

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE
PROVOZNĚ EKONOMICKÁ FAKULTA

KATEDRA INFORMAČNÍCH TECHNOLOGIÍ



DIPLOMOVÁ PRÁCE

Srovnání databázových systémů
MySQL a PostgreSQL

Autor: Bc. Marek Tůma

Vedoucí práce: Ing. Alexandr Vasilenko

© 2014 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra informačních technologií

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Tůma Marek

Informatika

Název práce

Srovnání databázových systémů MySQL a PostgreSQL

Anglický název

Comparison of MySQL and PostgreSQL database systems

Cíle práce

Diplomová práce je zaměřená na srovnání databázových strojů MySQL a PostgreSQL. Hlavním cílem práce je porovnat tyto databázové stroje z hlediska přijímaných SQL dialektů, rozšíření jejich nasazení a z hlediska výkonu při práci nad velkými objemy dat.

Díličí cíle práce jsou

- analýza vybraných odborných zdrojů zejména z hlediska metodiky měření výkonu databází,
- vytvoření testovacího prostředí,
- vytvoření testů dle vhodně navržené metodiky a jejich provedení.

Metodika

Metodika řešené problematiky je založená na analýze dostupných zdrojů, výběru vhodných hodnotících kritérií a poté porovnání daných databázových zdrojů podle zvolených kritérií. Pro praktické porovnání bude vybrán stroj, na kterém budou nainstalovány vhodně zvolené verze obou databázových strojů a porovnání provedeno. Nakonec budou formulovány závěry.

Harmonogram zpracování

- 1) Příprava a studium odborných zdrojů, upřesnění dílčích cílů práce a volba postupu řešení, zejména s ohledem na měření výkonu databázových strojů: 6/2013
- 2) Zpracování přehledu řešené problematiky dle informačních zdrojů: 7/2013 - 9/2013
- 3) Vypracování analytické části práce, diskuse a zhodnocení výsledků: 10/2013 - 11/2013
- 4) Tvorba finálního dokumentu diplomové práce: 12/2013 - 2/2014
- 5) Odevzdání diplomové práce a teze: 3/2014

Rozsah textové části

60 - 80 stran

Klíčová slova

MySQL, PostgreSQL, databáze, databázový stroj, SQL, dialekt, výkon, srovnání, rychlost, test

Doporučené zdroje informací

BALLIN, Derek J. a kolektiv. MySQL profesionálně: optimalizace pro vysoký výkon. Vyd. 1. Brno: Zoner Press, 2009, 712 s. Encyklopedie webdesignera. ISBN 978-80-7413-035-9.

SMITH, Gregory. PostgreSQL 9.0: high performance : accelerate your PostgreSQL system and avoid the common pitfalls that can slow it down. Birmingham: Packt Publishing, c2010, xiii, 442 s. ISBN 978-1-849510-30-1.

MySQL Documentation: MySQL Reference Manuals. ORACLE. [online]. [cit. 2013-02-04]. Dostupné z: <http://dev.mysql.com/doc/>

Documentation. THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. PostgreSQL [online]. [cit. 2013-02-04]. Dostupné z: <http://www.postgresql.org/docs/>

Performance test tools. MARK, Abedour. Opensourcetesting.org [online]. [cit. 2013-02-04]. Dostupné z: <http://www.opensourcetesting.org/performance.php>

Vedoucí práce

Vasilenko Alexandr, Ing.

Termín odevzdání

březen 2014



doc. Ing. Zdeněk Havlíček, CSc.
Vedoucí katedry



prof. Ing. Jan Hron, Dr.Sc., dr. h. c.
Děkan fakulty

V Praze dne 26. 11. 2013

Prohlášení

Prohlašuji, že svou diplomovou práci „Srovnání databázových systémů 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 20. 3. 2014

Bc. Marek Tůma

Poděkování patří Ing. Alexandru Vasilenkovi za vedení práce.
Také Bc. Rostislavovi Stříbrnému za konzultace ohledně Java vývojových nástrojů a
následné zapůjčení testovacího stroje. Také za přátelství a podporu během studia.
Největší poděkování však patří mé manželce Ing. Lucii Tůmové za trpělivost a podporu jak
během psaní práce, tak v průběhu celého studia.

Srovnání databázových systémů MySQL a PostgreSQL

Comparison of MySQL and PostgreSQL database systems

Souhrn

Práce se zaměřuje na srovnání databázových systémů MySQL a PostgreSQL. Nejprve rozebírá teoretická východiska – standardizaci a vývoj protokolu SQL, historii a vlastníka obou systémů, jejich forků (tj. odvozené větve vývoje), licence a shodu s SQL standardy. Poté představuje organizaci Transaction performance council a její testovací scénáře a metodiky pro měření výkonu databází. Přidává alternativní metodiky, které poskytují i měřicí nástroje zdarma a rozbor těchto nástrojů.

V praktické části se věnuje postupu instalace na různých operačních systémech, možnosti optimalizace výkonu a popisu k tomu nutných postupů a parametrů, dále je provedeno měření pomocí nástrojů database benchmark a pole position a srovnání výkonu změřeného těmito nástroji. Dále se zabývá problematikou provozních nákladů, ať už z hlediska nákladů na hardware a software, tak z hlediska nákladů na lidské zdroje.

Na závěr je provedena diskuse, která vypočítává klady a zápory obou databázových systémů a také metodik měření jejich výkonu, a jsou sepsány závěry. Je přidán také slovníček pro vysvětlení některých technologických pojmů.

Summary

This diploma thesis is based on comparison of MySQL and PostgreSQL database systems. At first, theoretical background is described – the standardisation of SQL protocol, history and owners of both systems, their forks (derived development branches), licences and compliance with the SQL standards. Thereafter the Transaction performance council is introduced along their testing scenarios and database performance metric methodologies. Alternated free and open source methodics are described as well.

In practical part is thesis focused on instalation steps of the database systems using varisous operating systems, possibilities of performance tuning, description of the steps needed and configuration change. Then the measurement is made using tools database benchmark and pole position and comparsion of performance measured by these tools. After that the thesis is focused on cost of operation including cost and availability of human resources on local market.

At the end discussion of measured resuluts is made, advantages and disadvantages of measurement tools are described and conclusions are formulated. Brief vocabulary of used technical terms is also included.

Klíčová slova: MySQL, PostgreSQL, databáze, databázový stroj, dialekt, výkon, srovnání, rychlost, test

Keywords: MySQL, PostgreSQL, database, database machine, dialect, performance, comparsion, speed, test

Obsah

| | |
|--|-----------|
| PROHLÁŠENÍ | 4 |
| 1 ÚVOD | 10 |
| 2 CÍL PRÁCE A METODIKA | 11 |
| 2.1 CÍLE PRÁCE | 11 |
| 2.2 METODIKA PRÁCE..... | 11 |
| 3 TEORETICKÁ ČÁST | 12 |
| 3.1 AKTUÁLNÍ STAV V LITERATUŘE | 12 |
| 3.2 JAZYK SQL, JEHO VÝVOJ A STANDARDIZACE..... | 13 |
| 3.3 MYSQL | 17 |
| 3.3.1 Zastřešující organizace..... | 19 |
| 3.3.2 Licence..... | 19 |
| 3.3.3 Aktuální verze | 21 |
| 3.3.4 Historie | 21 |
| 3.3.5 Forky (odvozené větve vývoje)..... | 22 |
| 3.3.6 Administrační nástroje..... | 24 |
| 3.3.7 Omezení | 25 |
| 3.3.8 Škálování | 26 |
| 3.3.9 Přijímaný SQL Dialekt | 27 |
| 3.3.10 Výhrady k MySQL..... | 29 |
| 3.4 POSTGRESQL..... | 30 |
| 3.4.1 Zastřešující organizace..... | 31 |
| 3.4.2 Licence..... | 31 |
| 3.4.3 Aktuální verze | 33 |
| 3.4.4 Historie | 33 |
| 3.4.5 Forky (odvozené databáze)..... | 34 |
| 3.4.6 Administrační nástroje..... | 35 |
| 3.4.7 Omezení | 36 |
| 3.4.8 Škálovatelnost..... | 36 |
| 3.4.9 Přijímaný SQL Dialekt | 37 |
| 3.4.10 Výhrady k PostgreSQL | 38 |
| 3.5 METODIKA MĚŘENÍ VÝKONU DATABÁZE..... | 40 |
| 3.5.1 TPC..... | 40 |
| 3.5.2 The Open Source Database Benchmark | 42 |

| | | |
|----------|---|-----------|
| 3.5.3 | <i>Standard Performance Evaluation Corporation (SPEC)</i> | 43 |
| 3.5.4 | <i>Database benchmark</i> | 44 |
| 3.5.5 | <i>Pole position</i> | 44 |
| 4 | PRAKTICKÁ ČÁST | 45 |
| 4.1 | DOSTUPNOST INSTALAČNÍCH BALÍKŮ | 45 |
| 4.2 | PODPOROVANÉ OPERAČNÍ SYSTÉMY | 45 |
| 4.3 | INSTALACE | 45 |
| 4.4 | DATABÁZOVÉ OPTIMALIZACE A JEJICH MĚŘENÍ | 47 |
| 4.5 | SROVNÁNÍ VÝKONNOSTI | 51 |
| 4.5.1 | <i>Database benchmark</i> | 51 |
| 4.5.2 | <i>Pole position</i> | 56 |
| 4.6 | EKONOMIKA PROVOZU | 69 |
| 5 | VÝSLEDKY A DISKUSE | 72 |
| 6 | ZÁVĚR | 76 |
| 7 | SLOVNÍČEK | 77 |
| | SEZNAM TABULEK | 78 |
| | SEZNAM OBRÁZKŮ | 79 |
| | SEZNAM POUŽITÝCH ZDROJŮ | 81 |

1 Úvod

Diplomová práce se zabývá srovnáním dvou známých open source databázových strojů. Toto téma se stává stále aktuálnějším s velkým rozvojem internetových služeb a takzvaného internetu věcí. V blízké budoucnosti se navíc dá předpokládat prudký rozvoj takzvaných wearables – tedy chytrých předmětů na denní nošení. Již dnes si své místo na slunci pomalu získávají chytré hodinky, chytré termostaty a další podobná zařízení, která produkují data, jež je třeba někde uchovávat, přenášet a zpracovávat. A právě open source databázové stroje jsou zpravidla první volbou pro nezávislé vývojáře tisíců aplikací pro tyto přístroje a navázané internetové služby.

Práce se bude zabývat srovnáním dvou nejznámějších – MySQL a PostgreSQL, které si v průběhu desítek let vývoje již získali svůj respekt v komunitě vývojářů, ale také jejich novými odnožemi, které se o uznání teprve snaží. MySQL si vydobyla své místo v softwarovém balíku LAMP a běží tak na množství webhostingů poskytovaných zdarma, čímž se často pro začínajícího programátora stává první branou k relačním databázím. PostgreSQL naproti tomu často představuje první krok ke „striktnějšímu SQL“ a více „enterprise“ databázím.

Pomocí vhodných nástrojů se práce bude snažit srovnat výkonnost obou typů databází v různých testovacích scénářích.

2 Cíl práce a metodika

2.1 Cíle práce

Diplomová práce je zaměřená na srovnání databázových strojů MySQL a PostgreSQL. Hlavním cílem práce je porovnat tyto databázové stroje z hlediska přijímaných SQL dialektů, rozšíření jejich nasazení a z hlediska výkonu při práci nad velkými objemy dat.

Dílčí cíle k tomuto vedoucí jsou:

- Analýza vybraných odborných zdrojů zejména z hlediska metodiky měření výkonu databází,
- vytvoření testovacího prostředí,
- vytvoření testů dle vhodně navržené metodiky a jejich provedení.

2.2 Metodika práce

Metodika řešené problematiky je založená na analýze dostupných zdrojů, výběru vhodných hodnotících kritérií, a poté porovnání daných databázových strojů podle zvolených kritérií.

Pro praktické porovnání bude vybrán stroj, na kterém budou nainstalovány vhodně zvolené verze obou databázových strojů a porovnání provedeno.

Na závěr budou formulovány závěry.

3 Teoretická část

Práce shrnuje základní poznatky použitelnosti databázových strojů MySQL a PostgreSQL z hlediska snadnosti jejich nasazení, přijímaných SQL dialektů a výkonu.

3.1 *Aktuální stav v literatuře*

Literatura, která by se zabývala přímým porovnáváním jednotlivých databázových enginů, je k dispozici zpravidla ve formě článků na odborných webových stránkách (1) (2) (3) (4) a také prací z vědeckých konferencí (5) a ve formě knih spíše nevychází. Je tomu tak pravděpodobně proto, že jednotlivé verze různě zastarávají, výkon jednotlivých databází hodně závisí na použitém hardwaru a zkušenosti databázového administrátora, který databázi spravuje (6) (7) (8).

Dále na tomto poli existuje profesní organizace TPC (Transaction performance council), sídlící v San Franciscu, která si klade za cíl vytvoření metodik pro porovnávání různých transakčních systémů. Členy této organizace jsou všichni velcí hráči na poli databází. Namátkou například Oracle, IBM, Microsoft, Cisco, Teradata a další.

TPC pravidelně publikuje výsledky testů na svých webových stránkách <http://www.tpc.org>

Protože získání oficiálních výsledků TPC může být finančně nákladná záležitost, vznikla i open source snaha o totéž - The Open Source Database Benchmark (dostupné na <http://osdb.sourceforge.net/>), zde je bohužel nutno konstatovat, že tato snaha zřejmě postupně odumírá, dokumentace je z roku 2002 a poslední commit do repozitáře z roku 2012.

Co se týče přijímaných dialektů jednotlivých databázových strojů, nejlepším zdrojem informací je oficiální dokumentace na webových stránkách

a to <http://www.postgresql.org/docs/9.3/interactive/index.html> pro PostgreSQL a <http://dev.mysql.com/doc/> pro MySQL.

Pro ladění různých nejasných chování a chyb mají oba projekty k dispozici vlastní vývojářská fóra (9) (10), případně se poté komunitou probírají i na jiných odborně zaměřených portálech (11) (12).

3.2 Jazyk SQL, jeho vývoj a standardizace

Protože se práce zabývá srovnáním SQL databází i z hlediska přijímaných SQL dialektů, je vhodné na začátek uvést vývoj tohoto jazyka, aby bylo možné dále srovnat shodu s jeho standardy, poměrně zajímavá je také problematika certifikační autority, protože se na tomto poli angažují jak ANSI (*American National Standards Institute*), tak ISO (*International Organization for Standardization*), ale například i národní FIPS (*Federal Information Processing Standards* - vydávaný vládou USA), kvůli čemuž se i stejný standard může označovat různě – viz dále Tabulka 1 – sloupce *označení* a *další označení*.

V roce 1970 publikoval v časopise vydávaném organizací *Association of Computer Machinery (ACM)* Dr. E. F. Codd stať „Relační model dat pro velké sdílené databáze“. Tento Coddův model je dnes široce akceptován jako definitivní model pro systémy řízení báze dat (SRBD).

Společnost IBM poté implementovala jazyk, který se pro operace nad tímto relačním modelem používal. Tento jazyk byl nazván SEQUEL (zkratka z *Structured English Query Language*), což bylo později zkráceno na SQL (ale stále čteno jako sequel). IBM v roce 1979 představila první komerčně dostupnou implementaci tohoto jazyka (13).

Roku 1986 byl tento jazyk přijat jako standard Americkou standardizační asociací (ANSI) pod označením SQL-86 (14). Mezinárodní standardizační organizace ISO ho přijala o rok později, tj. v roce 1987. Označení SQL-86 a SQL-87 tak představují název stejné specifikace pouze u dvou různých standardizačních autorit. V roce 1989 je dále SQL standard přijat i jako FIPS 127-1.

ISO spolu s IEC (*International Electrotechnical Commission – tedy mezinárodní elektrotechnická komise*) ustanovily technickou podkomisi pod označením *ISO/IEC JTC 1, Information technology, Subcommittee SC 32, Data management and interchange*, která

má na starosti vývoj SQL standardu - *ISO/IEC 9075*. Současnou verzí¹ je standard SQL:2011.

ISO/IEC 9075 je dále doplňován standardem *ISO/IEC 13249 SQL Multimedia and Application Packages*, o který se také stará již zmíněná komise, a který definuje další interface a balíčky založené na SQL. Cílem tohoto standardu je sjednocení přístupu k různým databázovým objektům, například textu, obrázkům, ale i rozměrným datům.

Historii vývoje SQL standardu od jeho prvotního přijetí až po dnešek ukazuje Tabulka 1.

| Rok | Název | Další označení | Obsah |
|-------------|--------------|---------------------------|--|
| 1986 | SQL-86 | SQL-87 | První formalizace ANSI, přijetí ISO. |
| 1989 | SQL-89 | FIPS127-1 | Přidání integritních omezení. Přijato jako FIPS 127-1 (FIPS jsou vydávány vládou USA). |
| 1992 | SQL-92 | SQL2, FIPS 127-2 | Velká revize (ISO 9075), <i>Entry Level</i> SQL-92 přijato jako FIPS 127-2. |
| 1999 | SQL:1999 | SQL3 | Další velká revize: přidány regulární výrazy, rekurzivní dotazy, triggery, neskalární datové typy a další. |
| 2003 | SQL:2003 | SQL 2003 | Přináší podporu pro XML (SQL/XML), window funkce, sekvence a sloupce s auto-generovanými hodnotami. |
| 2006 | SQL:2006 | SQL 2006 | Dále zlepšuje podporu XML, přináší podporu |

¹ psáno 2014

| | | | |
|-------------|----------|----------|---|
| | | | dotazovacího jazyka Xquery dle W3C. |
| 2008 | SQL:2008 | SQL 2008 | Přináší ORDER BY i mimo definice kurzoru, přináší trigger INSTEAD OF a příkaz TRUNCATE. |
| 2011 | SQL:2011 | SQL 2011 | Přináší lepší podporu pro časová (temporal) data. |

Tabulka 1 Vývoj SQL standardu

SQL-92 definovalo 3 úrovně shody s tímto standardem – Entry, Intermediate a Full. Většina databází z té doby se spokojila s úrovní shody „Entry“ (15).

Od standardu SQL-99 se definují jednotlivé balíky požadavků namísto pouhých tří úrovní shody. Některé se začaly označovat jako „CORE“ (základní), jiné jako „OPTIONAL“ (dobrovolné) – aby jednotlivé SQL implementace mohly být označovány jako vyhovující, bylo nutné, aby implementovaly všechny CORE funkčnosti.

Nejnovější standard SQL-2011 specifikuje dnes 9 oblastí, které představuje Tabulka 2, PostgreSQL definuje svou shodu se SQL standardem právě vůči těmto devíti oblastem, jak bude rozebráno později v příslušné sekci práce.

| Část | Označení | Obsah |
|-------------|--|---|
| 1 | ISO/IEC 9075-1:2011 Part 1: <i>Framework</i> (SQL/Framework) | Logické koncepty SQL |
| 2 | ISO/IEC 9075-2:2011 Part 2: <i>Foundation</i> (SQL/Foundation) | Hlavní zásady a prvky jazyka |
| 3 | ISO/IEC 9075-3:2008 Part 3: <i>Call-Level Interface</i> (SQL/CLI) | Interface komponenty (struktury, procedury, proměnné) pro SQL příkazy v jazycích ADA C, ADA C++, COBOL, Fortan... |
| 4 | ISO/IEC 9075-4:2011 Part 4: <i>Persistent Stored Modules</i> (SQL/PSM) | Standardizuje procedurální rozšíření SQL – řízení flow, podmínek, a další. SQL/PSM dále formalizuje deklaraci |

| | | |
|-----------|---|--|
| | | a údržbu uložených procedur (stored procedures) |
| 9 | ISO/IEC 9075-9:2008 Part 9: <i>Management of External Data</i> (SQL/MED) | Přináší rozšíření pro wrapery na data cizího původu, a datové typy „datalink“ pro možnost udržovat externí data. To jsou taková data, která jsou pro SQL přístupná, ale nejsou v něm uložena. |
| 10 | ISO/IEC 9075-10:2008 Part 10: <i>Object Language Bindings</i> (SQL/OLB) | Přináší podporu pro jazyk SQLJ – tedy SQL embedované v Javě (podobně jako část 3 pro další jazyky) |
| 11 | ISO/IEC 9075-11:2011 Part 11: <i>Information and Definition Schemas</i> (SQL/Schemata) | Definuje information schema, definition schema a přináší nástroje, jak udělat SQL databáze a objekty samopopisující. |
| 13 | ISO/IEC 9075-13:2008 Part 13: <i>SQL Routines and Types Using the Java Programming Language</i> (SQL/JRT) | Specifikuje možnost volat statické Java metody přímo ze SQL aplikací ('Java-in-the-database'). Přináší také možnost zavést Java třídy jako uživatelsky definované strukturované SQL datové typy. |
| 14 | ISO/IEC 9075-14:2011 Part 14: <i>XML-Related Specifications</i> (SQL/XML) | Specifikuje rozšíření pro XML v SQL databázích včetně datového typu XMLType. |

Tabulka 2 9 oblastí ISO SQL standardu

Obecně SQL jazyk můžeme podobně jako jiné jazyky rozdělit na tři části:

- DDL (data definition language), tj. Jazyk pro definici dat. V SQL se používají příkazy jako CREATE, ALTER a DROP na řízení tabulkových struktur,
- DML (data manipulation language), tj. Jazyk pro manipulaci dat. V SQL se používají příkazy SELECT, INSERT, UPDATE, DELETE, TRUNCATE,
- DCL (data control language), tj. Jazyk pro správu dat. Tím se definují práva pro jednotlivé uživatele a uživatelské role ve vztahu k jednotlivým tabulkám databáze. Používané příkazy jsou CONNECT, SELECT, INSERT, UPDATE, DELETE, EXECUTE a USAGE.

3.3 MySQL

MySQL²³ se prezentuje jako nejoblíbenější open source databáze na světě. Dle oficiálních prohlášení při ohlášení převzetí původní vývojářské společnosti MySQL AB dosahoval počet stažení této databáze v roce 2008 více než 100 milionů stažených kopií s přírůstkem 50 000 denně (16).



Obrázek 1 Logo MySQL

Dle dohledatelných čísel by měla být druhou vůbec nepoužívanější databází (následuje za Sllite, která je součástí každého Android a iPhone operačního systému) (17).

