

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Diplomová práce

Srovnání databází MySQL a PostgreSQL

Miroslav Sivoň

© 2015 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra informačních technologií

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Miroslav Sivoň

Informatika

Název práce

Srovnání databází MySQL a PostgreSQL

Název anglicky

MySQL a PostgreSQL databases comparison

Cíle práce

Diplomová práce je tematicky zaměřena na problematiku open-source databází MySQL a PostgreSQL. Hlavním cílem je analyzovat současné možnosti obou databázových systémů z pohledu analytika/vývojáře. Dílčí cíle jsou:

- vytvořit přehled řešené problematiky,
- analyzovat vybranou aplikaci,
- porovnat technologické možnosti:
 - migrace
 - výkonnost
 - bezpečnost
 - SQL normy
 - nástroje pro správu

Metodika

Metodika řešené problematiky diplomové práce je založena na studiu a analýze odborných informačních zdrojů. Praktická část práce je zaměřena na vypracování případové studie analyzující praktické zkušenosti a možnosti zkoumaných databázových systémů. Na základě syntézy teoretických poznatků a výsledků praktické části práce budou formulovány závěry diplomové práce.

Doporučený rozsah práce

60 – 80 stran

Klíčová slova

MySQL, PostgreSQL, Srovnání databázových systémů

Doporučené zdroje informací

1. Schwartz, Baron. Zaitsev, Peter. Tkachenko, Vadim. High Performance MySQL. 3. vydání. O'Reilly Media. 2012. 823 s. 978-1-4493-1428-6
2. Clarke, Justin. SQL Injection Attacks Defense. 2. vydání. Syngress Media, U.S. 2012. 576 s. 1-59749-963-3
3. Bell, Charles. Expert MySQL. 2. vydání. Apress. 2013. 640 s. 1-4302-4659-6
4. Hsu, Leo. Obe, Regina. PostgreSQL: Up and Running. 1. vydání. O'Reilly Media. 2012. 166 s. 1-4493-2633-1
5. Winand, Markus. SQL Performance Explained. 1. vydání. Markus Winand. 2012. 204 s. 3950307826

Předběžný termín obhajoby

2015/06 (červen)

Vedoucí práce

Ing. Alexandr Vasilenko

Elektronicky schváleno dne 31. 10. 2014

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 11. 11. 2014

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 31. 03. 2015

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Srovnání MySQL a PostgreSQL" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.03.2015

Poděkování

Rád bych touto cestou poděkoval Ing. Alexandr Vasilenko za trpělivost a vstřícnost. Renátě Kaplanové za kontrolu pravopisu a revizi dokumentu .

Srovnání databází MySQL a PostgreSQL

MySQL a PostgreSQL databases comparison

Souhrn

Diplomová práce je zaměřena na srovnání databázových systémů MySQL a PostgreSQL. V teoretické části jsou charakterizovány oba srovnávané databázové systémy. Je popsána obecné povědomí, současné použití, historie, architektura a jak funguje vykonání dotazu včetně připojení uživatele do databáze.

Praktická část se věnuje otestování kritérií, které jsou důležité pro Oracle databázový systém. Rozšířené schopnosti databázového systému a Migrace z Oracle databáze mají za úkol otestovat schopnost migrace z Oracle databáze případně ukázat, jak velké budou náklady na migraci. TCO spolu s Cenou za administraci databáze jsou finančními faktory. Pro vytvoření vah byla použita Saatyho vícekritériální analýza. Na závěr je vytvořena přehledná tabulka, která vyhodnocuje a dává odůvodněné vyhodnocení.

Summary

This thesis is focused on comparison of database systems MySQL and PostgreSQL. In the theoretical part are characterized both compared database systems. It describes the general awareness, current use, history, architecture, and how the execution of the query, including the user's connection to the database.

The practical part is devoted to testing criteria that are important for the Oracle database system. Enhanced capabilities of the database system and Migrating from Oracle databases are designed to test the ability of the Oracle database migration possibly or show how large will be the cost of migration. TCO along with the price for the database administration are financial factors. To create a balance has been used Saaty multi-criteria analysis. Finally, it made clear table, which evaluates and gives a reasoned evaluation.

Klíčová slova: MySQL, PostgreSQL, EnterpriseDB Advanced server, Srovnání databázových systémů, Oracle, RDBMS, Migrace, Benchmark, TCO, Bezpečnost, MariaDB

Keywords: MySQL, PostgreSQL, EnterpriseDB Advanced server, An comparison of database systems, Oracle, RDBMS, migration, Benchmark, TCO, Security, MariaDB

Obsah

Úvod.....	10
1. Cíl práce a metodika	12
1.1. Cíl.....	12
1.2. Metodika	12
2. Přehled řešené problematiky.....	13
Kapitola slouží k stručnému uvedení řešené problematiky.	13
2.1. Související teorie.....	13
2.1.1. Zátěžové testy	13
2.2. Mysql	13
2.2.1. Uživatelé	14
2.2.2. Historie.....	14
2.2.3. Současnost	16
2.2.4. Edice	18
2.2.5. Architektura	19
2.2.6. Průběh spuštění dotazu	25
2.3. PostgreSQL.....	26
2.3.1. Historie.....	27
2.3.2. Současnost	29
2.3.3. Architektura	30
2.3.4. Spuštění příkazu.....	32
3. Analytická část.....	36
3.1. Určení váhy kritérií.....	36
3.2. Bezpečnost.....	37
3.3. Vysoká dostupnost.....	39
3.4. Přškolení stavajících zaměstnanců.....	41
3.5. Možnosti databázového systému	43
3.6. Migrace z Oracle DB	45
3.7. Nástroje pro práci.....	53
3.8. Benchmark	55
3.9. Nástroje pro správu.....	56

3.10. TCO	60
4. Zhodnocení výsledků.....	61
5. Závěr	62
6. Seznam použitých zdrojů.....	63

Úvod

Diplomové práce se zabývá srovnáním dvou systémů řízení báze dat. Důvodem vzniku systému řízení báze dat (dále jen databázové systémy) byla potřeba rychle zpracovávat data, které zůstanou konzistentní. Taková potřeba zde byla od nepaměti, ale velké objemy dat přišly ve chvíli, kdy z ústavy USA vyplynulo, že se každých deset let bude sčítat obyvatelstvo USA. První sčítání obyvatel USA byla dokonce delší než samotný cyklus. Postupně se rychlost sčítání obyvatel USA podařilo zvládnout za neskutečné 2 roky. Při jedenáctém sčítání bylo poprvé použito strojové sčítání. Díky němu se podařilo zvládnout tento úkol za 6 týdnů. V této době se prozatím nejednalo o databázový systém jako takový, přesto to byl první krok co nejrychleji zpracovávat data. Doba pokročila a systémy řízení báze dat se uplatnila v mnoha dalších odvětvích, například v bankovníctví, zdravotnictví nebo také v obchodě, které patří do třívrstvé architektury.

Jednotlivé databázové systémy se od sebe dají oddělit pomocí několika charakteristik. Hlavní dělení je pomocí datových modelů, jenž dělí systémy řízení báze dat na hierarchický, síťový, relační, logický, objektový a nosql. Srovnání provádím na relačních datových modelech. Dalšími shodnými charakteristikami jsou například: otevřený kód – angl. open source, podpora transakčního zpracování a podpora obnovení konzistence dat po havárii. Na trhu je mnoho jiných řešení, jenž mají podobné charakteristiky – například Oracle database systém, DB2, MS SQL nebo H2. První tři jmenované příklady mají společný velký podíl na trhu a podporu velkých firem, které je vlastní. Čtvrtý má otevřený kód, ale zatím není nijak výrazně zastoupen na trhu.

Cíl práce je vytvoření studie, která bude sloužit jako podklad pro rozhodnutí o možnosti migrace na jiný databázový systém. Jejím cílem je vybrat správný software, který sníží náklady na pořízení a bude mít podobné vlastnosti jako Oracle databázový systém. Firmou je BSC Praha spol, s. r. o., jenž vyvíjí internetové bankovníctví. Při získání zakázky u menších bank se firma potýká s požadavkem na včlenění ceny licence Oracle databázového systému do prodaného řešení. Výsledná nabídka poté vyvolá tlak na snížení ceny řešení. Oracle databázový systém, který aplikace používá pro ukládání dat, je výborný a respektovaný, ale bohužel velmi drahý. Jeho pořízení však stojí tolik, co stála třetina zakázky jedné nejmenované banky. To byla částka, o kterou dostalo BSC Praha

spol., s. r. o. méně. Z tohoto důvodu budu porovnávat MySQL a PostgreSQL. Obě varianty jsou výrazně levnější na pořízení, jsou zdarma. Pro banky je důležitá podpora, tu rovněž obě firmy splňují. Firma BSC s.r.o se zajímá a je příznivá pro přechod z Oracle database systém a tudíž i vhodnými kandidáty.

Práce je rozdělena na část teoretickou a část praktickou. V teoretické části se zaměřím na definici pojmů, které se týkají obou variant porovnaných řešení, a jejich představení. Praktická část je zaměřena na analýzu, testování a ověření možnosti převést databázovou část z Oracle databáze na porovnávaná řešení. Praktická část je zakončena zhodnocením výsledků, který je reprezentován tabulkou tabulkami. Váhy jednotlivých kritérií byly stanoveny na základě Saatyho metody vícekritériální analýzy variant.

1. Cíl práce a metodika

1.1. Cíl

Diplomová práce je tematicky zaměřena na problematiku open source databází MySQL a PostgreSQL. Hlavním cílem je analyzovat současné možnosti obou databázových systémů z pohledu analytika/vývojáře. Dílčí cíle jsou:

- vytvořit přehled řešené problematiky,
- analyzovat vybranou aplikaci,
- porovnat technologické možnosti:
 - migrace
 - výkonnost
 - bezpečnost
 - SQL normy
 - nástroje pro správu

1.2. Metodika

Metodika diplomové práce je založena na mém studiu a analýze odborných informačních zdrojů. Praktická část práce je zaměřena na vypracování případové studie analyzující praktické zkušenosti a možnosti zkoumaných databázových systémů. Na základě syntézy teoretických poznatků a výsledků praktické části práce budou formulovány závěry této diplomové práce.

2. Přehled řešené problematiky

Kapitola slouží k stručnému uvedení řešené problematiky.

2.1. Související teorie

2.1.1. Zátěžové testy

Transaction Processing Performance Council (TPC) je nezisková organizace založená roku 1988. TPC publikuje řadu testů pro porovnávání efektivity konkrétních databází na konkrétních strojích. Pro takovéto porovnání je nutné stanovit metodiky a také měrné jednotky. TPC se snaží přiblížit svoje testovací data a operace k těm, které se často vyskytují v komerčních aplikacích (například vazby typu zákazník-zboží-obednávka). Základní jednotkou většiny testů je počet transakcí za minutu, přičemž konkrétní metodika testu určuje, co se považuje za transakci (např. tpmC – tedy transactions per minute u testu typu C). Vedlejší jednotkou je cena za transakci (\$/tpmC) a při porovnání konkrétních hardwarových konfigurací i doba, za kterou je schopen výrobce tuto konfiguraci dodat. Na základě některých těchto testů si i jednotliví výrobci databází připravují vlastní metriky, například nástroj pgbench pro PostgreSQL je volně založen na dnes již nepoužívaném TPC benchmarku typu B. (Tůma, 2014, s. 40)

Pro zátěžové testy OLTP databázových systémů se používá:
- TPC – C, simuluje objednávkový systém a zátěž od více uživatelů současně. Vyznačuje se vysokými nároky na diskové přenosy, transakční integritu a tabulkami s různou velikostí vložených dat. (Tůma, 2014, s. 40)

2.2. Mysql

MySQL, dle tvůrců
jde o
nejpopulárnější
open source
databázi.



The world's most popular open source database

Obrázek 1- Logo MySQL

Počáteční prioritou MySQL [My Ess Que Ell] bylo vytvořit jednoduchou, přívětivou a hlavně rychlou databázi pro jednoduché operace. Toto rozhodnutí mělo vliv

na ostatní vlastnosti, které by mohli narušovat zvolené priority nebo je přímo nepodporovaly – např. ACID¹, triggerů nebo konstraintů byly nadbytečnou vlastností. Proto se rozšířilo podvědomí o MySQL jako o jednoduché a rychlé databázi, která uměla základní prvky běžné pro ostatní OLTP databázové systémy. Dnes se jedná už jen o pověst, protože uvedené informace se nezakládají na pravdě a MySQL udělala velký skok kupředu. V průběhu času vývojáři rozvíjeli MySQL o chybějící funkcionalitu. U tohoto databázového systému se klade důraz na praktickou jednoduchost a rychlé zpracování základních operací, která je stále její největší prioritou.

2.2.1. Uživatelé

MySQL je zaměřena převážně na uživatele jednoduchých webových aplikací. Právě pro ně existují projekty jako například WAMP (Windows, Apache 2, Mysql/MairaDB a Php) nebo XAMP. Při zadání hesla do google: „webhosting with MySQL“ jsem dostal 730 tisíc výsledků. Dotaz zaměněný za PostgreSQL namísto MySQL zobrazí „jen“ 208 tisíc dotazů. Tento výsledek potvrzuje prvotní tvrzení.

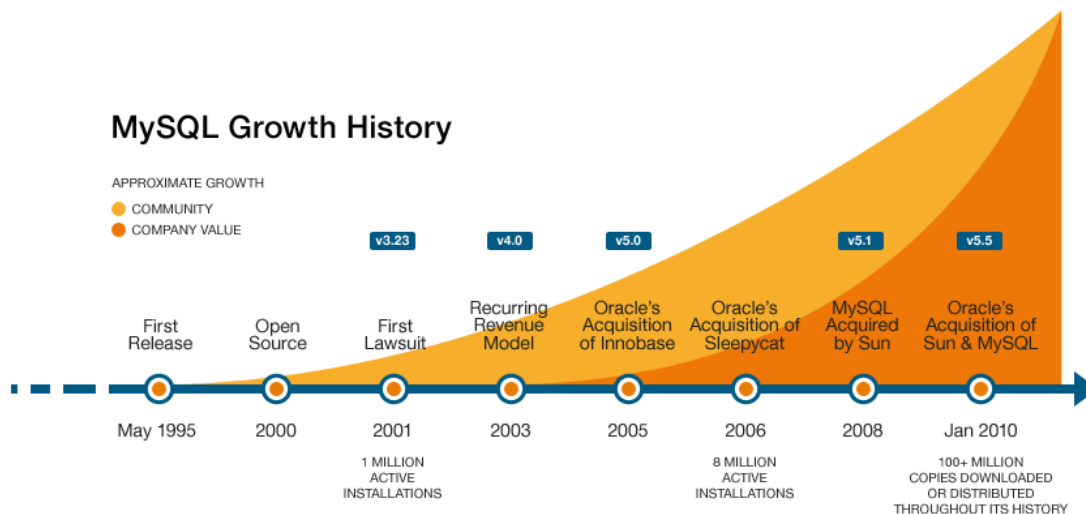
MySQL je používána i pro podnikovou sférou a to nejen pro účely webových stránek. Nejznámější společností je Facebook. Samotný facebook provozuje 1.800 MySQL databázových serverů (Miller, 14. 2009) Jsou zde i další společnosti ze sféry sociálních sítí jako je LinkedIN. Zástupcem retailových obchodníků je firma Walmart nebo Amazon a ze společností, které mají na starosti finance lze uvést Bank of Finland nebo Paypal.²

2.2.2. Historie

Historie MySQL začíná rokem 1995 kdy byla založena Švédská firma MySQL AB. Založili ji Michael Widenius (Monty), David Axmark and Allan Larsson.(Buytaert, 8.3.2010)

¹ ACID je zkratka pro Atomicity, Consistency, Isolation, Durability. Je to sada vlastností, která garantují, že transakce v databázovém systému bude fungovat dle definice.

² Další reference a zákazníci lze dohledat na [www http://www.mysql.com/customers/](http://www.mysql.com/customers/).



Obrázek 2 - Historický růst MySQL [zdroj: <http://www.databasefriends.co/2014/02/history-of-mysql.html>]

Historie změn v rámci jednotlivých verzí MySQL v tabulce 1 níže.

