu na obou zařízeních nepřekročí 10 Mbps. Zpravidla první výkyv je vždy ze strany master serveru, odesílající data, pak následuje výkyv na slave serveru. Výkyvy mohou být způsobeny náročnějším příkazem, resp. záznamem obsahující mnoho lomových bodů.




Graf 8: Rychlosti přenosu dat v rámci replikace vrstvy BPEJ pomocí MySQL

Graf 9 porovnává maximální naměřené přenosové rychlosti v rámci přenosu změn u vrstvy *BPEJ* pomocí MySQL. U žádné z testovaných podmínek naměřená hodnota nepřekročí 14 Mbps, pohybuje se kolem 10 Mbps. Lze pozorovat, že nejnižší hodnotu má master server při podmínce PC jako slave server. Je to dáno výkonem NB, který je mnohem nižší, než je tomu u PC. Master server je tedy svým výkonem relativně omezen, proto není schopen ukládat změny k sobě a zároveň odesílat změny na slave server tak rychle, jako tomu bylo právě u PC. Naopak nejvyšší naměřenou přenosovou rychlost má slave server při zapojení PC jako master server. Je to dáno faktem, že slave server získává příkazy o změnách velmi rychle díky výkonnému master serveru. Rychlost ale není až tak velká, aby změny nestíhal slave server k sobě zapisovat.



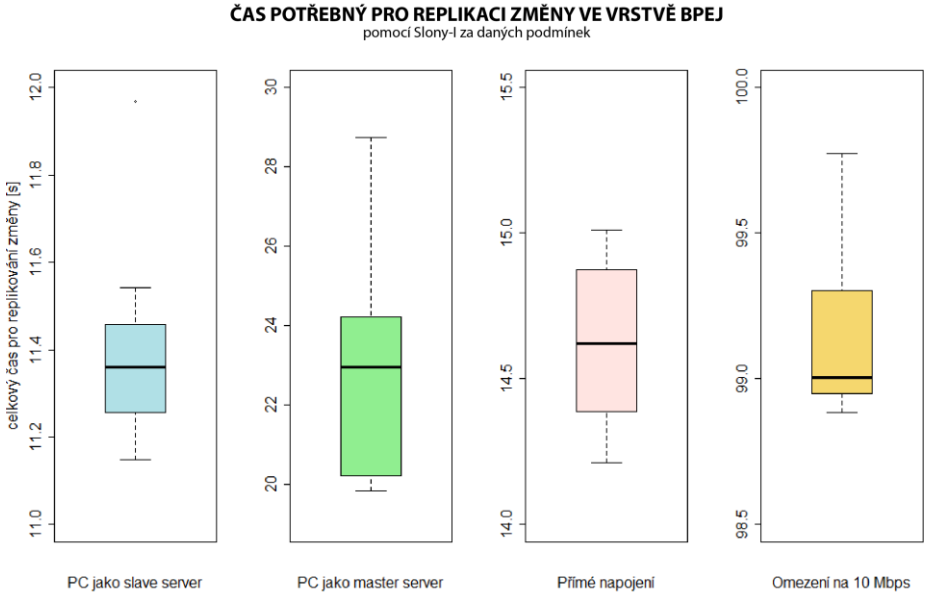

Graf 9: Maximální přenosové rychlosti během replikace změn v BPEJ za daných podmínek pomocí MySQL

Při porovnání obou grafů lze konstatovat, že u Slony-I se změna v datech replikuje co nejrychleji, jak systém umožňuje – rychlost přenosu dat dosahuje maximálních hodnot, která jsou zařízení na základě svých parametrů schopna. Tím je docíleno rychlého přenosu dat. Zatímco u MySQL je změna replikována kontinuálně řádek po řádku, rychlost přenosu je téměř sedmkrát nižší, než je tomu u Slony-I. Na základě tohoto kritéria lze tedy konstatovat, že při replikování změn v datech pomocí Slony-I velmi záleží na rychlosti připojení.

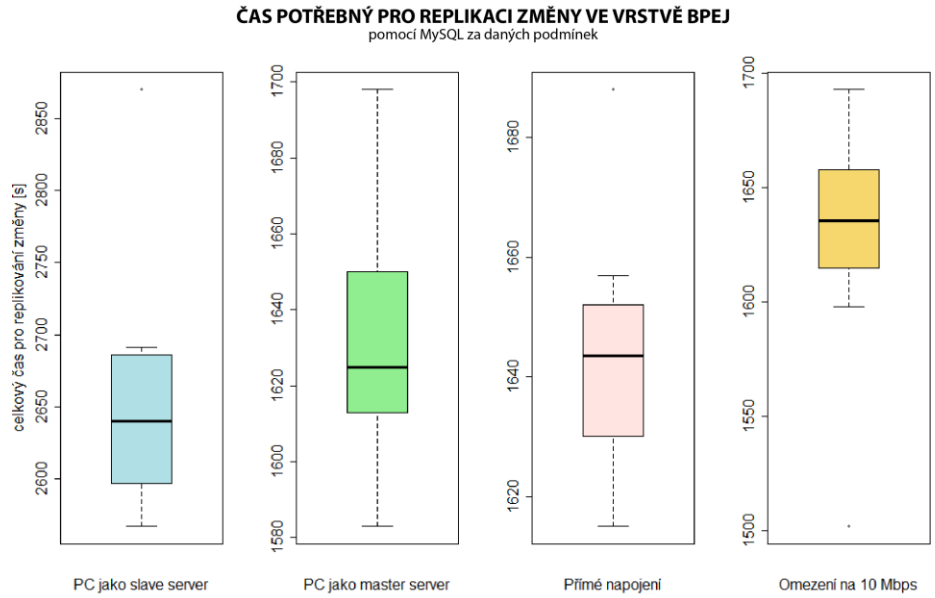
## 5.3 Čas přenosu

Nejdůležitějším kritériem pro porovnání obou replikačních systémů je právě čas. Čas, za který se daná změna replikuje na slave server. Níže jsou statistickými metodami vizualizovány výsledky za daných podmínek, které jsou v kapitole 4.1.4 zmíněny. K statistickému vyhodnocení byly použity boxploty, které však mají odlišné stupnice. Nelze je tedy přímo mezi sebou porovnat. U měření s omezenou přenosovou rychlostí byly hodnoty tak moc odlišné, že by společný boxplot byl téměř nečitelný. Proto pro vizuální porovnání hodnot mezi sebou byly použity grafy s průměrnými hodnotami za daných podmínek.

Graf 10 a 11 porovnává časy potřebné pro replikaci změn u vrstvy *BPEJ* pomocí Slony-I a MySQL. Oba grafy představují boxploty, ze kterých lze vyčíst základní statistické metody jako je medián, 1. a 3 kvartil a případné extrémní hodnoty. Na první pohled je patrný časový rozdíl v obou systémech. Mimo časový rozdíl je ale vidět rozdíl v rozptylu hodnot, který vznikl na základě 10 měření.

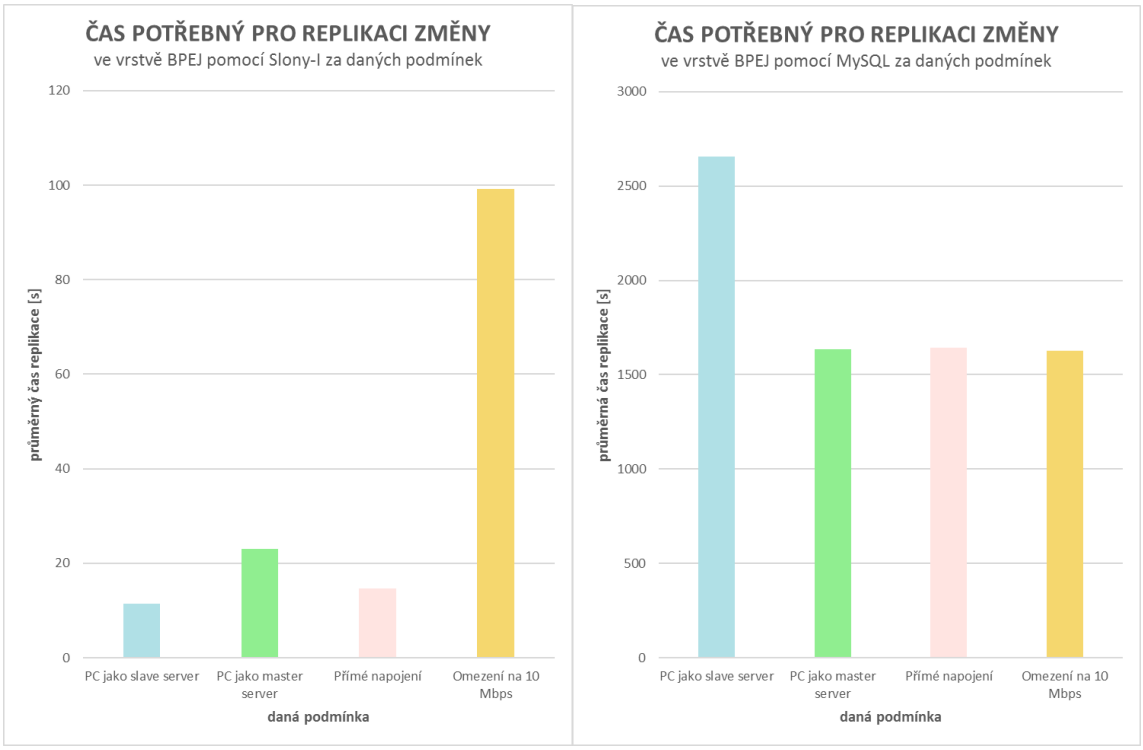



Graf 10: Boxploty časů potřebných pro replikaci změny ve vrstvě BPEJ za daných podmínek pomocí Slony-I