Podle vyjádření na oficiálních stránkách, používá tuto databázi mnoho z největších, či nejvíce rostoucích organizací pro úsporu času a nákladů při budování webových stránek,

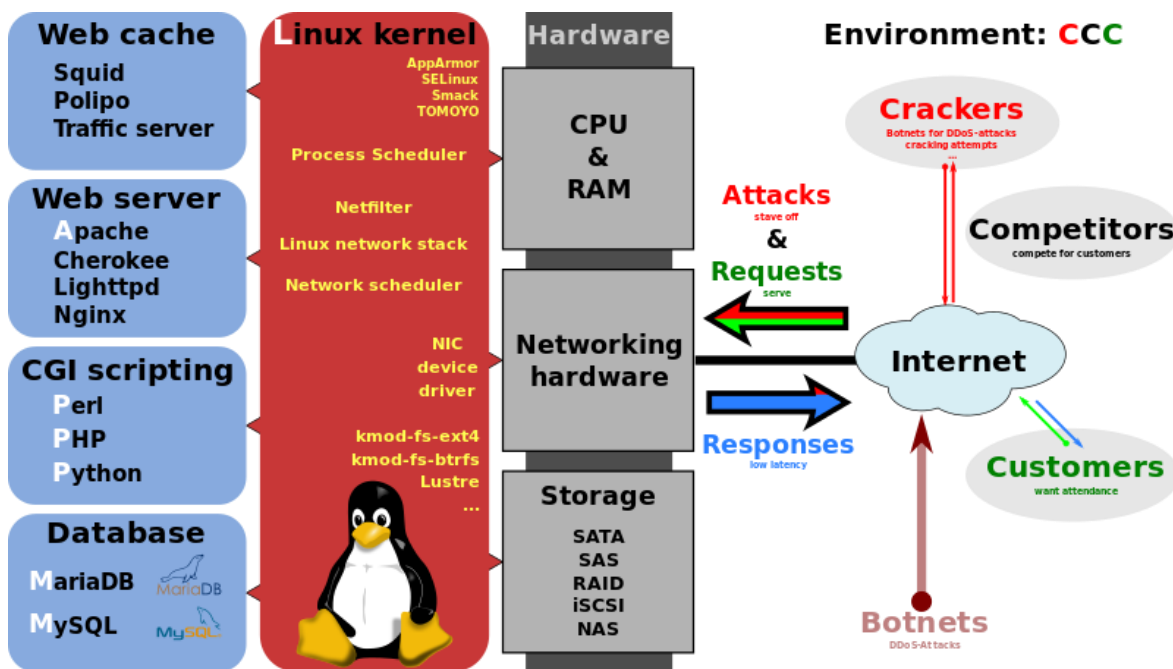
² (/maɪ ˈɛskjuːˈɛl/ "My S-Q-L" oficiálně, často také čteno jako /maɪ ˈsiːkwəl/ "My Sequel")

³ Název vznikl složením SQL a slova My, což je domácí zkratka jména Maria – jména dcery vývojáře Michaela Widenia, toto jméno také dalo vzniknout názvu forku MariaDB

informačních bussiness systémů, ale i balíkového software. Jako příklad jmenují Yahoo, Alcatel-Lucent, Google, Nokii, YouTube, Wikipedii a Booking.com (18). MySQL není jen databázový stroj, ale instalace obsahuje i množství nástrojů na testování, monitoring a reporting, a také je zde k dispozici možnost koupit si placenou licenci s podporou od společnosti Oracle – více viz v části licence.

Cena takového řešení je podle oficiálních stránek 41.388,- Kč bez DPH za rok (platí pro MySQL Standard Edition Subscription (1-4 socket server)) (19), více viz Tabulka 3.

Jeden z největších úspěchů MySQL je integrace do softwarového balíku LAMP, který je dnes považován za vstupní bránu k webovému vývoji. LAMP je původně zkratka pro Linux, Apache, MySQL, PHP, dnes je však možné jednotlivé části nahrazovat alternativami, viz Obrázek 2. Existují také alternativní balíky jako XAMP (platformě nezávislý), WAMP (pro platformu Windows) a další.



Obrázek 2 LAMP server a jeho okolí - autor Shmuel Csaba Otto Traian

Předností těchto balíků jsou svobodné licence na použitý software, což může mít za následek významné snížení TCO (celkových provozních nákladů) (18).

3.3.1 Zastřešující organizace

MySQL bylo původně založeno a vytvořeno ve Švédsku dvěma Švédy a Finem: Davidem Axmarkem, Allanem Larssonem a Michaelem „Monty“ Wideniusem.

Tito lidé spolupracovali již od osmdesátých let, v roce 1994 začali s vývojem MySQL, načež v roce 1995 založili společnost MySQL AB. Tato společnost byla v roce 2008 převzata společností Sun Microsystems (známou například vývojem jazyka a prostředí Java), která sama byla v roce 2009 prodána společnosti Oracle.

Tato akvizice byla dokončena v roce 2010 a vlastnická práva na databázi MySQL tak v současné době (2013) vlastní Oracle Corporation.

Oracle Corporation je významným hráčem na poli softwarových a zejména databázových produktů. V roce 2013 měla dle oficiálních výkazů 122 830 zaměstnanců rozmístěných po celém světě.

3.3.2 Licence

Svobodná licence

MySQL Community Edition je k dispozici pro „nekomerční“⁴ použití pod GPL (General Public Licence) verze 2 (20). Tato licence vyžaduje, aby veškerá odvozená díla byla k dispozici pod toutéž licencí, ale zároveň umožňuje bezproblémové tvoření odvozenin, více viz kapitolu 3.3.5 na straně 22, zároveň ale Oracle vydal takzvanou FOSS výjimku.

Tato FOSS výjimka (free open source software) umožňuje použití této GPL licencované databáze i pro software, který je sám šířen pod jinou svobodnou licencí,

⁴ Slovo nekomerční je zde použito ve smyslu, že tuto licenci může využít i majitel komerčních webových stránek, pokud MySQL slouží jako úložiště dat například pro jeho redakční systém. Licenční politika je poměrně složitá, a proto se doporučuje v případě nejasností kontaktovat obchodní oddělení společnosti Oracle.

za podmínek, že MySQL je jediná GPL součást a zároveň se nejedná o přímou odvozeninu (21).

Produkty, které MySQL využívají jako datový sklad a mohou být například připojeny i na jiné databáze (ať už forky MySQL, nebo i úplně jiné databázové stroje), obecně mohou také používat tuto bezplatnou verzi.

Zajímavostí na tomto poli je, že firma MySQL AB v roce 2002 žalovala u Massachusettského soudu firmu Progress software corporation (výrobce například software NuSphere), kdy došlo k předběžnému opatření – zákazu používání obchodních známek a soudkyně stanovila, že GPL může být vymáhána. Obě strany ovšem poté vstoupili v mimosoudní jednání a rozsudek nepadl. (22) .

Tato komunitní edice zahrnuje:

- **Zásuvnou architekturu úložných enginů včetně**
 - InnoDB
 - MyISAM
 - NDB (MySQL Cluster)
 - Memory
 - Merge
 - Archive
 - CSV
 - **MySQL Replication** pro zlepšení aplikačního výkonu a škálovatelnosti
 - **MySQL Partitioning** pro zlepšení výkonu a správy velkých databází
 - **Performance Schema** pro uživatelský a aplikační monitoring využívání zdrojů
 - **Information Schema** pro snadný přístup k metadatům
 - **MySQL Connectors** (ODBC, JDBC, .NET, etc) pro vývoj aplikací v různých programovacích jazycích.
 - **MySQL Workbench** pro vizuální modelování, SQL vývoj a administraci
- (20)

Komerční licence

Komerční licence je nutné využít v případě, že MySQL je dále prodáváno jako produkt, nebo integrální součást produktu (tedy produkt samotný by bez této databáze nemohl existovat). Tato licence v sobě zahrnuje i poskytnutí technické podpory a další služby.

| Verze | Cena roční licence pro 1-4 socketový server |
|--|--|
| MySQL Standard Edition | 2.000,- USD/rok |
| MySQL Enterprise Edition | 5.000,- USD/rok |
| MySQL Cluster Carrier Grade Edition | 10.000,- USD/rok |

Tabulka 3 Ceník komerčních licencí MySQL pro rok 2013 dle (18)

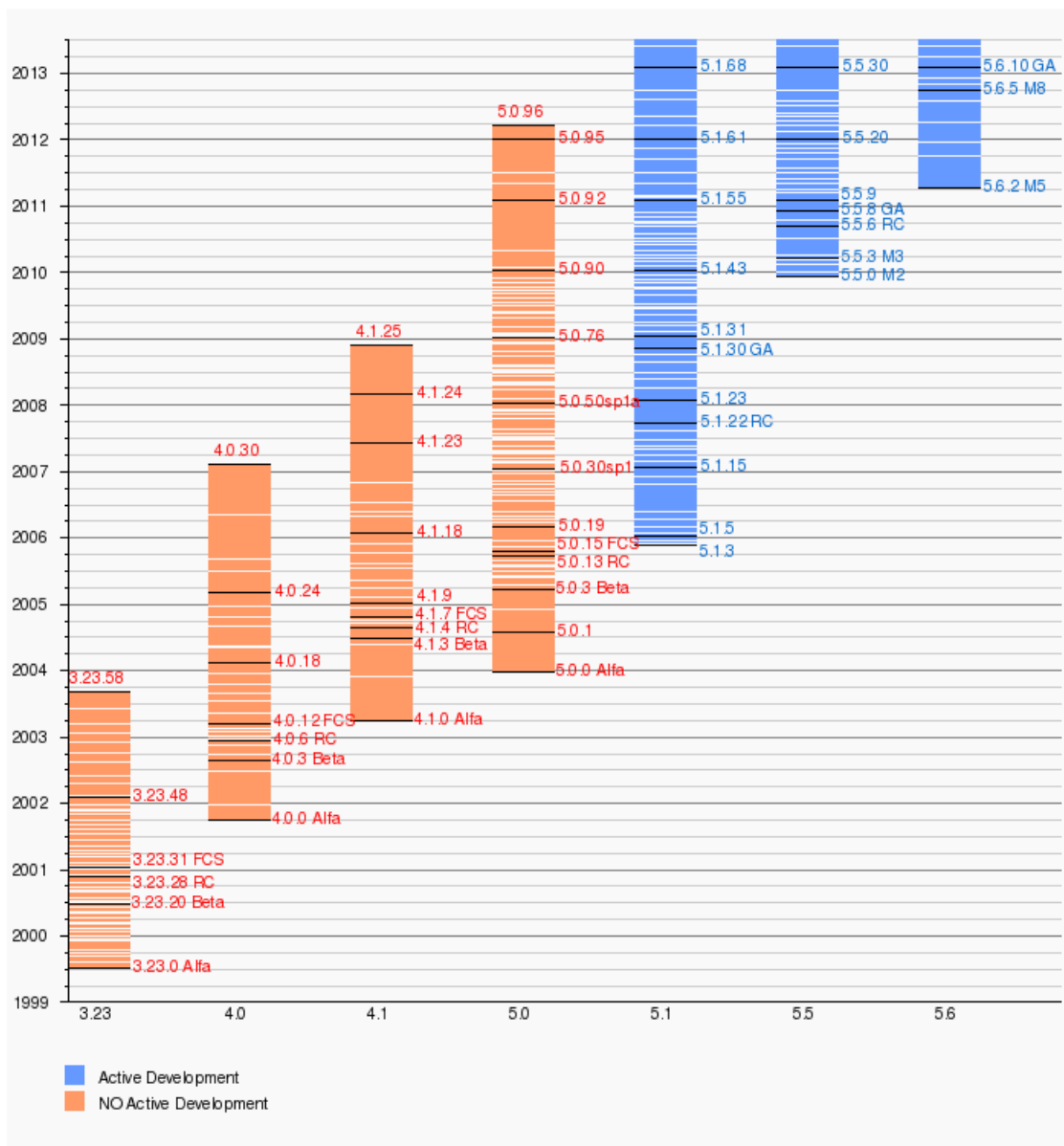
3.3.3 Aktuální verze

Aktuálně dostupná verze MySQL, která bude dále použita v praktické části je 5.6.15. Oracle již avizoval verzi 5.7, která je zatím k dispozici ke stažení v balíčcích pro vývojáře a není ještě označena jako stabilní (psáno prosinec 2013).

Verze 5.7 přinese optimalizaci úložných enginů pro určité druhy operací a zároveň odstraní některé limity. Co se týče přijímaných dialektů SQL, přináší nově možnost mít více triggerů na stejné operace nad tabulkou.

3.3.4 Historie

Jak již bylo řečeno, na MySQL se pracuje od roku 1994, 23. května 1995 byl k dispozici první interní build a od konce roku 1996 byla k dispozici pro veřejnost verze 3.19 - více k historii jednotlivých verzí viz Obrázek 3. Jak je na něm dobře vidět, v současné době podporované major verze jsou 5.1, 5.5 a 5.6.

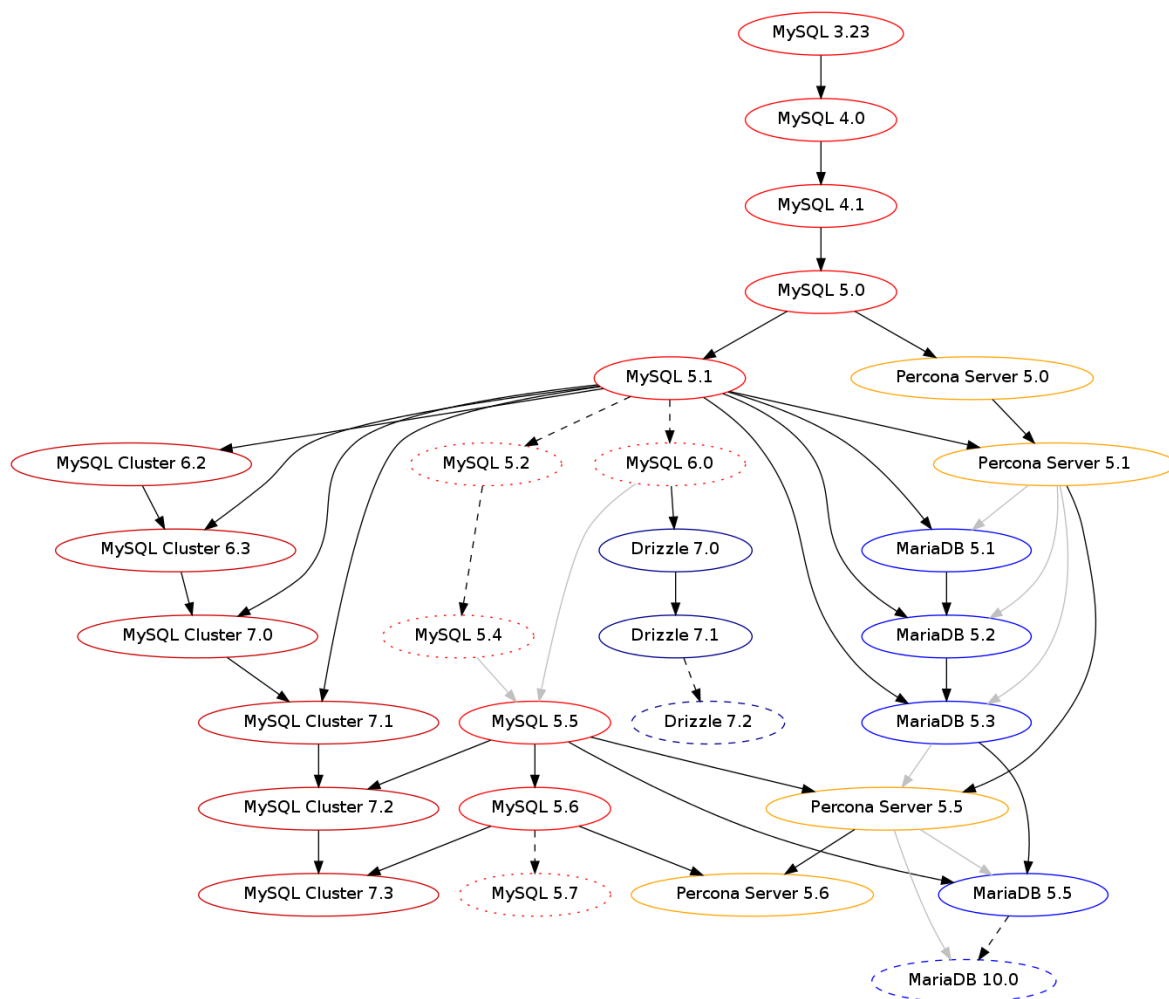


Obrázek 3 Historie vývoje verzí MySQL (zdroj Wikimedia Commons)

3.3.5 Forky (odvozené větve vývoje)

Od MySQL bylo odvozeno větší množství dalších vývojových větví – forků.

Některé z nich, včetně historie vývoje shrnuje Obrázek 4.



Obrázek 4 Vývoj MySQL a některých forků, autor Daniël van Eeden

Mezi důležité forky patří **Percona Server**, který vznikl zejména jako snaha o zrychlení práce nad většími objemy dat. Tohoto dosahují úpravou storage engine InnoDB, která se nazývá XtraDB. (23)

Další fork, který stojí za zmínku je **MariaDB**⁵. V roce 2009 Michael 'Monty' Widenus – jeden z původních autorů MySQL - odešel po prodeji své společnosti MySQL AB ze Sunu, aby se věnoval novému forku této databáze. V roce 2012 se přidávají David

⁵ Jméno opět odkazuje na stejnou Marii jako u MySQL

Axmark a Allan Larsson – zbylí z trojice původních vývojářů a představují **MariaDB Foundation**, která si klade za cíl bdít nad vývojem MariaDB.



Obrázek 5 Logo MariaDB

MariaDB je určitou odpovědí na otázky, co se stane s MySQL pod taktovkou Oracle. Protože zatímco MySQL Community edition zůstává (zatím) open source, Oracle přestal například zveřejňovat testovací scénáře pro jednotlivé reportované chyby.

MariaDB začíná svoje úspěšné tažení zejména na poli Linuxových distribucí, kde například Fedora ohlásila, že kvůli politice otevřenosti nahradí od verze 19 MySQL ve svých repozitářích právě forkem MariaDB (24).

V roce 2013 se MariaDB foundation, která poskytuje MariaDB pouze pod GPL licenci spojila se společností SkySQL, která začala pro MariaDB nabízet placenou podporu. (25)

Co se týče verzování, až do verze 5.5 MariaDB sleduje verzování MySQL, poté ovšem přechází na nové číslování, které začne verzí 10. Tato major verze se zatím nachází ve vývojářském stádiu. Vzhledem ke vzrůstající rozšířenosti a oblibě, budou testy v praktické části této práce provedeny i na MariaDB verzi 5.5.

Oba zmíněné forky jsou binárně kompatibilní s příslušnou verzí MySQL, tedy operace nad stejnými daty mají naprosto stejný výsledek jako MySQL. Tímto se snaží zaručit i snadnou migraci mezi jednotlivými databázovými enginy.

Příkladem forku, pro který toto neplatí, může být **Drizzle**. Ten se snaží odlehčit MySQL od funkcí, které nejsou potřeba pro vývoj jednoduchých webových stránek, jako jsou například uložené procedury, views, query cache a další.

3.3.6 Administrační nástroje

Nejpoužívanějšími administračními nástroji pro přímý přístup do databáze (tedy ne volání v rámci nějakého programu, informačního systému a podobně) budou pravděpodobně MySQL Workbench – klient od společnosti MySQL AB, který je dodáván

spolu s MySQL databází (opět zde existuje volná i placená verze). Z „tlustých“ klientů je možné zmínit například HeidiSQL, multiplatformní Navicat, ale k MySQL se umí připojit například i MS Access.

Z tenkých klientů za zmínku stojí phpMyAdmin, který je také součástí balíku LAMP, nebo projekt českého programátora Jakuba Vávry Adminer (ten je zajímavý tím, že jeho kompletní programový kód je pouhý jeden PHP soubor).

MySQL také nabízí textového klienta pro shell, s prostým pojmenováním mysql.

3.3.7 Omezení

Omezení MySQL jsou z velké části dána použitým storage enginem. Dále závisí na použitém systému souborů a mnoha konfiguračních možnostech. Některá tato omezení přehledně shrnuje Tabulka 4 a Tabulka 5. Například InnoDB zvládne obsloužit až 4 miliardy tabulek v jedné databázi.

Dále je omezení definice tabulky dáno maximální velikostí souboru typu .frm (činí 64 kB), který popisuje strukturu tabulky.

| Operační a souborový systém | Omezení velikosti souboru |
|------------------------------------|------------------------------------|
| Win32 w/ FAT/FAT32 | 2GB/4GB |
| Win32 w/ NTFS | 2TB (v některých případech i více) |
| Linux 2.2-Intel 32-bit | 2GB (LFS: 4GB) |
| Linux 2.4+ | 4TB (při použití ext3) |
| Solaris 9/10 | 16TB |
| Mac OS X w/ HFS+ | 2TB |

Tabulka 4 Omezení velikosti MySQL souborů dle souborového systému

| Typ limitu | Limit |
|--|---|
| Maximální velikost řádku | 65.535 bytů Bez ohledu na engine a bez započítání sloupců některých typů (např. text, blob mohou mít až 4 GB). |
| Maximální velikost pole | Opět dle typu – například celková délka všech varchar polí nesmí přesáhnout výše uvedených 65.535 bytů. |
| Maximální počet řádků tabulky | Obecně neomezeno. Tabulka je omezená maximální velikostí souboru a může být omezena některými dalšími věcmi – například délkou autoinkrementovaného primárního klíče. |
| Maximální počet sloupců v tabulce | Engine InnoDB umožňuje mít v tabulce až 1.000 sloupců . Efektivní omezení závisí na použitých datových typech; maximální velikost řádku je 65.535 bytů, což dává cca 85 sloupců typu varchar(255) v UTF-8. Do tohoto limitu se nezapočítávají některé typy sloupců (text, blob), které se uchovávají jinak. |
| Maximální počet indexů na tabulce | Nejméně 16 – všechny enginey. Pro InnoDB 64 (bez započtení primárního) |

Tabulka 5 Některá omezení MySQL, dle (9)

3.3.8 Škálování

MySQL nabízí škálování ve formě produktu MySQL Cluster. Jedná se vlastně o sdílený engine jménem NDB (zkratka z Network Database), který využívá dvoufázový commit a dosahuje tak synchronní replikace.

Časté je také využití asynchronní replikace (použití je například pro různě umístěné CDN servery), které je označováno jako „MySQL Cluster Replication“ nebo také "geographical replication". (26)

Minimální konfigurace pro toto použití obsahuje 3 stroje a to buď ve formě

- 2 × Datový uzel
- 1 × Aplikace + management uzel

Případně ve formě

- 2 × Datový uzel+ aplikace
- 1 × Management uzel

Oracle spolu s Intelem provedli serii benchmarků, aby vyzkoušeli výkon a škálovatelnost řešení MySQL Cluster na komoditních serverech vybavených procesory z rodiny Intel® Xeon® E5 poslední generace, podařilo se tak dosáhnout výkonu 4.3 miliardy plně konzistentních čtení (SELECT) za minutu a 1.2 miliardy plně transakčních zápisů (UPDATE) za minutu. (27)

3.3.9 Přijímaný SQL Dialekt

MySQL se dle dokumentace (28) snaží o vyhovění SQL standardům už od standardu SQL-92. Ovšem ne na úkor použitelnosti a rychlosti databáze. Také se nebojí rozšiřovat přijímaný dialekt o nové výrazy tam, kde vylepší použitelnost. Jako příklad uvádí proprietární příkaz HANDLER, který slouží jako přístup k nastavení jednotlivých storage enginů.