Verze (datum vydání)	Popis
Version 3.23 (2001)	Verze 3.23 se považuje za vznik Mysql. Není v ní o moc více než soubory obalené "SQL" jazykem. Podstatnou změnou je uvedení MyISAM jenž nahradilo ISAM. InnoDB bylo možné používat, ale nebyl poskytován ve standardním balíčku, takže pro použití uživatel musel stáhnout dodatečně a zkompilovat. Další velmi podstatným prvkem je full-text indexování a replikace. Replikace fungovala na jednoduchém principu. Jakákoliv operace DML a DDL byla logována do souboru. Tento soubor byl následně spuštěn na cílové databázi.
Version 4.0 (2003)	Verze 4.0 přináší nové prvky jako např. podpora UNION a multi-table DELETE. Replikace byla přepsána na používání dvou threadu místo jednoho. InnoDB bylo zahrnuto v instalačním balíčku. Uživatelé dočkali také podpory transakcí a row-level zámek v MySQL. Podstatnou novinkou byla query cache. Z bezpečnostních prvků uvedu podporu SSL připojení.
Version 4.1 (2005)	Hlavní změnou byla podpora unicode neboli podpora UTF8. Vývojáři přidali rozšíření syntaxe dotazů jako jsou poddotazy (angl. subqueries) a insert on duplicate key update.
Version 5.0 (2006)	U této databáze byly přidány "enterprise" prvky jako view, trigger, procedury a funkce. Databáze ISAM byla odebrána z releasu. Novinkou byla nová databáze jmenem Federated.
Version 5.1 (2008)	První release po koupi MySQL AB firmou Sun Microsystems.

	Verze přináší partitioning, row-based replikaci, mnoho plugin API, ale hlavně pluggable storage engine, což je hlavní charakteristický prvek MySQL databáze. Databáze BerkeleyDB byla odebrána a Federated byla označena za zastaralou (angl. deprecated). Oracle je vlastníkem Innobase Oy což znamená, že vlastní i InnoDB.
Version 5.5 (2010)	Sun Microsystems bylo koupeno firmou Oracle Inc. Oracle získal do svého portfolia HW, OS, Javu a opensource programy včetně MySQL. Verze se soustředila na vylepšení rychlosti, škálovatelnosti, replikace, partitioningu a vylepšené podpory pro MS Windows. InnoDB se stalo základní databází. Pro lepší tuning byl přidán PERFORMANCE_SCHEMA spolu s prvními možnostmi batchových operací. U této databáze byly přidány nové pluginy API pro replikaci, autentizaci a auditing. V rámci komerčních verzí Oracle přidal možnost vícesměrné replikace, vylepšenou autentizaci a thread pooling.
Version 5.6 (2013)	U této verze bylo provedeno mnoho postupných vylepšení, zejména musím zdůraznit vylepšení v oblasti query optimizéru, verze obsahuje nová API například pro full-text vyhledávání nebo byla přidána další statistiky do PERFORMANCE_SCHEMA.
Version 6.0 (zrušena)	V mezi čase vznikla verze 6.0.11 alfa - http://lists.mysql.com/packagers/418 - s mnoha úpravami - ty byly většinou nakonec zahrnuty do verze 5.5 a 5.6. Zrušením verze 6.0 vedení MySQL změnilo strategii uvolňování nových releasů na mnohem kratší interval.

Tabulka 1 - Historie změn (zdroj: Schwartz, 2012, 30 s.)

2.2.3. Současnost

Před koupí firmy Sun Microsystems firmou Oracle spoluzakladel databáze MySQL Michael "Monty" Widenius žádal své obchodní partnery, aby posílali svůj nesouhlas s akvizicí Evropské komisy. Monty měl za cíl udržet MySQL mimo dosah Oracle z důvodu ohrožení budoucnosti MySQL. Kromě technických a historických důvodů – především souvisejících s akvizicí InnoDB – byl hlavním důvodem této žádosti fakt, že Oracle a MySQL jsou velcí konkurenti. Markus použil příměr s operačními systémy založenými na platformě Unixem a Linuxem. Oracle Corp. má biliónové obraty. MySQL se pohybuje ve stovkách milónů. To je na jednu stranu hodně, aby MySQL byl považován za konkurenci a na druhou stranu žalostně málo, aby to ospravedlnilo kanibalizaci na výnosech z Oracle databáze. (Widenius, 12.12.2009) Oracle se v minulosti snažil bránit konkurenci s pomocí Express edice – ta je zdarma i ke komerčnímu užití. Zdarma je ale

pouze za podmínek různých omezení – jako například limitování místa a paměti. Markus nedokázal zabránit převzetí MySQL firmou Oracle. Následně se rozhodnul odejít a založil odvozenou MySQL databázi jménem MariaDB. Mimochodem Maria je jméno dcery Markuse a je jménem nové databáze. Zrušení MySQL se neuskutečnilo. Vývoj MySQL pokračuje dál.

Alternativy MySQL

Percona Server je zpětně kompatibilní s MySQL. Sám sebe označuje jako vylepšenou verzi MySQL, hlavně co se výkonosti týká. Místo InnoDB používá XtraDB, což to vylepšený InnoDB. I ten je zpětně kompatibilní a je schopen rozpoznat a převést tabulku vytvořenou v InnoDB automaticky do XtraDB. Percona Server stále používá MySQL jako vzor, ze kterého vychází.

Percona Server má tři hlavní cíle:

1. **Transparentost.** Transparentnosti se myslí zejména možností vidět interní možnosti a chování. Je to zvládnuto pomocí přidání prvků ve výstupu z příkazu SHOW STATUS. Dále byly přidány tabulky do INFORMATION_SCHEMA, ale hlavně je lépe popsán výstup z query logu.
2. **Výkonost.** Hlavní pilíř. Percona se snaží o stabilitu a předvídatelnost.
3. **Flexibilita.** Flexibilitou je myšleno odebrání limitů MySQL serveru. (Schwartz, 2012, 680 s.)

MariaDB. Byla vytvořena Monty Wideniusem, spoluzakladatelem MySQL. Monty vytvořil společnost se jménem Monty Program AB, která vyvíjí MariaDB. Jeho hlavní záměr je dodávat opravdu otevřenou databázi.

MariaDB používá XtraDB od Percony. Tím, ale rozdíly neskončili. Na rozdíl od Percona Serveru se MariaDB pustila do úprav celé databáze a to nejenom do změn v úložišti dat. Změny jsou převážně v optimizéru. MariaDB používá pro dočasné tabulky Aria uložení místo MyISA, přidala komunitní uložení dat jako je SphinxSE a PBXT. Stejně tak jako Percona Server i MariaDB se snaží o zpětnou kompatibilitu s MySQL. (Schwartz, 2012, 680 s.) Aktuálně rozdělila vývoj mezi verzi 5.6 a 10. Verze 10 se má odlišit od MySQL. Analogicky se nabízela verze 5.6. Marcus novou verzi databáze nechtěl spojovat s MySQL verzí 5.6. Důvodem je mnoho nových úprav a prvků. (The MariaDB

Blog, 14.1.2013) Verze 10 rovněž přilákala Google, který migruje jednotlivé aplikace z MySQL do MariaDB (Jelínek, 15.09.2013)

Drizzle.

Je to skutečná nezávislá větev, která neslouží pouze k obohacení MySQL jako je tomu u Percona serveru a v současné době stále je u MariaDB. Není zcela kompatibilní s MySQL. Drizzle byl vytvořen pro jednodušší fungování a soustředí se pro webové aplikace. Má změněnou SQL syntaxi. Zjednodušením se myslí například pouze podpora unicode nebo jediná možnost datového typu BLOB. Ten je tvořen pro 64bitový hardware a podporuje IPv6. Drizzle se soustředí na eliminaci prostých chyb MySQL server. Některé části nebyly zrovna nejlépe napsány – například triggerů – stávalo se, že i po nastavení sloupce jako not null byl sloupec bez hodnoty. Architektura je vytvořena jako mikrokernél. Core serveru je značně minimalizován a většina funkcí se stala pluginem. Plugin je licencován pod GP a vyvíjí ho početná komunita, která je početnější než MySQL nebo MariaDB. V případě tvorby komerční aplikace a nutnosti komunikovat s MySQL databází bylo nutné pořídit komerční variantu MySQL verze. Toto již dnes pravdou, protože Drizzle vytvořil svoji vlastní variantu knihovny - libmysql, která komunikuje MySQL jazykem, ale ta je licencována pod BSD. (Schwartz, 2012, 682 s.)

Výčet není kompletní. Byly představeny hlavní větve. Další větve budou ovedeny ve zkratce – jsou jimi například OurDelta, DorsalSoure atd. Neveřejné větve jsou například v Google, eBay nebo Facebooku. Každá s těchto firem má svoji upravenou verzi, která slouží jejich potřebám. Mimochodem jakmile si stáhnete MySQL přes repository linuxové distribuce. Nejedná se v té chvíli o MySQL jako takovou, ale o modifikovanou větev. (Schwartz, 2012, 683 s.)

Jak velký je rozdíl mezi těmito větvemi, se mi nepodařilo dopátrat.

MariaDB převzala roli hlavní databáze pro linuxové distribuce Fedora, openSUSE, RedHat (McAllister, 15.06.2013)

2.2.4. Edice

MySQL poskytuje různé verze pro potřeby svých zákazníků. Vlastnosti edicí lze dohledat v tabulce 2. Ceny jsou uvedeny pro servery, které mají 1-4 sockety.

Produkt	Popis	Licence	Cena
MySQL Cluster Carrier Grade Edition	High-performance, in-memory klaster; řešení nabízející vysokou dostupnost	komerční	\$10,000.00**
MySQL Enterprise Edition	Edice obsahuje MySQL Enterprise Monitor, backup, Security, Encryption, Audit atd. Nástroje nejsou dostupné pro MySQL Standard edici.	komerční	\$5,000.00**
MySQL Standard Edition	Obsahuje InnoDB úložiště dat, které dělá z této edice transakční databázi včetně podpory ACID. Edice obsahuje také možnost replikace.	komerční	\$2,000.00**
MySQL Classic	Embedded databáze pro ISVs, OEM a VARs vývoj aplikace využívajících čtení za pomoci MyISAM úložiště dat.	komerční	Na zavolání
MySQL Embedded (OEM/ISV)	Jakákoliv z výše uvedených edicí licencovaná jako OEM/ISV embedded aplikace	komerční	Na zavolání
MySQL Community Edition	Volně dostupná a modifikovatelná verze.	GPL	Zdarma
** Jedná se o katalogovou cenu, cena po slevě může být až o 40 procent nižší.			

Tabulka 2 - MySQL edice [zdroj: <https://www.mysql.com/products/>, upraveno]

2.2.5. Architektura

Architektura je odlišná ve dvou věcech od ostatních databázových systémů – PostgreSQL, DB2, Oracle, MSSQL. Za prvé neexistence schémat. Schéma je chápáno jako synonymum pro databázi (MySQL 5.6 Reference Manual, 23.11.2013). Za druhé pak množstvím formátů úložiště dat, které se dají použít ve vytvořené databázi pro jednotlivé tabulky a ovlivňují schopnosti MySQL. Z toho vyplývá, že jednotlivé v databázi může být 1 až N typů úložišť dat. Anglicky se úložiště dat nazývá storage engines. Popis vlastností úložišť a jejich specifikace rozepíší dále v této kapitole.

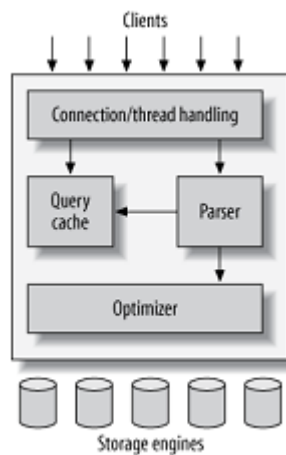
Obrázek 3 zobrazuje společně jednotlivé komponenty MySQL Serveru, které tvoří databázový systém. Dle funkčnosti jsou komponenty rozděleny do tří vrstev.

První vrstva obsahuje síťové služby – klient/server nástroje nebo nástroje pro připojení do databáze a jejich obsluhu, autentifikaci a zabezpečení (SSL).

Druhá vrstva je základem MySQL Serveru. Najdete v ní parsování dotazů, jejich analýzu a optimalizaci. Také je zde uložena paměť (key_buffer_size) pro rychlejší

odpověď na dotazy, které jsou v ní již obsaženy. Druhou oblastí jsou zabudované funkce – například datum, čas nebo šifrování. V této vrstvě jsou také uloženy procedury, trigger a view.

Ve třetí vrstvě jsou datová úložiště. Server s nimi komunikuje skrz API. API poskytuje jednotný interface. Z API není poznat, o jaké datové úložiště se jedná. API mají možnost provést operace jako je „begin transaction“ nebo „fetch the row that has this primary key“. Datová úložiště nekomunikují pomocí SQL resp. ani napřímo spolu navzájem. To platí pro všechny datové úložiště kromě InnoDB. MySQL prozatím neimplementovalo komunikaci s tímto datovým úložištěm do MySQL Serveru. API jednoduše odpovídají na dotazy. (Schwartz, 2012, 2 s.)



Obrázek 3 - Logický pohled na MySQL Server architekturu [zdroj: Schwartz, B.. 2012, 2 s.]

Datová úložiště

V rámci této subkapitoly uvedu základní popis jednotlivých typů datových úložišť.

MySQL ukládá svou databázi jako podsložku v její datové složce na disku. Když vytvořím tabulku, tak se její definice zapíše do souboru s příponou frm a bude mít jméno tabulky. Pro zjištění základních vlastností tabulky lze použít následující příkaz:

```

mysql> SHOW TABLE STATUS LIKE 'user' \G
*****
1. row
*****

Name: user
Engine: MyISAM
  
```

```
Row_format: Dynamic
Rows: 6
Avg_row_length: 59
Data_length: 356
Max_data_length: 4294967295
Index_length: 2048
Data_free: 0
Auto_increment: NULL
Create_time: 2015-01-24 18:07:17
Update_time: 2015-01-24 21:56:29
Check_time: NULL
Collation: utf8_bin
Checksum: NULL
Create_options:
Comment: Users and global privileges
1 row in set (0.00 sec)
```

InnoDB úložiště

Toto úložiště je hlavním, nejdůležitějším a nejužitečnějším úložištěm v MySQL. Podporuje transakční zpracování, které není omezeno délkou transakce. Spíše se transakce dokončí než, aby byly změny vráceny zpátky. InnoDB je hlavně ve verzi 5.5+ známé pro vysokou rychlost zpracování příkazů. V případě výpadku se provede automatická obnova. Nepotvrzené transakce budou zahozeny.

Ukládá data v InnoDB se rozlišuje na logické a fyzické. Fyzické uložení je pomocí datových souborů na discích. Logické uložení je pomocí tablespace, kterým je přiřazen jeden a více datových souborů – data files. Uložení dat tabulek a dat indexů je rozděleno do různých souborů, důvodem je zvýšení výkonosti.

InnoDB podporuje MVCC³ pro dosažení velkého souběhu činnosti jednotlivých příkazů/dotazů, dále podporuje izolaci. Výchozí izolací je repeatable read. Vlastním

³ MVCC neboli Multiversion concurrency control je metoda jak udržet doovou konzistenci při přístupu více procesů do stejné tabulky

mechanismem zabraňuje phantom reads. Mechanismus je založen na vkládání zámku do neošetřených hodnot v indexových hodnotách.

Indexy nad tabulkami jsou vytvořeny jako klusterované indexy. Výsledkem je velmi rychlé vyhledávání pomocí primárního klíče. Další indexy vytvořeny nad tabulkou obsahují primární klíč. Jamile je primární klíč velký, ostatní indexy jsou rovněž velké, doporučuje se vytvářet co nejmenší primární klíč. MySQL dovoluje vytvářet identické indexy, které jsou nad stejnými sloupci. Tyto indexy jsou aktivní – probíhají nad nimi operace pro udržení odpovídajícího stavu. Druhý index nebude při dotazech použit a tím se stává zbytečným. Důsledkem je snížení eventuální výkonosti. Navíc tento index zabírá diskový prostor. (Schwartz, 2012, 16 s.)

InnoDB podporuje zálohování online.

MyISAM

Verze 5.1 to byla výchozí datové úložiště. MyISAM umožňuje fulltextové vyhledávání, komprimaci a GIS funkce. Od InnoDB se liší tím, že neumožňuje transakční zpracování a řádkové zamykání tabulek. Zamykání probíhá na úrovni celé tabulky. Další nevýhodou, která přispěla k velmi špatné pověsti je neschopnost regenerovat na výpadek databázového systému. MyISAM se nedoporučuje použít jako hlavní datové úložiště. Pro webové stránky, jejíž obsah se nemění a nevstupuje sem více uživatelů najednou, je to výborné datové úložiště, které nemá konkurenci v rychlosti provádění dotazů.