Graf 11: Boxploty časů potřebných pro replikaci změny ve vrstvě BPEJ za daných podmínek pomocí MySQL

Další série dvou grafů zobrazují průměrné hodnoty časů naměřených během daných podmínek. Na grafu vlevo je patrné, že nejdéle trvala replikace pomocí Slony-I při omezené přenosové rychlosti. Slony-I se snaží pro přenos změn vždy použít celou šířku pásma a pokud je omezená, je logické, že se zvýší doba replikace (viz předchozí podkapitola). Pokud se tedy přenosová rychlost omezí téměř na minimum, logicky se čas potřebný pro replikaci razantně prodlouží. Následně lze zjistit, že replikace probíhala rychleji při prohozených rolích zařízení – kdy PC představoval slave server. U Slony-I se nejdříve změny provedou na master server a až pak se změna replikuje na další server. Je to dáno mechanismem asynchronní replikace. Závěrem tedy záleží na slave serveru a jeho schopnosti změny k sobě lokálně zapisovat. Dále lze z grafu vyčíst, že při přímém napojení obou zařízení mezi sebou byl čas kratší. Není zde vliv routeru, proto jsou dosaženy maximální přenosové rychlosti, kterých jsou zařízení schopna. Zatímco u Slony-I nejdéle trval přenos při omezené rychlosti, u MySQL byl největším faktorem výkon zařízení. Nejdelší čas byl u prohozených rolích. Může být důvodem, že změna v datech se provádí na master serveru, ale zároveň se v ten samý okamžik již odesílá na slave server. Záleží tedy na výkonu zařízení, které má roli master serveru. PC v roli master serveru mělo viditelně kratší čas replikace. Další podmínky nehrály téměř žádnou roli. Důvodem je základní přenosová rychlost u MySQL, která nepřesáhla 10 Mbps. Proto omezení, ani typ připojení nijak neovlivnil měřený čas.




Graf 12: Průměrné časy replikace změn za testovaných podmínek

Tab. 3 zobrazuje statistickou analýzu vytvořenou nad naměřenými časy během replikace změny geometrie ve vrstvě *BPEJ*. Medián, 1. a 3. kvartil, maximum a minimum lze vyčíst z boxplotů, proto jsou v tabulce vynechány. Pro vypočítání popisných statistik bylo použito MS Excel. Rozptyl (někdy také označením variance) nám říká, jak moc jsou hodnoty rozptýleny. Intervalové rozpětí je rozdílem maximální a minimální hodnoty. Směrodatná odchylka nám říká, jak moc se od sebe hodnoty liší. Je tedy logické, že čím nižší hodnota, tím více si jsou jednotlivé hodnoty bližší a naopak.

Ze všech analýz je patrné, že vyšší hodnoty jsou u MySQL. Je to způsobeno dobou replikace, která je oproti Slony-I velmi dlouhá. U kolísavých hodnot doby je rozptyl naměřených hodnot vyšší, než je tomu u Slony-I, u kterého se ve většině případů hodnoty lišily řádově o jednu s. U Slony-I je výjimkou měření při PC jako master server. Hodnoty zde více kolísaly než u ostatních podmínek. Může to být dáno jistými procesy běžícími na pozadí, které měření ovlivnily. U obou systémů byla naměřena nejnižší směrodatná odchylka během přímého napojení. Důvodem může být absence právě routeru, který přenosovou rychlost omezuje v rámci pravidel QoS (*Quality of Service*). Avšak bez jeho vlivu byla přenosová rychlost mnohem stálejší a následně i doba replikace se snížila.

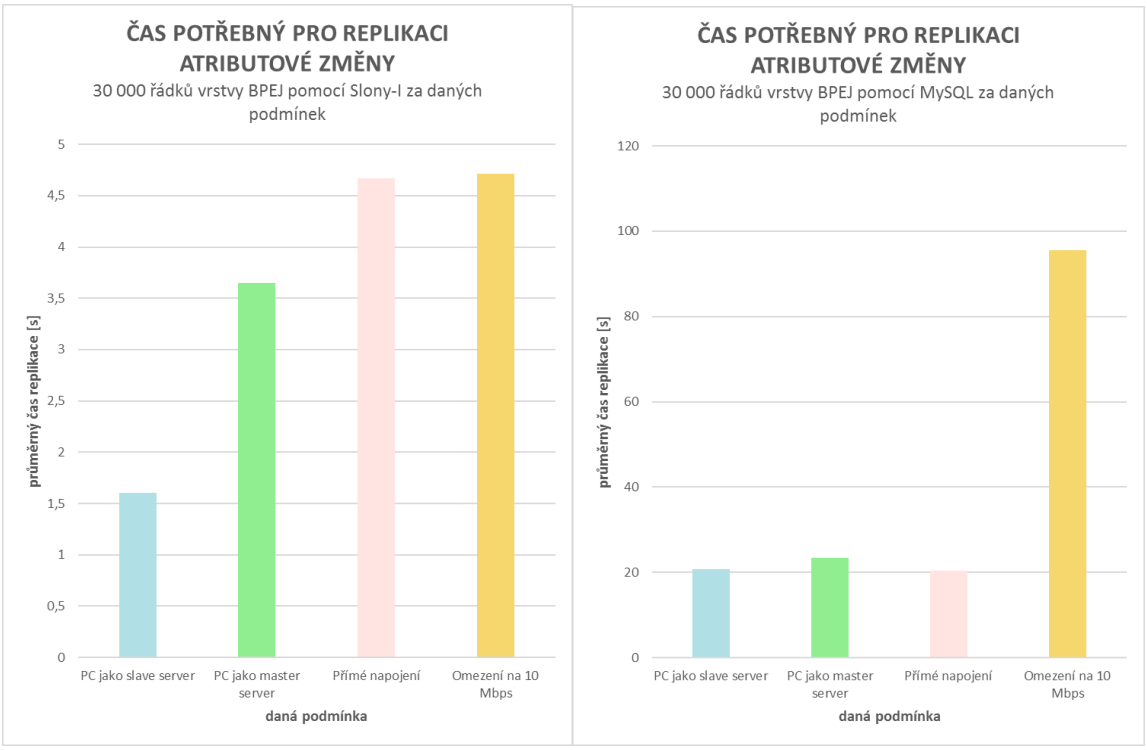
Tab. 3: Statistická analýza naměřených hodnot pro oba replikační mechanismy [s]

| databázový systém | Slony-I (PostgreSQL) | | | | MySQL | | | |
|---|---|---|---|---|---|---|---|---|
| podmínka | PC jako slave server | PC jako master server | Přímé napojení | Omezení 10 Mbps | PC jako slave server | PC jako master server | Přímé napojení | Omezení 10 Mbps |
| průměr | 11,40 | 23,00 | 14,63 | 99,14 | 2656,50 | 1633,30 | 1643,70 | 1628,10 |
| rozptyl | 0,05 | 7,39 | 0,07 | 0,07 | 6668,25 | 926,01 | 370,41 | 2451,89 |
| intervalové rozpětí | 0,82 | 8,91 | 0,80 | 0,89 | 303,00 | 115,00 | 73,00 | 191,00 |
| směrodatná odchylka | 0,22 | 2,72 | 0,26 | 0,26 | 81,66 | 30,43 | 19,25 | 49,52 |

## 5.4 Atributová změna

Pro atributovou změnu v negeometrických datech byly nastaveny stejné podmínky jako při změně geometrie v kapitolách výše. Celkový čas je nižší, než tomu bylo u změny geometrie. Musí se brát v potaz, že při změně geometrie každý záznam obsahoval souřadnice více lomových bodů, které se musely přepsat. S tím souvisí i samotná velikost atributu, která je větší u geometrického atributu. Zatímco u změny negeometrického atributu platí, co záznam, to změna pouze jedné hodnoty. Sledovat využití procesoru nebo přenosovou rychlost nebylo vhodné z důvodu rychlosti replikace.