Aby mohli podporovat jak kritické aplikace, tak naopak ty, které potřebují rychlé úložiště pro logování, případně pro rychlé použití na webu, podporují jak transakční, tak netransakční zpracování.

Dále uvádí, že podporují standardy ODBC⁶ úrovně 0 až 3.51.

Ačkoliv bylo původně designováno pro středně velké databáze, dnes má nasazení až o velikosti 1 TB. Přičemž si ale stále uchovává schopnost být optimalizováno i na velikost použití v přenosných zařízeních.

⁶ ODBC se specializuje na middleware API pro přístup k databázím.

Pro větší adhezi k standardům se dá MySQL pustit v takzvaném „ansi mode“ a to buď parametrem

```
Mysqld -ansi
```

Což je vlastně alias k

```
--transaction-isolation=SERIALIZABLE --sql-mode=ANSI
```

Případně i při běhu programu následujícím SQL scriptem:

```
SET GLOBAL TRANSACTION ISOLATION LEVEL SERIALIZABLE;  
SET GLOBAL sql_mode = 'ANSI';
```

Dokumentace v tomto bodě bohužel neříká, na jakou konkrétní verzi standardu ANSI se tímto krokem přepíná (27). Nicméně, použití je dobré zejména tam, kde se do budoucna předpokládá možná nutnost přenosu dané aplikace i na jinou databázi a proto v ní není vhodné používat mysql proprietární doplňky.

Co se týče rozšíření SQL standardu, MySQL nabízí přímo tomuto tématu dedikovanou sekci dokumentace (29).

Za zmínku stojí například příkaz SHOW pro snadnější práci s informacemi v information schematu – použití například

```
SHOW TABLES
```

vrátí SQL seznam všech tabulek a views ve zvolené databázi.

Dále pak například možnost psát příkazy UPDATE ve stejném stylu jako INSERT tedy:

```
UPDATE tabulka SET (sloupec1,sloupec2) VALUES  
(hodnota1,hodnota2) WHERE ...
```

Jako doplněk ke standardnímu

```
UPDATE tabulka SET sloupec1=hodnota1,sloupec2=hodnota2  
WHERE...
```

Také nabízí možnost ovlivnit pořadí jednotlivých sloupců v tabulce příkazy typu

```
UPDATE tabulka ADD COLUMN sloupec AFTER jiny_sloupec
```

Což na jednu stranu odporuje představě, že na pořadí sloupců nemá záležen, případně si je určí vývojář pomocí vyjmenování v příkazu SELECT, nicméně zdá se, že nachází uplatnění.

3.3.10 Výhrady k MySQL

MySQL ve výchozí konfiguraci obsahuje některé změny oproti SQL standardům.

Jedná se například o použití operátoru pro spojování řetězců dvě svislé čáry - || jako alias k OR, jak je zvykem jej používat v jiných programovacích jazycích. Na jeho místo zavádí funkci CONCAT o libovolném množství parametrů. V MySQL se tak standardní

```
"řetězec1" || "řetězec2" || "řetězec3"
```

Píše jako

```
CONCAT("řetězec1", "řetězec2", "řetězec3")
```

Další výhradou je chování k cizím klíčům v storage enginech mimo InnoDB (které se k nim chová správně). MySQL při použití engine MyISAM definici těchto klíčů ani nepoužije, ani neuloží, přičemž ani nezahlásí error syntaxe. Zřejmě je tak proto, že parser SQL syntaxe nemá přístup k informaci o tom, jaký storage engine je použit pro danou tabulku.

Obecně je také možné kvůli možnosti přítomnosti tabulek s více různými storage enginy dosáhnout stavu, kdy transakční rollback nevrátí databázi do stavu před startem transakce (30).

3.4 PostgreSQL

PostgreSQL⁷ je výkonný, opensourcový relační databázový systém. Má za sebou více než 15 let aktivního vývoje a silnou architekturu, který mu zajistila silnou reputaci. Mezi hlavní přednosti patří spolehlivost, datová integrita a správnost. Lze jej provozovat na všech hlavních operačních systémech ať už Windows, Linux, či UNIX (např AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64).

Plně vyhovuje standardům ACID, nabízí podporu pro cizí klíče, joiny, view, triggerly i uložené procedury (dokonce ve více programovacích jazycích). Obsahuje většinu SQL:2008 datových typů včetně INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL a TIMESTAMP. Také nabízí uložení velkých binárních objektů, jako jsou obrázky, zvuky nebo video. Má nativní programovací interface pro C/C++, Javu, .Net, Perl, Python, Ruby, Tcl, ODBC a další. Také nabízí výtečnou dokumentaci.



Obrázek 6 Logo PostgreSQL

PostgreSQL také nabízí další funkčnosti, které se často připisují databázím označovaným jako enterprise. Mezi tyto funkčnosti patří Multi-Version Concurrency Control (MVCC), point in time recovery, tablespaces, asynchronous replication, nested transactions (savepoints), online/hot backups, a sophisticated query planner/optimizer a write ahead logging pro prevenci chyb. Dále nabízí mezinárodní znakové sady, multibyte character encoding, podporu pro Unicode a třídění dle místních nastavení (např. v češtině znak CH až za H – přestože první písmeno je vlastně C), case sensitivity a formátování.

⁷ Čte se 'poust.gres ,kju: 'el

PostgreSQL je systém škálovatelný jak ve velikosti dat, tak v počtu současně připojených uživatelů. Největší produkční nasazení tohoto systému přesahuje velikost 4 TB dat (31), největší databázi založenou na PostgreSQL je ale zřejmě databáze Yahoo, která má až 2 PB (tedy 2 tisíce TB) dat, ta je ale již silně modifikovaná od původního systému a PostgreSQL je použit jen pro nejvyšší přístupovou vrstvu k datům (32).

3.4.1 Zastřešující organizace

The PostgreSQL Global Development Group není organizací v pravém slova smyslu. Je to otevřená komunita, do které může přispívat každý, podobně, jako je to například při vývoji Linuxu.

Finance jsou získávány formou sponzoringu; z těch nejdůležitějších sponzorů jmenujme například EnterpriseDB, 2ndQuadrant, Red Hat, Coomand Prompt, Dalibo, Conova,HP, Huawei, Skype, Google.

3.4.2 Licence

PostgreSQL je vydávána pod licencí **PostgreSQL License** a **liberal Open Source license** (jsou velmi podobné známějším BSD a MIT licencím). Tato licence dává každému právo používat, kopírovat, měnit a distribuovat tento software i jeho dokumentace k jakémukoliv účelu, bez poplatku a bez sepsané smlouvy. Jediné, co je vyžadováno, je přibalení následující noticky ke všem kopiím, kde se *University of California* (kde vývoj PostgreSQL započal) zříká jakékoliv zodpovědnosti:

Copyright pro části vyvinuté od roku 1996 drží *The PostgreSQL Global Development Group*, pro části z roku 1994-6 *The Regents of the University of California*.

THE UNIVERSITY OF CALIFORNIA ("KALIFORNSKÁ UNIVERZITA") NENÍ V ŽÁDNÉM PŘÍPADĚ ODPOVĚDNA ŽÁDNÉ TŘETÍ OSOBE ZA PŘÍMOU, NEPŘÍMOU, ZVLÁŠTNÍ, NAHODILOU NEBO VÝSLEDNOU ŠKODU, VČETNĚ UŠLÉHO ZISKU, ZPŮSOBENOU UŽITÍM TOHOTO SOFTWARE A DOKUMENTACE K NĚMU, A TO I V PŘÍPADĚ, ŽE THE UNIVERSITY OF CALIFORNIA BYLA INFORMOVÁNA O MOŽNOSTI VZNIKU TAKOVÉ ŠKODY.

THE UNIVERSITY OF CALIFORNIA ZEJMÉNA NEPOSKYTUJE JAKÉKOLI ZÁRUKY, A TO NEJEN ZÁRUKY OBCHODOVATELNOSTI A VHODNOSTI TOHOTO

VÝROBKU KE SPECIFICKÝM ÚČELŮM. NÍŽE UVEDENÝ SOFTWARE JE POSKYTNUT "JAK STOJÍ A LEŽÍ" A THE UNIVERSITY OF CALIFORNIA NENÍ POVINNA ZAJISTIT JEHO ÚDRŽBU, PODPORU, AKTUALIZACI, VYLEPŠENÍ NEBO MODIFIKACI. (33), překlad Pavel Stěhule.

Lze tedy nahlédnout, že tato licence je mnohem svobodnější, než obě formy licence u MySQL. I proto se někdy označuje licenční rozdíl mezi těmito dvěma databázemi okřídleným rčením, že MySQL je opensourcový produkt, zatímco PostgreSQL opensourcový projekt.

Spolu se samotnou PostgreSQL obsahuje základní instalační balík další sadu podpůrných nástrojů pro práci s touto databází, jedná se zejména o:

- **PgAdmin** – nástroj pro správu databází a dat, který má přehledné GUI
- **stackbuilder** – pro instalaci doplňků třetích stran do PostgreSQL
- **pgbench** – nástroj pro měření výkonu databáze založený na TPC-B testu
- **pg_upgrade** – nástroj pro upgrade jednotlivých databázových clusterů na novější verze databáze,
- **pg_dump** – nástroj, který umožňuje uložit databázová data, včetně struktury celé databáze do binární podoby pro potřeby zálohování, či přenosu
- **pg_restore** – nástroj pro obnovení databáze z dumpu získaného pomocí pg_dump,
- **pg_test_timing** – nástroj pro měření správnosti dat operačního systému z hlediska času – například, jestli se čas v systému nikdy neposunul zpět, což by mohlo způsobit nekonzistenci v datech
- **pg_config** – nástroj pro zjištění aktuální konfigurace PostgreSQL serveru
- **pg_ctl** – nástroj pro správu PostgreSQL serveru

3.4.3 Aktuální verze

Aktuální stabilní major verze je 9.3, která byla vydaná 9. 9. 2013. Oproti předchozí verzi přinesla například lateral join, trigger na DDL události (vytvoření tabulky a podobně), materializované pohledy (views) a updatovatelné pohledy (views)⁸.

Komunita PostgreSQL obvykle vydává jednu major verzi za rok a další (9.4) je naplánována na třetí čtvrtletí roku 2014, jedinou ze zatím potvrzených změn, které bude obsahovat je možnost měnění některých konfiguračních direktiv přímo pomocí SQL dotazů.

Podporovanými verzemi (leden 2013) jsou 8.4, 9.0, 9.1, 9.2 a 9.3 v aktuálních minor verzích, více viz Obrázek 7

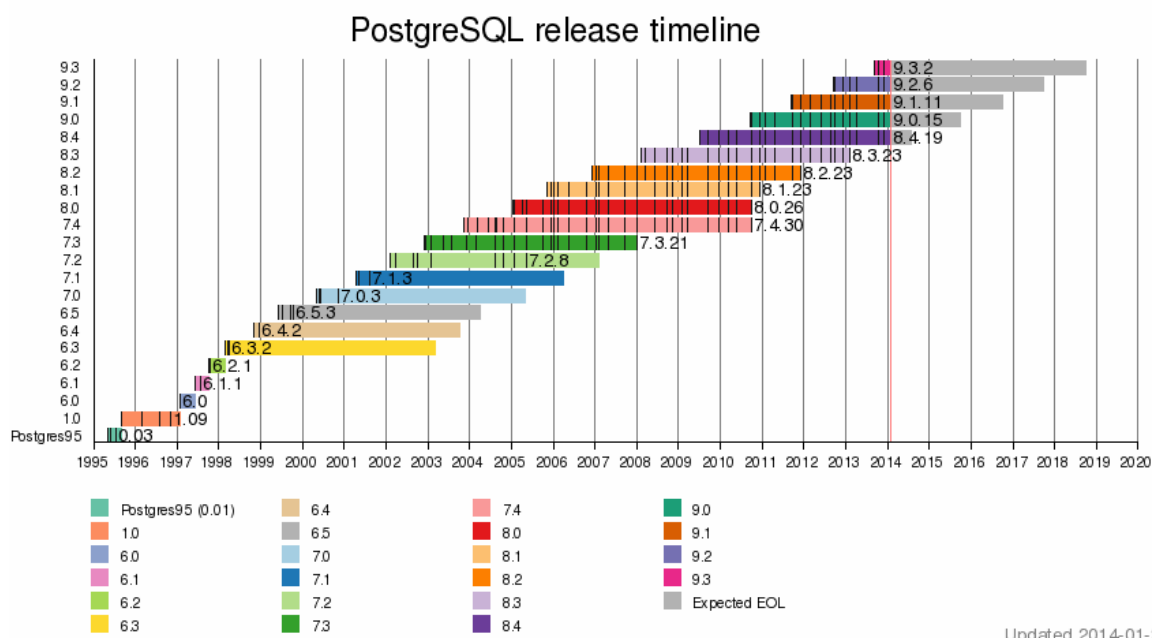
3.4.4 Historie

Vývoj předchůdce PostgreSQL INGRES (zkratka z Interactive Graphics Retrieval System) začal na University of California v Berkley již v sedmdesátých letech. V roce 1985 se Michael Stonebraker (výzkumník v oblasti databází) pouští do vedení projektu Post-Ingres, který si kladl za cíl odstranit některá omezení Ingresu. Zkráceně se tedy tomuto systému začalo říkat **Postgres**. Postgres již byla relační databáze, tj. rozuměla vztahům mezi tabulkami, umožňovala vytvářet složitější datové struktury a vlastní datové domény. V roce 1993 byla výzkumníky na Kalifornské univerzitě vydána verze 4, kterou tam projekt skončil. Protože byly zdrojové kódy uvolněny pod MIT licenci, mohli postgraduální studenti Andrew Yu a Jolly Chen nahradit původní dotazovací jazyk QUEL rozšířenějším SQL. Pod jménem Postgre95 již publikovali tento databázový systém na webu.

Roku 1996 se Postgres přejmenoval na PostgreSQL, když bylo již jasné, že Postgres95 je jméno, které neodolá zubu času, a také aby bylo již z názvu patrné, že

⁸ Existují domněnky, že k tomuto kroku komunita PostgreSQL přistoupila, aby bylo možné nad touto databází provozovat systémy SAP. (zdroj <http://www.root.cz/clanky/novinky-v-pripravovanem-postgresql-9-3/>)

podporuje SQL. Od roku 1996 se také vývoj přesouvá mimo zdi univerzity a stabilizuje se převzatý kód, který je v lednu 1997 vydán pod názvem PostgreSQL 6.0. Od této doby se o vývoj PostgreSQL stará komunita organizovaná přes internet.



Obrázek 7 Historie verzí PostgreSQL (zdroj Wikimedia Commons)

3.4.5 Forky (odvozené databáze)

Vzhledem k otevřené licenční politice a Open Source přístupu ke zdrojovým kódům, vzniká velké množství forků PostgreSQL. Některé časem odumírají, případně se jejich kód stane součástí příští verze, jiné dále pokračují vlastním směrem a od původního PostgreSQL se více odchyľují.

Příkladem zaniklých forků může být například **NuSphere UltraSQL**. Tento fork byl odvozen z PostgreSQL verzí 7.2 a 7.3 a přinášel podporu pro běh pod Windows32 (rok 2002, 2003). Kód byl následně včleněn do verze 8.0. Dalším příkladem ze stejné doby je **Red Hat Database**. Přinášela lepší administrační nástroje, které byly opět později využity v základním projektu (například visual query analyser pro pgAdmin).

Ze stále žijících projektů lze jmenovat například **Fujitsu Supported PostgreSQL**, což je fork, který obsahuje vlastní storage engine. Fujitsu také poskytuje uživatelskou

podporu, konzultace, obnovu dat, hosting, migrace, školení a systémovou integraci pro PostgreSQL.



Obrázek 8 Logo Fujitsu supported PostgreSQL

Dále **PostGIS** (dříve samostatný fork, dnes ve formě Add-on k hlavní vývojové větvi PostgreSQL), který přináší podporu pro geografické operace. Je používán například ve Francouzském národním geografickém institutu, nebo britské společnosti InfoTerra.



Obrázek 9 Logo PostGIS

Zajímavým projektem je také **BigSQL**, které přináší spojení PostgreSQL s technologií Hadoop, pro distribuovaná spojení a rozprostírání zátěže na více počítačů.

Ze stejné oblasti stojí za zmínku ještě fork **Everest**, který není veřejně k dispozici, vlastněný společností Yahoo. V současnosti dosahuje velikosti více než 2PB, v roce 2008 dosahoval 24 miliard operací denně a běží na více než 1000 strojích. (32) (34)

3.4.6 Administrační nástroje

Pro PostgreSQL je k dispozici velké množství různých administračních nástrojů založených na různých technologiích.

Základním klientem je **psql** – klient pro příkazovou řádku, kromě spouštění SQL příkazu poskytuje i administrační rozhraní samotného databázového systému, nápovědu, scriptování, automatické doplňování jmen objektů a další.

Z tlustých klientů je možno jmenovat opensourcový **pgAdmin** – vyvíjený taktéž komunitou, nebo proprietární **EMS SQL Manager for PostgreSQL**, oba tyto klienty

umožňují pokročilé datové modelování (např. Visual database designer a Visual Query builder), importování, exportování a transformace dat.

Pro PostgreSQL také vznikl fork slavného phpmyadminu jménem **phpPgAdmin**. V roce 2002 byl jeho kód od základu přepsán a jedná se tak již o samostatný produkt, který ale umožňuje vše, co jeho protějšek pro MySQL.

3.4.7 Omezení

Reálná omezení pro PostgreSQL jsou dána použitým operačním systémem, velikostí paměti, síťovou propustností a dalšími hardwarovými a softwarovými vlivy.

Některá omezení vyplývající ze samotné architektury PostgreSQL jsou shrnuta v Tabulka 6.

| Typ limitu | Limit |
|-----------------------------------|--|
| Maximální velikost databáze | Bez omezení |
| Maximální velikost tabulky | 32 TB |
| Maximální velikost řádku | 1.6 TB |
| Maximální velikost pole | 1 GB |
| Maximální počet řádků tabulky | Bez omezení |
| Maximální počet sloupců v tabulce | 250 – pro sloupce kolem 8 KB velikosti až 1.600 pro jednodušší datové typy. |
| Maximální počet indexů na tabulce | Bez omezení |

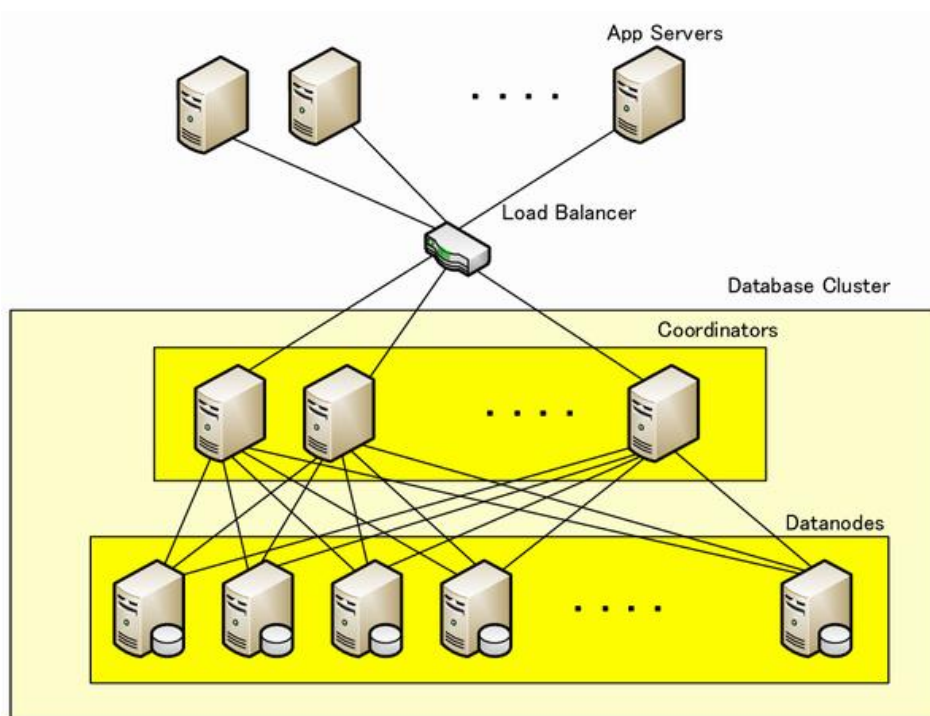
Tabulka 6 Některá omezení PostgreSQL dle (31)

3.4.8 Škálovatelnost

PostgreSQL jako takový nabízí sadu nástrojů pro škálování zejména z pohledu vysoké dostupnosti aplikací. Jedná se například o nástroj „log shipping“, který nabízí streaming replication, cascading replication a synchronous replication. Je možné takto nakonfigurovat prostředí, kde když selže jeden stroj, na kterém běží databáze, jeho úlohu

automaticky přebírá záložní, aniž by běžný uživatel poznal výpadek, toto řešení se také často označuje jako „warm standby“ (35).

Protože PostgreSQL (ani MySQL) nenabízí intraquery (tedy uvnitř-dotazový) paralelismus, vznikl na jeho základech projekt Postgres-XC, který nabízí podporu pro škálování z pohledu práce nad obrovskými daty. Přináší zlepšení zejména při provádění analytických dotazů a umožňuje zapojit potenciálně až 30 serverů naráz, potenciální schéma funkce tohoto řešení ukazuje Obrázek 10. (36)



Obrázek 10 Schéma funkce Postgres-XC clusteru dle (36)

3.4.9 Přijímaný SQL Dialekt

Jádro PostgreSQL pokrývá části 1, 2, 9, 11, a 14 SQL-2011 standardu (který popisuje Kapitola 3.2 - Tabulka 2). Část 3 je pokrytá ODBC driverem a část 13 pomocí PL/Java plug-inu, ale ani pro jednu z těchto částí se dnes nezkoumá a neověřuje úplně přesná shoda. Části 4 a 10 PostgreSQL neimplementuje vůbec.

PostgreSQL splňuje většinu hlavních požadavků standardu SQL:2011. Ze 179 povinných částí jich PostgreSQL splňuje 160. Kromě toho splňuje také mnoho požadavků,

kteře jsou označeny jako nepovinné. PostgreSQL dokumentace dále tvrdí, že žádná jiná databáze nesplňuje tento standard v jeho úplnosti (15).