MyISAM ukládá data tabulek a data v indexech do dvou souborů. Příponou je MYD resp MYI. Datový typ umožňuje ukládat statickou a dynamickou velikost řádků. Technicky je to zajištěno pointery, které jsou 6bytové. Tabulky mohou být velké až do 256TB. (Schwartz, 2012, 18 s.)

Archive

Datové úložiště Archive podporuje operace typu Insert a Select. Díky kompresy řádků pomocí kompresní funkce zlib má mnohem menší velikost na soborovém systému. Dotazy na data vyžadují načtení celé tabulky. Datové úložiště podporuje zámky nad jednotlivými řádky a má speciální paměť pro inserty. Dalším prvkem je možnost bulk insertů. Archive není transakční databází, a už jeho název zcela jasně napovídá co je jeho úkolem, a ten plní velmi dobře. (Schwartz, 2012, 19 s.)

The Blackhole engine

V překladu černá díra nemá za úkol ukládat data a není k tomu připravena. Jakýkoliv insert je zahozen. Dotazy proti tomuto úložišti dat jsou ukládány do logu. Blackhole engine je populární pro možnost jednoduché replikace nebo pro audit. Nicméně v praxi jsou s ním problémy. (Schwartz, 2012, 20 s.)

The CSV engine

Datové úložiště pro CSV formát. Není nad ním podpora indexů. (Schwartz, 2012, 20 s.)

The Federated engine

Datové úložiště se chová jako proxy server pro ostatní servery. Otevře sezení klientovi na další server, kde spustí příkaz nad danou tabulkou. Federated engine je zamčený ve výchozím nastavení. (Schwartz, 2012, 20 s.)

The Memory engine

V MySQL jsou pojmenovány jako memory tabulky, obecně se používá pojem HEAP. Této datové úložiště je užitečné zejména pro rychlost vyhledávání neměných dat. Ty jsou, jak název napovídá, uloženy v paměti. Výhodou je rychlejší přístup k datům. Paměť je rychlejší pro přístup než I/O na disku. Nevýhodou je, že data jsou po restartu ztracena. Datové úložiště podporuje hash indexy. Memory engine nepovoluje datový typ BLOB a TEXT. Varchary jsou přeměny na chary, aby byla dodržena pevná délka řádků. (Schwartz, 2012, 20 s.)

MySQL používá Memory engine jako dočasné tabulky pro mezi výpočty dotazů a jejich řazení. V případě, že obsahují BLOB nebo se nevejdou do paměti, je použit MyISAM. Memory engine však není z povahy věci dočasná tabulka, kterou uživatel vyvolá za pomoci příkazu: „CREATE TEMPORARY TABLE“. I v tomto případě je použito úložiště MyISAM. Hlavní rozdíl mezi nimi je, že dočasná tabulka je viditelná pouze pro dané sezení a memory engine je viditelný pro všechny sezení.

NDB Cluster engine

MySQL AB koupila NDB database od Sony Ericsson v roce 2003 a použila NDB database jako své řešení klasterového systému. Cluster storage engine je interface mezi SQL a MySQL a protokolem NDB. Klaster je kombinací MySQL serveru, NDB Cluster engine, distribuovaného, nic nesdílejícího, tolerantního k chybám řešení splňujícímu možnosti vysoce dostupného řešení. (Schwartz, 2012, 21 s.)

Pluggable Storage Engines.

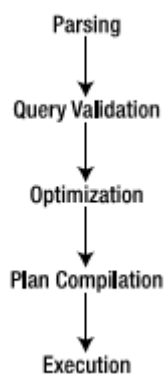
Nejpoužívanější engine jsou MyISAM, InnoDB a Memory. Každý engine má odlišné vlastnosti. Při návrhu aplikace se nemusíme omezovat pouze na jeden engine. Databáze resp. Schéma může využít výhody každého typu z nich. Nejpoužívanější kombinací je InnoDB pro transakční zpracování spolu s Memory enginem.

Indexy

MySQL má podporu B-tree indexů, R-tree a clustered indexů. V InnoDB se používají clusterované indexy. O tabulkách lze mluvit jako o index organized tables. (Ovais Tariq, 2011)

2.2.6. Průběh spuštění dotazu

Klient zadá příkaz ve své aplikaci resp. klientovi. Pomocí síťové vrstvy proběhne připojení. Při připojení dostane klient vlastní thread. V komunitní verzi server pouze ukládá do paměti thready, tím odbourává nutnost jejich znovu vytvoření v případě připojení dalšího klienta. V komerčních verzích od verze 5.5 existuje plugin, který umožňuje thread-pooling⁴, je to vylepšená verze a jakmile dojde k připojení, server bude autentifikovat nového klienta. Autentifikace je na základě jména, počítače a hesla. Připojení je možné s X.509 certifikátem (SSL). Po připojení server vždy před každým příkazem ověří oprávnění uživatele – například jestli uživatel má možnost číst tabulku. (Schwartz, 2012, 4 s.)



Obrázek 4 Proces spuštění příkazu [Bell, Charles., 2013, 31 s.]

Jakmile příkaz dorazí na sever přes síť, je na serveru, aby příkaz interpretoval a spustil. Na obrázku 4 jsou vidět jednotlivé činnosti. Prvním krokem je přeložení logického příkazu z SQL do příkazového (anglicky query tree) stromu v relační algebře. Krok zahrnuje také rozpadnutí příkazu na jednotlivé části a je proveden pomocí parseru. V druhém kroku je příkaz přeložen z příkazového stromu v logické algebře do fyzického plánu. Ke každému příkazovému stromu lze vytvořit více možností fyzického plánu, proto dalším krokem je najít nejlepšího z těchto plánů. Proces se jmenuje query optimization neboli optimalizace dotazu. (Bell, 2013, 31 s.)

⁴ Thread-pool je onačení pro frontu threadu. Každý thread buď pracuje na přiděleném úkolu nebo čeká. Pomocí parametrů jej lze ovlivnit. Více se lze dočíst zde: <https://mariadb.com/kb/en/mariadb/threadpool-in-55/>

Důležité je, najít optimální plán. Na tomto místě si dovolím paralelu s šachovým programem. Počítač může hledat vhodný tah vteřiny, ale i desítky minut. Rozdíl je v počtu možných kroků resp. reakcí oponenta na mé kroky atd. Stejně tak to má i optimizér. Nicméně v jeho případě čekat na nějaký dotaz hodinu, když samotný dotaz by se provedl (i ten nejhorší možný) za hodinu. Například optimizér dojde za hodinu k plánu, který vykoná za 10 min, výsledek je 70 min namísto hodiny a vteřiny. Z tohoto důvodu optimizér má za úkol v omezeném čase najít optimální plán. Důvod, proč optimizér by mohl hledat hodinu je zejména v případě spojení různých tabulek pomocí join. MySQL spojuje tabulky pomocí metody nested loop. Spojení probíhá tak, že hodnota jedné tabulky je porovnávána cyklem vůči druhé tabulce. Tato metoda se vyplatí v případě, že první tabulka je malá. Pokud máme dvě tabulky, jsou dvě možnosti. V případě třech tabulek je kombinace 6. Při deseti tabulkách je možností jak provést spojení dokonce 3.628.800. Kombinace je prováděná pomocí n-faktoriál, kde n je počet tabulek spojení. Propočítávání těchto možností a hledání v nich optimálního plánu se říká search space. (Schwartz, 2012, 226 s.)

2.3. PostgreSQL

PostgreSQL [vyslovuje se Post-Gres-Q-L], rovněž známá jako Postgres, je open source



Obrázek 5 – Logo PostgreSQL

databáze typu ORDBMS, která se vyvinula z projektu Ingres. Zajímavostí je, že na databázi Ingres pracoval sám Codd, autor principu relační databáze. (Stěhule, 22. 05. 2012)

Počáteční vizí PostgreSQL bylo vytvořit pokročilejší, ale stabilnější databázi, která bude co nejvíce odpovídat SQL standardu. To celé za cenu, že bude pomalejší. I Postgres udělal velký krok v rychlosti. Na rychlosti databáze se velmi tvrdě pracovalo, takže rychlost komplexních dotazů a zpracování dotazů pod velkým množstvím uživatelů je rychlejší než u Mysql.

„Databáze PostgreSQL” je navržena a optimalizována pro tzv. podnikové prostředí. To se vyznačuje požadavkem na maximální možnou spolehlivost a dobrý výkon při

souběžném zpracování komplikovanějších SQL příkazů bez ohledu na objem dat. Kromě toho PostgreSQL nabízí více než obvyklý komfort a široký repertoár funkcí.“ (Stěhule, 2013)

PostgreSQL je projektem, který je vyvíjen komunitou vývojářů a několika málo firm. V průběhu času se mnoho firem hlásilo k podpoře – Sun, Red Hat. Většina z nich, ale od tohoto záměru z různých důvodů nakonec ustoupila, totiž největším problémem byl pro ně nalézt obchodní model. Firma EnterpriseDB ho nakonec ale našla. Modifikovala PostgreSQL, aby byl kompatibilní s Oraclem. Jejich cílem je přitáhnout co nejvíce zákazníků Oracle. Tato firma dnes láká zákazníky na nulovou licenci, s 3x menšími náklady na podporu. EnterpriseDB podpořila vývoj tohoto projektu a to zaměstnáním vývojářů PostgreSQL a přidala prvky do databáze – například PL/pgSQL debugger. (Stěhule, 2013)

2.3.1. Historie

„PostgreSQL vychází z databáze POSTGRES – výzkumného prototypu DBMS nové generace. Z postgresu byl převzat silný datový model a bohatý soubor datových typů a jeho dotazovací jazyk PostQuel byl nahrazen rozšířenou podmnožinou jazyka SQL. PostgreSQL lze používat bez omezení a jeho zdrojové kódy jsou volně k dispozici. (Stěhule, 22. 05. 2012)

Autory první verze PostgreSQL 1.01 byli Andrew Yu and Jolly Chen. Původní kód Postgresu, ze kterého PostgreSQL vychází, je výsledkem úsilí mnoha studentů a programátorů pracujících pod vedením prof. Michaela Stonebrakera na University of California v Berkley. (Stěhule, 22. 05. 2012)

Původní název software z Berkley byl Postgres. Po přidání jazyka SQL se název změnil na Postgres95. Koncem roku 1996 byl RDBMS přejmenován na PostgreSQL.“ (Stěhule, 2015)

V roce 1991 a 92 Pamela Marcera a Siliva de Hoop implementovali datové typy, které umožňují podporovat geografické informační systémy (GIS). Na základě této podpory byl Postgres použit v projektu Sequiia 2000 – cílem tohoto projektu bylo vytvoření informačního systému pro stadium globálních změn. Díky projektu se zdvojnásobil počet uživatelů. (Stěhule, 22. 05. 2012)

Do verze 4 byl Postgres zcela akademickým projektem.

V první polovině devadesátých let se potkal David Hughes s vývojáři PosQUELU. Tato databáze se mu však ukázala jako příliš příliš komplexní a pomalá. David Hughes proto hledal jinou databázi, a Michael Winenius (spoluzakladatel MySQL) zase hledal někoho, kdo má interface. Na základě jejich spolupráce vznikla první verze MySQL 1.0. (Stěhule, 22. 05. 2012)

V roce 1995 byl implementován jazyk SQL do PosQUELU, pročištěn kód a jeho optimalizace. Databáze byla přejmenována na PostgreSQL95. (Stěhule, 22. 05. 2012)

Další verze PostgreSQL vznikaly běžným vývojem, níže uvádím tabulku 3 s verzemi databáze a hlavními změnami. Nejedná se o úplný výčet změn, ale pouze o zásadní změny.

Verze (datum vydání)	Změny
PostgreSQL 7.4 (Release date: 2003-11-17)	Uložení informace o použitém kódování do dumpu databáze a automatické dekódování při načtení dumpu
PostgreSQL 8.1 (Release date: 2005-11-08)	Rozšíření PL/pgSQL
PostgreSQL 8.2 (Release date: 2006-12-05)	Implementace operátoru IS NOT DISTINCT FROM
PostgreSQL 8.3 (Release date: 2008-02-04)	Rozšíření PL/pgSQL, příkaz MOVE v PL/pgSQL, prototyp SQL/XML
PostgreSQL 8.4 (Release date: 2009-07-01)	Práva na sloupce, paralelní obnova
PostgreSQL 9.0 (Release date: 2010-09-20)	Pravidla hesel, podpora 64-bit Windows,
PostgreSQL 9.1 (Release date: 2011-09-12)	Synchronní replikace, SE Linux integrace
PostgreSQL 9.2 (Release date: 2012-09-10)	Index-only čtení, nativní podpora JSON
PostgreSQL 9.3 (Release date: 2013-09-09)	data checksums, materializovaná view,
PostgreSQL 9.4 (Release date: 2014-12-)	Linux huge page podpora, JSONB data

18)	type, ALTER SYSTÉM pro změnu konfigurace databáze
-----	---------------------------------------------------

Tabulka 3 - Hlavní změny ve verzích PostgreSQL [zdroj: autor]

2.3.2. Současnost

Vzhledem k licenční politice komunity, která ji vyvíjí a přehlednému kódu, není divu, že PostgreSQL má sama mnoho větví. Některé z nich obstály, jiné skončily v propadlišti dějin. Nově vzniklé databáze jsou rozděleny do dvou druhů – upravené pro OLTP⁵ a OLAP⁶.

Databáze pro OLTP je reprezentována databází PostgrePlus resp. PostgrePlus Advanced Server, za níž stojí firma EnterpriseDB. Druhým zástupce je databáze Mammoth PostgreSQL. Také stojí za zmínku Yahoo! Everest, největší databáze postavená na PostgreSQL – její velikost je cca 5-10 peta-byte. Everest je silně upravená tak, aby se řádky chovali jako sloupce.

Zastupci OLTP jsou Netezza (IBM), Greenplum Database a Amazon Redshift. U Amazon Redshift je zajímavé, že je vytvořena pro Cloud a pro její vytvoření byla použita databáze ParACCEL. (Hsu, Leo. Obe, Regina., 2012, 1. s)

Největším trhem pro Postgres je Severní Amerika a Japonsko. Dnes se přidává i Jižní Korea. Uživatelé jsou převážně akademici. PostgreSQL se používá pro DNS servery. Z komerčních firem PostgreSQL používá Twitter, Instagram a Skype.

Hlavní větve PostgreSQL databázového systému:

- PostgreSQL Plus Advanced Server od firmy EnterpriseDB. Srovnání vlastností oproti komunitní verzi PostgreSQL lze nalézt na stránce <http://www.enterprisedb.com/products-services-training/products/postgres-plus-Advanced-server#different>
- Mammoth PostgreSQL je PostgreSQL vylepšený o Mammoth replikaci.
- Postgres spolus open source ad only od EnterpriseDB
- Hadoop

⁵ OLTP neboli Online Transaction Processing je databáze určená k online zpracování a vyhledávání dat

⁶ OLAP neboli Online Analytical Processing je databáze, ve které se provádějí složité dotazy nad agregovanými daty. Databáze je určena pro aplikace Business Intelligence.

- Výčet

2.3.3. Architektura

PostgreSQL používá multigenerační architekturu převzatou z Ingres databáze. Záznamy se neupravují, ale vytváří se nová kopie. V rámci odmazávání starých záznamů slouží proces VAKUUM.

Z obrázku 6 mi architektura PostgreSQL připomíná Oracle až na to, že Oracle má více hlavních procesů než PostgreSQL. Hlavní proces, Postmaster, se stará o databázi, připojení klienta a jeho alokování paměti. Od verze 8.4 je aliasem postgres procesu. Postmaster se stará o přístup k datům uživatelů – kontroluje oprávnění uživatelů. Má pod kontrolou datové soubory. (Stones, Matthew, 2007, 58 s.) PostgreSQL se skládá z datových a konfiguračních souborů a spolu s nainstalovaným softwarem. Po spuštění procesu Postmaster jsou spuštěny procesy a sdílené paměti.

Sdílená paměť (anglicky shared memory) má tyto části:

- Shared buffer je největší částí sdílené paměti. Výsledek každého dotazu je načten do shared_buffers. Shared buffers jsou paměti datových stránek. Vyjimku tvoří dočasná paměť, která není sdílená – temp_buffer.
- Wal⁷ buffers – paměťová oblast, která slouží k ukládání změn. Změny z wal bufferu jsou zapisovány do wal souborů (logovacích souborů).
- Clog buffers – je zkratkou pro commit log. Paměť drží informaci o transakcích záznamů. Paměť spolupracuje s procesem VACUUM.
- Lock space – chrání sdílené paměťové bloky proti zápisům od ostatních sezení. V Lock space se používá mechanismus známý jako „lightweight“ zámek. Tabulka, které se týká daný zámek, používá „heavyweight“ zámek.
- Další prvky paměti jsou: work mem; maintenance work mem a sys cache. (Shaik, 2013)

Nesdílené bloky paměti má každý klient. Klient za pomoci svého procesu přistupuje do této části paměti (tzn. procesy na pozadí).