Graf 13 zobrazuje průměrné časy potřebné pro replikaci atributových změn pomocí obou replikačních mechanismů. Opět se u replikace pomocí Slony-I potvrzuje, že nejrychlejší čas je u podmínky PC jako slave server. Důvod je stejný – záleží na výkonu slave serveru, jak rychle si k sobě ukládá změny. Tentokrát omezení na 10 Mbps mělo minimální vliv z důvodu rychlosti přenosu. Zajímavým zjištěním je fakt, že přímé napojení zařízení trvalo stejnou dobu jako při omezené rychlosti. Zatímco omezení přenosové rychlosti při změně geometrie v rámci MySQL nemělo téměř žádný vliv, u atributové změny je omezení velmi patrné. Při omezení na 10 Mbps se celkový čas prodloužil až čtyřikrát. Ostatní podmínky měly minimální vliv. Dokonce již nebyl tak markantní rozdíl mezi časy při PC jako slave server. Důvodem může být, že negeometrická změna není tolik náročná, jako je změna geometrie v důsledku velikosti atributu.




Graf 13: Průměrné časy atributové změny za daných podmínek

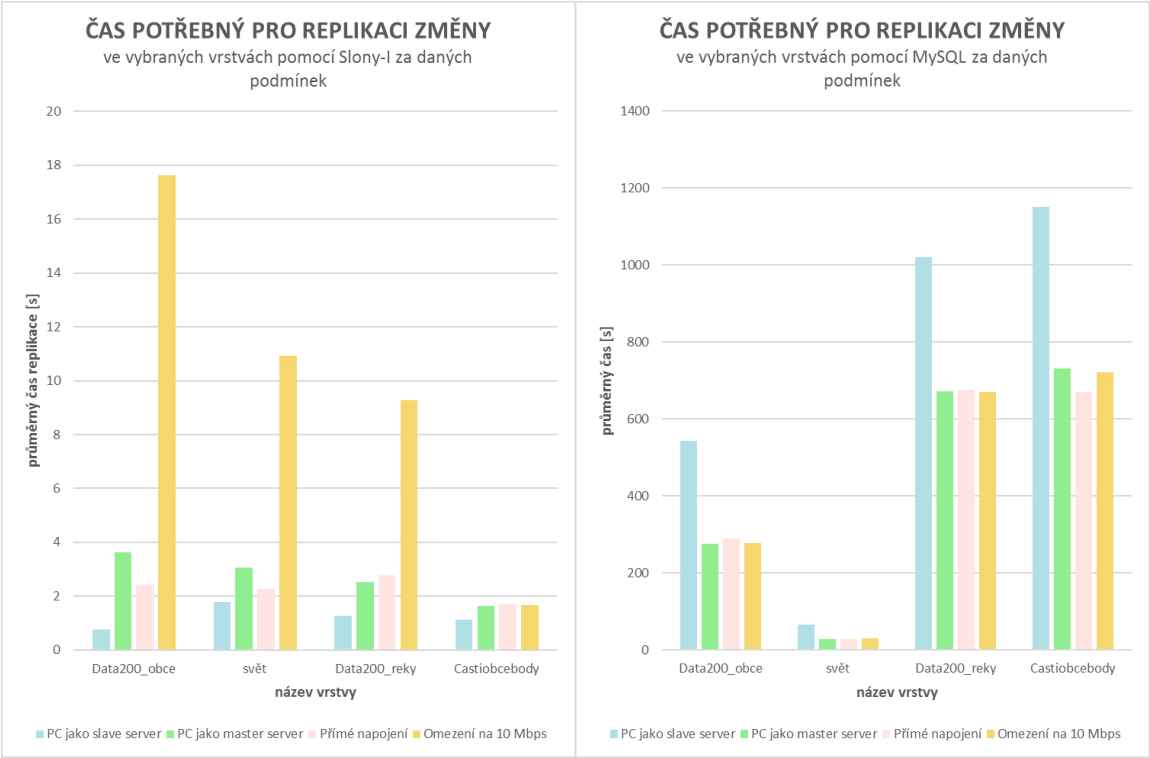
## 5.5 Formát a množství dat

Pro další srovnání byly použity ostatní vrstvy kromě BPEJ. V Tab. 4 jsou pro zopakování údaje o množství lomových bodů a množství prvků vybraných vrstev. Množství lomových bodů se při testování ukázaly jako podstatné proměnné. Množství lomových bodů a počet záznamů měly vliv na čas replikace.

Tab. 4: Údaje o testovaných vrstvách

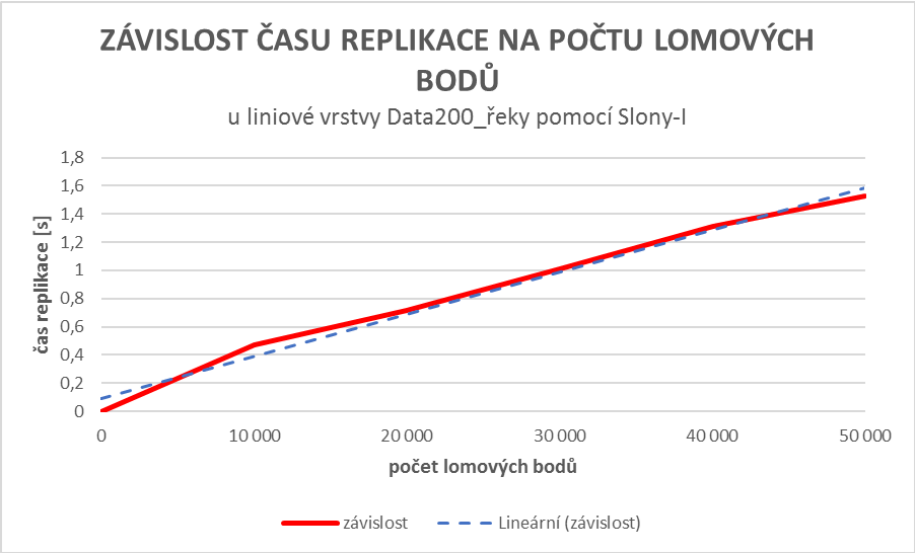
| název vrstvy | množství lomových bodů | počet záznamů |
|---|---|---|
| Data200_obce | 672 299 | 6 353 |
| svět | 411 132 | 1 |
| Data200_reky | 338 959 | 14 606 |
| Castiobcebody | 15 092 | 15 092 |

Graf 14 porovnává průměrné časy jednotlivých vrstev za stejných podmínek, které byly použity pro vrstvu *BPEJ*. Vlevo je graf replikace pomocí Slony-I. Při podmínce s omezenou přenosovou rychlostí je krásně vidět, že doba replikace byla nejdelší u vrstvy s největším počtem lomových bodů – polygonová vrstva *Data200_obce*. Naopak nejkratší doby bylo dosaženo u bodové vrstvy *Castiobcebody*, která má nejméně bodů ze všech vrstev – jedná se pouze o body. Časy u ostatních podmínek nepřekročí 4 s. U MySQL nejdéle trvala bodová vrstva *Castiobcebody*, razantně nejkratší čas byl naměřen u polygonové vrstvy *svět*.




Graf 14: Průměrné časy replikace vybraných vrstev za daných podmínek

U vrstvy nezáleží prakticky na jejím formátu, zda se jedná o bodovou, liniovou či polygonovou. Nezáleží ani na množství prvků, kterými je tvořena. Záleží pouze na **počtu lomových bodů**, kterými jsou dané prvky tvořeny. Dokonce na základě měření byla zjištěna lineární závislost mezi časem a počtem lomových bodů (viz Graf 15). Čím více změn v lomových bodech se najednou provede, tím déle bude replikace trvat. Naopak u MySQL záleží na **počtu záznamů** dané vrstvy.