PostgreSQL v dokumentaci předkládá velmi podrobné tabulky jak těch funkcí, které podporuje, tak těch, které ne. Také je nutné zmínit, že mnoho z těch částí, ve kterých standardu nevyhovuje, řeší některým ze svých rozšíření. Jako příklad je možno uvést namátkou požadavek *F202 - TRUNCATE TABLE: identity column restart option* (restart hodnoty identifikačního sloupce jako parametr příkazu TRUNCATE), který se dá řešit SELECTem příslušné sekvence a nastavení jejího parametru - nextval (příští použitá hodnota sekvence).

Příkladem vlastního rozšíření SQL syntaxe může být příkaz SELECT DISTINCT ON. SQL standard specifikuje pro příkaz SELECT možnost parametru DISTINCT, který způsobí, že řádky, které mají stejné hodnoty ve všech sloupcích, se zobrazí jen jednou – bez tohoto parametru pak právě tolikrát, kolikrát v tabulce jsou. Rozšíření DISTINCT ON umožňuje určit jen ty sloupce, které se mají pro toto porovnání unikátnosti použít.

Dalším příkladem je například rozšíření klauzule UPDATE o parametry FROM, který umožňuje provést UPDATE jen na některé řádky tabulky, které vrátí část příkazu za FROM (zapisuje se stejně jako standardní SELECT) a RETURNING, který pokud je použit, modifikuje příkaz UPDATE tak, že vrátí jako výsledek příkazu modifikované řádky tak, jak by je vracel příslušný SELECT - ušetří se tak jeden dotaz do databáze.

Pravděpodobně nejvyužívanějším rozšířením je umožnění parametrů LIMIT a OFFSET přímo do standardního zápisu příkazu SELECT – tedy bez definice kurzoru, jak by to v tomto případě formuloval standard.

3.4.10 Výhrady k PostgreSQL

Existuje několik věcí, ve kterých se PostgreSQL od standardu odchyluje. Jako příklad je možno uvést konverzi neuvozovkovaných literálů na malá písmena – uvozovkovaný literál se bere tak, jak je, tedy se zřetelem k velikosti písmen a nezkoumá se, jestli literál není klíčovým slovem. Identifikátor, IDENTIFIKÁTOR ale i IDENTiFIKátor jsou tedy hodnoceny jako shodné s "identifikátor", ale ne "IDENTIFIKÁTOR", což jde proti standardu, který vyžaduje konverzi na velká písmena.

Další problémy může způsobit, že definici UNICODE PostgreSQL chápe jako UTF-8 (což je dlužno dodat asi pravděpodobně nejčastější použití unicode), ale může způsobit problémy při komunikaci se systémy v UNICODE UTF-16.

PostgreSQL také chápe každý neoznačený integer jako 4 bitový, jednoduchý násobek:

```
SELECT 256 * 256 * 256 * 256;
```

způsobí

```
ERROR: integer out of range
```

Toto je možno snadno řešit přetypováním, nicméně může nezkušeného programátora snadno zmást.

3.5 Metodika měření výkonu databáze

Tato kapitola se zabývá metodikami měření výkonu databáze, společnostmi a softwarem, který se k tomuto měření dá použít.

3.5.1 TPC

Transaction Processing Performance Council (TPC) je nezisková organizace založená roku 1988. Za poslední dvě dekády, měla organizace TPC nezanedbatelný vliv na používání oborových benchmarků v oboru počítačů. Výrobci používají TPC benchmarky, aby ukázali, jak si vedou jejich existující produkty v porovnání s jejich konkurenty a také, aby monitorovali a zlepšovali výkonnost produktů, které právě vyvíjí. Mnoho nakupujících využívá výsledky TPC benchmarků jako jedno z hledisek při rozhodování o nákupu nových počítačových systémů.

Oblast informačních technologií se rapidně vyvíjí a klade vysoké nároky na výzkumníky i ostatní pracovníky, aby vytvářeli inovativní techniky sloužící k hodnocení, měření a charakterizování komplexních systémů. TPC je i nadále odhodlána vyvíjet nové benchmarky tak, aby držela krok s těmito rychlými změnami. Jednou z cest jak toho dosáhnout je pořádání Technology Conference on Performance Evaluation and Benchmarking (TPCTC). Na této konferenci umožňuje TPC prezentovat nové nápady a debatovat mnoha odborníkům jak z průmyslu, tak výzkumu (37).

Členové

Jak již bylo řečeno, TPC je příspěvkovou organizací, která sdružuje množství firem zabývajících se databázovým zpracováním dat. Zakladateli jsou Omri Serlin – ředitel ITOM International, Co, který přesvědčil prvních 8 společností k založení TPC, a Tom Sawyer – první TPC auditor a autor prvního TPC benchmarku, další členi viz Obrázek 11.

Full Members

| | | | | |
|---|---|---|--|---|
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  | | | | |

Obrázek 11 Někteří členové TPC (Zdroj (38))

Aktuální typy testů

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-objednávka). 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 (39).

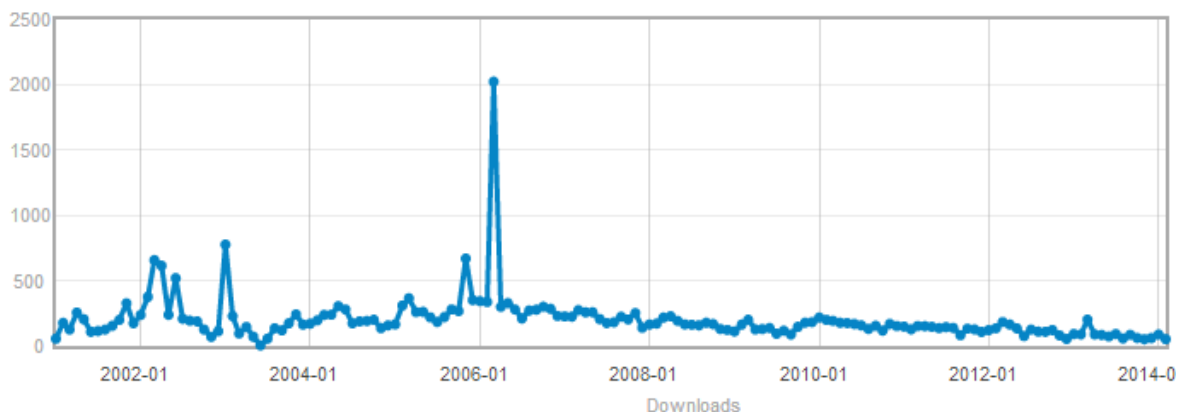
Aktuálně používané metriky (rok 2014) jsou

- TPC – C, která simuluje objednávkový systém a zátěž od více uživatelů současně, vyznačuje se vysokými nároky na diskové přenosy, transakční integritu, a více tabulek naráz o různých množstvích obsažených dat.
- TPC – DS, simuluje databáze pro podporu rozhodování (decision support, odtud zkratka) nad rozsáhlými daty. Vyznačuje se velkými datovými objemy, dotazy o značné komplexnosti (ad-hoc, reportování, OLAP, data-mining) a velkou zátěží na procesor a vstupy a výstupy.

- TPC-E, simuluje databázi brokerské firmy, která musí zvládat velké množství „malých“ dotazů - provedení obchodu. Zároveň také obsahuje některé dotazy pro analýzu trhu. Jednotkou tohoto testu je počet uzavřených transakcí za sekundu (tps),
- TPC – H, která je také metrikou pro podporu rozhodování a zaměřuje se zejména na práci s konkurenčními transakcemi, která modifikují stejná data. Jednotkou je kompozitní počet vykonaných dotazů za hodinu v závislosti na velikosti databáze (QphH@Size),
- TPC – VMS (Virtual Measurement Single System Specification), která přináší metodiku, jak otestovat výše zmíněné testy současně. Provádí se zejména pro otestování výkonu strojů, které slouží pro virtualizaci. Vytvoří se 3 virtuální stroje se stejně nastavenými konfiguracemi databáze a pustí zároveň TPC – C,E a DS (případně H). Primární metrikou je nejnižší výsledek z těchto 3 virtualizovaných databází,
- TPC – Energy, která se zabývá energetickou náročností běhu jednotlivých databází. Měří se spotřeba serveru, diskových polí a dalších součástí, které jsou nezbytné pro provedení testů.

3.5.2 The Open Source Database Benchmark

Protože získání oficiálních výsledků TPC může být finančně nákladná záležitost – TPC totiž poskytuje metodiku zdarma, nástroje a konzultace jsou ale placené - vznikla i open source snaha o totéž - The Open Source Database Benchmark (dostupné na <http://osdb.sourceforge.net/>), zde je bohužel nutno konstatovat, že tato snaha zřejmě postupně odumírá, dokumentace je z roku 2002 a poslední commit do repositáře z roku 2012. Také počet stažení se pomalu, ale jistě snižuje.



Obrázek 12 Počet stažení Open Source Database Benchmark (zdroj sourceforge.net)

Tento komunitní projekt nabízí nástroj k porovnání výkonů databází, ale na svých stránkách žádné výsledky nepublikuje, což omlouvá tím, že někteří databázoví vydavatelé nedávají k publikaci takovýchto výsledků licenční svolení.

Benchmarkovací nástroj je ke stažení ve formě zdrojových kódů v jazyce C++. Aktuálně nejnovější verze ke stažení je 0.9.

3.5.3 Standard Performance Evaluation Corporation (SPEC)

SPEC je nezisková organizace, která vznikla, aby vytvořila, udržovala a prosazovala standardizovaný set relevantních benchmarků, které mohou být aplikovány na nejnovější generace vysoce výkonných počítačů. SPEC vytváří benchmarkovací scénáře a také poskytuje revize výsledků, které jim zasílají jejich členské organizace.

Mezi členy patří většina velkých softwarových a hardwarových společností, například: Acer Inc., Amazon Web Services, Inc., Apple Inc., ARM , Cisco Systems, Inc., Dell, Inc., Hitachi Ltd., HP ,Intel, Lenovo, Microsoft , NVIDIA, Oracle , Red Hat, Samsung, SAP AG , SUSE, Symantec Corporation a Twitter, Inc.

Dále se zde angažují univerzity a výzkumné týmy z celého světa. (38)

SPEC poskytuje testovací nástroje, které se zaměřují zejména na testování výkonnosti aplikačního stroje jako celku, ne tedy jen na jeho databázovou část. Přesto mohou být dobrými nástroji pro testování vyladění a rychlosti konkrétně nasazené databáze, a proto si v tomto přehledu zaslouží alespoň zmínku.

3.5.4 Database benchmark

Dalším open source benchmarkem, který si bere některé postupy z metodiky organizace TCP je benchmark pojmenovaný jednoduše **Database benchmark**. Nástroj je vhodný zejména pro testování NoSQL databází, protože testuje pouze rychlost zápisu a čtení. Technicky se jedná o aplikaci napsanou v jazyce Java, která provádí operace insert a select. V současnosti je ve verzi 2.0 a je ke stažení na stránkách database-benchmark.com, kde je k němu k dispozici i jednoduchý návod. Jedná se o zdrojové kódy v jazyce C# a také už zkompilevanou aplikaci.

Samotná aplikace je snadno použitelná (stačí jen na správná místa doplnit databázového uživatele a heslo), má grafické uživatelské prostředí a výstupem jsou přehledné tabulky a grafy.

3.5.5 Pole position

Pole position je další z databázových benchmarků, vytvořených vývojáři pro vývojáře (jeho autory jsou Andrew Zhang, Carl Rosenberger, Christian Ernst, Fabio Roger, Joerg Plewe, Patrick Roemer, Pieter van Zyl, Richard Liang a Travis Reeder).

Aplikace je dostupná ve formě zdrojových kódů v jazyce Java s předpřipravenými soubory projektu pro editor Eclipse

Samotná aplikace potom běží v příkazové řádce a výstupem je PDF s přehlednými grafy obsahujícími i vysvětlivky, co je jednotlivými testy měřeno.

Pole position nabízí tuto sadu testů:

- Complex - write, read, query, update and delete complex object graph
- Nestedlists - write and read nested lists
- Inheritancehierarchy - writes, reads, queries and deletes objects with a 5 level inheritance structure
- Flatobject - write, query, update and delete indexed flat objects

4 Praktická část

4.1 Dostupnost instalačních balíků

Pro linuxové systémy je MySQL k dispozici v jejich repositářích. V některých distribucích je ovšem kvůli transparentnosti vývoje nahrazována forkem MariaDB.

Pro Windows a Mac OS systémy se na stránkách obou produktů dají stáhnout instalátory s přehledným GUI. Oracle /vlastník MySQL/ v tomto případě ovšem klade určité překážky – vyžaduje registraci uživatele spolu s ověřením jeho e-mailu a při stažení balíku předkládá dotazník, na jaký typ provozu bude stažená databáze používána.

Komunita PostgreSQL tyto překážky neklade a instalátory se dají stahovat bez omezení.

4.2 Podporované operační systémy

MySQL je možné provozovat na všech hlavních operačních systémech (40). Přímou vyjmenovává: Oracle Linux, Oracle Solaris, Red Hat, SuSE, Debian, Microsoft Windows a Microsoft Windows Server, IBM AIX, Apple Mac OS, FreeBSD, Fedora, OpenSuSE, CentOS a HP-UX.

PostgreSQL je možné také provozovat na všech hlavních operačních systémech (31) – přímo vyjmenovává Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) a Windows.

U obou databázových strojů konkrétní podporované verze operačních systémů závisí na konkrétní verzi příslušné databáze, právě kvůli podpoře starších verzí OS jsou podporovány bezpečnostními a výkonnostními updaty i některé starší verze databází (viz Obrázek 3, Obrázek 7).

4.3 Instalace

Na linuxových strojích je instalace poměrně jednoduchou záležitostí vzhledem k linuxovým balíčkovacím repositářům. Pro **MySQL** je tak dostačující použití následujícího scriptu:

```
sudo apt-get install mysql-server mysql-client
```

a kvůli bezpečnosti je vhodné nastavit základnímu uživateli heslo:

```
sudo mysqladmin -u root -h localhost password 'mypassword'
```

Dále je třeba nainstalovat potřebné obslužné knihovny pro programovací jazyk dle výběru příslušného programátora (zde PHP):

```
sudo apt-get install php5-mysql
```

To je pro instalaci vše, dál je vhodné nainstalovat některý obslužný program pro administraci jednotlivých databází (jsou popsány v kapitole 3.3.6), ale lze pracovat i bez něj. A samozřejmě databázi nakonfigurovat dle požadavků konkrétního použití.

PostgreSQL se instaluje podobně jednoduše:

```
sudo apt-get install postgresql postgresql-contrib
```

```
sudo passwd postgres
```

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

A opět může následovat konfigurace a instalace obslužných programů (jsou popsány v kapitole 3.4.6).

Pro Windows jsou k dispozici přehledné instalátory s GUI, stáhnutelné na oficiálních stránkách obou databází, které uživatele provedou instalací krok za krokem.

Mac OS X Server obsahuje databázi PostgreSQL spolu s jejími obslužnými nástroji pro příkazovou řádku už v základní instalaci. Dále je k dispozici opět GUI instalátor PostgreSQL a balíku nástrojů pgAdmin a StackBuilder, které je vhodné použít zejména při potřebě upgrade databáze na novější verzi. MySQL je v základní instalaci tohoto serveru také a opět je k dispozici balíček s GUI pro instalaci novějších verzí.

Ve všech těchto systémech je výchozím portem, na kterém databáze naslouchají 3306 pro MySQL a 5432 pro PostgreSQL. Při instalaci je třeba dát pozor, jestli tyto porty nejsou použity některým jiným programem.

Ve věci snadnosti instalace jsou tedy databázové stroje MySQL a PostgreSQL srovnatelné.

4.4 Databázové optimalizace a jejich měření

Oba zkoumané databázové stroje nabízí nástroje k měření vlastního výkonu. Jejich použití je vhodné zejména tam, kde se ladí výkon databáze na konkrétním hardware, dá se jimi například rychle změřit, jestli měla změna některého z nastavení příznivý dopad na výkon.

MySQL nabízí k dodatečnému stažení (tj. není součástí běžného instalačního balíku) nástroj **DBT2 Benchmark Tool**. Tento nástroj umožňuje provádět sadu testů, které jsou inspirovány testy definovanými TPC (popisuje je Kapitola 3.5.1).

Je k dispozici ve formě sady .sh scriptů a k jejich spuštění na operačním systému Windows je tedy nutný některý z nástrojů, který zpřístupňuje linuxové funkce – například Cygwin (41).

U **MySQL** lze ladit výkonnost pro konkrétní stroj úpravami konfiguračního souboru *my.cfg*, který se většinou nachází v instalačním adresáři databáze, lze mu ale během instalace určit jiné umístění. Toto umístění se pak dá přímo za běhu databáze zjistit příkazem

```
$ mysqld --help -verbose
```

Který ukáže umístění, ve kterých MySQL hledá daný konfigurační soubor. Na linuxových strojích se například hledá standardně postupně ve složkách:

```
/etc/my.cnf  
/etc/mysql/my.cnf  
$MYSQL_HOME/my.cnf  
[datadir]/my.cnf  
~/my.cnf
```

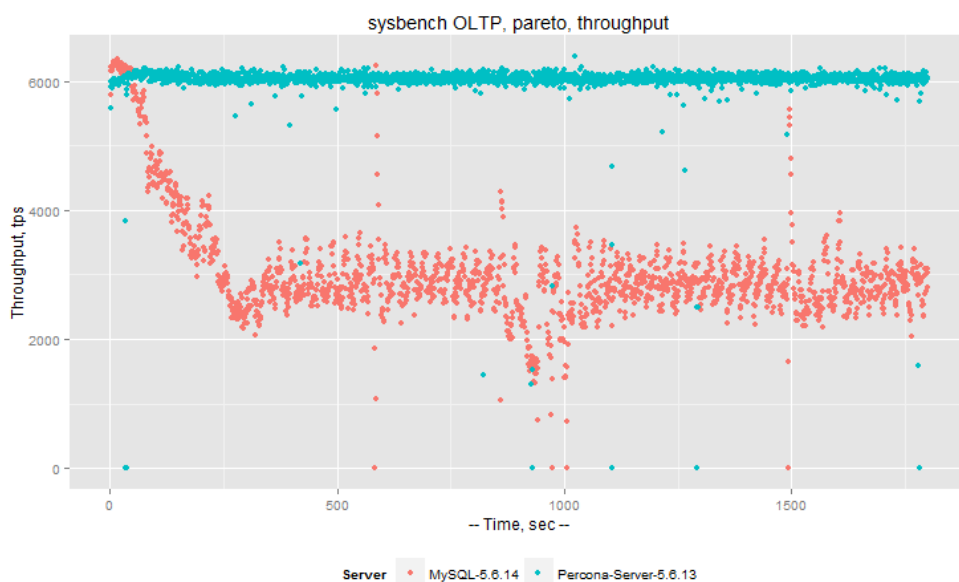
S tím, že každá z nich může obsahovat soubor *my.cfg* a později definované hodnoty přebíjejí ty dřívější. Lze tak mít různé konfigurace per databáze, a dokonce i per uživatel.

Na strojích s windows je potom možné tento soubor najít ve složkách

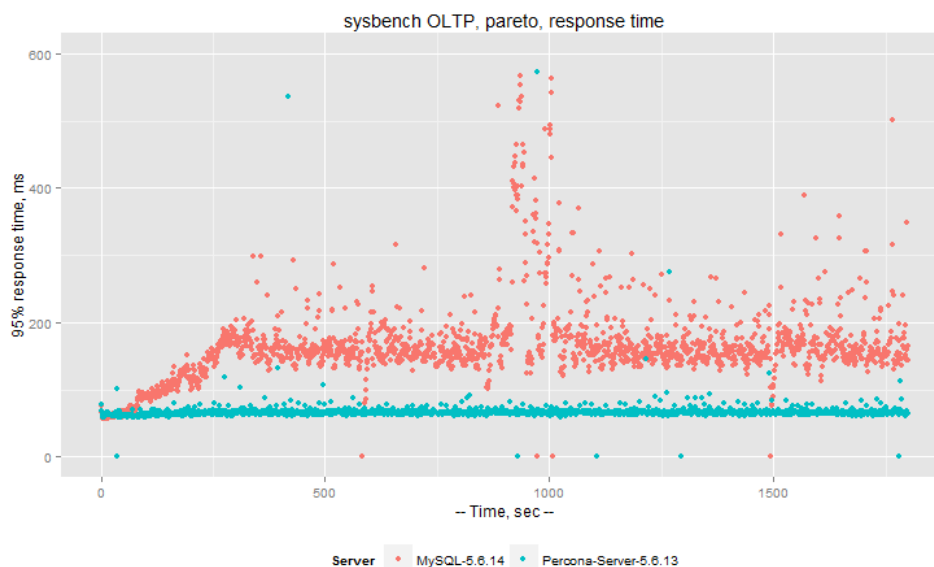
```
c:\my.cnf nebo c:\Windows\my.ini.
```

Nejdůležitější parametry pro ladění výkonu **MySQL** databáze jsou *key_buffer_size* a *table_cache*, měly by být nastaveny tak, aby bylo dost paměti pro všechna databázová spojení. Jejich maximální počet definuje parametr *max_connections*. V souboru *my.cfg* lze dále měnit i nastavení samotných úložných enginů. Za zmínku stojí zejména parametry *innodb_buffer_pool_size* (pro InnoDB) a *key_buffer* (pro MyISAM) (42).

Výkon **MySQL** je také možné zvýšit nasazením některého z forků této databáze, které se na výkon zaměřují. Například **Percona server** uvádí výkonnost v benchmarcích vyšší a dobu odezvy nižší, než standardní instalace (23):



Obrázek 13 Percona server test výkonu vůči MySQL (zdroj (23))



Obrázek 14 Percona server - test doby odezvy vůči MySQL (zdroj (23))

PostgreSQL nabízí již v rámci běžné instalace nástroj **pgbench**, který se také inspirovuje testy od TPC, a je přímo založený na starším standardu TPC-B. Jeho spuštění žádné dodatečné nástroje nevyžaduje, a proto je možné zde ukázat příklad jeho volání.

Po vstoupení do složky bin adresáře instalace PostgreSQL lze pgbench spustit prostým zadáním pgbench do příkazové řádky. To se ale snaží připojit jménem uživatele právě přihlášeného na počítači, proto je někdy třeba přidat parametr `-U` pro změnu uživatele například takto:

```
pgbench -U postgres
```

Výsledkem (po zadání příslušného hesla) je potom

```
starting vacuum...end.
transaction type: TPC-B (sort of)
scaling factor: 4
query mode: simple
number of clients: 1
number of threads: 1
number of transactions per client: 10
number of transactions actually processed: 10/10
```

```
tps = 14.776283 (including connections establishing)
```

```
tps = 18.192789 (excluding connections establishing)
```

Poslední dva řádky ukazují vlastní výkon databáze. Jednotky jsou stejné jako u TPC testů – počet transakcí za sekundu. Předposlední řádek počítá sekundy včetně času na zahájení transakce a proto je jeho hodnota menší, než posledního řádku, který tento čas nezohledňuje. Pro škálování tohoto testu je možné použít parametry

-*t* pro zvýšení počtu transakcí per klient,

-*c* pro určení počtu paralelně běžících databázových klientů a

-*s* pro zvednutí „scale faktoru“ tedy právě škálování počtu provedených transakcí.