⁷ Wal je anglická zkratka pro Write-Ahead Logging

Základní procesy:

- Postmaster – řídí přístup uživatelů k datům; startuje instanci; připojuje nové uživatele a vytváří jim nesdílenou paměť a proces.
- Checkpointer – zařizuje checkpointy.
- Wal proces – zapisuje změny z Wal bufferu do Wal logů.
- Writer proces – zapisuje změny do datových souborů.
- Autovacuum launcher proces – spouští Vacuum.
- Stats collector – stará se o sběr statistiky. Aktuální statistiky jsou dávají podklad Optimizéru. (Shaik, 2013)

Soubory a složky dostupné v PGDATA složce:

Složky:

- Base – složka pro každou databázi
- Global – informace o klasteru
- PG_xlog – složka s Wal soubory
- Pg_tblspc – symbolický link na tablespace lokaci
- Pg_clog – transakční status komitovaných dat
- Pg_multixact – podsložka obsahuje multitransakční data (používají se pro sídlené zámky řádků)
- Pg_notify – záznam o LISTEN/NOTIFY stavech dat
- Pg_sat_tmp – dočasné soubory pro subsystém
- Pg_subtrans – data o subtransakcích
- Pg_twophase – statusy souboru pro připravené transakce

Konfigurační soubory:

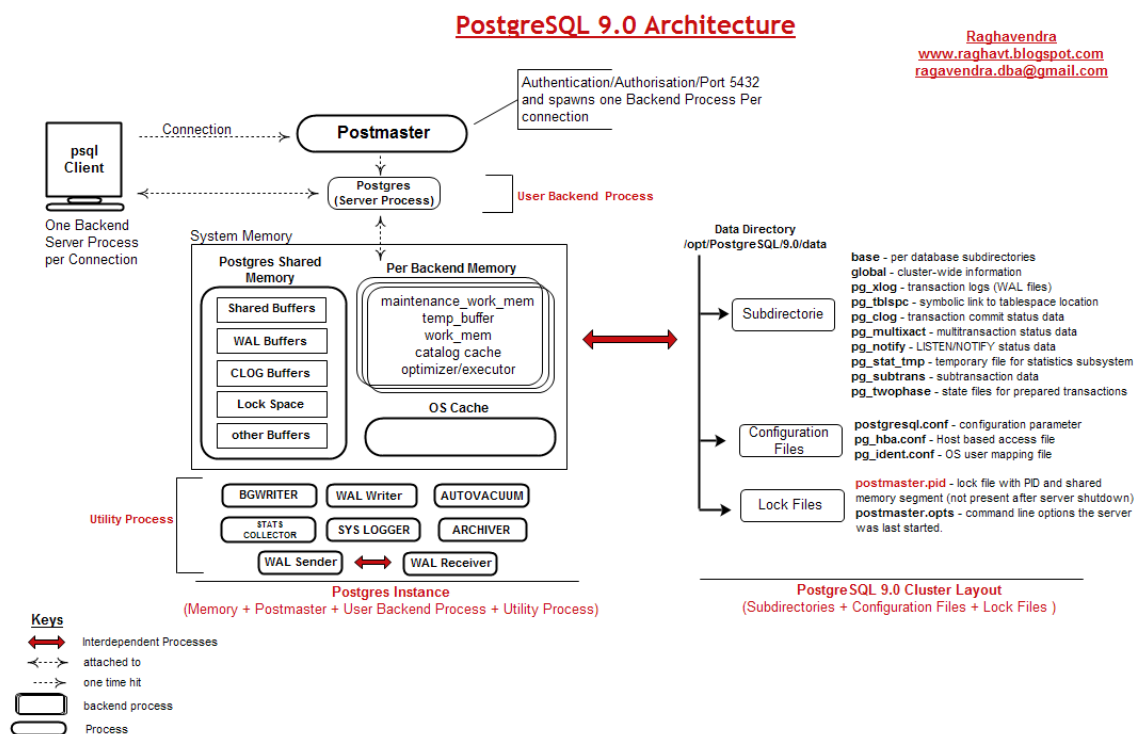
- Postgresql.conf – soubor obsahuje parametry databáze
- Pg_hba.conf – soubor specifikuje právo přístupu k databázi
- Pg_ident.conf – soup

Zámky:

- Postmaster.pid – soubor obsahuje data potřeba ke správě databáze, jako jsou čísla procesu, odkaz na paměťové bloky, IP adresu atd.

- Postmaster.opts – soubor má záznamy o parametrech, se kterými byla databáze spuštěna
- Postgresql.auto.conf – udržuje informace o změněných parametrech pomocí příkazu ALTER SYSTÉM (PostgreSQL 9.4.1 Documentation, 2014)

Pro vytvoření nové databáze se používá vestavěná databáze temp1, která předává základní tabulky. Nová databáze je vytvořena spolu s 60 základními tabulkami.

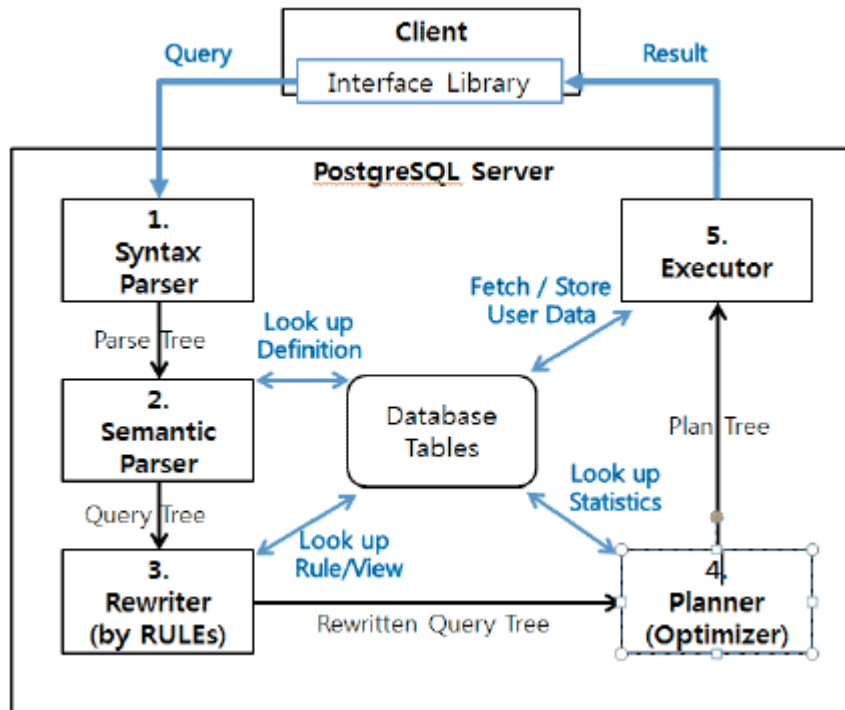


Obrázek 6 Architektura PostgreSQL 9.x [zdroj: Shaik Baji, 2013]

2.3.4. Spuštění příkazu

Klient se připojí k databázi pomocí psql nebo web server ovladače. Připojuje se přes TCP/IP na port (základním portem je 5432) na svoji databázi přes databázový systém. Postmaster proces poslouchá na portu např. 5432 a ve chvíli, kdy přijde požadavek na připojení k databázi, Postmaster přečte požadavek – kam se klient chce připojit. Zde přichází provedení autentifikace a autorizace. Postmaster přečte požadavek a připojí sezení k nově vytvořenému server procesu a nesdílené paměti. Jakmile server proces propojí, tedy

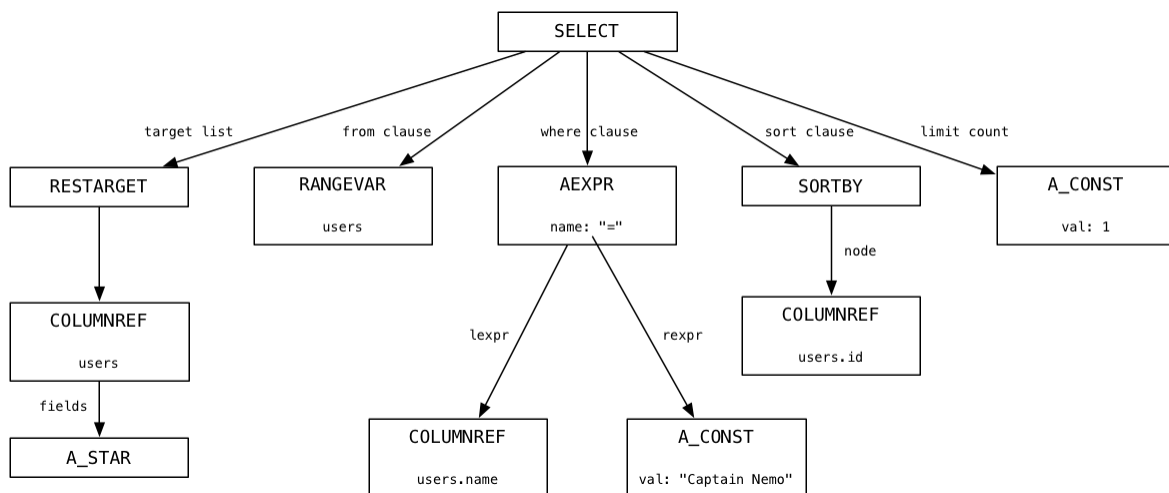
klienta s databází (připojení pracuje napřímo), postmaster se přestane angažovat. Počet server procesů může být až 1024, pokud to OS povolí. (Shaughnessy, P., 13. 11. 2014)



Obrázek 5- Kroky PostgreSQL při spuštění dotazu [Kim Sung Kyu, 14. 05. 2013]

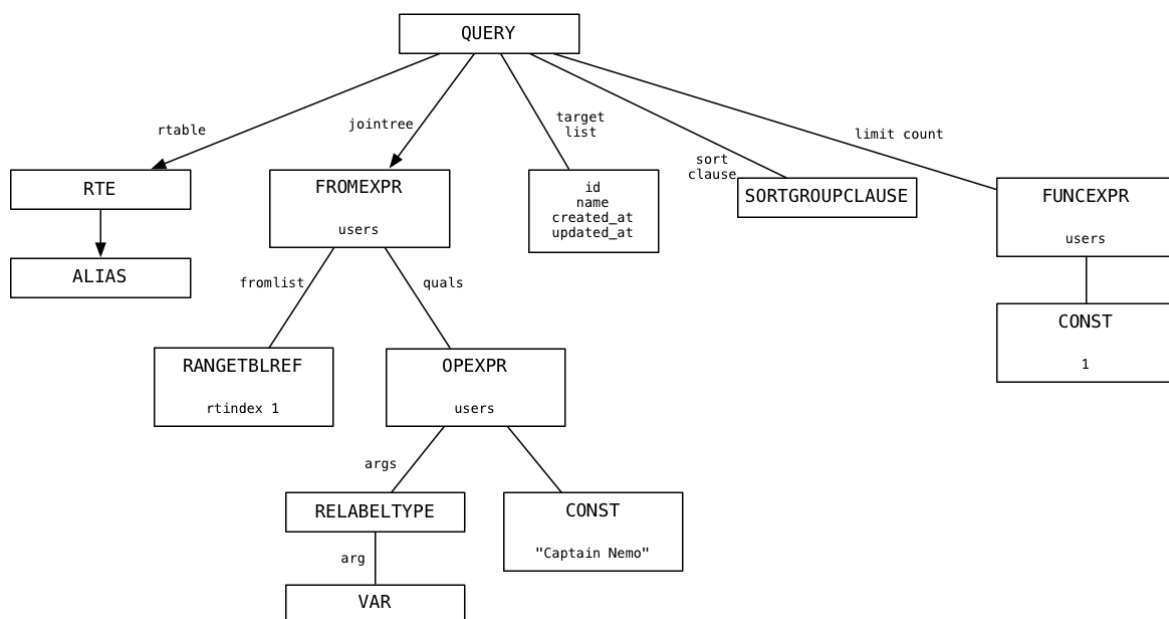
Na obrázku 5 je uveden proces spuštění příkazu (dotazu). Klient nejprve zadá pokyn. První krok – syntax parser – zkontroluje syntax příkazu a pokud příkaz je v pořádku, je vytvořen Parse Tree. Na parsování je použita technologie Bison⁸. Parse Tree je vytvořen na základě pravidel. Pokud by dotaz vypadal například: `select * from users where name = 'Catain Nemo' order by id asc limit 1` - výsledek Parse tree je zobrazen na obrázku 6 (Shaughnessy, 13. 11. 2014)

⁸ BISON - <http://www.gnu.org/software/bison/>



Obrázek 6 Parse Tree [Shaughnessy, P., 13. 11. 2014]

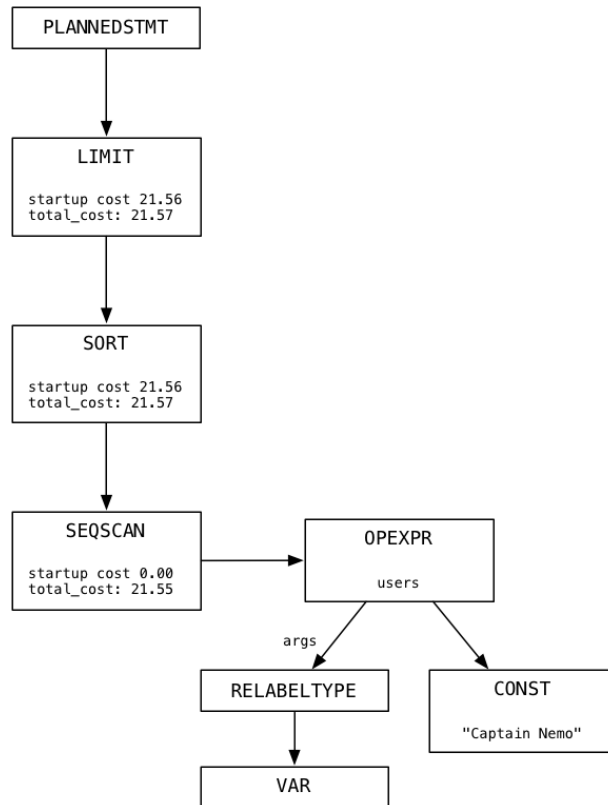
Z Parse Tree PostgreSQL přegeneruje na query tree s pomocí C funkce `pg_analyze_and_rewrite`. Obrázek 6 je toho výsledkem. (Shaughnessy, 13. 11. 2014)



Obrázek 7 query tree diagram [Shaughnessy, P., 13. 11. 2014]

Poslední krok před spuštěním dotazu je vytvoření plánu dotazu – obrázek 7. Plán dotazu je vytvořen pomocí funkce `pg_plan_queries`. Těmito třemi kroky PostgreSQL přepsal a optimalizoval dotaz, který dokáže najít hledaná data. Spuštění dotazu je na základě vytvořeného plánu. Jakmile PostgreSQL prohledává tabulku USERS, prohledá jí celou bez ohledu na to, že hledáme jen první řádek, který záznam obsahuje. To je docela neefektivní

postup, protože při představě velikosti tabulek, které má Facebook nebo na v našem případě Instagram, půjde o velmi pomalý postup. (Shaughnessy, 13. 11. 2014) Z tohoto odvozují, že právě tento důvod vývojáři v Yahoo změnili řádky za sloupce. (Cunningham, 28. 05. 2008) a tímto dosáhli mnohem větší rychlosti při čtení dotazů.



Obrázek 8 - Plan [<http://patshaughnessy.net/assets/2014/10/13/plan-tree.png>]

3. Analytická část

V předchozí části jsem představil dva databázové systémy. Podle zdrojů se u nich prokázala různorodost a zacílení na různé trhy. V této kapitole ověříme tvrzení a učiníme závěr.

Ověření proběhne na MySQL verze 5.6. a PostgreSQL Advanced Server verze 7.4 od EnterpriseDB. EnterpriseDB byla zvolena z důvodu rozšíření své kompatibility s Oracle databázovým systémem a také z důvodu podpory. V rámci instalace byla zvolena kompatibilita s Oracle databázovým systémem.