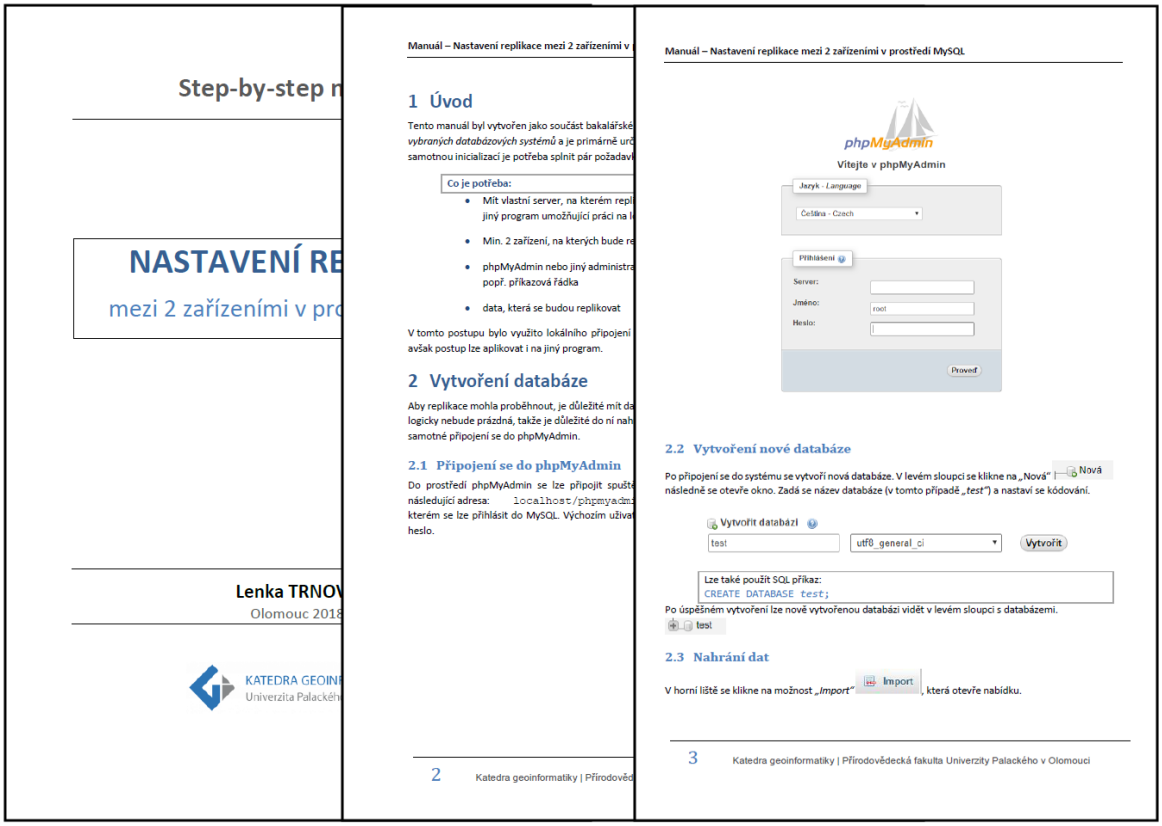

Graf 15: Závislost času replikace na počtu lomových bodů u liniové vrstvy Data200 řeky

# 6 PŘÍPADOVÁ CVIČENÍ

Součástí práce bylo také vytvoření step-by-step návodů pro studenty popisující postup replikace a sepsání praktické použitelnosti daných replikačních mechanismů. Použitelnost byla stanovena na základě vlastního testování a sledovaných kritérií.

## 6.1 Step-by-step návody

V rámci bakalářské práce byly vytvořeny podrobné step-by-step návody od importu dat do databáze až ke zdárnému spuštění replikace. Návody jsou obohaceny o množství snímků obrazovky, které se snaží celý postup více zjednodušit. Inicializace replikace je provedena na konkrétních datech (bodová vrstva *Castiobcebody*), které bude možno s návody stáhnout společně s ostatními testovanými daty v této práci. Oba dokumenty, rozdělené na databázové systémy, jsou v elektronické příloze bakalářské práce. Dokumenty PDF jsou volně ke stažení z webových stránek, které byly vytvořeny společně s bakalářskou prací. Webové stránky jsou umístěny na stránkách katedry[2]. Na webových stránkách je možné si stáhnout vzorové textové soubory pro nastavení master a slave serveru v prostředí PostgreSQL. Na Obr. 5 je možné vidět ukázku vzhledu výsledných step-by-step návodu, konkrétně MySQL replikace.



Obr. 5: Ukázka step-by-step návodu

[2] http://www.geoinformatics.upol.cz/dprace/bakalarske/trnova18/

## 6.2 Praktické využití

Na základě testování byly zjištěny výhody a nevýhody obou replikačních mechanismů. Je potřeba mít na paměti zásadní rozdíl mezi nimi. Slony-I je asynchronní replikace, zatímco MySQL je streaming. Už na základě tohoto rozdílu je možný rozdíl v jejich využití. Záleží, zda je potřeba replikovat všechna data v rámci celé databáze (jak umožňuje streaming replikace) nebo pouze vybraná data (asynchronní replikace).

Praktickým využitím replikace pomocí Slony-I může být situace, kdy server přijímá data ze senzoru. Senzorem může být měřič kvality ovzduší, který v předem nadefinovaných hodinách změří dané proměnné (např. teplotu, tlak a vlhkost vzduchu). Po změření senzor hodnoty odešle na server, který je označen jako master. Na master serveru se ukládají všechna primární data ze senzoru. Jsou zde v neupravené podobě, nepracuje se s nimi. Pro klienty bude zřízen slave server, na který se budou replikovat data. Klient zvolí, jaká data potřebuje, bude se přímo dotazovat na slave server pomocí SQL dotazů nebo se umožní vygenerování exportu dat. Klient získá potřebná data, se kterými může pracovat offline. Tím, že se na data na master serveru fyzicky nesahá, se zajistí jejich správnost a konzistence. V případě, že klient chybně upraví data, se nic na straně master serveru nestane, chyba zůstane pouze u klienta. Pokud se k datům dotazuje velmi mnoho klientů, je vhodným řešením rozložení zátěže. To se zajistí vytvořením kopie master serveru.

Praktickým využitím MySQL streaming replikace by mohla být georeplikace, kdy je potřeba replikovat všechna data na velké vzdálenosti. Slave server by obsahoval všechna data z databáze master serveru. Bylo by vhodné, aby se nejednalo o data, která je nutná velmi často aktualizovat. Vzhledem ke vzdálenostem přenosu by docházelo ke zpomalování celého systému, kdyby se data neustále měnila. Bez častých aktualizacích by slave server sloužil jako lokální kopie master serveru. Zajistilo by se rychlé dotazování klientů, kteří by se připojovali právě k slave serveru, který je lokalizován v jejich blízkosti. Není totiž výhodné, aby se data získávala vždy z master serveru, který může být vzdálený i tisíce km.

# 7 VÝSLEDKY

Výsledky představují shrnutí testování databází, které je popsáno v kapitole 4.3. K cílovému zhodnocení byla zahrnuta kritéria, která byla navržena na začátku práce. Zhodnocení je rozděleno na oba replikační mechanismy, které se mezi sebou liší.

## 7.1 Zhodnocení použitelnosti Slony-I

Z výsledků je patrné, že replikace změn pomocí extenze Slony-I je velmi rychlá i v případě, že se najednou replikuje velký objem změn. Ovšem i díky testování lze konstatovat, že rychlost závisí na přenosové rychlosti. Dá se konstatovat, že u tohoto způsobu replikace je možné provádět synchronizace mezi servery méně často. Interval se odvíjí od počtu změn, které se v datech provádí. Pokud se jedná o data, která se mění zřídka a jejich aktuálnost není podmínkou, lze synchronizovat po delší době. Aby ale byla replikační rychlost únosná, je potřeba promyslet, kolik dotazujících serverů může v jeden moment být. Byla zjištěna lineární závislost replikačního času na počtu změněných lomových bodů. Aby replikační čas nepřekročil 1 s, lze relativně provést řádově desetitisíce změn v lomových bodech prostorových dat. Konkrétní hodnotu nelze určit, hlavním faktorem je, kolik lomových bodů je obsaženo v každém záznamu. Rychleji se replikuje velká změna v jednom záznamu než několik změn ve velkém množství záznamů. Není reálné, aby se najednou provedly změny v tisíce lomových bodech, většina uživatelů provede pár změn a následně změny uloží. Dále je nutné do replikační doby zahrnout počet uživatelů. Je logické, že při větším množství dotazování se hodnota replikovaných změn razantně sníží.