Vyškálování je tedy možné provést například takto:

```
pgbench -U postgres -c 10 -t 1000
```

Výsledek pak vypadá například takto:

```
starting vacuum...end.
```

```
transaction type: TPC-B (sort of)
```

```
scaling factor: 4
```

```
query mode: simple
```

```
number of clients: 10
```

```
number of threads: 1
```

```
number of transactions per client: 1000
```

```
number of transactions actually processed: 10000/10000
```

```
tps = 737.257426 (including connections establishing)
```

```
tps = 759.030022 (excluding connections establishing)
```

Ladění výkonu **PostgreSQL** se provádí úpravami souboru *postgresql.conf*, který se obvykle nachází v data adresáři příslušné instalace databáze. Nejdůležitějšími konfiguračními direktivami jsou *shared_buffers*, která určují velikost cache pro databázová data (Na 32 bitových strojích je maximální efektivní nastavení kolem 2,5 GB) a *effective_cache_size*, která stanovuje, kolik diskového prostoru je možné využít pro

cachování běhu PostgreSQL. Maximální množství připojených klientů se nastaví pomocí parametru *max_connections*.

4.5 Srovnání výkonnosti

Pro srovnání výkonnosti obou databázových strojů je potřeba je oba nainstalovat do stejného počítače a provést srovnatelné testy pro obě databáze. Pro tento účel byl použit standardní domácí stroj Lenovo IdeaCentre H520 s 64 bitovým operačním systémem Windows 8.1.

| | |
|-----------------------|--|
| Operační paměť | 4 GB |
| Procesor | Dvoujádrový Intel Pentium CPU G2030 @ 3.00GHz |
| Pevný Disk | 500 GB, Souborový systém NTFS |

Tabulka 7 Parametry testovacího stroje

Pro dále uvedené testy byly databáze použity v jejich výchozích nastaveních.

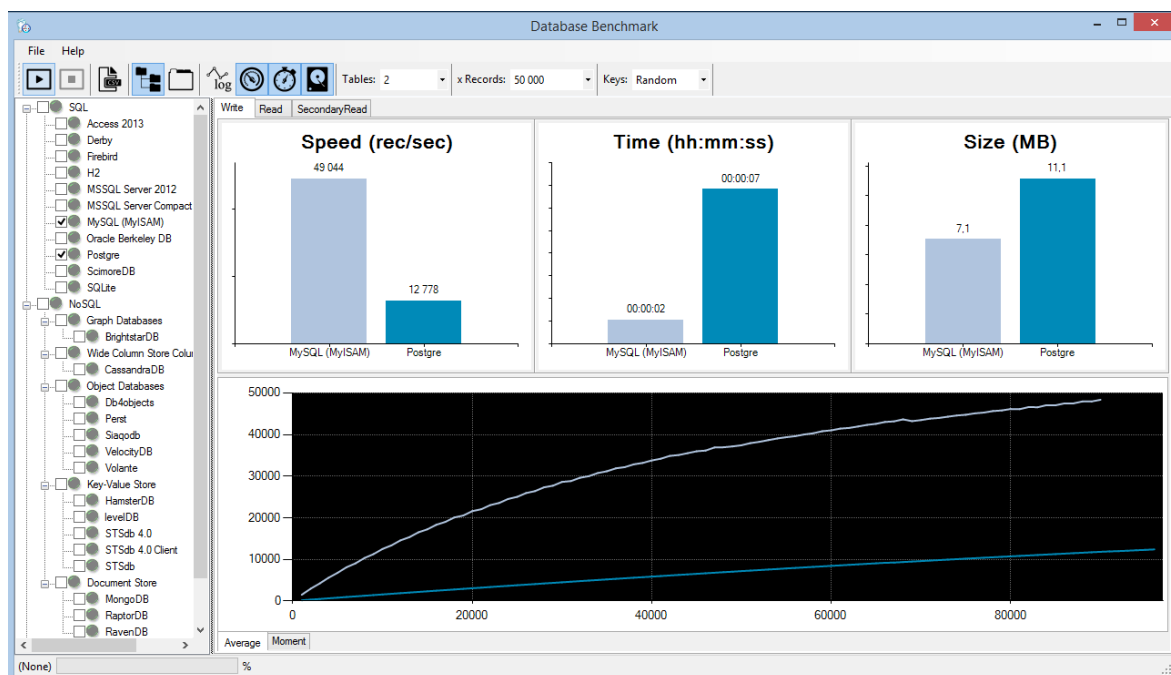
K jednotlivým testům je na tomto místě ještě důležité uvést, že v nich nejsou z hlediska důležitosti srovnávána absolutní čísla, která se mohou lišit na základě mnoha vnějších okolností – jsou například ovlivněny už jen správou jednotlivých zdrojů, které přiděluje operační systém – ale porovnávají se výsledky vůči sobě zejména s ohledem na jejich řád. Tj. jestli některá databáze vykazuje naprosto jiný výsledek, než druhá (případně jiný engine).

4.5.1 Database benchmark

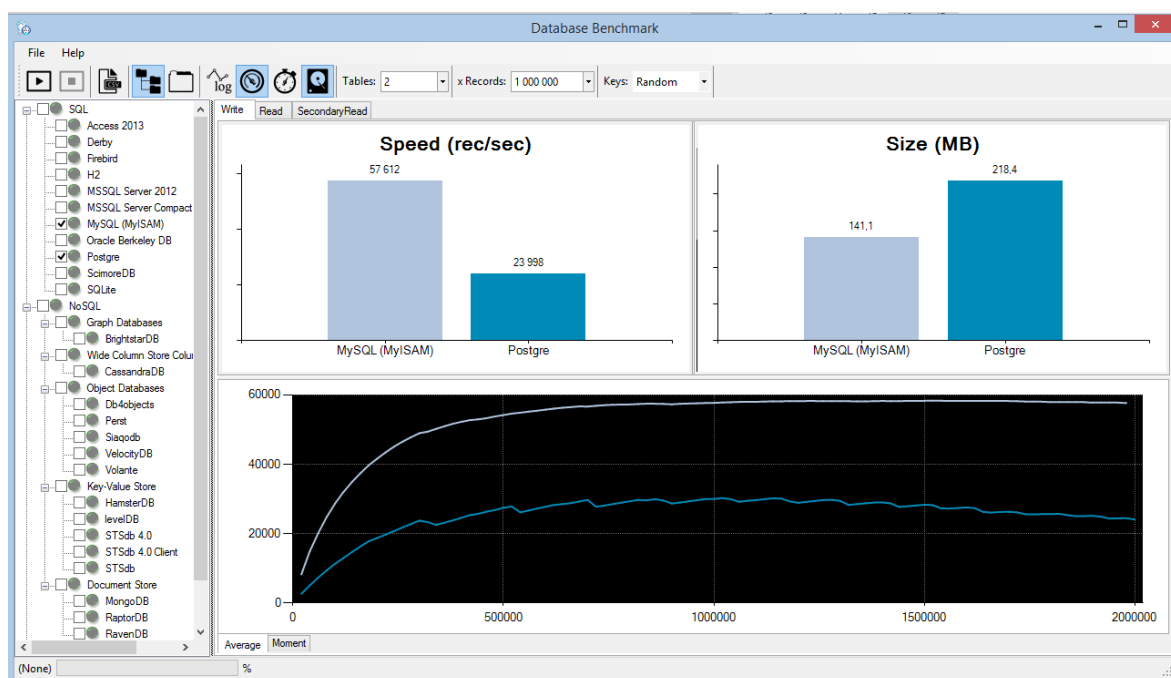
Jako první je proveden test pomocí nástroje **database benchmark**, jehož použití je z diskutovaných nástrojů nejjednodušší, protože jeho distribuční balíček už obsahuje zkompilevanou a spustitelnou verzi. Po navolení správného přihlašovacího jména a hesla (a kontrole, že obě databáze běží) je možné vybrat příslušnou škálu testu.

Zvoleným testovacím scénářem byly zápisy do 2 tabulek. Nejprve s 50 000 záznamy per tabulka, a poté 1 000 000 záznamů. V největším grafu je na svislé ose

vynesena rychlost zápisu v záznamech za sekundu, na vodorovné je poté počet již zapsaných řádků.



Obrázek 15 Database benchmark, testovací scénář 1.1 (zdroj: vlastní tvorba)



Obrázek 16 Database benchmark, testovací scénář 2.1 (zdroj: vlastní tvorba)

Výsledky jsou vzhledem k použitému výchozímu MySQL engine MyISAM nepřekvapující. Tento engine je značně jednoduchý a narušil od engine PostgreSQL například neumí hlídat datovou integritu a má další omezení. Pro lepší srovnání bylo třeba přidat podporu pro engine InnoDB. Po konzultaci s dokumentací na stránkách projektu (43) byla vytvořena nová třída `MySQLBenchmarkInnodb` jako potomek `MySQLBenchmark` takto:

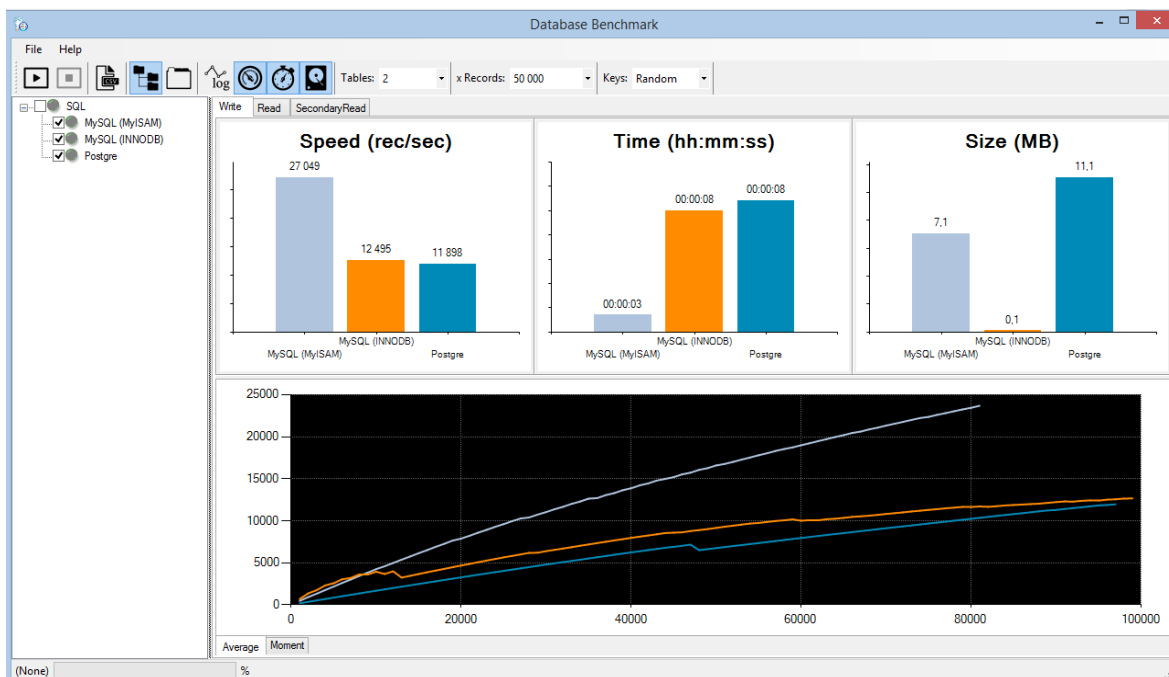
```
namespace DatabaseBenchmark.Databases
{
    public class MySQLBenchmarkInnodb : MySQLBenchmark
    {
        public MySQLBenchmarkInnodb()
        {
            Category = "SQL";

            Description = "MySQL InnoDB + .NET Connector";
            Website = "http://www.mysql.com/";
            Color = System.Drawing.Color.DarkOrange;
            Requirements = new string[] { "MySQL.Data.dll" };

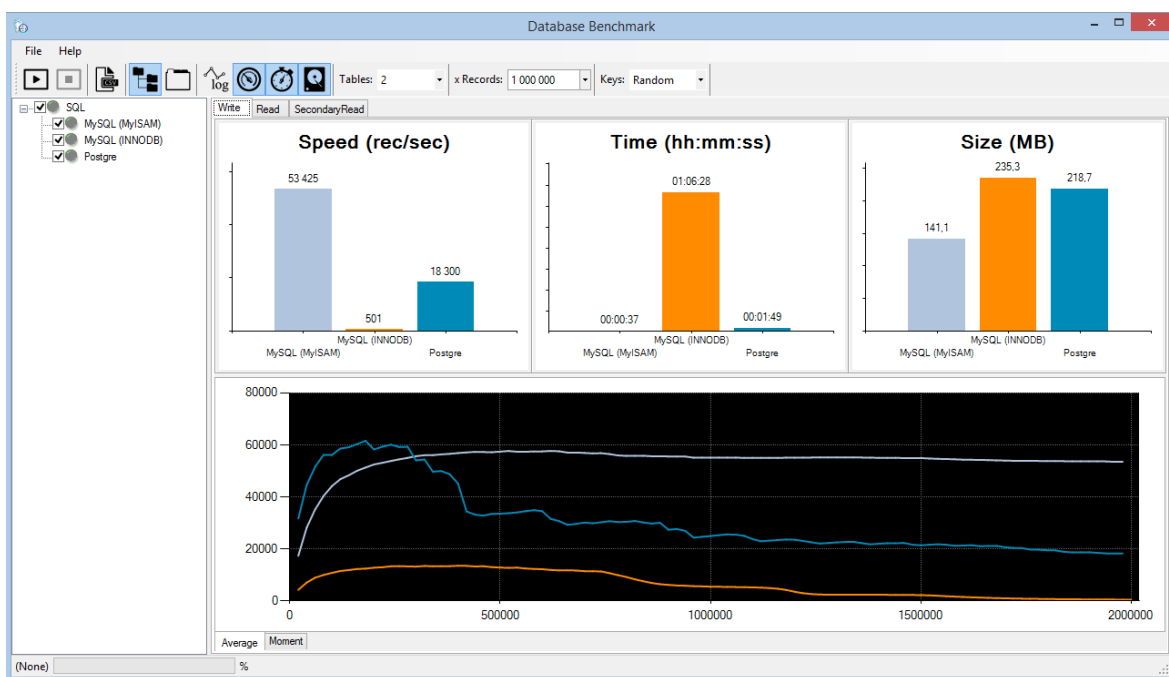
            StorageEngine = MySQLStorageEngine.INNODB;
            DatabaseName = String.Format("MySQL ({0})",
StorageEngine);

            MySqlConnectionStringBuilder cb = new
MySQLConnectionStringBuilder();
            cb.ConnectionLifeTime = 0;
            cb.DefaultCommandTimeout = 0;
            cb.Server = "localhost";
            cb.UserID = "root";
            cb.Password = "adminpass";
            cb.Database = "test2";
            ConnectionString = cb.ConnectionString;
        }
    }
}
```

Výsledky stejného testovacího scénáře poté vypadají takto (v menu nalevo si lze všimnout, že nástroj byl zkompilován již bez podpory pro další databáze):



Obrázek 17 Database benchmark, testovací scénář 1.2 (zdroj: vlastní tvorba)



Obrázek 18 Database benchmark, testovací scénář 2.2 (zdroj: vlastní tvorba)

Jak již bylo řečeno, Database benchmark testuje hlavně rychlost zápisu, rychlost čtení a poté velikost výsledné databáze. Činí tak vytvořením tabulek o osmi sloupcích různých datových typů:

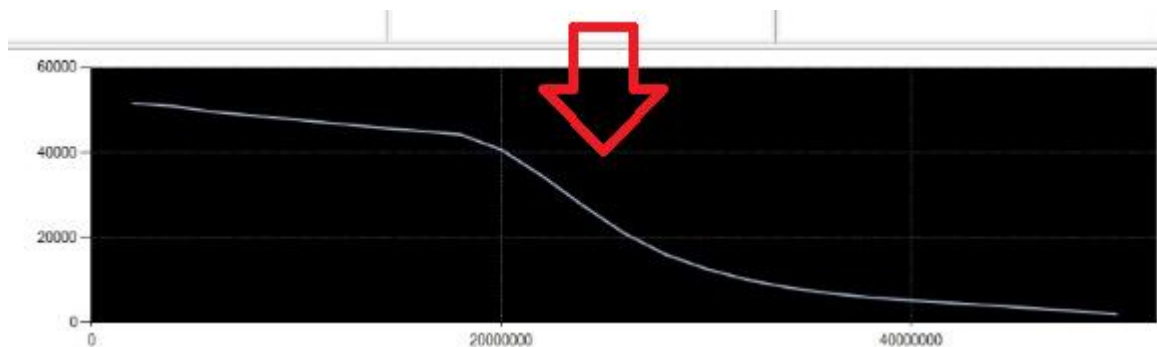
```

helper.AddField("ID", DbType.Int64, 0);
helper.AddField("Symbol", DbType.String, 255);
helper.AddField("Time", DbType.DateTime, 0);
helper.AddField("Bid", DbType.Double, 0);
helper.AddField("Ask", DbType.Double, 0);
helper.AddField("BidSize", DbType.Int32, 0);
helper.AddField("AskSize", DbType.Int32, 0);
helper.AddField("Provider", DbType.String, 255);

```

Tyto tabulky jsou poté plněny pseudonáhodnými daty z předpřipravených seznamů dat, aby bylo možné provést měření rychlosti příkazu SELECT.

Tabulka 8. Řádek maximální rozumný insert udává počet záznamů, po jejichž vložení došlo k prudkému poklesu výkonnosti vkládání dat o několik řádů. Jak toto zpomalení může vypadat demonstruje Obrázek 19, kde je provedený test pro databázi MySQL s enginem MyISAM. Na svislé ose je rychlost ukládání záznamů v záznamech za sekundu, na vodorovné potom počet již založených řádků.



Obrázek 19 Database benchmark, pokles rychlosti insertu (zdroj: vlastní tvorba)

| Operace | MySQL-MyISAM | MySQL-InnoDB | PostgreSQL |
|---------------------------------|--------------|---------------|-------------|
| Insert 100 K záznamů | 3 s | 8 s | 8 s |
| Insert 2 M záznamů | 37 s | 3988 s | 109 s |
| Maximální rozumný insert | Cca 20 M záz | Cca 800 K záz | Cca 4 M záz |
| Select 2 milionu záznamů | 106 K záz/s | 14 K záz/s | 73 K záz/s |

Tabulka 8 Výsledky testů database benchmark

4.5.2 Pole position

Jak již bylo řečeno v kapitole 3.5.5, měřící nástroj Pole position je poskytován ve formě zdrojových kódů v jazyce Java s připravenými soubory projektu pro editor Eclipse. Pro provedení pokusu byl stažen tento editor ve verzi *Eclipse Standard 4.3.2* a po neúspěšné snaze o build aplikace také na radu profesionálního Java vývojáře ve verzi *Eclipse IDE for Java EE Developers*. Ani v jednom z těchto IDE (*Integrated development environment*) se bohužel nepodařilo aplikaci sestavit. Následovala druhá konzultace s již zmíněným vývojářem a na jeho radu bylo vyzkoušeno i běhové prostředí pro Java aplikace (JVM – java virtual machine) ve verzi *Java Platform (JDK) 7u5*. Ani s tímto nastavením sestavení aplikace na stroji s operačním systémem Windows nebylo úspěšné.

Následovalo tedy zapůjčení stroje s nainstalovaným operačním systémem Ubuntu ve verzi 13.10 a po zkompileování, které proběhlo již bez problémů, bylo možné spustit test. Použité verze databází jsou stejné jako u předchozího testu (PostgreSQL 9.3 a MySQL 5.6), Java driver použitý v tomto nástroji ovšem tuto PostgreSQL verzi ještě nezná a proto ji ve výsledcích testu identifikuje jako 9.1.12. V testech MySQL je použit úložný engine InnoDB, který je od MySQL verze 5.5 výchozí pro každou tabulku, která nemá při vytváření příkazem CREATE TABLE engine specifikovaný. Podpora pro MyISAM vzhledem k nutnosti zápůjčky cizího stroje nebyla dopsána – stroj byl použit na spuštění databázového testu tak, jak jej autoři dali k dispozici. Parametry druhého testovacího stroje ukazuje Tabulka 9.

| | |
|-----------------------|---|
| Model | Lenovo ThinkPad T530 |
| CPU | Intel(R) Core(TM) i7-3520M CPU @ 2.90GHz |
| Operační paměť | 12 GB |
| Pevný Disk | TOSHIBA MK5061GS 500 GB, Souborový systém Ext4 |

Tabulka 9 Parametry druhého testovacího stroje

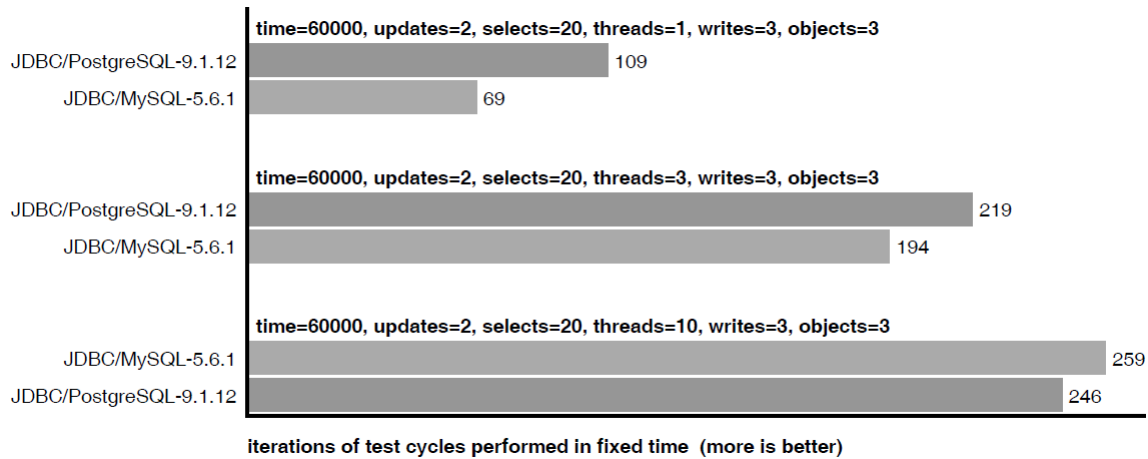
Nástroj **pole position** nabízí oproti předchozímu **database benchmark** i testování dotazů běžících ve více vláknech – toto je uvedeno jako parametr „threads“ v parametrech jednotlivých testů. Ostatní parametry udávají počet dotazů jednotlivého typu, který se v daném scénáři pro každé běžící vlákno provádí. Tyto parametry pro jednotlivé testy si může nastavit osoba, která testy provádí v souboru settings/Cirucits.properties. Jednotlivé parametry jsou dobře okomentované a díky tomuto nastavení se tak testy dají dobře škálovat. Příklad nastavení testu flatobject:

```
# flatobject
#
# [objects]: number of objects to store
# [selects]: number of queries to be run against all objects
# [updates]: number of updates and deletes to be run
# [commitinterval]: when to perform an intermediate commit during write
and delete
flatobject.objects=30000,100000,300000
flatobject.selects=3000,3000,3000
flatobject.updates=3000,3000,3000
flatobject.commitinterval=10000,10000,10000
```

Následující testování bylo s ohledem na obecnost porovnání provedeno s výchozím nastavením všech testů.