V analytické části budu srovnávat oba databázové systémy na základě těchto kritérií:

- Bezpečnost
- Cena administrace databáze (DBA cost)
- Škálovatelnost
- Migrace z Oracle databáze
- Nástroje pro práci
- Rozšířené schopnosti databázového systému
- Rychlost
- TCO – Total Cost of Ownership
- Nástroje pro správu

Každá podkapitola obsahuje tři sekce. První sekcí popisuje kritérium a jeho význam pro hodnocení. Druhá sekce ukazuje rozdíl mezi porovnávanými databázemi. Ve třetí sekci jsou ohodnoceny řešení obou databázových systémů na stupnici od 1 do 10, kde 10 je nejvyšší. Základní hodnocení je 5. Takové hodnocení znamená, že kritérium bylo splněno uspokojivě. A migrace z Oracle databázového systému neznamenevá žádný závažný problém.

3.1. Určení váhy kritérií

Váhy kritérií byly vytvořeny na základě Saatyho metody vícekritériální analýzy. V tabulce 3. Důležitost je řazena sestupně – od 1 do 9.

Dle tabulky 4 je hlavním kritériem je „Rozšířené schopnosti databázového systému“, následující kritérium je „Migrace z Oracle databáze“, které na sebe navazují. TCO je hlavní důvod, proč vznikla myšlenka na migrace z Oracle databázového systému. Proto je kritérium zapsáno v horní polovině. Poslední dva důležité kritéria jsou „Nástroje pro práci“, které umožňují administrátorovi efektivně spravovat databázový systémem respektive podporu aplikace. Posledním důležitým kritériem je „Bezpečnost“.

Název	Zkratka	Důležitost
Rozšířené schopnosti databázového systému	V1	1
Migrace z Oracle databáze	V2	2
TCO – Total Cost of Ownership	V3	3
Nástroje pro správu	V4	4
Bezpečnost	V5	5
Cena administrace databáze (DBA cost)	V6	6
Škálovatelnost	V7	7
Nástroje pro práci	V8	8

Tabulka 4 - Kritéria řazená dle důležitosti [zdroj: autor]

Tabulka 5 představuje jednotlivé váhy kritérií. Každá váha má svůj geometrický průměr, který bude sloužit pro výsledné srovnání.

	V1	V2	V3	V4	V5	V6	V7	V8	Geometrický průměr
V1	1	2	3	4	5	6	7	8	3,76
V2	1/2	1	2	3	4	5	6	7	2,66
V3	1/3	2/3	1	2	3	4	5	6	1,89
V4	1/4	1/2	3/4	1	2	3	4	5	1,35
V5	1/5	2/5	3/5	4/5	1	2	3	4	0,99
V6	1/6	1/3	1/2	2/3	5/6	1	2	3	0,74
V7	1/7	2/7	3/7	4/7	5/7	6/7	1	2	0,58
V8	1/8	1/4	3/8	1/2	5/8	3/4	7/8	1	0,47

Tabulka 5 - Váha kritérií dle Saatyho metody [zdroj: autor]

3.2. Bezpečnost

Bezpečnost aplikace pro bankovníctví je jedna z hlavních kritérií. Data chrání programy, metodiky a správci databázových systémů.

MySQL

Autentifikaci připojení k databázovému systému se potvrzuje pomocí jména a hesla, SHA-256 klíčem, PAM modulem, Socket Peer-Credential autentifikací. Nativní je pouze jméno heslo. Ostatní jsou pluginy. S pomocí PAM modulu lze ověřit připojení i pro LDAP. Plugin PAM je v komerční verzi. Percona. Ale vydala svůj plugin. Lze použít ten. MySQL podporuje i šifrování spojení SSL.

Pro ztížení útoků lze od verze 5.6 použít `validate_password` plugin. Ten zajistí nastavení komplexity hesla a jeho délku platnosti.

Role v MySQL neexistují. Všechny práva jsou dávány přímo uživatelům. Právo pro manipulaci s tabulkami, procedurami atd. jsou standardní – `SELECT`, ..., `CALL`.

Pro auditování lze použít celou řadu nástrojů – například od McAfee nebo od Oracle pod jménem MySQL Enterprise Audit. Základní auditování lze ukládat do logu nebo do tabulek. V případě logu se zapisuje v XML formátu. Zapisuje se připojení uživatelů, DML a DLL operace.

PostgreSQL

Autentifikace připojení uživatele lze provést pomocí jména a hesla anebo adresy serveru, LDAPu, PAMu, Kerberosu, SSL certifikátem, Ident server autentifikace a UNIX doménového socketu. Proti poslouchání na síti je možné komunikovat s pomocí SSL.

Heslo je uloženo pomocí MD5. Jestliže chceme zjistit, jaké heslo použil náš kolega, můžeme tímto příkazem zkusit napodobit: `„select 'md5'||md5('user_one') as "pokus", passwd as "originální heslo" from pg_shadow where username='user_one';“`. Sílu hesla lze nastavit pomocí modulu `passwordcheck`, který je nutné dodatečně povolit. Důvodem pro komplexitu hesla je ochrana proti brutálforce útokům (Clarke, 2012, 519 s.) Jeho používání se nedoporučuje, protože např. Kerberos má mnohem lepší ochranu.

Na výběr jsou dvě základní vlastnosti funkce, které developer může v rámci PostgreSQL vytvořit – důvěryhodná (angl. `trusted`) a nedůvěryhodná (angl. `untrusted`). Rozdíl mezi těmito funkcemi je, že `untrusted` mají možnost komunikace s OS, takže v krajním případě mohou formátovat disk nebo se pokusit o spojení s dalšími PC. Používání `untrusted` funkce je potřeba hlídat a omezit je jak nejvíce to jde.

PostgreSQL má role, do kterých lze přidávat oprávnění a role pak dává jednotlivým uživatelům. PostgreSQL má schopnost omezovat přístup do jednotlivých schémat a k jednotlivým objektům – nastavením práva `SELECT`, `UPDATE`, `DELETE`,

INSERT a ALL. Fine-Grain access neboli možnost omezit čtení řádků pro jednotlivé uživatele (Oracle tuto funkcionalitu nazývá virtual private database) bude přístupné od verze 9.5. PostgreSQL Advanced server však nabízí taktéž tuto technologii, ale pro implementaci se musí nastavit SE Linux.

Audit je zajištěn pomocí rozšíření pgaudit nebo vlastní implementací pomocí triggerů nad tabulkami. Pgaudit zapisuje do logů. Každý objekt v databázi má vlastní log. Loguje se připojení, DML operace a DDL operace.

EnterpriseDB vytvořila nástroj jménem SQL/Protect. Nástroj zabraňuje útočnickům škodit, například pomocí SQL Injection. Chování aplikace může být nastaveno neaktivní nebo učí se režim. Záznamy o své činnosti zapisuje k pozdější analýze. Je propojen s Postgres Enterprise Managerem, pro posílení bezpečnosti se na Linuxových platformách může použít SELinux.

Vyhodnocení

K ověření připojení je nejlepší PostgreSQL, MySQL nutí koupit uživatele komerční licenci, aby mohl použít PAM autentifikaci. V nastavení vlastností hesel je evidentně lepší MySQL. Nakonfigurovat komplexnost hesla a jeho délku platnosti zabere mnohem méně času uživateli než v PostgreSQL. PostgreSQL spoléhá na externí ověřování a jejich vlastní nastavení. Taktéž audit je lepší u MySQL. Chybějící role, budoucí vývoj a žádné další rozšíření je důvodem, aby vyhrál PostgreSQL – 7, MySQL 4.

<http://www.ibm.com/developerworks/opensource/library/os-postgresecurity/index.html>

3.3. Vysoká dostupnost

Zajištění vysoké dostupnosti – nejlépe 99.99% je něco, co si přeje každý manager. Zbytečné odstávky nebo závady snižují výnos a spokojenost zákazníků se službou. Bankovní sektor si nemůže dovolit dlouhodobý výpadek a národní regulátoři žádají banky, aby měly vytvořeno řešení pro případ výpadku. Ne všechny banky toto řešení mají. Není tedy podmínkou.

Vysokou dostupnost lze zajistit pomocí HW a SW. Vysoká dostupnost může být active-active, active-pasive a acitve-cold. Ve všech jmenovaných řešeních je aktiv první server. Tedy ten, na který přistupují klienti. Existují dvě možnosti jak implementovat active-active

řešení. První je pomocí sdíleného disku. Druhé řešení disk nesdílí. Obecně je první řešení mnohem rychlejší a vzhledem k faktu, že ho neimplementovalo mnoho firem, je i poměrně těžší na vytvoření. Active-cold řešení má druhý server vypnutý. Druhý server může být nastaven, aby v případě výpadku se nastroval a nahradil první server. Takové nastavení umožňují například Windows OS nebo Red Hat. Active-pasive má oba servery zapnuté. Na druhém serveru je databáze, která přijímá data a ty aplikuje. Většinou se takovému řešení říká replikace – v Oracle světě je to dataguard.

Doplňující technikou je přestavěný server, který slouží jako interface. Klienti jsou k němu připojeni. Tento server zpravuje jejich požadavky a poté je posílá na databázové servery.

MySQL

Pro MySQL existují tři řešení vysoké dostupnosti. Nejedná se o celkový seznam, protože existuje více produktů replikace, než jsou zde uvedeny:

- Master-slave replikace. Jednoduché na implementaci. Data ze zdroje jsou přenášena a aplikována na její kopii. Nevýhodou tohoto řešení je replikace probíhá jen jednou cestou. Použít můžeme xtrabackup od Percony⁹
- NDB Cluster (MySQL Cluster), který lze zakoupit s edicí MySQL Cluster Carrier Grade Edition. Share nothing je architekturou, kterou je vytvořen. Data jsou držena (uchovávaná) v paměti. V intervalech probíhá jejich ukládání na disk. Mezi nody probíhá synchronní replikace. Commit je dvoufázový. Skládá se ze serveru, který je určen pro data, dále serveru pro SQL, a ještě je nutný i management servery. Doporučeny jsou alespoň čtyři servery. Výhodou je běžný HW – tzn. SAN není potřeba.
- Galera cluster může být kombinován s MySQL, MariaDB a Percona serverem. Na rozdíl od NDB Clusteru je architektura dosti jednodušší. Galera neobsahuje ani management server a ostatní servery jsou vloženy do jednoho serveru. Opětovně se jedná o synchronní replikaci. Opět je zde dvoufázový commit. Galera může být spuštěna i mezi dvěma datacentry. Nevýhoda Galera clusteru jsou velké frekvence commitů a také velké mazání dat.

⁹ Více i o řešení od Percony je na adrese http://www.percona.com/doc/percona-xtrabackup/2.2/howtos/setting_up_replication.html

PostgreSQL

Pro PostgreSQL neexistuje clusterové řešení – active-active. Pro Postgres lze použít replikaci – řešení active-pasive – nebo active-cold. Replikace může být logická (za pomoci Logical Streaming repliacation, jenž je dostupná od verze 9.4) a fyzická. U fyzické replikace se aplikují na standby databázi záznamy z WAL souborů. Logická replikace je výhodná z důvodu možnosti mít standby databázi otevřenou pro čtení. Lze jí použít k vytváření reportu bez toho, aby se zatížil primární server. Docílit vysoké dostupnosti lze s pomocí Linux HA nebo Windows řešením.

PostgreSQL Advanced server používá i řešení založené na xDB Replication. Řešení umožňuje multi-master replikaci. Doporučené jsou až tři instance PostgreSQL Advanced serveru. (xDB Replication FAQs, 2015)

Vyhodnocení

Vzhledem k technologii active-active clusteru je MySQL vítěz. PostgreSQL nemá tuto technologii a není ani v implementačním plánu. Jako svoji náhradu používá multi-master replikaci.

MySQL 4, PostgreSQL 3.

3.4. Přeskolení stávajících zaměstnanců

Pro poskytování prvního a druhého stupně podpory musí firma buď zaměstnat nové zaměstnance nebo investovat peníze do přeskolení současných zaměstnanců. Vyjdeme-li z podmínek firmy BSC Praha, spol. s r. o., máme zaměstnány dva seniorní zaměstnance pro podporu Oracle databázových systémů. Zaměstnanci jsou ochotní podporovat databázové systémy PostgreSQL nebo MySQL a není nutné přijmout nového zaměstnance.

Ústní školení má být v Českém jazyce – tím je vyloučeno školení přes internet, ty jsou nabízeny pouze v anglickém jazyce. Školení by mělo naučit zaměstnance základní a pokročilé administraci. Administraci se myslí instalace a konfigurace databázového systému, zálohování a migrace dat, optimalizace SQL operací atd.

MySQL

Pro MySQL jsem našel školení u firmy Gopas. Přesně spňuje očekávání. Školení a jeho cena je vyspána v tabulce 6. Celkova cena školení je 65.703,- Kč s DPH. Každé školení ma více kurzů, takže je pro firmy vhodné, aby oba administrátoři absolvovali školení v jiném termínu už z důvodu vysoké absence.

ID	Jméno	Dnů	Cena vč. DPH	Cena bez DPH
GOC661	Administrace MySQL	5	23595	19500
GOC663	Jazyk SQL pro pokročilé v MySQL	3	16335	13500
GOC664	Optimalizace SQL dotazů v MySQL	2	9438	7800
GOC665	Pokročilá administrace MySQL	3	16335	13500
	Suma	13	65703	54300

Tabulka 6-Školení pro MySQL [zdroj: autor]

PostgreSQL

Nabídky ohledně tohoto školení na trhu jsou minimální. V současné době jsem na internetu objevil pouze jedno školení a to u firmy O.K. Systems. Za toto pětidenní školení Administrace PostgreSQL účtují 12 342 Kč s DPH na účastníka kurzu. Druhá možnost je si pozvat Pavla Stěhuleho na školení do firmy. Ten si za každý den účtuje 12 tisíc Kč s DPH. Školení je pro 4 osoby s tím, že počet zaměstnanců se dá ještě navýšit. Jeho účtovaná částka je relativně vysoká vůči školení od firmy O.K., ale je zde záruka, že školení provede největší propagátor na PostgreSQL v ČR. Ze stránek <http://postgres.cz/wiki/%C5%A0kolen%C3%AD> nelze zjistit počet dní, která jsou pro na školení vyčleněny. Osobně si myslím, že školení – kompletní se dá zvládnout za 10 dní. Pesimistický odhad je 14 dní. Vezmu-li delší variantu, pak výsledná částka je výsledná 168 tisíc korun s DPH.

Jako další možnost proškolení zaměstnanců je samo-vzdělávání. Většinou z knih a internetu. Co se odborných knih týká, má MySQL mnohem lepší. Pročítáním PostgreSQL jsem se mnohdy dozvěděl úplné základy o SQL. Obecně je na internetu více lidí, kteří píší blogy o MySQL – google má přibližný počet výsledku na heslo „blog mysql“ 35,9 milionů - PostgreSQL se stejným dotazem jen 2,7 milionů. Expertní podpora komunity je spíše na straně PostgreSQL.

Výsledná cena za školení MySQL pro dva lidi je 131.406 Kč s DPH oproti 168.000,- Kč s DPH.

MySQL 7, PostgreSQL 5.

3.5. Možnosti databázového systému

Tato kapitola je jedná z nejdůležitějších. Některé položky se nahradit dají, jiné jen za změnu kódu, některé vůbec – například SELECT without table. Přesto pokud by zvolená databáze neumožňovala potřebnou funkcionalitu, musela by se aplikace přepsat. S přepsáním aplikace v drobnostech se počítá, ale nesmí dojít k tomu, aby existovalo víc verzí aplikací – každá pro svůj databázový systém. Aplikace v dnešní době používají Spring nebo Hibernate¹⁰, jejichž výhodou je právě multiplatformnost.

MySQL i PostgreSQL umožňují partitioning. EnterpriseDB ersp. PostgreSQL používá spojování menších tabulek s hlavní tabulkou. Oproti Oracle databázovému systému je těžkopádné řešení.

Tabulka 7 zachycuje rozdílné možnosti mezi MySQL a PostgreSQL servery. Všechny shodné záznamy byly smazány, aby tabulka nebyla dlouhá. Záznamy u niž se objevuje závorka, databáze umožňuje, ale s výhradou. Funkce, které nejsou podporovány – například windows functions – byla zatržena jako ne i přesto, že existuje možnost jí docílit.