Co se využití procesoru týče, je vhodné, aby master server představoval zařízení, které bude sloužit pouze jako databázový server. Již z předchozích kapitol bylo možné z grafů odečíst, že využití procesoru může při náročných operacích vystoupat až na třetinu celého možného zatížení. Pokud by tedy na pozadí běžely náročné procesy, mohly by způsobit přetížení procesoru a celkové zpomalení přenosu.

Vzhledem k výsledkům při prohození rolí (PC jako slave server) došlo k zjištění, že rychlost přenosu je značně rychlejší. Důvodem je výkonnější slave server, který zvládá mnohonásobně rychleji zapisovat změny ve vrstvě. Dá se říci, že při replikaci pomocí Slony-I nemusí být master server výkonným zařízením. Záleží pouze na slave serveru, jak rychle je schopen zapisovat změny k sobě na disk.

Při omezení rychlosti se celková doba replikace prodloužila. Lze vyvodit další doporučení, kterým je připojení. Při vyšších přenosových rychlostech se změna bude replikovat rychleji, při nižších rychlostech tomu bude naopak.

Při použití Slony-I hraje důležitou roli typ připojení. Záleží na šířce a propustnosti pásma zvoleného připojení. Server běžně mívá 1 Gbps a router bývá alespoň 10 Gbps, aby se do sítě dalo připojit více serverů. Takovéto využití replikace slouží spíše pro vytvoření kopie hlavního serveru. Je jasné, že při přímém připojení bude přenos replikovaných změn nejrychlejší. Ve většině případech se replikace používá pro replikování změn na konkrétní vzdálenost, je tedy potřeba započítat vliv routeru, vždy je přenosová rychlost snížena.

Podle použití je v neposlední řadě výhodou, ale zároveň i nevýhodou, že pomocí Slony-I se replikují pouze vybrané tabulky. V případě, že se klient ve formě webového serveru dotazuje k slave serveru, má možnost zvolit, jaké tabulky, resp. data chce k sobě zreplikovat. Konkrétní použití bylo uvedeno v kapitole 6.2.

## 7.2 Zhodnocení použitelnosti MySQL

U MySQL velký objem změn najednou, především v geometrii prostorových dat, je příliš náročný. Je náročný z pohledu celkového trvání operace, která může trvat i desítky min. Doporučuje se replikovat častěji po malých částech. Zaprvé se častější replikací zajistí aktuálnost dat, ale také se razantně sníží celkový čas potřebný pro replikaci změn.

Během měření bylo zjištěno, že aby replikace běžela 1 s, musely by se provést změny maximálně u desítky záznamů. Očekává se, že tolik změn se najednou neprovede, avšak je nutné vědět, kolik změn lze maximálně provést, aby doba replikace byla stále přijatelná. Opět se musí počítat s počtem dotazujících serverů, díky kterým se počet změn provedených najednou razantně sníží.

U MySQL replikace průměrné zatížení procesoru bez větších výkyvů nepřekročilo hodnotu 10 % celkového zatížení. Je tedy patrný rozdíl oproti Slony-I. V tomto případě lze říct, že teoreticky na pozadí mohou běže další procesy, které jistým způsobem zvyšují zátěž procesoru. Master server tedy nemusí být zařízením, na kterém běží pouze replikace. Lze teoreticky s ním dále pracovat, aniž by byl ovlivněn čas replikace.

Na základě prohození rolí master a slave serveru je doporučeno, aby master server byl tvořen výkonným zařízením. Od jeho výkonu se odvíjí celkový čas potřebný pro replikování změn. Čím výkonnější master server je, tím rychleji je schopen změny ukládat k sobě a zároveň rozesílat na další servery. V testování při prohozených rolích čas narostl o více jak 40 % oproti výchozímu nastavení rolí.

K replikaci i velkého objemu dat najednou není potřeba silného připojení k internetu. I při omezení přenosové rychlosti replikování změny trvalo prakticky stejnou dobu. Při replikaci atributových změn, které jsou časově méně náročné, se čas odvíjí od připojení k internetu. Během testování přenosová rychlost nepřesáhla 10 Mbps. Lze tedy říci, že stačí MySQL omezit šířku pásma na tuto hodnotu a zbytek pásma přenechat dalším procesům.

MySQL je streaming replikace, to znamená, že slave server je věrnou kopií master serveru. Replikace probíhá 1:1, takže celá databáze se všemi tabulkami se vždy kopíruje na slave server. Záleží opět jako u Slony-I na použití. Konkrétní použití bylo uvedeno v kapitole 6.2.

# 8 DISKUZE

Cílem práce bylo zhodnocení replikačních mechanismů. Celé testování probíhalo na lokální síti a na dvou konkrétních zařízeních. Je nutné brát v potaz, že jiných výsledků bude dosaženo na jiném zařízení. Výsledky by se měly lišit hodnotami časů, avšak nemělo by se lišit chování replikace. Během testování byla snaha, aby v pozadí neběžely žádné procesy, které by měření ovlivnily. Ani grafy tuto možnost nepotvrdily. Není však možné s jistotou říci, že takový proces na pozadí neprobíhal.

Během práce se naskytlo několik problémů, které znepříjemnily celý postup. Místy i způsobily velké zpoždění v práci. Nejhorším problémem byly nucené aktualizace systému na straně NB, které vždy způsobily absolutní nefunkčnost napojení replikace v prostředí PostgreSQL. Velkou výhodou bylo, že většina chyb je již řešena na webových diskuzích, bylo tedy snadnější se dopátrat chyby. Někdy ale jediným řešením bylo přeinstalování celé služby PostgreSQL. Prostředí MySQL bylo mnohem přívětivější, co se řešení problémů týče. Navíc se žádné větší neobjevily. V prostředí phpMyAdmin je nejdůležitějším krokem správně zvolit možnost, že se nemají replikovat databáze s výjimkou konkrétní databáze. Výchozí hodnotou je totiž opačná možnost, kdy se replikují všechny databáze s výjimkou konkrétní zvolené. Proto docházelo lehce k nefunkčnosti replikace.

Součástí práce bylo vytvoření návodů jak replikaci v daném replikačním mechanismu inicializovat. Očekává se, že takto vytvořené návody budou sloužit studentům KGI. Byla snaha udělat návody co nejpodrobnější, aby bylo od začátku až do konce jasné, jak se má postupovat. Jedinou dá se říct nevýhodou je, že v návodu byly použity konkrétní programy a je přizpůsoben platformě MS Windows. Musí se počítat s tím, že budou jisté odlišnosti pro Linux.

Představa využití práce je při volbě replikačního mechanismu na základě informací sepsaných v této práci. V práci bylo testování změn zaměřeno na extrémní případy. Není reálné, aby se najednou provedlo např. 400 000 změn v lomových bodech. Avšak na základě těchto extrémních případů lze vyvodit závěry, jak se daný replikační mechanismus chová. Kdyby byly měřeny reálné změny (např. změna jednoho lomového bodu, jednoho záznamu), měření by nebylo vypovídající. Celkový přenos by trval řádově jednotky ms až jednu s. Součástí práce jsou také jistá doporučení, jak z daného mechanismu „dostat co nejvíce“, resp. aby replikovaný čas byl co nejkratší a práce s tímto systémem byla co nejspolehlivější.

Testování se mohlo pojmout jiným způsobem, např. využitím nástrojů určených pro testování zátěže (např. *pgbench*). Nelze opomenout, že nástroje pracují s fiktivními daty, která jsou náhodně generována na základě nastavení zátěže. Tyto nástroje nepracují se skutečnými daty.

V návaznosti na tuto práci by bylo možné otestovat další replikační mechanismy, které oba databázové systémy nabízejí. V rámci bakalářské práce nebylo možné, aby bylo otestováno více replikačních mechanismů. Ať už z důvodu složitosti nastavení systémů při různých podmínkách, tak z pohledu komplexnosti práce. Pro testování byly vybrány dva mechanismy, které se mezi sebou liší, a právě díky této odlišnosti lze vyvodit jiná využití.

# 9 ZÁVĚR

Hlavním cílem bakalářské práce bylo zhodnocení replikací v databázových systémech PostgreSQL a MySQL. Prvním krokem bylo studium literatury a dostupných dokumentací týkající se daných databázích. V rešerši byly krátce vysvětleny základní pojmy potřebné k další práci. Krátce byla zmíněna úzká hrdla systému, která mohou ovlivnit chod samotné replikace. V neposlední řadě se v rešerši zmínilo dosavadní hodnocení databázových systémů, ze kterých bylo částečně možné vytvořit vlastní kritéria. Po navržení kritérií následovala praktická část. Začalo také primární seznámení se s nástroji pro monitorování vybraných kritérií a programy pro práci s databázemi.