ComplexConcurrency test

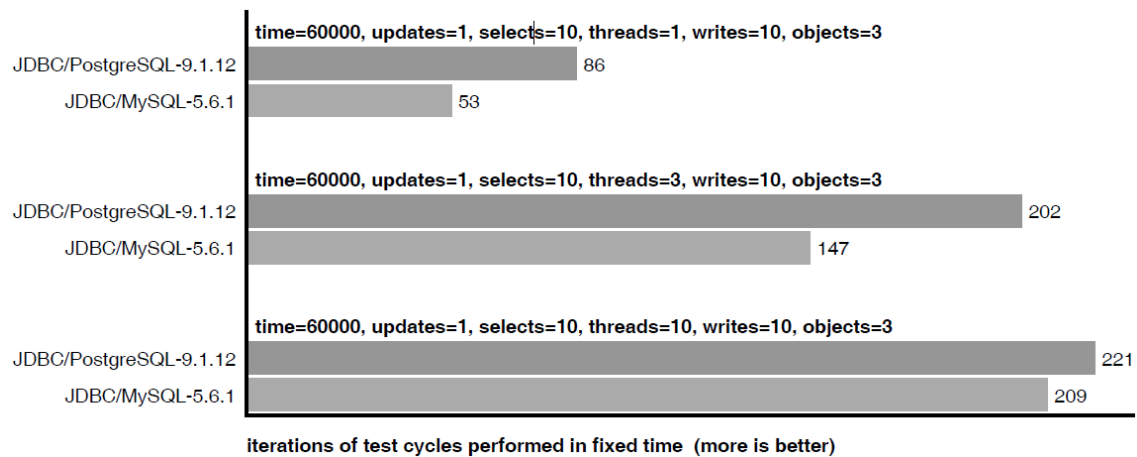
Prvním testem, který **pole position** provádí je test ComplexConcurrency. Tento test se sestává z dotazů typu insert, select, update a delete v autory nazvaném komplexním scénáři a počítá se, kolik iterací tohoto scénáře zvládne daný databázový stroj za předem fixně zvolený čas. Jak ukazuje Obrázek 20, výsledky testu jsou v tomto ohledu řádově srovnatelné pro obě databáze.



Obrázek 20 Pole position - ComplexConcurrency test (zdroj: vlastní tvorba)

InsertCentricConcurrency test

Dalším testem, který **pole position** provádí je InsertCentricConcurrency test. Tento test se zaměřuje na vložení (insert) většího množství objektů, opět za fixní množství času. Výsledky, jak ukazuje Obrázek 21, jsou opět řádově srovnatelné, ačkoliv MySQL je v tomto testu o něco horší než PostgreSQL.

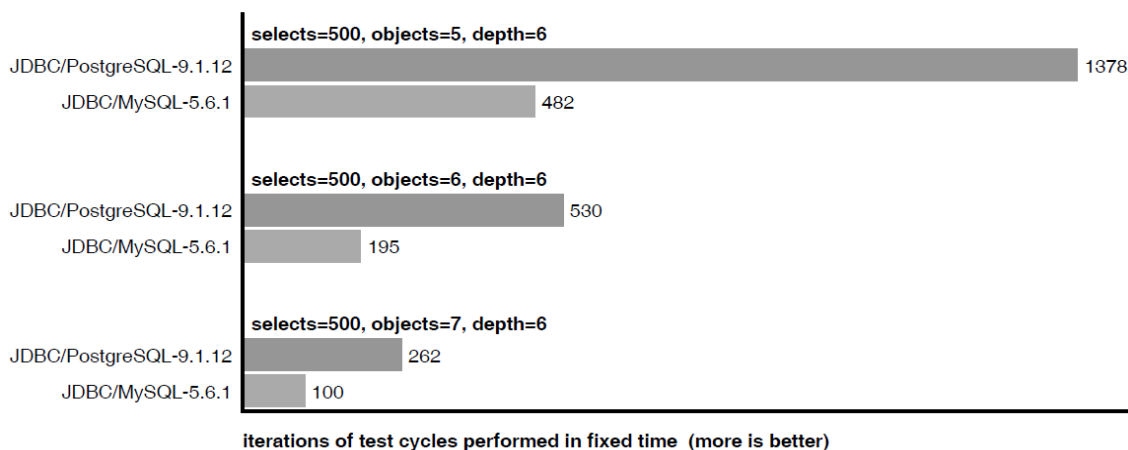


Obrázek 21 Pole position - InsertCentricConcurrency test (zdroj: vlastní tvorba)

Complex test

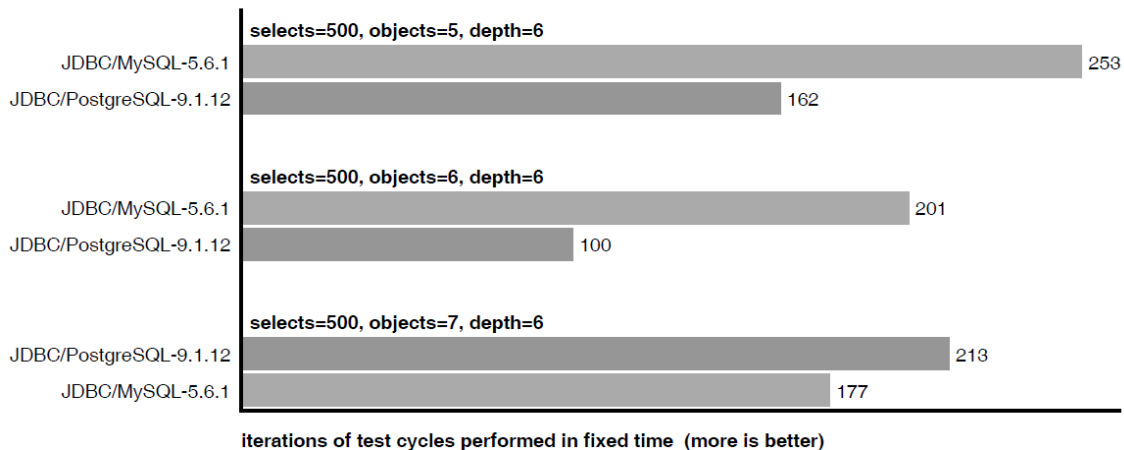
Dalším testem prováděným nástrojem **pole position** je test nazvaný Complex. Provádí zápis, čtení, úpravy a mazání na komplexnějších objektech grafu. Tento test je dále rozdělen do sekcí write, read, query, update a delete. Výsledky jednotlivých scénářů shrnuje Obrázek 22 až Obrázek 25. Jak lze dobře vidět, PostgreSQL v těchto testech řádově překonalo MySQL při testu typu write, update i delete. Pouze u testu typu query (provádí příkazy select) byla výkonnost u menšího počtu objektů vyšší než u PostgreSQL.

write



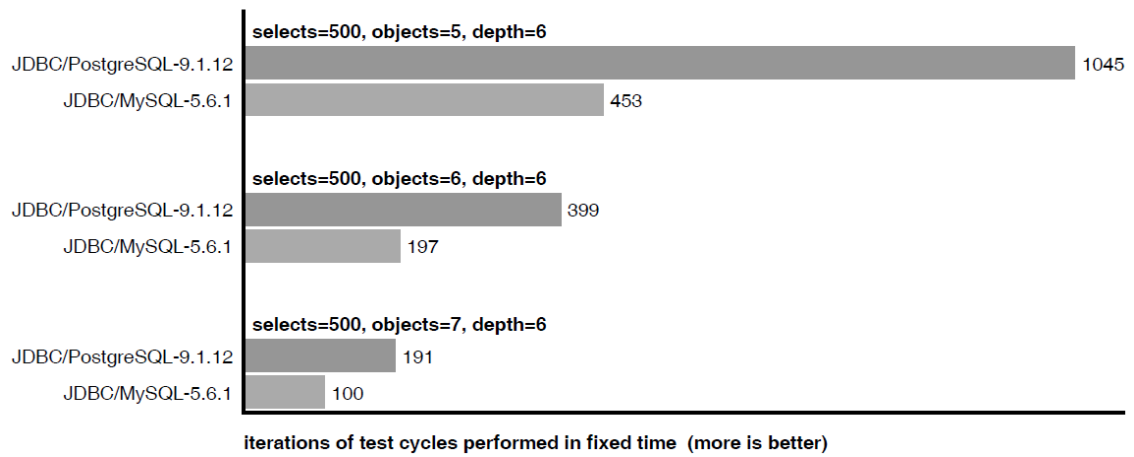
Obrázek 22 Pole position - Complex test – write (zdroj: vlastní tvorba)

query



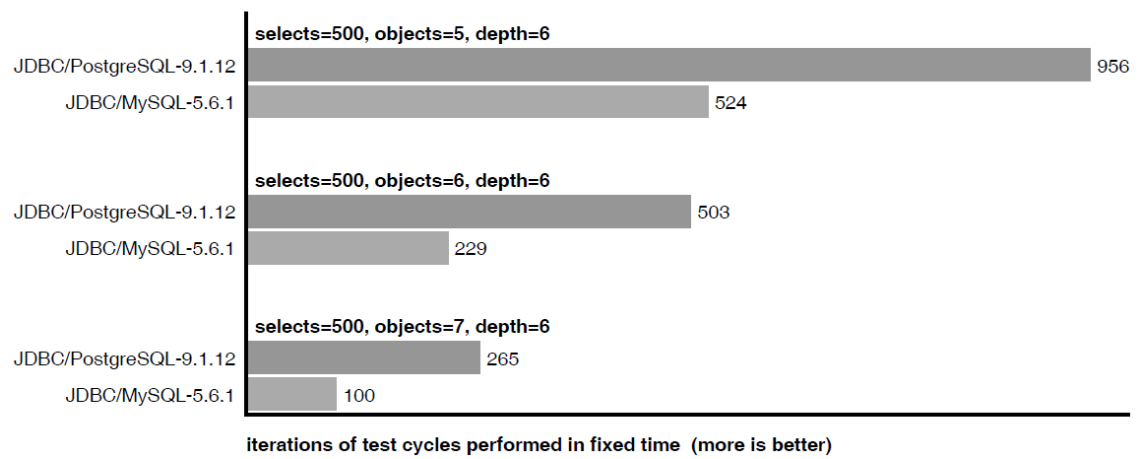
Obrázek 23 Pole position - Complex test – query (zdroj: vlastní tvorba)

update



Obrázek 24 Pole position - Complex test – update (zdroj: vlastní tvorba)

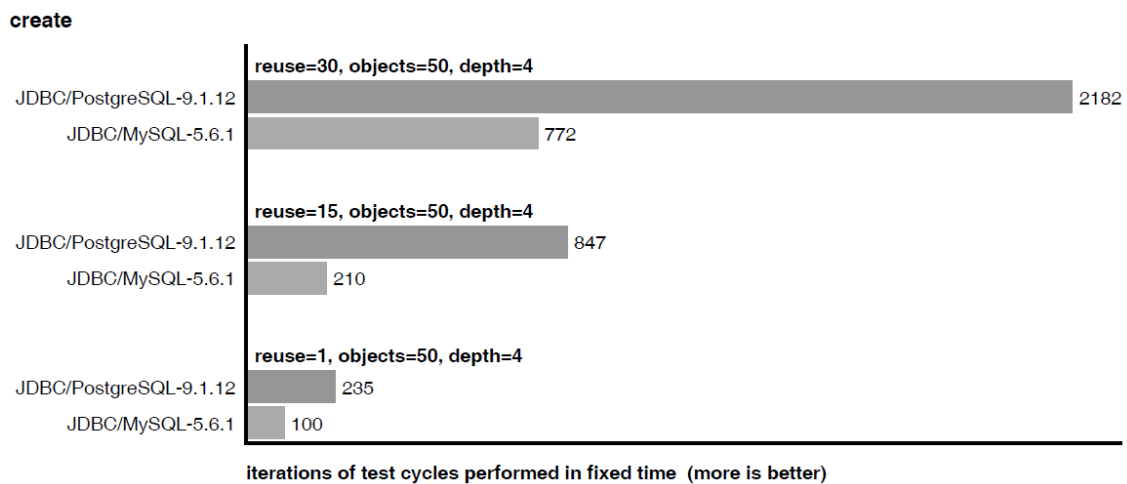
delete



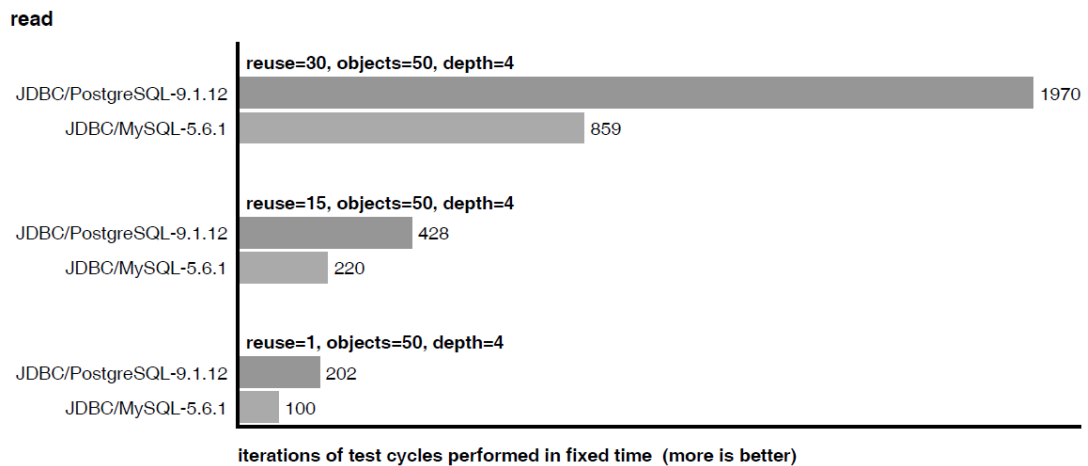
Obrázek 25 Pole position - Complex test – delete (zdroj: vlastní tvorba)

NestedLists test

Dalším typem testu, který provádí nástroj **pole position**, je test NestedLists. Ten se sestává z psaní a čtení vzájemně propojených seznamů – obsahují odkazy na objekty do stejné tabulky. Podobně jako předchozí test provádí testovací scénáře write, update, delete a query (tj. typu select). Výsledky měření ukazuje Obrázek 26 až Obrázek 29. Výsledky potvrzují měření předchozího testu, PostgreSQL je v těchto scénářích jasně dominující, v některých případech až několikanásobně.

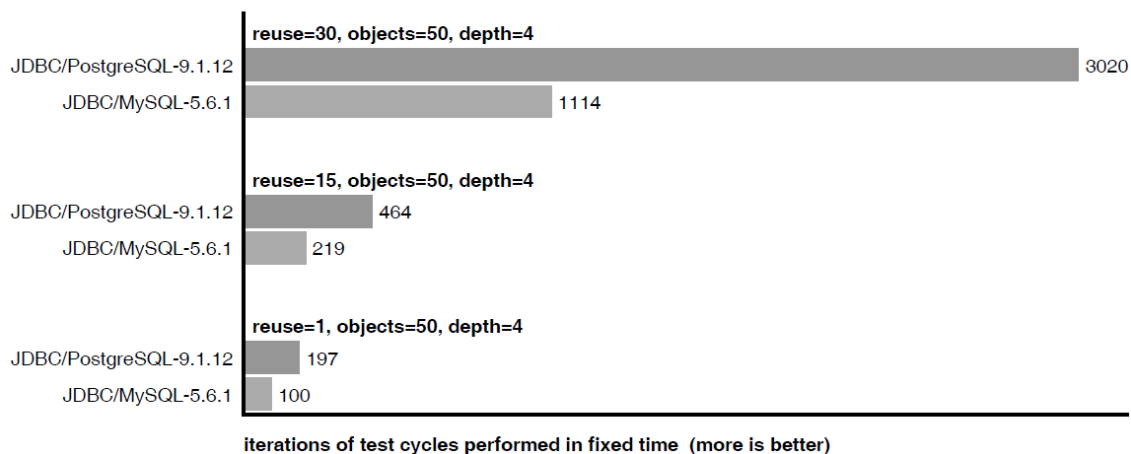


Obrázek 26 Pole position - NestedList test – create (zdroj: vlastní tvorba)



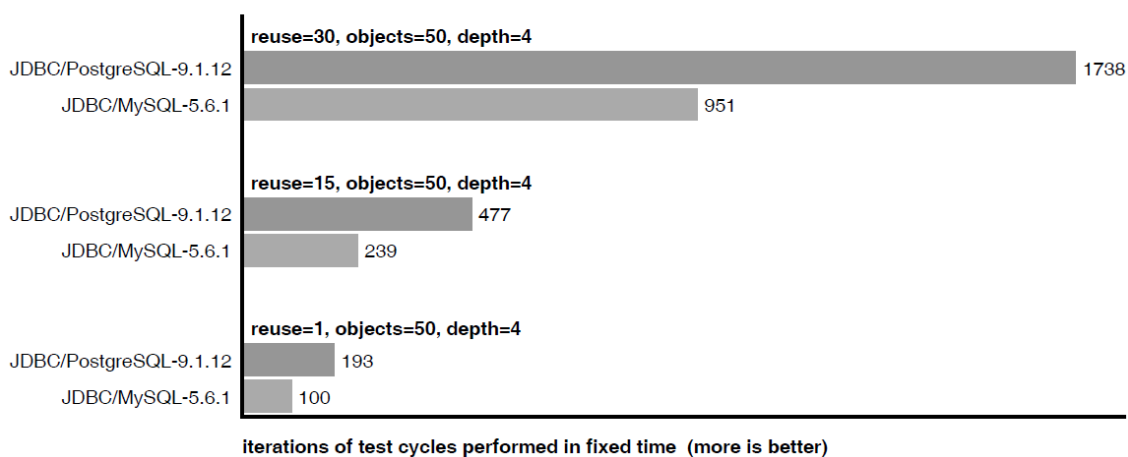
Obrázek 27 Pole position - NestedList test – read (zdroj: vlastní tvorba)

update



Obrázek 28 Pole position - NestedList test – update (zdroj: vlastní tvorba)

delete



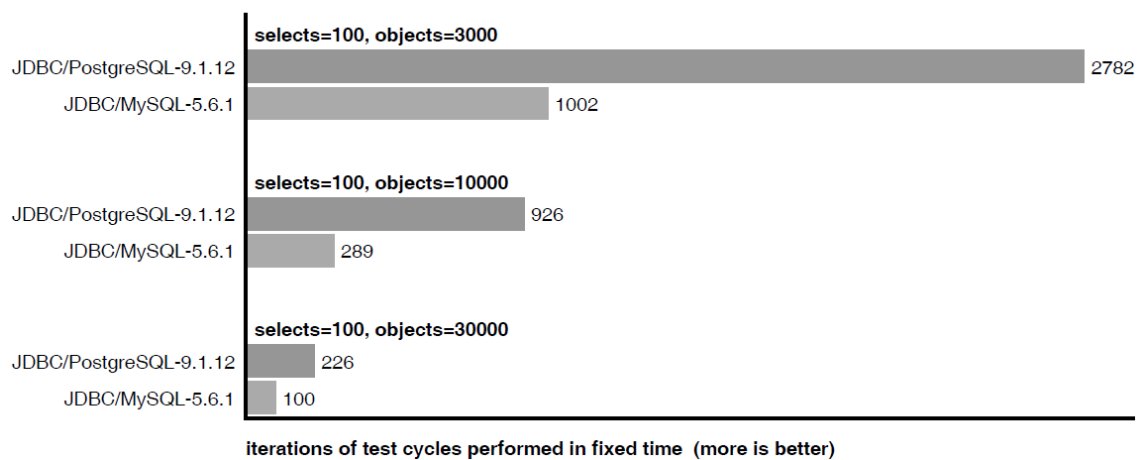
Obrázek 29 Pole position - NestedList test – delete (zdroj: vlastní tvorba)

InheritanceHierarchy test

Dalším testem, který je možné provést pomocí nástroje **pole position** je InheritanceHierarchy test. Opět se sestává z již více testovacích scénářů a to: write, read, query a delete a je prováděn pro velké množství objektů se strukturou s 5 úrovněmi hierarchie (odkazy na rodiče jednotlivých záznamů). Výsledky testu ukazuje Obrázek 30 až Obrázek 33.