¹⁰ Frameworky na aplikačním serveru vytváření SQL dotazy na základě Java kódu a informace o použité databázi

Vlastnosti	Postgres	MySQL
Queries		
Window functions	Ano	Ne
Common Table Expressions	Ano	Ne
Recursive Queries	Ano	Ne
Filtered aggregates	Ano	Ne
GROUP BY .. ROLLUP	Ne	Ano
SELECT without table	Ano	Ne
Tuple updates	Ano	Ne
Query variables	Ne	Ano
Regular Expressions	Postgres	MySQL
Substring	Ano	Ne
Replace	Ano	Ne
Constraints	Postgres	MySQL
Deferred constraints	Ano	Ne
Check constraints	Ano	Ne
Check constraints using custom functions	Ano	Ne
Exclusion constraints	Ano	Ne
Statement based constraint evaluation	Ano	Ne
Indexing	Postgres	MySQL
Partial index	Ano	Ne
Descending Index	Ano	Ne
Index on expression	Ano	Ne
Index using a custom function	Ano	Ne
Clustered index	Ne	Ano
DML	Postgres	MySQL
Writeable CTEs	Ano	Ne
TRUNCATE table with FK	Ano	Ne
Read consistency during DML operations	Ano	Ne
MERGE support	Ne	Ano
SELECT .. FOR UPDATE NOWAIT	Ano	Ne
RETURNING clause	Ano	Ne
Data Types	Postgres	MySQL

User defined datatypes	Ano	Ne
Domains	Ano	Ne
Arrays	Ano	Ne
IP address	Ano	Ne
BOOLEAN	Ano	Ne
Interval	Ano	Ne
Range types	Ano	Ne
DDL	Postgres	MySQL
Transactional DDL	Ano	Ne
Functions as column default	Ano	Ne
Sequences	Ano	Ne
Cascading DROP	Ano	Ne
DDL Triggers	Ano	Ne
TRUNCATE Trigger	Ano	Ne
Custom name for PK constraint	Ano	Ne
ALTER a table used in a view	Ne	Ano
Add table column at specific position	Ne	Ano
Programming	Postgres	MySQL
Table functions	Ano	Ne
Custom aggregates	Ano	Ne
Function overloading	Ano	Ne
User defined operators	Ano	Ne
Statement level triggers	Ano	Ne
Dynamic SQL in functions	Ano	Ne
Dynamic SQL in triggers	Ano	Ne
Delete triggers fired by cascading deletes	Ano	Ne
Built-in scheduler	Ne	Ano
Views	Postgres	MySQL
Triggers on views	Ano	Ne
Views with derived tables	Ano	Ne
JOINs and Operators	Postgres	MySQL
CROSS JOIN	Ano	Ne

FULL OUTER JOIN	Ano	Ne
LATERAL JOIN	Ano	Ne
INTERSECT	Ano	Ne
EXCEPT	Ano	Ne
ORDER BY ... NULLS LAST	Ano	Ne
IS DISTINCT FROM	Ano	(Ne)
BETWEEN SYMMETRIC	Ano	Ne
OVERLAPS	Ano	Ne
Other	Postgres	MySQL

Schemas	Ano	Ne
NoSQL Features	Postgres	MySQL
JSON	Ano	Ne
Key/Value storage	Ano	Ne
Administration	Postgres	MySQL
User groups / Roles	Ano	Ne
High Availability	Postgres	MySQL
Multi-master replication	Ne	Ano

Tabulka 7 - Přehled vlastností MySQL a PostgreSQL [zdroj: dbms_comparison, upraveno]

Vyhodnocení

MySQL neumí sekvence – tato technika byla nahrazena sloupcem s vlastností autoincrement, deferred constraints, index on expression a full outer join. Také zde nejsou pole a vlastní datové typy. Problémem je hlavně deferred indexes a arrays (pole). Právě proto se o PostgreSQL tvrdí, že je: “the most Advanced database”.

Vyhrává zcela jasně PostgreSQL s 6 body. MySQL dostalo 3 bodů.

3.6. Migrace z Oracle DB

Tato podkapitola volně navazuje na předchozí podkapitulu. Podkapitola se zabývá možnostmi převedení aplikace z Oracle databázového systému. Kapitola je klíčová, protože pokud by se nepovedlo přenést objekty a data, můžeme migraci zcela škrtnout.

Zajistit převedení lze buď online pomocí streamovací technologie – například Oracle GoldenGate. Druhou možností je vygenerování DDL a DML příkazů (zaváděcí skript) a ten předělat pro novou databázi. Třetí variantou je nástroj, jenž konvertuje rozdíly automaticky.

Datové typy

Tabulka 8 zobrazuje základní datové typy MySQL, PostgreSQL a Oracle databázových systémů.

MySQL	Postgres	Oracle	Velikost
číselné datové typy			
BIGINT	BIGINT	NUMBER (19,0)	8 B
BIT	BIT	RAW	
DECIMAL(M,D)	DECIMAL	FLOAT(24), BINARY_FLOAT	
DOUBLE	DOUBLE PRECISION	FLOAT(24), BINARY_FLOAT, BINARY_DOUBLE	8 B
DOUBLE PRECISION	DOUBLE PRECISION	FLOAT(24), BINARY_DOUBLE	8 B
FLOAT(25<=X <=53)	REAL	FLOAT(24), BINARY_FLOAT	8 B
FLOAT(X<=24)	REAL	FLOAT, BINARY_FLOAT	4 B
INT	INT	NUMBER (10,0)	4 B
INTEGER	INTEGER	NUMBER (10,0)	4 B
MEDIUMINT	INTEGER, BIGINT	NUMBER (7,0)	3 B
NUMERIC	NUMERIC	NUMBER	variabilní
REAL	REAL	FLOAT(24), BINARY_FLOAT	8 B
Smallint	Smallint	NUMBER(5,0)	2 B
TINYINT	TINYINT	NUMBER(3,0)	1 B
datumové typy			
DATE	DATE	DATE	3 B
DATETIME	TIMESTAMP	DATE	8 B
TIMESTAMP	TIMESTAMP	DATE	4 B
TIME	INTERVAL	DATE	3 B
YEAR	NUMBER	NUMBER	1 B

datové typy			
BLOB	BYTEA	RAW, BLOB	variabilní
CHAR(m)	CHAR	CHAR	variabilní
ENUM			variabilní
LOB	BYTEA	RAW, BLOB	variabilní
LONGTEXT	TEXT	RAW, CLOB	variabilní
MEDIUMBLOB	BYTEA	RAW, BLOB	variabilní
MEDIUMTEXT	TEXT	RAW, CLOB	variabilní
SET			variabilní
TEXT	TEXT	VARCHAR2, CLOB	variabilní
TINYBLOB	BYTEA	RAW, BLOB	variabilní
TINYTEXT	TEXT	VARCHAR2	variabilní
VARCHAR(m)	VARCHAR(m)	VARCHAR2, CLOB	variabilní

Tabulka 8 - Datové typy [zdroj: autor]

Migrované objekty

Pro migraci byla vytvořena aplikace, která simuluje knihovnu. Obsahuje následující tabulky:

- autor: id, jmeno, prijmeni, narozeni
- ctenari: id, jmeno, prijmeni, RC, registrace,pobočka_id, stav
- knihy: id, jmeno_knihy, isbn, datum_vydani, autor_id
- knihy_pobočka: id, KNIHA_ID, pobočka_id, autor_id, pocet_knih, vypujceno_knih
- platby: id, ctenar_id, PLATBA, datum, duvod, stav
- pobočky: id, jmeno, ulice,mesto
- vypujcky: id, ctenar_id, kniha_id, pobočka_id, datum, stav
- zamestnanci: id, jmeno, prijmeni, pobočka_id, ctenari_id

Triggery detekují změny tabulek a mění data na jiných místech. Jejich popis je:

- Zaměstnanec si nesmí půjčit knihu
- Když pokuta, čtenář má změněn status na dlužníka
- Když se stane čtenář zaměstnancem, smažou se mu všechny poplatky

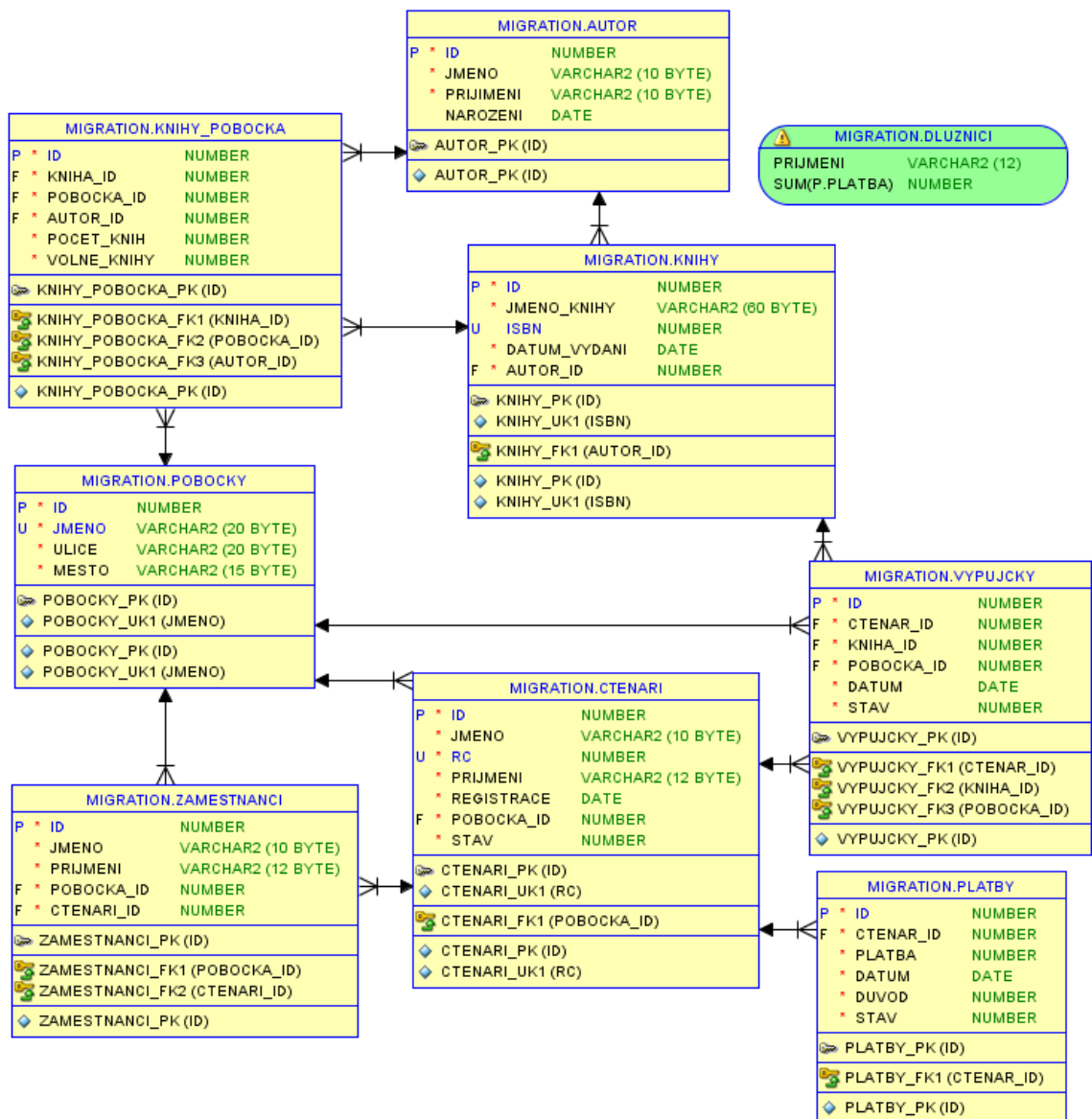
Procedury slouží k náhodnému vygenerování dat. Nepoužívají se žádné Oracle procedury. Pomocí nich se plní tyto tabulky:

- a) platby
- b) vypujcky
- c) zamestnanci - vyberou se z pracovníku
- d) knihy - pobočky

Tabulky obsahují konstrainty, primární klíče, sekvence. Jejich podobu lze vidět na obrázku Obrázek 5. Zelenou barvu zachycuje view Dlužníci, které je složeno z tohoto dotazu:

```
select * from (  
Select C.Prijmeni, Sum(P.Platba) From Platby P, Ctenari C  
Where P.Ctenar_Id = C.Id  
AND P.Stav = 1  
Group By C.Prijmeni  
Order By 2 Desc, 1 Asc)  
where rownum < 11;
```

Rownum se nepoužívá ani MySQL ani PostgreSQL. Join není dle ANSI SQL, ale podle Oracle metodiky.

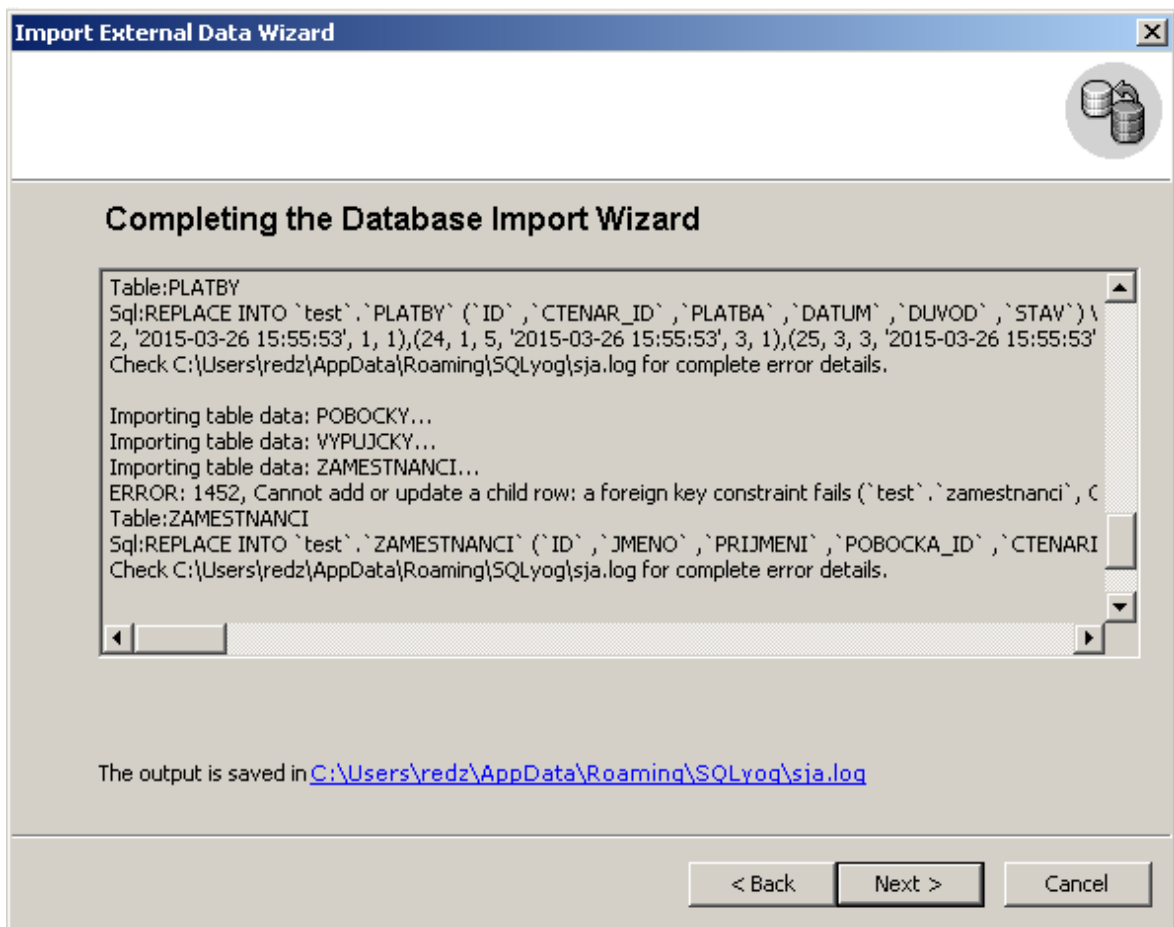


Obrázek 7 Logický model migrované aplikace [zdroj: autor]

MySQL

MySQL Workbench a MySQL migration toolkit jsou dva nástroje od Oracle Corp. Inc., které podporují migraci z jiných databázových systémů. Nepodporují však import z Oracle. Nebyl nalezen software, který by umožnil migraci objektů a dat a byl zdarma. Na základě recenzí byl vybrán SQLyog, který umožňuje 14 denní zkušební verzi. V ní však umožňuje přesunout pouze dvě tabulky v jednu chvíli. Proto byl SQLyog zakoupen.