Na základě rešerše a diplomové práce Mgr. Markéty Solanské byly inicializovány dva vybrané replikační mechanismy. Asynchronní logická replikace pomocí Slony-I a streaming logická replikace v MySQL. Testování probíhalo na dvou zařízeních – PC a NB, které pracovaly v rámci lokální sítě. Testovaná data byla vybrána z BPEJ, ArcČR 500, Data200 a NaturalEarth. Data se lišila strukturou a objemem, nejdůležitějšími proměnnými byl počet lomových bodů a počet záznamů.

Na datech byly provedeny změny geometrie tak, že se celá vrstva posunula v prostředí QGIS. Ihned se ukázalo, která data dělají největší potíže, co se zpracování požadavku týče. Vrstva *BPEJ* byla ihned vyhodnocena jako nejnáročnější, proto vstupovala jako hlavní prvek pro srovnání obou replikačních mechanismů. Během testování se sledovala kritéria a vždy bylo provedeno 10 měření, aby bylo možné vytvoření statistické analýzy. Součástí testování bylo také vytvoření různých podmínek, díky kterým byla zjištěna další východiska. Např. díky prohozeným rolím zařízení, kdy PC představovalo slave server, bylo zjištěno následující. Zatímco v prostředí extenze Slony-I byla replikace rychlejší při prohozených rolích (výkonnější PC je schopno rychleji k sobě změny zapisovat), u MySQL čas narostl. Druhou operací pro replikace byla změna čistě číselného atributu u nejnáročnější vrstvy – *BPEJ*, kdy se najednou změnila hodnota pro 30 000 řádků.

Po dokončení testování byla sesbírána všechna naměřená data a byla z nich vytvořena série grafů, boxplotů a tabulek. Díky nim bylo možné oba replikační mechanismy porovnat jak mezi sebou, tak v rámci různých podmínek. Nejdůležitější rozdíly mezi oběma systémy je způsob inicializace replikace. U Slony-I se změna prvotně zapíše na master serveru a až následně se posílají změny v datech na slave server. U MySQL je opačný přístup. V momentě, kdy se provádí změna na master serveru, se již provádí na slave serveru. Oba systémy mají své výhody i nevýhody, detailněji byly popsány v kapitole 7.

Součástí práce bylo vytvoření step-by-step návodů pro studenty. Oba dokumenty obsahují detailní popis jak replikaci díky danému replikačnímu mechanismu spustit. Návod je obohacen o sérii snímků, aby bylo pochopení postupu co nejjednodušší.

# POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE


ABDUL MOIZ, Salman, Sailaja P., Venkataswamy G. a Supriya N. PAL. Database Replication: A Survey of Open Source and Commercial Tools. International Journal of Computer Applications. 2011, 13(6), 1-8. DOI: 10.5120/1788 2469. ISSN 09758887.

ABRAMOVA, Veronika, Jorge BERNARDINO a Pedro FURTADO. Which NoSQL Database? A Performance Overview. 2014

BELL, Charles, Mats KINDAHL a AND LARS THALMANN. MySQL High Availability. Sebastopol, CA: O'Reilly Media, 2010. ISBN 9780596807306.

BOICEA, Alexandru, Florin RADULESCU a Laura Ioana AGAPIN. MongoDB vs Oracle -- Database Comparison. In: 2012 Third International Conference on Emerging Intelligent Data and Web Technologies. IEEE, 2012, s. 330-335. DOI: 10.1109/EIDWT.2012.32. ISBN 978-1-4673-1986-7.

BÖSZÖRMENYI, Zoltan. PostgreSQL replication: understand basic replication concepts and efficiently replicate PostgreSQL using high-end techniques to protect your data and run your server without interruptions. Birmingham: Packt Publishing, c2013. ISBN 978-1-84951-672-3.

Database replication [online]. Copyright © 1997 Oracle Corporation. [cit. 2017-11-15]. Dostupné z: https://docs.oracle.com/cd/A59447_01/nt_804ee/doc/database.804/a58227/ch_repli.htm

DOLGIKH, Maria. Vysoká dostupnost v relačních databázových systémech. Brno, 2014. Diplomová práce (Mgr.). Masarykova univerzita, Fakulta informatiky.

eKatalog BPEJ [online]. Copyright © VÚMOP v.v.i. [cit. 2017-10-22].
Dostupné z: https://bpej.vumop.cz/

KABAKUS, Abdullah Talha a Resul KARA. A performance evaluation of in-memory databases. Journal of King Saud University – Computer and Information Sciences. 2017, 29(4), 520-525. DOI: 10.1016/j.jksuci.2016.06.007. ISSN 13191578.

MAZILU, Marius Cristian. Database Replication [online], Database Systems Journal, 1, issue 2. 2010 [cit. 2017-10-20].
Dostupné z: http://www.dbjournal.ro/archive/2/4_Cristian_Mazilu.pdf

MySQL: Developer Zone [online]. Copyright © 2017, Oracle Corporation and [cit. 2017-10-22]. Dostupné z: https://dev.mysql.com

MySQL: MySQL 5.7 Reference Manual [online]. Copyright © 2018, Oracle Corporation and [cit. 2017-10-22].
Dostupné z: https://dev.mysql.com/doc/refman/5.7/en/replication-solutions.html

OBE, Regina O. a Leo S. HSU. PostgreSQL: up and running. Second edition. Beijing: O'Reilly, 2015. ISBN 978-144-9326-333.

OBE, Regina O. a Leo. S. HSU. PostGIS in action. London: Pearson Education [distributor], c2011. ISBN 19-351-8226-9.

pgpool Wiki. [online]. [cit. 2017-10-22].
Dostupné z: http://www.pgpool.net/mediawiki/index.php/Main_Page

phpMyAdmin. phpMyAdmin [online]. Copyright © 2003 [cit. 2017-10-22].
Dostupné z: https://www.phpmyadmin.net/

PostGIS — Spatial and Geographic Objects for PostgreSQL [online]. Copyright ©S
[cit. 2017-11-15]. Dostupné z: https://postgis.net/stuff/postgis-2.3.pdf\

Replication — Apache CouchDB 2.1 Documentation. [online]. Copyright © Copyright 2018, Apache Software Foundation. [cit. 2017-12-5].
Dostupné z: http://docs.couchdb.org/en/2.1.1/replication/index.html

Replication, Clustering, and Connection Pooling – PostgreSQL wiki. [online]. Dostupné z: https://wiki.postgresql.org/wiki/Replication,_Clustering,_and_Connection_Pooling

SCHÖNIG, Hans-Jürgen. PostgreSQL Replication: Leverage the power of PostgreSQL replication to make your databases more robust, secure, scalable, and fast. 2. vyd. Packt Publishing, 2015. ISBN 978-1-78355-060-9.

SOLANSKÁ, Markéta. Synchronizace a replikace geodat v prostředí Esri platformy. Olomouc, 2014. diplomová práce (Mgr.). Univerzita Palackého v Olomouci. Přírodovědecká fakulta

Streaming Replication – PostgreSQL wiki. [online]. [cit. 2017-12-5].
Dostupné z: https://wiki.postgresql.org/wiki/Streaming_Replication

SQL Server Replication | Microsoft Docs. [online]. [cit. 2017-12-5].
Dostupné z:
https://docs.microsoft.com/en-us/sql/relational-databases/replication/sql-server-replication

TIWARI, Shashank. Professional NoSQL. Hoboken, N.J: Wiley, 2011. ISBN 9780470942246.