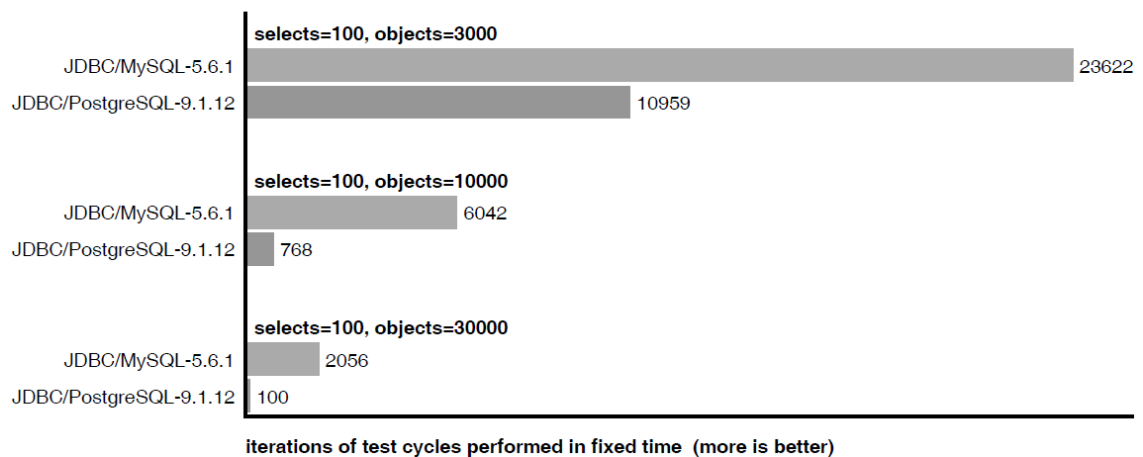
Ukazuje se, že v tomto typu testů PostgreSQL dominuje ve scénáři write a delete, zatímco MySQL je až několikanásobně výkonnější ve scénáři read a query.

write



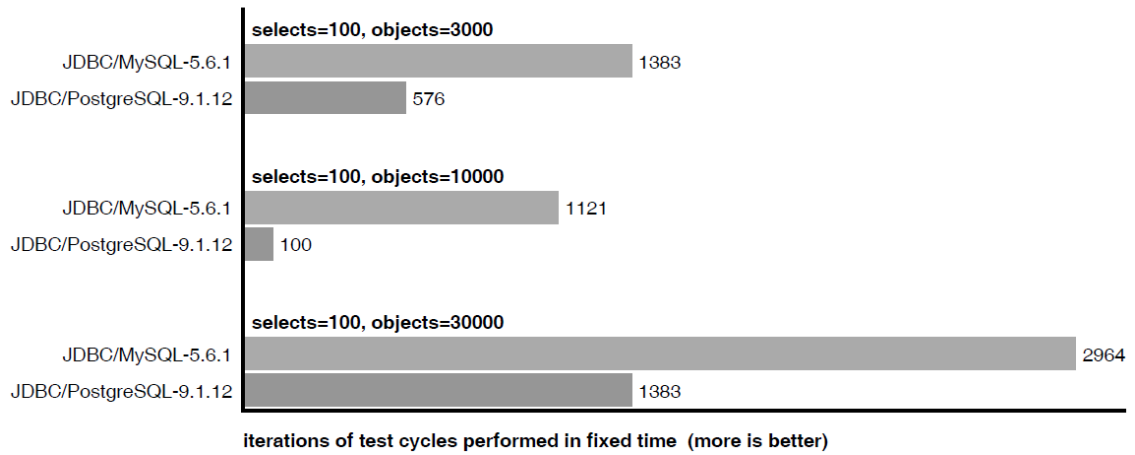
Obrázek 30 Pole position - InheritanceHierarchy test – write (zdroj: vlastní tvorba)

read



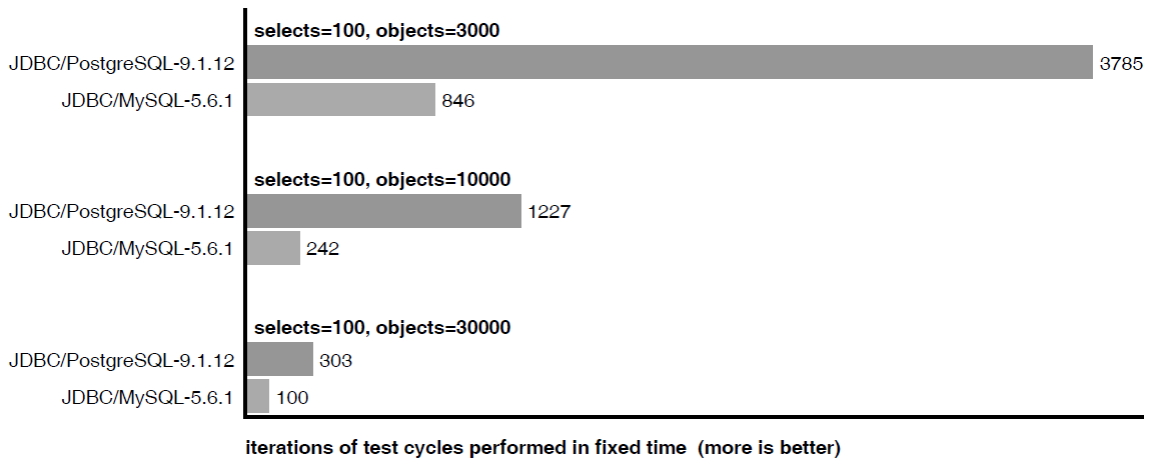
Obrázek 31 Pole position - InheritanceHierarchy test – read (zdroj: vlastní tvorba)

query



Obrázek 32 Pole position - InheritanceHierarchy test – query (zdroj: vlastní tvorba)

delete

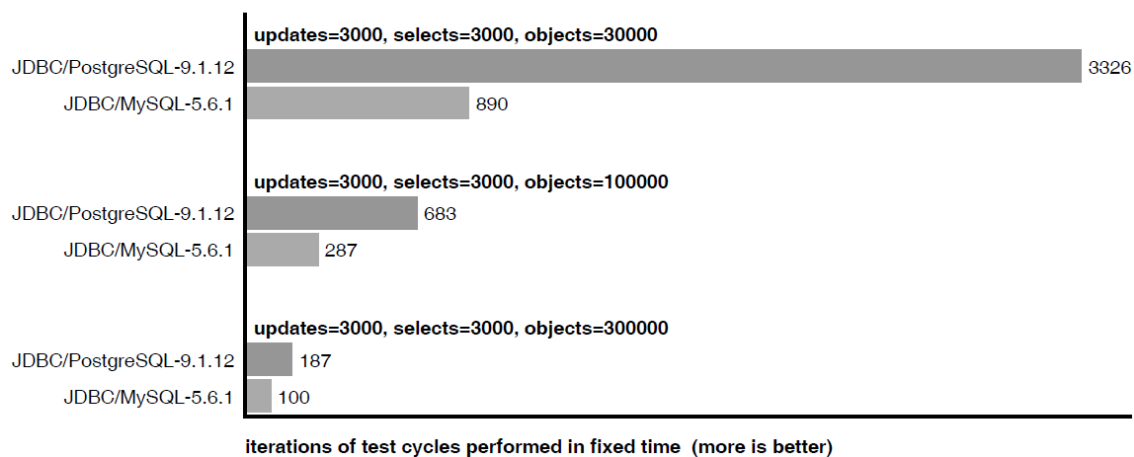


Obrázek 33 Pole position - InheritanceHierarchy test – delete (zdroj: vlastní tvorba)

FlatObject test

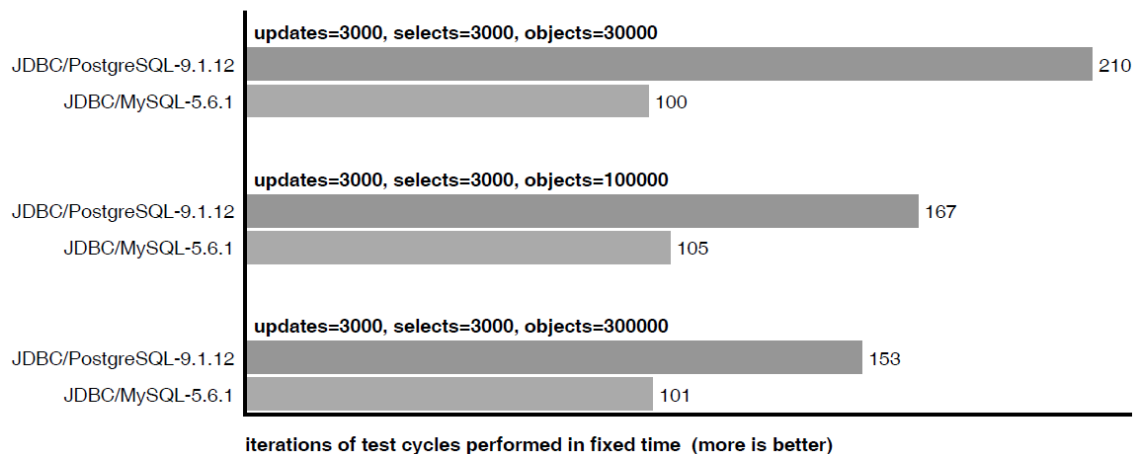
Posledním testem prováděným pomocí nástroje **pole position** je test typu FlatObject. Tento test vytváří velké množství indexovaných dat jednoduché struktury. Opět obsahuje pět scénářů – write, queryIndexedString, queryIndexedInt, update a delete. Výsledky ukazuje Obrázek 34 až Obrázek 38. PostgreSQL v tomto typu testu opět dominuje MySQL ve všech typech scénářů.

write



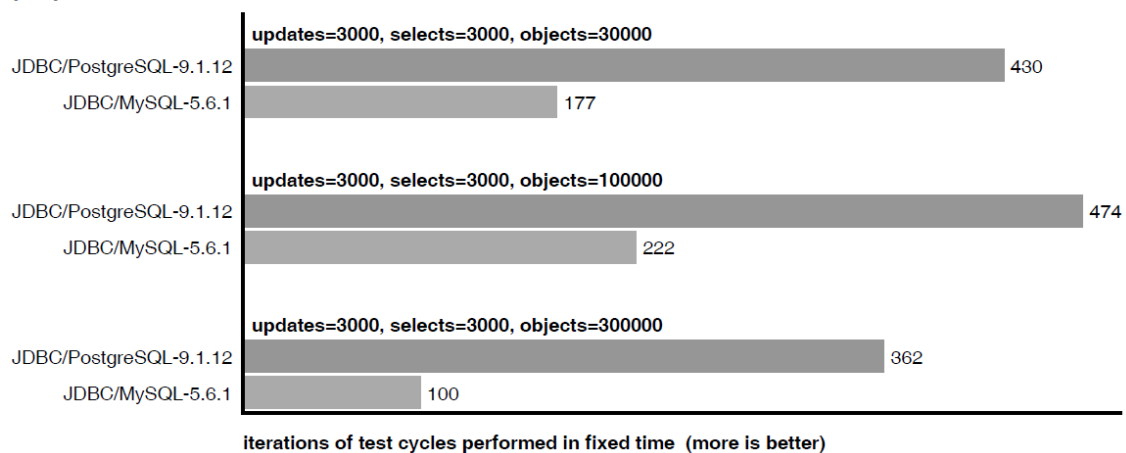
Obrázek 34 Pole position - FlatObject test – write (zdroj: vlastní tvorba)

queryIndexedString



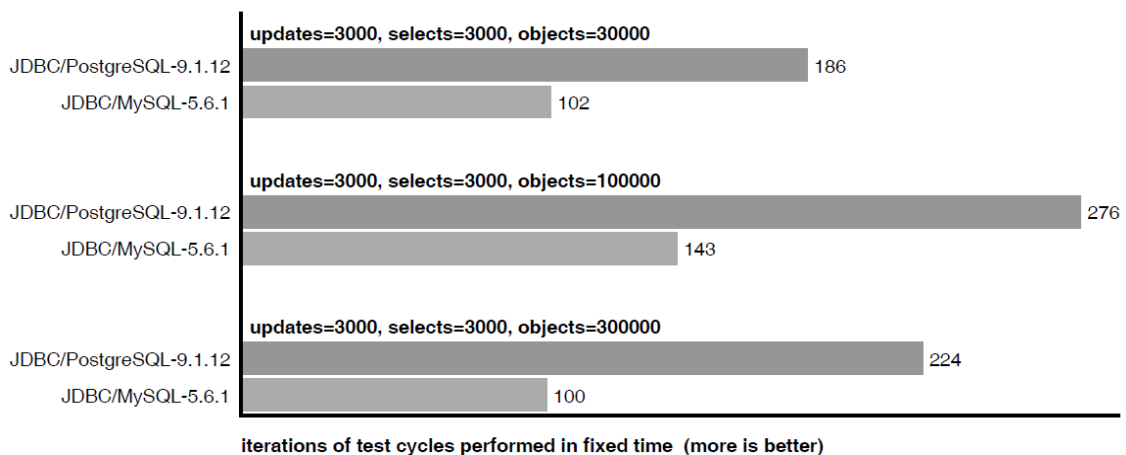
Obrázek 35 Pole position - FlatObject test – queryIndexedString (zdroj: vlastní tvorba)

queryIndexedInt



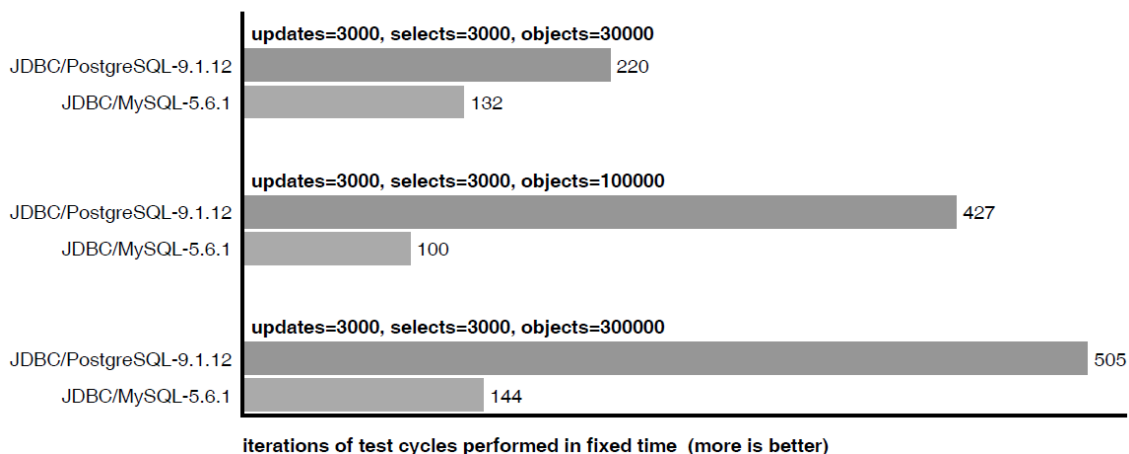
Obrázek 36 Pole position - FlatObject test – queryIndexedInt (zdroj: vlastní tvorba)

update



Obrázek 37 Pole position - FlatObject test – update (zdroj: vlastní tvorba)

delete



Obrázek 38 Pole position - FlatObject test – delete (zdroj: vlastní tvorba)

Shrnutí

Výsledky testu shrnuje Tabulka 10. PostgreSQL se v tomto typu testů ukazuje jako výkonnější, než MySQL využívající úložný engine InnoDB.

| Test | Dominující varianta |
|--------------------------------------|---------------------|
| ComplexConcurrency test | -- |
| InsertCentricConcurrency test | PostgreSQL |
| Complex test - write | PostgreSQL |
| Complex test - query | -- |
| Complex test - update | PostgreSQL |
| Complex test - delete | PostgreSQL |
| NestedLists test - create | PostgreSQL |
| NestedLists test - read | PostgreSQL |
| NestedLists test - update | PostgreSQL |
| NestedLists test - delete | PostgreSQL |
| InheritanceHierarchy test - write | PostgreSQL |
| InheritanceHierarchy test - read | MySQL |
| InheritanceHierarchy test - query | MySQL |
| InheritanceHierarchy test - delete | PostgreSQL |
| FlatObject test - write | PostgreSQL |
| FlatObject test – queryIndexedString | PostgreSQL |
| FlatObject test – queryIndexedInt | PostgreSQL |
| FlatObject test – update | PostgreSQL |
| FlatObject test – delete | PostgreSQL |

Tabulka 10 Shrnutí výsledků pole position

4.6 *Ekonomika provozu*

Ekonomiku provozu obou databází lze rozdělit na částku za infrastrukturu (hardware, software, služby) a náklady na odborníky, kteří se o danou infrastrukturu budou starat. Dle (44) tvoří největší část nákladů v IT sektoru právě mzdy a odměny odborníků.

Ceny za provoz hardware a software

Jak již bylo uvedeno v kapitolách 3.3.2 a 3.4.2, v základním režimu je licence na oba zkoumané databázové stroje zdarma a rozpočtovou položku tak netvoří. Pro „více komerční použití“ Oracle nabízí MySQL enterprise edition, která s cenami začíná zhruba na 40 000 Kč ročně.

Co se týče ceny za provoz hardware, buď se databáze provozují na strojích dedikovaných přímo jim, a tam se bude cena lišit dle jejich vybavení (zhruba od 20 000 do 200 000 Kč za stroj), které jsou pro použití na síti zpravidla ustájeny v některém datacentru formou takzvaného serverhousingu, kdy tento poskytovatel pronajímá klientům místo, konektivitu a elektřinu a klient si stroj spravuje sám. Tento serverhousing stojí zpravidla nízké desítky tisíc ročně (začíná kolem 1000 Kč za měsíc, liší se dle odběru elektřiny a dalších poskytovaných služeb) a opět vyjde stejně pro oba typy databáze.

Další možnou formou provozu databáze na síti je webhosting – tedy zpravidla pronajímané místo pro běh webových stránek. Zde MySQL převládá nad PostgreSQL jak rozšířením nabídky (webový portál hostingy.cz, který se věnuje této problematice, eviduje 107 hostingových nabídek pro MySQL a 39 pro PostgreSQL), tak cenou, kdy provoz MySQL nabízí některé hostingy i zdarma, pokud si u nich člověk objedná jiné služby. Webhostingové služby stojí zpravidla od několika set, do několika tisíc korun ročně, opět dle dalších připojených služeb. V současné době je také moderní přístup SaaS (*Software as a service*), kdy zákazník platí jen za zdroje, které skutečně spotřeboval. Ceny jsou v tomto případě například⁹ 2 Kč za započatý GB a den (společný diskový prostor pro data, databáze, emaily).

⁹ Zdroj: ceník společnosti IGNUM pro program Webcloud

V případě provozu na interní síti do nákladů samozřejmě spadají ještě náklady na zařízení této sítě a další, ale ty se již nebudou lišit v závislosti na použité databázi.

Náklady na IT odborníky

Jak již bylo řečeno, platové, mzdové a jiné náklady na IT odborníky jsou zpravidla největší výdajovou položkou na provoz databázové infrastruktury. Nejprve se poďme podívat na stranu poptávky.

Jedním z největších personálních portálů v ČR je jobs.cz, který umožňuje vyhledávat v nabídkách práce dle mnoha parametrů, včetně maximální nabízené mzdy.

Jobs.cz nabízel k 19. 3. 2014 při vyhledávání na klíčové slovo **MySQL** a pro práci na klasickou pracovní smlouvu (tedy ne na ŽL) **89 nabídek** z toho rozložení maximální hrubé mzdy:

- 45 nabídek na 10 000 Kč a více,
- 44 nabídek na 20 000 Kč a více,
- 40 nabídek na 30 000 Kč a více,
- 33 nabídek na 40 000 Kč a více,
- 20 nabídek na 50 000 Kč a více a
- 6 nabídek na 70 000 Kč a více.

Pro **PostgreSQL** je za stejných podmínek nabízeno **36 nabídek** a z toho

- 12 na 10 000 Kč a více,
- 12 na 20 000 Kč a více,
- 12 na 30 000 Kč a více,
- 10 na 40 000 Kč a více,
- 9 na 50 000 Kč a více a
- 4 na 70 000 Kč a více.

Mnoho nabídek na jobs.cz nemá uvedené mzdové rozpětí a pak se při filtrování dle mzdy neobjeví v součtech. Nicméně i tak si lze cirká udělat představu o mzdách nabízených těmito odborníkům. Lze vidět, že zatímco plat přesahující 50 000 je nabízen u tří čtvrtin nabídek na PostgreSQL, u MySQL je to jen u necelé poloviny nabídek.

Na stranu nabídky IT odborníků se lze podívat například na portálu společnosti Coolpeople, která sdružuje nabídky a poptávky prací nezávislých odborníků pracujících zpravidla formou obchodních smluv na konkrétních projektech. Na adrese coolpeople.cz lze pak tuto databázi filtrovat dle kompetencí a klíčových slov, čímž lze získat tato data:

Coolpeople má v databázi k 19. 3. 2014 **69953** IT odborníků z toho

- **17293** uvádí alespoň nějakou znalost MySQL a **611** seniorní znalost při filtrování dle kompetence
- Pro PostgreSQL není filtrování dle kompetencí k dispozici
- **19398** výsledků pro filtrování dle klíčového slova MySQL a
- **2267** výsledků pro filtrování dle klíčového slova PostgreSQL.

Dle vyhledávání na těchto dvou portálech lze tedy nahlédnout, že PostgreSQL odborníků je zřejmě na trhu méně a je pro ně i méně pracovních příležitostí, nicméně pro firmu jejich zaměstnání pravděpodobně vyjde nákladově hůře.

Pro srovnání - Český statistický úřad za rok 2012 eviduje průměrnou měsíční mzdu IT odborníků nad hranicí 42 tisíc Kč hrubé měsíční mzdy (45).

5 Výsledky a diskuse

V předchozích kapitolách byla provedena řada zkoumaných databázových strojů srovnání z různých hledisek.

Co se týče omezení velikosti výsledné databáze a maximální velikosti běžících databází, vítězí PostgreSQL (Kapitola 3.4.7).

PostgreSQL také nabízí méně omezující licenci na své použití (Kapitola 3.4.2) a také může být zajímavé pro aplikace geografických informačních systémů, ať už díky vlastním datovým typům (Kapitola 3.4.9) či přímo forku PostGIS, který se této oblasti věnuje (Kapitola 3.4.5).

Co se týče srovnání výkonnosti, zde vítězí MySQL s enginem MyISAM, který ale nenabízí mnoho z výhod relačních databází – například hlídání datové integrity (Kapitola 4.5.1). Pokud se přistoupí k porovnání úložného enginu InnoDB, vítězí spíše PostgreSQL (Kapitola 4.5.1 a 4.5.2).

Co se týče snadnosti instalace a podporovaných operačních systémů, jsou na tom obě databáze stejně (Kapitola 4.3 a 4.2), MySQL je na tom o něco hůře s dostupností instalačního balíčku pro Windows (Kapitola 4.1).

Výsledky srovnání výkonnosti

V praktické části práce bylo provedeno testování výkonnosti pomocí dvou nástrojů – database benchmark a pole position. Pro nástroj database benchmark byla dopsána podpora pro úložný engine InnoDB a bylo tak provedeno srovnání MySQL s úložnými enginy InnoDB i MyISAM a PostgreSQL.

V prvním testovacím scénáři, při zápisu 50 000 záznamů do 2 tabulek (tj. celkem 100 000 záznamů do databáze), průměrná rychlost zápisu do MyISAM dosahovala 27 049 záznamů za sekundu. Do InnoDB 12 495 záznamů za sekundu a do PostgreSQL 11 898 záznamů za sekundu. Celková rychlost provedení tohoto zápisu pak byla 3 s pro MyISAM, 8 s pro InnoDB a 8 s pro PostgreSQL. Výsledná velikost databáze byla 7,1 MB pro MyISAM, 0,1 MB pro InnoDB a 11,1 MB pro PostgreSQL.

Ve druhém testovacím scénáři byl proveden zápis 1 000 000 záznamů opět do 2 tabulek (tj. celkem 2 000 000 záznamů). Průměrná rychlost zápisu do MyISAM

dosahovala 53 425 záznamů za sekundu. Do InnoDB 501 záznamů za sekundu a do PostgreSQL 18 300 záznamů za sekundu. Celková rychlost provedení tohoto zápisu pak byla 37 s pro MyISAM, 1 h 6m a 28s s pro InnoDB a 1m 49 s pro PostgreSQL. Výsledná velikost databáze byla 141,1 MB pro MyISAM, 253,3 MB pro InnoDB a 218,7 MB pro PostgreSQL.