Postup byl zvolen dle blogu <http://tkurek.blogspot.cz/2013/04/migrate-oracle-to-mysql.html>. U všech tabulek se musel změnit datový typ pro number. SQLyog se snažil použít float místo integer. Migrace však skončila katastrofou. View a materializované view byly převedeny na klasické tabulky. Nejedná se o chybu MySQL, které view podporuje, ale o chybu nástroje. Ani data tabulek nebyla převedena v pořádku. V rámci migrace se objevilo několik chyb, které lze vidět na obrázku Obrázek 9.



Obrázek 8 - Převod tabulek a pohledů pomocí programu SQLyog [zdroj: autor]

Bohužel došlo ke špatnému výběru migračního nástroje. Za předpokladu, že i další nástroje fungovali stejně, bylo nejjednodušší udělat migraci manuálně resp. s použitím vlastních skriptů.

Bez ohledu na použitý nástroj, MySQL nemá podpora PL/SQL. procedury a funkce se musí upravit manuálně, aby vyhovovali MySQL jazyku.

PostgreSQL

EnterpriseDB poskytuje migrovací nástroj, který lze nainstalovat při instalaci database. Proces migrace je popsán na stránce (eng) http://www.enterprisedb.com/docs/en/9.1/migrate/Postgres_Plus_Migration_Guide-04.htm#P48_4192. PostgreSQL nepodporuje Advanced queue, kompresi dat a specifické nastavení tabulek (pctfree, pctused) od Oracle. Pokud se vyskytne takový object, Postgres buď automaticky konvertuje funkcionalitu – případ nastavení tabulek – nebo ignoruje (komprese). Samozřejmě vždy napíše informaci do logu. Migrační nástroj lze ovládat pomocí přepínačů.

Enterprise Advanced server podporuje systémových procedur z Oracle (například dbms_output, dbms_crypto). Jejich výčet je uveden na stránce (eng) http://www.enterprisedb.com/products-services-training/products/postgres-plus-Advancedd-server?quicktabs_Advancedservertab=3#packages. Také má běžné pohledy jako například v\$session a obsahuje všechny pohledy all_%, dba_%, user_%. SQL obsahuje podporu nvl, decode a podporu date/time funkcí. Tím pomáhá zachovat logiku, která je ukrytá v procedurách a package. Podpora buld-in procedur a dalších prvků je dostatečně velká, aby se zde dala přenést aplikace, která má logiku v PL/SQL jazyce.

Výsledek přenesení schématu:

Sequences: 6 out of 6

Tables: 8 out of 8

Constraints: 28 out of 28

Triggers: 3 out of 3

Views: 0 out of 1

Procedures: 3 out of 3

Users: 1 out of 1

Total objects: 50

Successful count: 49

```
Failed count: 1
Invalid count: 0
```

Problém nastal v případě view jenž bylo označeno chybou, nevznikl však v kritických místech, které byly vyznačeny. Postgres nepoznal výstup ze subselectu:

```
Creating View: DLUZNICI
MTK-15008: Error Creating View: DLUZNICI
DB-42703: ERROR: column "SUM(P.PLATBA)" does not exist at
position 70
-- CREATE VIEW DLUZNICI (PRIJMENI,"SUM(P.PLATBA)") AS
-- Line 2: select PRIJMENI,"SUM(P.PLATBA)" from (
--                                     ^
```

View stačilo přepsat a opětovná migrace proběhla úspěšně:

```
create view DLUZNICI as
select PRIJMENI, sum
from (Select C.Prijmeni,Sum(P.Platba) as sum
From migration.Platby P, migration.Ctenari C Where
P.Ctenar_Id = C.Id AND P.Stav = 1
Group By C.Prijmeni
Order By 2 Desc,1 Asc)
where rownum < 11;
```

Dalším testem bylo přenesení materializovaného view, které se nepřeneslo z důvodu nepodporování refreshované metody ‘force’. Po upravení metody na ‘complete’ bylo materializované view úspěšně přeneseno na PostgreSQL.

Vyhodnocení

V tomto testu PostgreSQL zvítězil, na základě testu přesunu objektů a dat z Oracle databázového systému. EnterpriseDB poskytuje připravený nástroj pro migraci z jiných databází. MySQL má svůj migration toolkit taktéž, ale migraci z Oracle databázového systému nepodporuje. Koupený nástroj pro MySQL nefunguje.

Druhým důvodem je podpora PL/SQL resp. kompatibilita Postgresu s Oracle databázovým systémem. Na základě tohoto testu mohu doporučit migrovat aplikaci na MySQL, která používá databázový systém pouze jako úložiště dat. EnterpriseDB Advanced Server lze doporučit jako opravdovou náhradu.

MySQL 4, PostgreSQL 9

3.7. Nástroje pro práci

Kategorie nazvaná 'Nástroje pro práci' je doplňkovým kritériem, které je nejméně důležité. Nástroje pro správu se rozdělují do tří kategorií – GUI klient, nástroj pro správu, nástroj pro modelování a monitoring.

První kategorie je schválně pojmenovaná s GUI, aby se jednalo grafický program. Samozřejmě lze pracovat i s příkazovým řádkem, ale pro lepší přehlednost se hodí klient s uživatelským rozhraním. Kritériem pro klienta je doplňování a dokončování SQL příkazů. Druhým požadavkem je, aby cena byla pokud možno zdarma.

MySQL

Seznám klientů včetně drobného hodnocení (eng) pro MySQL lze nalézt na adrese <http://www.databasejournal.com/features/mysql/article.php/3880961/Top-10-MySQL-GUI-Tools.htm>.

Vhodným klientem je SQL Workbench/J, jenž je kompatibilní i s PostgreSQL a dalšími databázovými systémy. Klienta zdarma bylo velmi těžké najít.

Základní administrační nástroje jsou phpMyAdmin pro MySQL. Jde o webového klienta. PhpMyAdminu konkuruje Adminer, jenž lze použít pro administraci PostgreSQL taktéž. Prozatím nemá takovou popularitu, ale vzhledem k jeho výčtu popsaných schopností a jeho zastoupení v některých balíčcích WAMP, může jít o vhodný produkt. Srovnání Admineru a phpMyAdminu je na stránce <http://www.adminer.org/cs/phpmyadmin/>.

Nástrojem pro vytváření, změnu a design tabulek pro MySQL je velmi oblíbený MySQL WorkBench. Je zdarma ke stažení. Je velmi intuitivní, rychlý a jednoduchý. Vytvořené tabulky lze vygenerovat v SQL formátu.

Monitorovacím nástrojem, jenž se používá skrz webové rozhraní je MySQL Enterprise Manger od firmy Oracle Corporation Inc. Je to placený software, který vychází z Oracle Enterprise Manageru. Umožňuje vytvářet reporty, nastavovat metriky upozornění, ukazovat graf zátěže HW, SW a databázového systému. V neposlední řadě umožňuje jejich správu. Může monitorovat jednu a více databázových prostředí.

PostgreSQL

Seznam klientů PostgreSQL je na adrese <http://www.postgresql.com/journal/index.php?/archives/133-Database-Administration,-Reporting,-and-Light-application-development.html>.

PgAdmin3 je nativním klientem pro PostgreSQL. Obě kritéria splňuje. Je rychlý, ale trochu nepřehledný. Hlavně z důvodu velké komplexnosti.

Základní administrační nástroj je pgAdmin3 pro Postgres. Oproti phpMyAdminu není webovým rozhraním. Jeho nevýhodou je mírná kostrbatost. Například při zadávání dotazů se spouští další okno, ale zase pro Administrační rozhraní umožňuje navíc nastavovat konfiguraci databázového systému. Z mého pohledu jde o velkou pomoc správcům databázových systémů. I Postgres má podobného klienta, který je pojmenován jako PhpPgAdmin.

Nástrojem pro vytváření, změnu a design tabulek pro PostgreSQL byl PostgreSQL Database Modeler. MySQL Workbench a PostgreSQL Database Modeler jsou vyrovnanými nástroji, co se svých možností a ovladatelnosti týče.

Monitorovacím ekvivalentem MySQL Enterprise Manger je Postgres Enterprise Manager od firmy EnterpriseDB. Podobnost jména není náhodná a je vidět, že se EnterpriseDB inspirovala u Oracle Corporation Inc. Oba jsou to úchvatné programy, které umožňují správu a monitorování. Jako nástroje přidali další možnosti – převážně se jedná o dashboardy s grafy a také umožňují sledovat HW a OS statistiky. Pro oba platí nutnost zakoupení této licence. Postgres Enterprise Manager lze provozovat 60 dní zdarma.

Výsledek

Porovnání je velmi těsné. Největší rozdíl vidím v klientovi. PgAdmin3 a je mnohem lepší co se svých vlastností a přehledu týče než PhpMyAdmin. Podobného klienta MySQL

nenabízí. V porovnání Enprise Manageru je MySQL o trochu dále, tedy co se týče jeho vlastností.

MySQL 5, PostgreSQL 4

3.8. Benchmark

Kapitola nazvaná Benchmark je o testování rychlosti databázových systémů. Výsledky testování nelze brát jako důkaz, že jedna databáze je skutečně rychlejší než druhá. Důvodem může být špatné nastavení. Případně nevyhovující test. Z tohoto důvodu problému s testem byl TPC – C test, který lze simulovat s pomocí aplikace HammerDB.

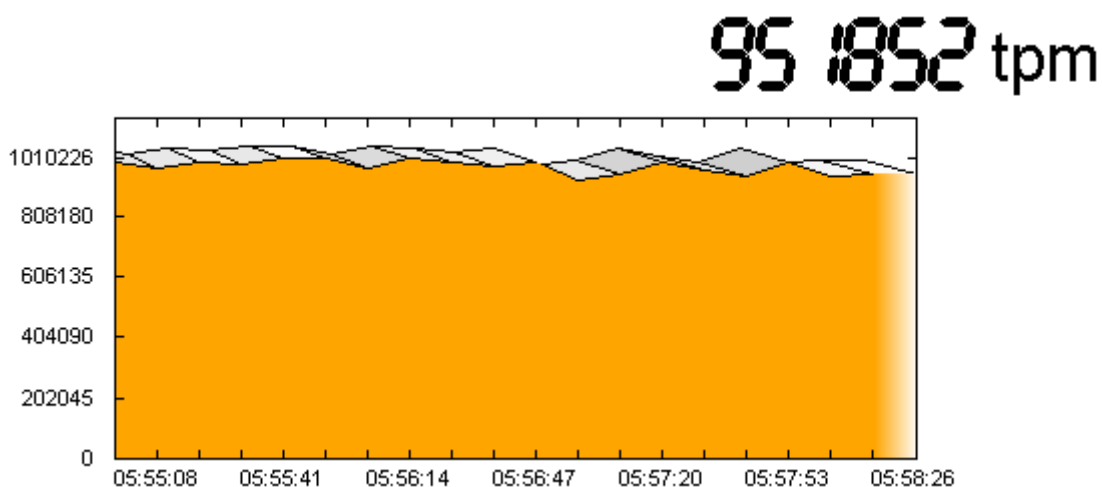
Pro srovnání výkonosti byl použit PC s nainstalovanými Windows 7 64 bit. PC má 8 GB RAM, SDD HDD a Intel® Core™2 Quad CPU Q66000 2.4 Ghz.

Test byl vytvořen jeden Warehouse. Pro test bylo vytvořeno 20 virtuálních uživatelů.

MySQL

Testování proběhlo s InnoDB. Ze základu jsem změnil innodb_buffer_pool_size na 2G a innodb_log_file_size na 200M.

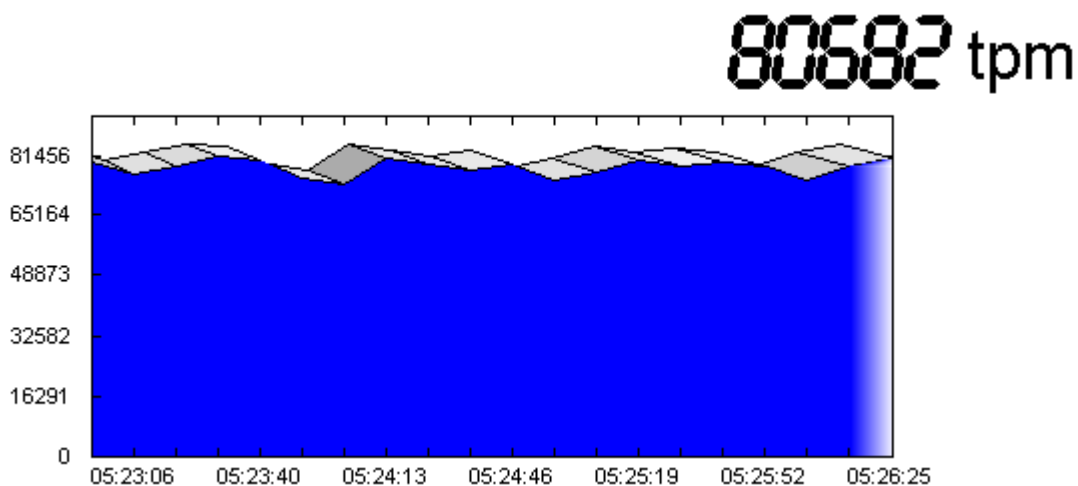
TPM je počet zpracovaných transakcí za minutu. Pro MySQL obrázku 8 je aktuální hodnota 95 tisíc. Nejvyšší hodnota byla přes milión transakcí. (<http://www.tpc.org/tpcc/detail.asp>)



Obrázek 9 - MySQL benchmark [zdroj: autor]

PostgreSQL

Nastavení dvou hlavních parameteru je `shared_buffers` bylo na 2 GB, `effective_cache_size` na 4 GB.



Obrázek 10 - PostgreSQL benchmark [zdroj: autor]

Vyhodnocení

Rozdíl výsledku testu si neumím vysvětlit. Bez ohledu na změnu parametrů (`shared_buffers` a `effective_cache_size`) PostgreSQL dopadnul 10 krát hůře. Dle jiných benchmarku byl MySQL o něco rychlejší, nikde to však nebylo 10x více. Z toho důvodu kategorie Benchmark nebude hodnocena.

3.9. Nástroje pro správu

Náplň databázového administrátora je udržet databázi bez chyb. K tomu mu stačí číst log a zajišťovat opravu chyb. Kontroluje místo na disku. Zajišťuje zálohování a obnovu databází. Další aktivitou je ladění výkonu.

V rámci této podkapitoly budou představeny jednotlivé nástroje pro MySQL a PostgreSQL. Pro exekuční plán byl použit dotaz:

```
EXPLAIN
SELECT x.prijmeni,
       x.sum
```



```

FROM ( SELECT c.prijmeni,
          sum(p.platba) AS sum
        FROM migration.platby p,
          migration.ctenari c
        WHERE p.ctenar_id = c.id AND p.stav = 1
        GROUP BY c.prijmeni
        ORDER BY sum(p.platba) DESC, c.prijmeni) x
LIMIT < 11;

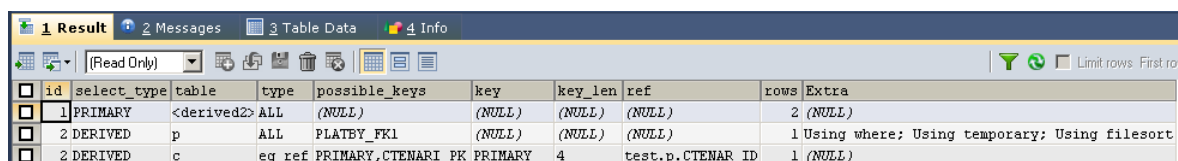
```

MySQL

Pro zajištění zálohování (fyzického) bylo použito Percona XtraBackup, který lze použít pro Perconu, MariaDB a MySQL. MySQL v Enterprise edici poskytuje vlastní nástroj MySQL Enterprise Backup. Hlavním důvodem je umožnění inkrementálních záloh. Toto nástroj od Oracle neumí a zálohování a obnova databáze je zde spuštěna pomocí příkazového řádku. Funkcionalita je popsána zde <http://www.percona.com/software/percona-xtrabackup>. Díky tomuto nástroji dokáže MySQL zálohovat za běhu databáze a lze nastavit pouze jednotlivé tabulky.

Logické zálohování – dump databáze – lze provést přes PhpMyAdmin. Variabilita výstupu je velká – například lze nastavit v jakém formátu bude výstup dat nebo jestli bude před tvorbou tabulek předcházet DROP. Dump obsahuje rovněž data z blobu a clobu.