Úzké hrdlo (Bottleneck) - ManagementMania.com. [online]. Copyright © 2011 [2017-10-24]. Dostupné z: https://managementmania.com/cs/uzke-hrdlo

VONDRA, Tomáš. Replikace v PostgreSQL [online]. 2011 [cit. 2017-10-22]. Dostupné z: https://www.cspug.cz/files/cspug-2011-04-19-vondra-replikace.pdf

What is a Database Trigger? - Essential SQL. Essential SQL – Your search for SQL training is over, [online]. [cit. 2017-12-5]. Dostupné z: https://www.essentialsql.com/what-is-a-database-trigger/

# PŘÍLOHY

# SEZNAM PŘÍLOH

**Vázané přílohy:**

Příloha 1 Dodatkové tabulky, boxploty a grafy

Příloha 2 Soubory potřebné pro inicializaci replikace v prostředí Slony-I

**Volné přílohy**

Příloha 3 Poster

Příloha 4 CD

**Popis struktury CD**

- Adresáře:
  - Web
  - Přílohy
    - Poster
    - Step-by-step návody
      - MySQL
      - Slony-I
  - Data
    - Vstupni_data
    - Vystupni_data
- Text práce

Tab. 5: Naměřené hodnoty při změně geometrie u daných vrstev – PC jako master server

| databázový systém | název vrstvy | čas replikace [s] | využití procesoru (master) [%] | využití procesoru (slave) [%] | posílání dat (master) [Mbps] | přijímání dat (slave) [Mbps] |
|---|---|---|---|---|---|---|
| Slony-I (PostgreSQL) | BPEJ | 23,00 | 27,85 | 27,16 | 34,76 | 38,41 |
| | Data200_obce | 3,62 | 23,15 | 23,54 | 68,95 | 79,60 |
| | svět | 3,06 | 20,11 | 13,23 | 58,80 | 49,13 |
| | Data200_reky | 2,50 | 25,61 | 22,28 | 50,00 | 75,05 |
| | Castiobcebody | 1,65 | 22,70 | 16,54 | 17,60 | 15,90 |
| Slony-I při omezení na 10 Mbps | BPEJ | 99,14 | 33,37 | 1,81 | 10,25 | 9,74 |
| | Data200_obce | 17,62 | 25,36 | 2,56 | 10,50 | 10,76 |
| | svět | 10,93 | 21,53 | 1,81 | 10,58 | 9,87 |
| | Data200_reky | 9,29 | 33,34 | 2,59 | 10,65 | 10,36 |
| | Castiobcebody | 1,67 | 33,65 | 1,50 | 26,10 | 15,90 |
| MySQL | BPEJ | 1633,30 | 1,41 | 4,29 | 0,91 | 0,91 |
| | Data200_obce | 276,20 | 1,25 | 3,89 | 0,71 | 0,72 |
| | svět | 29,50 | 26,38 | 1,60 | 3,01 | 3,88 |
| | Data200_reky | 671,20 | 1,08 | 2,95 | 0,28 | 0,28 |
| | Castiobcebody | 731,10 | 1,36 | 4,47 | 0,22 | 0,23 |
| MySQL při omezení na 10 Mbps | BPEJ | 1628,10 | 1,67 | 4,85 | 0,83 | 0,85 |
| | Data200_obce | 278,20 | 1,68 | 3,67 | 0,70 | 0,74 |
| | svět | 30,20 | 27,30 | 1,36 | 3,31 | 2,44 |
| | Data200_reky | 670,80 | 1,38 | 3,01 | 0,35 | 0,33 |
| | Castiobcebody | 722,40 | 1,30 | 4,69 | 0,20 | 0,25 |

Tab. 6: Naměřené hodnoty při atributové změně 30 000 řádků vrstvy BPEJ – PC jako master server

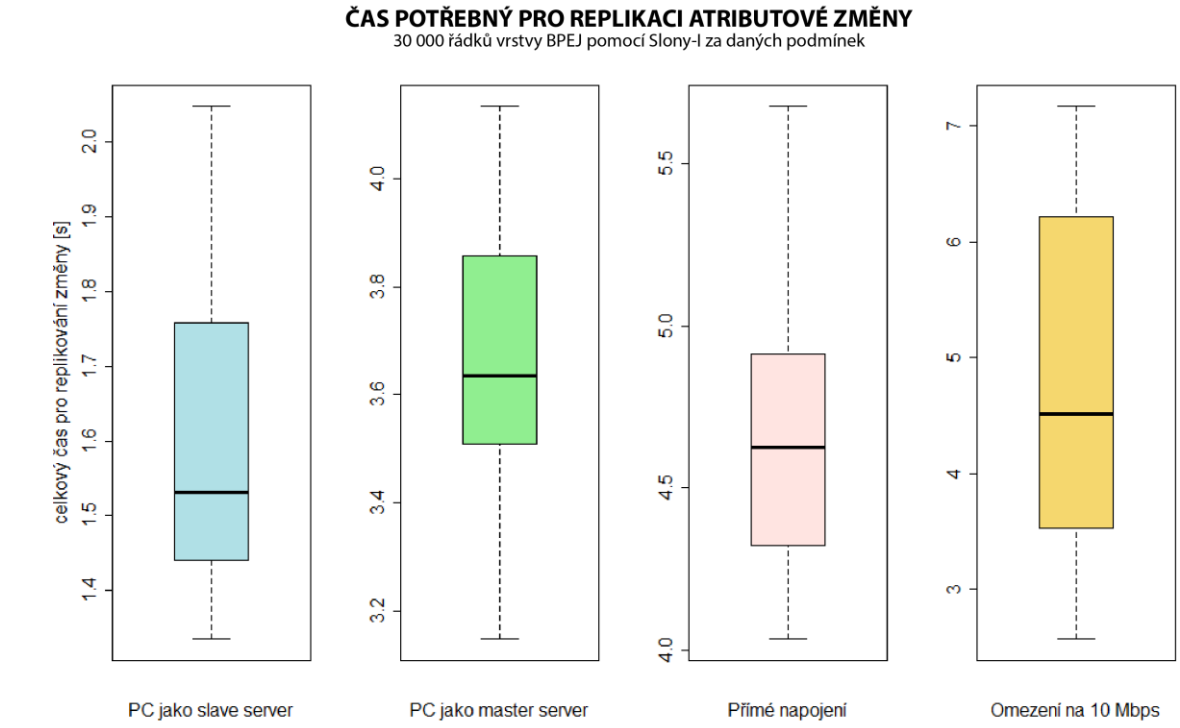
| databázový systém | čas replikace [s] | využití procesoru (master) [%] | využití procesoru (slave) [%] | posílání dat (master) [Mbps] | přijímání dat (slave) [Mbps] |
|---|---|---|---|---|---|
| Slony-I (PostgreSQL) | 3,65 | 1,55 | 8,10 | 27,90 | 28,20 |
| Slony-I při omezení na 10 Mbps | 4,71 | 1,97 | 6,50 | 9,97 | 16,90 |
| MySQL | 23,40 | 4,50 | 14,25 | 65,27 | 77,03 |
| MySQL při omezení na 10 Mbps | 95,60 | 2,79 | 2,46 | 10,38 | 10,48 |

Tab. 7: Naměřené hodnoty při změně geometrie u daných vrstev – PC jako slave server