Pro ladění dotazů MySQL nabízí hned tři doplňující nástroje, které jsou vně databázového systému. Pro ruční ladění dotazu lze analyzovat jeho kroky pomocí příkazu EXPLAIN (Winand, 2012, 188 s.). Výstup příkazu vidíme na obrázku Obrázek 8. Z výstupu lze poznat, které tabulky se používají, jaký může být použit index a jaký je použit ve skutečnosti. V tomto případě je to primární index CTERARI_PK. Dále je vidět další nestandardní funkce musí být použity pro dosažení výsledku.



id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	PRIMARY	<derived2>	ALL	(NULL)	(NULL)	(NULL)	(NULL)	2	(NULL)
2	DERIVED	p	ALL	PLATBY_FK1	(NULL)	(NULL)	(NULL)	1	Using where; Using temporary; Using filesort
2	DERIVED	c	eq_ref	PRIMARY,CTENARI_PK	PRIMARY	4	test.p.CTENAR_ID	1	(NULL)

Obrázek 11 - Výstup příkazu EXPLAIN v MySQL

Druhý nástroj pro ladění je Profiler. Slouží pro zaznamenávání příkazu, které byly spuštěny spolu s jejich časem. Pokud chceme analyzovat zvolený příkaz, stačí spustit příkaz: „SHOW PROFILE CPU FOR QUERY 43;“ výstup zobrazí jednotlivé úkoly, které byly provedeny pro vykonání příkazu. Detail výstupu lze vidět na obrázku. Podobným nástrojem je INFORMATION_SCHEMA. Jeho výhodou je, že lze psát dotazy a tím si výstup upravit, aby zcela vyhovoval. INFORMATION_SCHEMA, obsahuje také proceslist, pomocí kterého lze zjistit právě běžící příkazy. Druhý a třetí nástroj analyzuje dotazy ex-post.

PostgreSQL

Nabízí se dva programy, které dokáží zálohovat PostgreSQL fyzicky za běhu. EnterpriseDB dodává EDB Backup and Recovery Tool. Stáhnout se dá pouze pro Linuxové distribuce. Proto i v tomto případě byl použit „neoficiální“ program. Komunita nabízí program Barman. Nyní je ve verzi 1.4.0. I on v nynější verzi podporuje inkrementální zálohy. Umí point in time obnovu.

Logické zálohování je podobné tomu z MySQL. Jmenuje se SQL Dump a používá se pomocí příkazové řádky pg_dump. SQL Dump umí najednou zálohovat pouze jednu databázi.

Explain existuje pro PostgreSQL také. Jeho výstup lze vidět na obrázku 9. Na první pohled je výstup mnohem víc komplexní, protože se v něm zobrazuje, jakým způsobem byl dotaz vyhotoven, typ spojení tabulek a cena operací.

Datový výstup		Výklad	Zprávy	Historie
	QUERY PLAN			
	text			
1	Subquery Scan on x	(cost=3.28..3.30 rows=1 width=74)		
2	Filter:	(rownum < 11)		
3	-> Sort	(cost=3.28..3.29 rows=1 width=74)		
4	Sort Key:	(sum(p.platba)), c.prijmeni		
5	-> HashAggregate	(cost=3.26..3.27 rows=1 width=74)		
6	Group Key:	c.prijmeni		
7	-> Hash Join	(cost=1.60..3.26 rows=1 width=74)		
8	Hash Cond:	(c.id = p.ctenar id)		
9	-> Seq Scan on ctenari c	(cost=0.00..1.47 rows=47 width=74)		
10	-> Hash	(cost=1.59..1.59 rows=1 width=64)		
11	-> Seq Scan on platby p	(cost=0.00..1.59 rows=1 width=64)		
12	Filter:	(stav = 1::numeric)		

Obrázek 12 - Explain plan z PostgreSQL [zdroj: autor]

EnterpriseDB nabízí profiler, který spojuje spuštěné příkazy a ke zvolnému dotazu zobrazuje explain plan z obrázku 9. a jeho grafickou podobu.

INFORMATION_SCHEMA je nahrazují Pg pohledy. Obdobou Statspack reportu nebo jeho novější verze AWR reportu je Enterprise EDRITA report. Pomocí příkazu: „SELECT * FROM edbsnap();“ lze vytvořit snapshot. Pro vyvolání slouží příkaz „SELECT * FROM edbreport(7,8);“ Vytváření snapshotu lze autorizovat, aby byly vytvořeny v časových intervalech. EDRITA poskytuje ucelené (komplexní) informace o systému a jeho zátěži. Například na jaké objekty se nejvíce čeká.

Vyhodnocení

Na první pohled je patrný rozdíl mezi výstup Explain plánu mezi MySQL PostgreSQL. Na obrázku 8 jsou uvedeny tabulky názvem ‘c’ a ‘p’. Explain plán u MySQL neumožňuje poznat, jestli došlo ke správnému postupu při skládání dotazu. Jestli byly správně použity metody joinu. PostgreSQL s tím problém nemá. EDRITA sice nedosahuje úrovně AWR, přesto je to výborný nástroj. Takový u MySQL není.

MySQL 2, PostgreSQL 4.

3.10. TCO

V případě programů, které jsou open source, nemá smysl řešit cenu pořízení. Programy budou zdarma. To neplatí v případě požadované podpory. Podpora pro MySQL lze koupit u firmy Oracle. U PostgreSQL je situace výrazně složitější. Lze získat podporu z řady firem podporujících komunitní vývoj. V rámci této práce byl použit databázový systém od firmy EnterpriseDB a zjištěna podpora u této společnosti.

Pro kalkulaci ceny zajímá obě společnosti počet socketů a jader. Zvolil jsem 2 sockety a 4 jádra a podporu na 3 roky. Edici MySQL je nastavena na Enterprise Edition (1-4 Socket Server). U PostgreSQL je zvolena PostgreSQL Advanced server.

ni jedna ze společností nežadá o zaplacení licence. Obě společnosti chtějí zaplatit poplatky za podporu a údržbu. Z tabulek vyšlo, že **MySQL Enterprise Edition stojí 15 tisíc dolarů, PostgreSQL Advanced server stojí 41.400 dolarů**. Výsledné číslo nebude konečné, protože obchodníci jsou schopni dát klidně 75% slevu v případě migrace z Oracle. To platí hlavně u EnterpriseDB. U MySQL si nemyslím, že by někdo slevu nabízel, spíš naopak.

Vyhodnocení

Ve zvolené konfiguraci je, že MySQL více než 2x levnější. Avšak oproti Oracle databázovému systému se jedná i tak o pořádnému ušetření.

MySQL 9, PostgreSQL 7

4. Zhodnocení výsledků

V níže uvedené tabulce 9 se zobrazuje testování a hodnocení z předchozí kapitoly. Váha je převzata z tabulky 5. Sloupec MySQL a PostgreSQL má hodnoty z vyhodnocení jednotlivých kritérií. Sloupec MySQL výsledek a PostgreSQL výsledek je vypočítán jako součin váhy a hodnocení kritéria.

Název	Zkratka	Váha	MySQL	MySQL výsledek	PostgreSQL	PostgreSQL výsledek
Rozšířené schopnosti databázového systému	V1	3,76	3	11,29305	6	22,5861036
Migrace z Oracle databáze	V2	2,66	4	10,64719	9	23,9561805
TCO – Total Cost of Ownership	V3	1,89	10	18,85884	7	13,2011887
Nástroje pro správu	V4	1,35	2	2,7066	4	5,41319979
Bezpečnost	V5	0,99	4	3,959386	7	6,92892481
Cena administrace databáze (DBA cost)	V6	0,74	7	5,199003	5	3,71357359
Škálovatelnost	V7	0,58	4	2,306916	3	1,73018697
Nástroje pro práci	V8	0,47	5	2,352719	4	1,8821753
Součet				57,32371		79,4115332

Tabulka 9 - MySQL vs. PostgreSQL [zdroj: autor]

Na základě tabulky 9 vyhrálo PostgreSQL zastoupené PostgreSQL Advanced serverem. Obě finanční kategorie vyhrálo MySQL. Zejména v kategorii TCO byla suverénní.

MySQL a Oracle vlastní stejný majitel – firma Oracle Corporation Inc. Oracle podporuje pouze migraci z MySQL. Potvrzuje se názor Michael Wideniuse, že MySQL škodí Oracle Corporation Inc.

5. Závěr

Cílem mé diplomové práce bylo srovnat dva databázové systémy za účelem možnost nahradit Oracle databázový systém.

Srovnání bylo rozděleno na část teoretickou a praktickou. V teoretické části byly představeny oba databázové systémy. Byla popsána jejich obecná charakteristika, minulost, současnost, architektura a průběh spuštění dotazu včetně popisu procesu připojení k databázovému systému. Praktická část se zabývala analýzou jednotlivých kritérií. V rámci analýzy byla ověřena možnost migrace na srovnávané databázové systémy. Zjištěna cena za provozování porovnávaných řešení. Cena se dělí mezi vyškolení a náklady na pořízení respektive udržování databázového systému. Potvrdilo, se že řešení od Oracle je mnohem dražší a migrace může být uskutečněna.

Práce splnila vytyčené cíle. Tabulka 9 Dle tabulky 9 vyšel nejlépe PostgreSQL zastoupený PostgreSQL Advanced serverem. Databázový systém prokázal, že může nahradit Oracle. Nemůže se s ním srovnávat zejména v nástrojích na správu a možnostech škálovatelnosti.

MySQL může být také použito jako náhrada, ale při migraci se musí počítat se zvýšenými náklady na přeno objektů a dat. V případě, že by šlo o aplikaci, která má logiku ukrytou v PL/SQL je migrace nevhodná. MySQL na rozdíl od PostgreSQL Advanced serveru nemá kompatibilitu s PL/SQL od firmy Oracle Coporation Inc.

MariaDB má funkcionalitu, která je dostupná pouze v komerční verzi MySQL. Je nabízena firmou Red Hat jako hlavní databáze. Oproti MySQL se rychle rozvíjí.

Diplomovou práci lze rozšířit o reálnou migrace s uživatelskými daty. Tím se ověří nejen funkčnost migrace a její rychlost, ale může se otestovat i aplikace oproti novému databázovému systému. Další možností je otestovat rychlost zpracování dotazů za pomoci reálného provozu. To se dá dosáhnout pomocí aplikace jMeter, která je založena na javě. Od jMeteru se dají nahrát uživatelské akce, které mohou být znovu spuštěny.

6. Seznam použitých zdrojů

BELL, Charles. Expert MySQL. 2 vydání. Apress. 2013. 640 s. 1-4302-4659-6

CLARKE, Justin. SQL Injection Attacks Defense. 2. vydání. Syngress Media, U.S. 2012. 576 s. 1-59749-963-3

CUNNINGHAM, Lewis. World's Largest Database Runs on Postgres?. [ONLINE] 28.5.2008. [Citace: 25. 2 2015.], <http://it.toolbox.com/blogs/oracle-guide/worlds-largest-database-runs-on-postgres-24979>

dbms_comparison, [ONLINE] 2015, [Citace: 30. 1 2015], http://www.sql-workbench.net/dbms_comparison.html

Dries Buytaert. The history of MySQL AB. [ONLINE] 3.8.2010. [Citace: 10. 1 2015.], <http://buytaert.net/the-history-of-mysql-ab>

EnterpriseDB. PostgreSQL 9.4.1 Documentation. [ONLINE] 2014. [Citace: 25. 2 2015.], <http://www.enterprisedb.com/docs/en/9.4/pg/storage-file-layout.html>

EnterpriseDB. xDB Replication FAQs. [ONLINE] 2015. [Citace: 30. 1 2015], <http://www.enterprisedb.com/faq-xdb-multi-master>

HSU, Leo. OBE, Regina. PostgreSQL: Up and Running. 1. vydání. O'Reilly Media. 2012. 166 s. 1-4493-2633-1

JELÍNEK, Lukáš. Google migruje na MariaDB. [ONLINE] 15.09.2013. [Citace: 5. 1. 2015]. <http://www.linuxexpres.cz/novinky/google-migruje-na-mariadb?highlightWords=mariadb>

KIM, Kyu. PostgreSQL at a glance. [ONLINE] 2013. [Citace: 30. 1 2015]. <http://www.cubrid.org/blog/dev-platform/postgresql-at-a-glance/>

MariaDB. MariaDB 10.0 and MySQL 5.6. [ONLINE] 14.1.2013. [Citace: 5. 1 2015.], <http://blog.mariadb.org/mariadb-10-0-and-mysql-5-6/>

MCALLISTER, Neil. Red Hat to ditch MySQL for MariaDB in RHEL 7. [ONLINE] 15.6.2013. [Citace: 25. 2 2015.], http://www.theregister.co.uk/2013/06/15/red_hat_to_ditch_mysql_for_mariadb_in_rhel_7/

MILLER, Richard. Who Has the Most Web Servers?. [ONLINE] 14.5.2009. [Citace: 5. 1 2015.], <http://www.datacenterknowledge.com/archives/2009/05/14/whos-got-the-most-web-servers/>

Oracle Corporation. Tuning Server Parameters. MySQL. [Online] 23.11.2010. [Citace: 25. 2 2015.], <http://dev.mysql.com/doc/refman/5.0/en/create-database.html>

OVAIS, Tariq. Understanding InnoDB clustered indexes, [ONLINE] 2011. [Citace: 25. 2. 2015.]. <http://www.ovaistariq.net/521/understanding-innodb-clustered-indexes>

STONES, Richard. NEIL, Matthew. Beginning Databases with PostgreSQL: From Novice to Professional. 2. vydání. O'Reilly Media. 2007. 664 s. 1590594789

SHAIK , B. PostgreSQL Architecture. [ONLINE] 2013. [Citace: 25. 2 2015.]. <https://bajis.wordpress.com/2013/09/20/postgresql-architecture/>

SHAUGHNESSY, Pat. Following a Select Statement Through Postgres Internals. [ONLINE] 13. 11. 2014. [Citace: 21. 3 2015.]. <http://patshaughnessy.net/2014/10/13/following-a-select-statement-through-postgres-internals>

SCHWARTZ, Baron. ZAITSEV, Peter. TKACHENKO, Vadim. High Performance MySQL. 3 vydání. O'Reilly Media. 2012. 823 s. 978-1-4493-1428-6

STĚHULE, Pavel. Desatero. [ONLINE] 2015, [Citace: 25. 2 2015.]. <http://postgres.cz/wiki/Desatero>

STĚHULE, Pavel. Historie projektu PostgreSQL. [ONLINE] 22. 05. 2012. [Citace: 21. 3 2015.]. <http://www.root.cz/clanky/historie-projektu-postgresql/>

STĚHULE, Pavel. Pavel Stěhule patche. [ONLINE] 2013, [Citace: 20. 3 2015.]. http://postgres.cz/wiki/Pavel_St%C4%9Bhule_patche

TŮMA, Marek. Srovnání MySQL a PostgreSQL. 2014. Diplomová práce: CZU. PEF. 85 s.

WIDENIUS, Michael. Help saving MySQL. [ONLINE] 12.12.2009. [Citace: 5. 1 2015.]. <http://monty-says.blogspot.cz/2009/12/help-saving-mysql.html>

WINAND, Markus. SQL Performance Explained. 1. vydání. Markus Winand. 2012. 204 s. 3950307826

Seznam obrázků

Obrázek 1- Logo MySQL	13
Obrázek 2- Historický růst MySQL.....	15
Obrázek 3- Logický pohled na MySQL Server architekturu.....	20
Obrázek 4- Proces spuštění příkazu.....	25
Obrázek 5- Logo PostgreSQL.....	26
Obrázek 6- Architektura PostgreSQL 9.x	32
Obrázek 7- Logický model migrované aplikace.....	49
Obrázek 9- Převod tabulek a pohledů pomocí programu SQLyog.....	50

Obrázek 10- MySQL benchmark.....	55
Obrázek 11- PostgreSQL benchmark	56
Obrázek 12- Výstup příkazu EXPLAIN v MySQL.....	57
Obrázek 13 - Expalin plan z PostgreSQL.....	59

Seznam tabulek

Tabulka 1 - Historie změn	16
Tabulka 2 - MySQL edice	19
Tabulka 3 - Hlavní změny ve verzích PostgreSQL	29
Tabulka 4 - Kritéria řazená dle důležitosti.....	37
Tabulka 5 - Váha kritérií dle Saatyho metody	37
Tabulka 6- Školení pro MySQL	42
Tabulka 7 - Přehled vlastnosti MySQL a PostgreSQL	45
Tabulka 8 - Datové typy	47
Tabulka 9 - MySQL vs. PostgreSQL.....	61