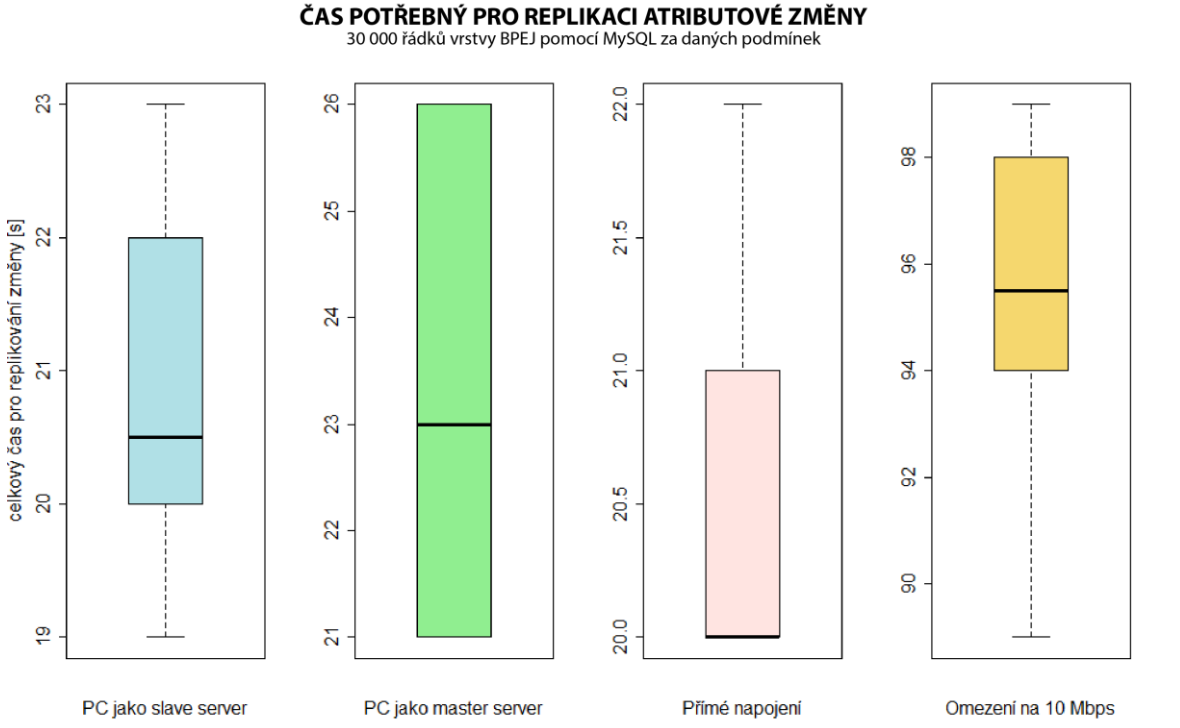
| databázový systém | název vrstvy | čas replikace [s] | využití procesoru (master) [%] | využití procesoru (slave) [%] | posílání dat (master) [Mbps] | přijímání dat (slave) [Mbps] |
|---|---|---|---|---|---|---|
| Slony-I (PostgreSQL) | BPEJ | 11,40 | 9,85 | 10,88 | 80,43 | 88,51 |
| | Data200_obce | 1,93 | 13,83 | 9,56 | 71,38 | 71,35 |
| | svět | 1,75 | 4,50 | 6,63 | 56,47 | 57,50 |
| | Data200_reky | 1,22 | 20,06 | 12,80 | 47,77 | 55,10 |
| | Castiobcebody | 0,75 | 5,80 | 7,95 | 9,65 | 9,25 |
| Slony-I při omezení Na 10 Mbps | BPEJ | 97,56 | 3,02 | 2,41 | 10,41 | 10,40 |
| | Data200_obce | 16,80 | 2,38 | 2,70 | 10,34 | 10,50 |
| | svět | 10,47 | 1,87 | 1,66 | 10,31 | 10,70 |
| | Data200_reky | 9,14 | 2,27 | 3,94 | 10,83 | 10,80 |
| | Castiobcebody | 1,90 | 1,20 | 6,80 | 21,50 | 17,40 |
| MySQL | BPEJ | 2627,80 | 2,65 | 0,69 | 0,50 | 0,48 |
| | Data200_obce | 543,20 | 2,56 | 1,33 | 0,40 | 0,78 |
| | svět | 65,50 | 38,09 | 2,60 | 4,50 | 4,10 |
| | Data200_reky | 1021,30 | 2,35 | 1,05 | 0,22 | 0,22 |
| | Castiobcebody | 1151,60 | 2,41 | 0,77 | 0,15 | 0,15 |
| MySQL při omezení Na 10 Mbps | BPEJ | 2666,50 | 2,78 | 0,64 | 0,51 | 0,49 |
| | Data200_obce | 523,80 | 2,74 | 1,20 | 0,45 | 0,45 |
| | svět | 65,80 | 32,48 | 2,30 | 4,20 | 3,80 |
| | Data200_reky | 1027,20 | 2,64 | 3,04 | 0,23 | 0,21 |
| | Castiobcebody | 1153,00 | 3,27 | 1,29 | 0,14 | 0,14 |

Tab. 8: Naměřené hodnoty při atributové změně 30 000 řádků vrstvy BPEJ – PC jako slave server

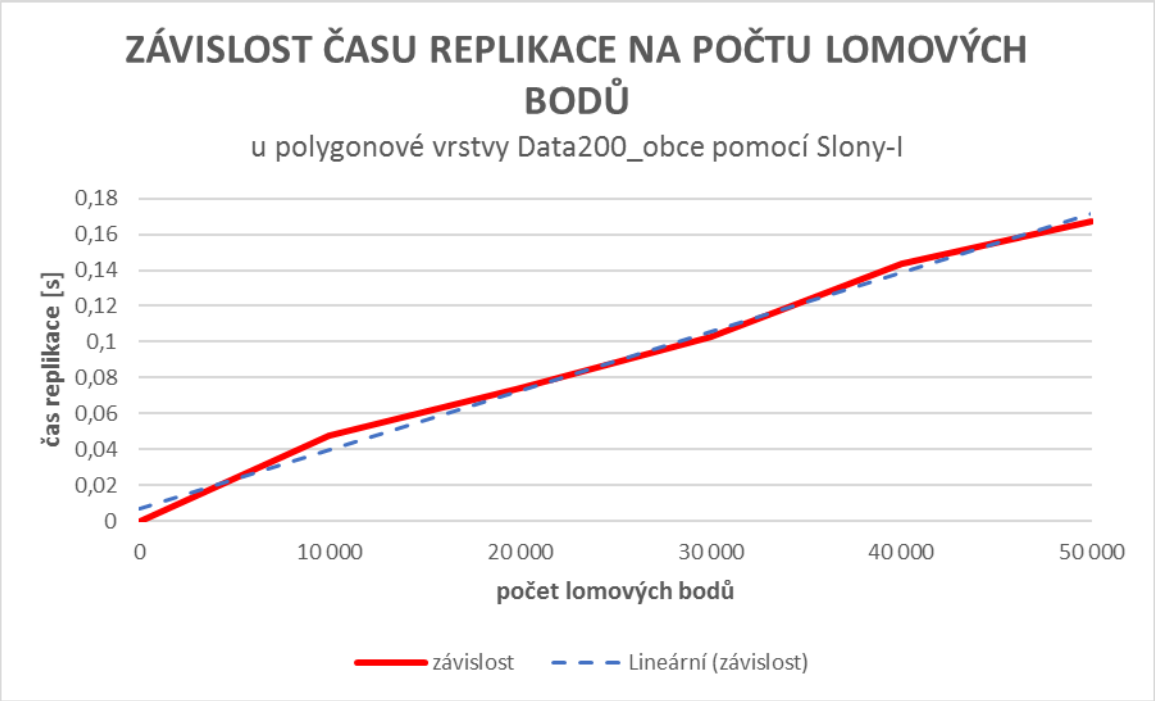
| databázový systém | čas replikace [s] | využití procesoru (master) [%] | využití procesoru (slave) [%] | posílání dat (master) [Mbps] | přijímání dat (slave) [Mbps] |
|---|---|---|---|---|---|
| Slony-I (PostgreSQL) | 1,6 | 5,82 | 0,4 | 23 | 18,8 |
| Slony-I při omezení na 10 Mbps | 2,67 | 6,68 | 1,2 | 10,4 | 10,5 |
| MySQL | 20,8 | 5,81 | 8,7 | 83,48 | 85,61 |
| MySQL při omezení na 10 Mbps | 96,1 | 1,81 | 2,59 | 10,35 | 10,35 |




Graf 16: Boxploty časů potřebných pro replikaci atributové změny 30 000 řádků vrstvy BPEJ za daných podmínek pomocí Slony-I




Graf 17: Boxploty časů potřebných pro replikaci atributové změny 30 000 řádků vrstvy BPEJ za daných podmínek pomocí MySQL




Graf 18: Závislost času replikace na počtu lomových bodů u polygonové vrstvy Data200_obce

*Příloha 2: Soubory potřebné pro inicializaci replikace v prostředí Slony-I*

**Soubor master.txt.**

```
# nazev clusteru
cluster name = replika_test;
# definice jednotlivych uzlu v clusteru:
# 1) informace o master databazi
node 1 admin conninfo = 'dbname = master host = 10.0.0.35/32 port = 5432 user = postgres password = postgres';
# 2) informace o slave databazi
node 2 admin conninfo = 'dbname = slave1 host = 10.0.0.36/32 port = 5432 user = postgres password = postgres';

# inicializace clusteru
init cluster (id = 1, comment = 'master');
store node (id = 2, comment = 'slave', event node = 1);

# vytvoreni replikacni sady
create set (id = 1, origin = 1, comment = 'Tabulky k replikaci');
# pridani tabulky, ktera nese nazev nasi vrstvy
set add table (set id = 1, origin = 1, id = 1, fully qualified name = 'public.castiobcebody', comment = 'Castiobcebody');

# cesta k ulozisti
store path (server = 2, client = 1, conninfo = 'dbname = slave host = 10.0.0.36/32 port = 5432 user = postgres password = postgres');
store path (server = 1, client = 2, conninfo = 'dbname = master host = 10.0.0.35/32 port = 5432 user = postgres password = postgres');

store listen (origin = 2, provider = 2, receiver = 1);
store listen (origin = 1, provider = 1, receiver = 2);
```

**Soubor slave.txt.**

```
# nazev clusteru
cluster name = replika_test;

# definice jednotlivych uzlu v clusteru
# master databaze
node 1 admin conninfo = 'dbname = master host = 10.0.0.35/32 port = 5432 user = postgres password = postgres';
# slave databaze
node 2 admin conninfo = 'dbname = slave host = 10.0.0.36/32 port = 5432 user = postgres password = postgres';

subscribe set (id = 1, provider = 1, receiver = 2, forward = yes);
```