Dále byla testována rychlost čtení po zápisu z testovacího scénáře 2, tedy ze dvou milionů záznamů ve dvou tabulkách. Rychlost čtení dosahovala cca 106 000 záznamů za sekundu pro MyISAM, 14 000 záznamů za sekundu pro InnoDB a 73 000 záznamů pro PostgreSQL.

Posledním testem provedeným s pomocí nástroje database benchmark byl test, kdy dojde k prudkému poklesu výkonnosti při zápisu mnoha záznamů (simuluje se situace například logování většího množství záznamů ze senzorů). V tomto testu se ukázalo, že na testovacím stroji u MyISAM dochází k prudkému poklesu výkonnosti kolem 20 milionu vložených záznamů, u InnoDB u 800 tisíc vložených záznamů a u PostgreSQL u 4 milionu záznamů.

Při testování pomocí nástroje pole position bylo provedeno celkem 6 typů testů, 19 testovacích scénářů a každý z těchto scénářů byl škálován na 3 různé velikosti zpracovávaných dat. Výsledky jednotlivých testů se zabývá kapitola 4.5.2. Tento nástroj umožnil měřit i výkon pro více konkurenčních vláken přístupu k databázovým datům, ovšem neumožnil měřit výkon pro úložný engine MyISAM. Srovnány tedy byly pouze MySQL s úložným enginem InnoDB a PostgreSQL.

V *ComplexConcurency* testu se u menšího množství vláken (1 a 3 vlákna) ukázalo jako výkonnější PostgreSQL, zatímco u 10 vláken bylo předstiženo MySQL. Výsledky ovšem jsou řádově srovnatelné (rozdíl se pohybuje do 30 %).

V *InsertCentricConcurrency* testu PostgreSQL předstihlo MySQL a je dominující pro všechny tři úrovně škálování. Řádově jsou ovšem výsledky srovnatelné (rozdíl maximálně 40 %).

V *Complex* testu ve scénáři *write* se PostgreSQL ukazuje jako řádově výkonnější (od 2,8 násobku rychlosti u nejmenší škály, po 2,6 násobek u největší škály). Ve scénáři *query* se u menší škály testu ukázalo jako zhruba dvakrát výkonnější MySQL,

zatímco větší škály již bylo předstíženo PostgreSQL. Ve scénáři *update* se opět jako cca 2× výkonnější ukazuje PostgreSQL, ve scénáři *delete* opět PostgreSQL (od 1,8 násobku u nejmenší škály, do 2,6 násobku výkonnosti MySQL u největší).

V *NestedLists* testu se ve všech scénářích (*create*, *read*, *update*, *delete*) ukazuje PostgreSQL jako výkonnější (od 2,8 násobku u nejmenší škály do 1,9 násobku u největší).

V *InheritanceHierarchy* testu ve scénáři *write* projevilo jako výkonnější PostgreSQL (od 2,7 násobku do 2,2 násobku výkonu MySQL). Ve scénáři *read* se jako výkonnější ukazuje MySQL (od 2,1 násobku až po 20 násobek u největší škály). Ve scénáři *query* také MySQL (od 2,1 násobku do 11,2 násobku) a ve scénáři *delete* se naproti tomu opět ukazuje výkonnější PostgreSQL (cca 3 násobně).

Posledním provedeným testem byl *FlatObject* test.

Zde se ve scénářích *write*, *queryIndexedString*, *queryIndexedInt*, *update* i *delete* ukazuje jako výkonnější PostgreSQL (cca dvojnásobně až třinásobně ve všech scénářích).

Lze tedy říci, že PostgreSQL je výkonnější oproti MySQL s úložným enginem InnoDB ve většině testovacích scénářů a může tak být vhodné pro aplikace s větší zátěží na databázi.

Výsledky ekonomického srovnání

Co se týče ekonomiky provozu, jsou náklady na licence srovnatelné, pouze pokud by příslušné nasazení nesplnilo licenční podmínky pro provoz zdarma, je nutné u MySQL zakoupit licenci od cca 40 tisíc Kč bez DPH/rok.

Ceny za hardware, případně jinou infrastrukturu jsou u obou databázových systémů srovnatelné.

Náklady na odborné pracovníky jsou u PostgreSQL o něco vyšší, než u MySQL a také počty odborných pracovníků na trhu jsou menší (viz kapitolu 4.6.).

Výsledky srovnání jednotlivých metodik měření výkonu databázového stroje

Zřejmě nejvíce propracovanou metodiku nabízí konsorcium TPC. Samotná metodika je zdarma a u firem, kterým záleží na výkonu databáze, nabízí dobrou možnost srovnat si jednotlivé databázové stroje z hlediska výkonnosti ještě před vytvořením vlastní

databáze. Také nabízí pohledy na to, jak je možné vytvořit si zátěžové a výkonnostní testy pro samotnou aplikaci.

SPEC metodiky jsou poté vhodné pro testování výkonnosti aplikačního stroje jako celku.

Open source metodiky jsou vhodné jako rychlý test (například na slabých strojích) a jejich velkou výhodou je snadnost použití. Jsou tedy vhodné zejména pro studenty, začínající administrátory, malé projekty a firmy, které si nemohou dovolit investovat značné finanční prostředky do implementace komplexních testovacích nástrojů.

6 Závěr

Hlavním cílem práce je srovnání dvou open source databázových systémů. Toto bylo provedeno jak po stránce teoretické, tak praktické. Nejprve byla provedena řešerše této problematiky, potom vytvořeno testovací prostředí s nainstalováním obou databází v aktuálních verzích (PostgreSQL 9.3 a MySQL 5.6) a pomocí vybraných nástrojů byly změřeny výsledky výkonnosti. Samotné výsledky praktické části jsou samozřejmě odpovídající jen pro měřený stroj, nicméně navrženými postupy se může inspirovat a nechat se vést každý informační architekt, či databázový specialista, při hledání vhodné databáze pro svůj projekt.

Výsledky porovnání jednotlivých hodnotících kritérií lze nalézt v kapitole 5. Na základě těchto výsledků lze pomocí srovnávacích metodik (například AHP) provést výběr vhodného databázového stroje pro případné konkrétní nasazení.

Obecně se dle zjištěných skutečností potvrdil předpoklad, že MySQL je pro jeho snadný provoz a dostupnost aplikačních hostingů vhodné použít na jednodušší webové projekty (při použití engine InnoDB), či na databáze typu KVS (při použití engine MyISAM), který je také levnější na provoz.

Pro komplexnější analytické použití, bezpečnost dat a informační systémy je naopak zpravidla vhodným databázovým strojem PostgreSQL, jehož provoz je ale dražší zejména na odměnách pro potřebné IT odborníky.

Na závěr lze konstatovat, že cíle práce byly splněny a práce splňuje stanovený rozsah.

Práci by bylo možné dále rozšiřovat a doplňovat zejména o další srovnání výkonu dalšími metodikami, přidání porovnávání výkonnosti jednotlivých forků a případně znovu měření opakovat při vydávání nových verzí zkoumaných databázových strojů.

7 Slovníček

Tento slovníček poskytuje jednoduchý výklad pojmů používaných v práci. Tyto termíny jsou v práci použity pro jejich notorickou známost a mezinárodní použití. Čtenář je tedy může snadno použít jako klíčová slova pro další vyhledávání informací o řešené problematice.

Benchmark je měřicí scénář zaměřený zejména na rychlost a výkon. Často se tímto slovem také označují samotné výsledky testů, nebo použité nástroje.

Databázový stroj (též Systém řízení báze dat, databázový systém) je softwarové vybavení zajišťující práci s daty.

Fork je odvozená větev vývoje software (zejména u open-source). Zpravidla je vytvářen nezávisle a jinými vývojáři, než hlavní větev.

GUI (Graphical User Interface) je grafické uživatelské rozhraní. Pojem se používá jako protiklad textovému uživatelskému rozhraní (což je například ovládání pomocí příkazové řádky).

NoSQL databáze jsou databáze, které nepodporují dotazovací jazyk SQL. Často jsou takto označovány databáze, které vznikly až po nástupu SQL jako technického standardu. Většinou umožňují rychlejší práci s jednoduchými daty (ukládání a vyhledávání) za cenu snížení ostatních možností (např. Analýza dat, referenční integrita), které poskytují SQL databáze.

SQL (Structured Query Language) je dotazovací jazyk pro práci s daty v relačních databázích.

Seznam tabulek

| | |
|---|----|
| Tabulka 1 Vývoj SQL standardu | 15 |
| Tabulka 2 9 oblastí ISO SQL standardu | 16 |
| Tabulka 3 Ceník komerčních licencí MySQL pro rok 2013 dle (18)..... | 21 |
| Tabulka 4 Omezení velikosti MySQL souborů dle souborového systému | 25 |
| Tabulka 5 Některá omezení MySQL, dle (9) | 26 |
| Tabulka 6 Některá omezení PostgreSQL dle (31) | 36 |
| Tabulka 7 Parametry testovacího stroje..... | 51 |
| Tabulka 8 Výsledky testů database benchmark | 56 |
| Tabulka 9 Parametry druhého testovacího stroje..... | 56 |
| Tabulka 10 Shrnutí výsledků pole position | 68 |

Seznam obrázků

| | |
|---|----|
| Obrázek 1 Logo MySQL | 17 |
| Obrázek 2 LAMP server a jeho okolí - autor Shmuel Csaba Otto Traian | 18 |
| Obrázek 3 Historie vývoje verzí MySQL (zdroj Wikimedia Commons)..... | 22 |
| Obrázek 4 Vývoj MySQL a některých forků, autor Daniël van Eeden | 23 |
| Obrázek 5 Logo MariaDB | 24 |
| Obrázek 6 Logo PostgreSQL..... | 30 |
| Obrázek 7 Historie verzí PostgreSQL (zdroj Wikimedia Commons) | 34 |
| Obrázek 8 Logo Fujitsu supported PostgreSQL | 35 |
| Obrázek 9 Logo PostGIS | 35 |
| Obrázek 10 Schéma funkce Postgres-XC clusteru dle (36)..... | 37 |
| Obrázek 11 Někteří členové TPC (Zdroj (38))..... | 41 |
| Obrázek 12 Počet stažení Open Source Database Benchmark (zdroj sourceforge.net) | 43 |
| Obrázek 13 Percona server test výkonu vůči MySQL (zdroj (23)) | 48 |
| Obrázek 14 Percona server - test doby odezvy vůči MySQL (zdroj (23)) | 49 |
| Obrázek 15 Database benchmark, testovací scénář 1.1 (zdroj: vlastní tvorba) | 52 |
| Obrázek 16 Database benchmark, testovací scénář 2.1 (zdroj: vlastní tvorba) | 52 |
| Obrázek 17 Database benchmark, testovací scénář 1.2 (zdroj: vlastní tvorba) | 54 |
| Obrázek 18 Database benchmark, testovací scénář 2.2 (zdroj: vlastní tvorba) | 54 |
| Obrázek 19 Database benchmark, pokles rychlosti insertu (zdroj: vlastní tvorba) | 55 |
| Obrázek 20 Pole position - ComplexConcurrency test (zdroj: vlastní tvorba) | 58 |
| Obrázek 21 Pole position - InsertCentricConcurrency test (zdroj: vlastní tvorba)..... | 59 |
| Obrázek 22 Pole position - Complex test – write (zdroj: vlastní tvorba) | 60 |
| Obrázek 23 Pole position - Complex test – query (zdroj: vlastní tvorba) | 60 |

| | |
|--|----|
| Obrázek 24 Pole position - Complex test – update (zdroj: vlastní tvorba)..... | 61 |
| Obrázek 25 Pole position - Complex test – delete (zdroj: vlastní tvorba)..... | 61 |
| Obrázek 26 Pole position - NestedList test – create (zdroj: vlastní tvorba) | 62 |
| Obrázek 27 Pole position - NestedList test – read (zdroj: vlastní tvorba)..... | 62 |
| Obrázek 28 Pole position - NestedList test – update (zdroj: vlastní tvorba) | 63 |
| Obrázek 29 Pole position - NestedList test – delete (zdroj: vlastní tvorba) | 63 |
| Obrázek 30 Pole position - InheritanceHierarchy test – write (zdroj: vlastní tvorba)..... | 64 |
| Obrázek 31 Pole position - InheritanceHierarchy test – read (zdroj: vlastní tvorba) | 64 |
| Obrázek 32 Pole position - InheritanceHierarchy test – query (zdroj: vlastní tvorba)..... | 65 |
| Obrázek 33 Pole position - InheritanceHierarchy test – delete (zdroj: vlastní tvorba)..... | 65 |
| Obrázek 34 Pole position - FlatObject test – write (zdroj: vlastní tvorba)..... | 66 |
| Obrázek 35 Pole position - FlatObject test – queryIndexedString (zdroj: vlastní tvorba) .. | 66 |
| Obrázek 36 Pole position - FlatObject test – queryIndexedInt (zdroj: vlastní tvorba)..... | 67 |
| Obrázek 37 Pole position - FlatObject test – update (zdroj: vlastní tvorba)..... | 67 |
| Obrázek 38 Pole position - FlatObject test – delete (zdroj: vlastní tvorba)..... | 67 |

Seznam použitých zdrojů

1. MySQL vs PostgreSQL. *wikivs*. [Online] 2013. [Citace: 10. 12 2013.]
2. ŠUŇAVEC, Ján. MySQL vs PostgreSQL vs Firebird. *root.cz*. [Online] 2005. [Citace: 10. 12 2013.] <http://www.root.cz/clanky/mysql-vs-postgresql-vs-firebird/>.
3. STĚHULE, Pavel. MySQL vs PostgreSQL vs Firebird II. *root.cz*. [Online] 2005. [Citace: 10. 12 2013.] <http://www.root.cz/clanky/mysql-vs-postgresql-vs-firebird-ii/>.
4. KOLÁŘ, Radim. 100MB TPC-B test databází. *linuxoft.cz*. [Online] 2010. [Citace: 10. 12 2012.] http://www.linuxsoft.cz/article.php?id_article=1771.
5. NAMBIAR, Raghunath a POESS, Meikel. *Performance Evaluation and Benchmarking Second TPC Technology Conference, TPCTC 2010, Singapore, September 13-17, 2010*. místo neznámé : Springer, 2011. 9783642182051.
6. ORACLE. Tasks of a Database Administrator. *Oracle® Database Administrator's Guide*. [Online] 2008. [Citace: 10. 12 2013.] http://docs.oracle.com/cd/B28359_01/server.111/b28310/dba002.htm#ADMIN11031.
7. LAROCK, Thomas. YOU'RE DOING IT WRONG: 5 FACTORS THAT AFFECT DATABASE PERFORMANCE. *SQL Rockstar*. [Online] 2012. [Citace: 10. 12 2013.] <http://thomaslarock.com/2012/08/youre-doing-it-wrong-5-factors-that-affect-database-performance/>.
8. BEAUCHEMIN, Bob. How Data Access Code Affects Database Performance. *MSDN Magazine*. [Online] 2009. [Citace: 10. 12 2013.] <http://msdn.microsoft.com/en-us/magazine/ee236412.aspx>.
9. MySQL komunita. MySQL Forums. *MySQL*. [Online] [Citace: 10. 12 2012.] <http://forums.mysql.com/>.
10. Komunita PostgreSQL. Resources. *PostgreSQL*. [Online] [Citace: 10. 12 2013.] <http://www.postgresql.org/community/resources/>.
11. Stack exchange inc. MySQL tagged questions. *StackOverflow*. [Online] [Citace: 10. 12 2013.] <http://stackoverflow.com/questions/tagged/mysql>.

12. —. PostgreSQL tagged questions. *StackOverflow*. [Online] [Citace: 10. 12 2013.] <http://stackoverflow.com/questions/tagged/postgresql>.
13. Oracle Corporation. History of SQL. *Oracle® Database SQL Reference*. [Online] [Citace: 15. 2 2014.] <http://www.stanford.edu/dept/itss/docs/oracle/10g/server.101/b10759/intro001.htm>.
14. SEIB, Carrie a FLORIN, Corinne. American National Standards Institute. X3H2 Records, 1978-1995. Finding Aid. *Charles Babbage Institute Center for the History of Information Technology*. [Online] 2003. [Citace: 12. 2 2014.] <http://special.lib.umn.edu/findaid/xml/cbi00168.xml>.
15. The PostgreSQL Global Development Group. Appendix D. SQL Conformance. *PostgreSQL*. [Online] [Citace: 10. 2 2014.] <http://www.postgresql.org/docs/9.3/static/features.html>.
16. Sun Microsystems, Inc. Sun to Acquire MySQL. *MySQL*. [Online] 16. 1 2008. [Citace: 12. 12 2013.] <http://www.mysql.com/news-and-events/sun-to-acquire-mysql.html>.
17. KOETSIER, John. Android reaches massive 80% market share, Windows Phone hits global high, iPhone languishes. *Venturebeat*. [Online] 1. 8 2013. [Citace: 13. 12 2013.] <http://venturebeat.com/2013/08/01/android-reaches-massive-80-market-share-windows-phone-hits-global-high-iphone-languishes/>.
18. Oracle corporation inc. About MySQL. *MySQL*. [Online] [Citace: 13. 12 2013.] <http://www.mysql.com/about/>.
19. —. ORACLE store. *ORACLE store*. [Online] 2013. [Citace: 14. 12 2013.] <https://shop.oracle.com/>.
20. —. MySQL Community Edition. *MySQL*. [Online] 2013. [Citace: 14. 12 2013.] <http://www.mysql.com/products/community/>.
21. —. FOSS License Exception. *MySQL*. [Online] 23. 2 2012. [Citace: 14. 12 2013.] <http://www.mysql.com/about/legal/licensing/foss-exception/>.
22. United States District Court, D. Massachusetts. *PROGRESS SOFTWARE CORP. v. MYSQL AB*. 195 F.Supp.2d 328 (2002), 195 F.Supp.2d 328 (2002). Massachusetts : United States District Court, D., 28. 2 2002.

23. Percona LLC. Benchmarks For Percona Server. *Percona*. [Online] 2013. [Citace: 10. 12 2013.] <http://www.percona.com/software/percona-server/benchmarks>.
24. Red Hat, Inc. Replace MySQL with MariaDB. *Fedora*. [Online] 12. 7 2013. [Citace: 10. 12 2013.] <https://fedoraproject.org/wiki/Features/ReplaceMySQLwithMariaDB>.
25. VAUGHAN-NICHOLS, Steven J. Look out, Oracle: SkySQL and MariaDB join forces. *ZDNet*. [Online] 23. 4 2013. [Citace: 10. 12 2013.] <http://www.zdnet.com/look-out-oracle-skysql-and-mariadb-join-forces-7000014411/>.
26. Oracle Corporation. MySQL Cluster CGE. *MySQL*. [Online] 2014. [Citace: 12. 2 2014.] <http://www.mysql.com/products/cluster/>.
27. Oracle Corporation. MySQL Cluster Benchmarks: Oracle and Intel Achieve 1 Billion Writes per Minute. *MySQL*. [Online] 2014. [Citace: 12. 2 2014.] <http://www.mysql.com/why-mysql/white-papers/mysql-cluster-benchmarks-1-billion-writes-per-minute/>.
28. Oracle Corporation. MySQL Standards Compliance. *MySQL*. [Online] 2014. [Citace: 16. 2 2014.] <https://dev.mysql.com/doc/refman/5.6/en/compatibility.html>.
29. —. MySQL Extensions to Standard SQL. *MySQL*. [Online] 2014. [Citace: 16. 2 2014.] <https://dev.mysql.com/doc/refman/5.6/en/extensions-to-ansi.html>.
30. BARWICK, Ian. MySQL Gotchas. *SQL-info.de*. [Online] 28. 1 2014. [Citace: 16. 2 2014.] <http://sql-info.de/mysql/gotchas.html>.
31. The PostgreSQL Global Development Group. About. *PostgreSQL*. [Online] [Citace: 10. 12 2013.] <http://www.postgresql.org/about/>.
32. LAI, Eric. Size matters: Yahoo claims 2-petabyte database is world's biggest, busiest. *Computerworld*. [Online] 22. 5 2008. [Citace: 17. 12 2013.] http://www.computerworld.com/s/article/9087918/Size_matters_Yahoo_claims_2_petabyte_database_is_world_s_biggest_busiest.
33. The PostgreSQL Global Development Group. Licence. *PostgreSQL*. [Online] [Citace: 5. 12 2013.] <http://www.postgresql.org/about/licence>.

34. BERKUS, John. Elephant Roads: a tour of Postgres forks. *Slideshare*. [Online] 10 2009. [Citace: 20. 12 2013.] <http://www.slideshare.net/pgconf/elephant-roads-a-tour-of-postgres-forks>.
35. The PostgreSQL Global Development Group. Log-Shipping Standby Servers. *PostgreSQL*. [Online] [Citace: 15. 2 2014.] <http://www.postgresql.org/docs/9.3/static/warm-standby.html>.
36. Postgres-XC team. Home. *Postgres-XC Wiki*. [Online] [Citace: 15. 2 2014.] http://postgresxc.wikia.com/wiki/Postgres-XC_Wiki.
37. TPC. TPTC Charter. *TPC*. [Online] [Citace: 5. 1 2014.] <http://www.tpc.org/tpctc/default.asp>.
38. Standard Performance Evaluation Corporation. The SPEC Consortium: Members and Associates. *Standard Performance Evaluation Corporation*. [Online] [Citace: 22. 2 2014.] <http://www.spec.org/consortium/>.
39. The PostgreSQL Global Development Group. pgbench. *PostgreSQL*. [Online] [Citace: 15. 1 2014.] <http://www.postgresql.org/docs/9.3/static/pgbench.html>.
40. Oracle Corporation. Supported Platforms: MySQL Database. *MySQL*. [Online] 2014. [Citace: 28. 2 2014.] <http://www.mysql.com/support/supportedplatforms/database.html>.
41. Red Hat, Inc. Cygwin. *Cygwin*. [Online] [Citace: 22. 2 2014.] <http://cygwin.com/>.
42. Oracle Corporation. Tuning Server Parameters. *MySQL*. [Online] 2014. [Citace: 22. 2 2014.] <https://dev.mysql.com/doc/refman/5.0/en/server-parameters.html>.
43. Implementation. *Database benchmark*. [Online] [Citace: 12. 2 2014.] <http://database-benchmark.com/implementation/>.
44. BERÁNEK, Ondřej. Podnikáte v oblasti IT? Můžete pro svou firmu získat dotace z EU. *podnikatel.cz*. [Online] 6. 8 2010. [Citace: 13. 3 2012.] <http://www.podnikatel.cz/clanky/it-podnikani-dotace-eu-finance/>.

45. SKARLANDTOVÁ, Eva. Počet IT odborníků v České republice stále stoupá. *Český statistický úřad*. [Online] 11. 5 2013. [Citace: 19. 3 2014.] http://www.czso.cz/csu/tz.nsf/i/pocet_it_odborniku_v_ceske_republice_stale_stoupa_20131105. ČSÚ TZ 131105.
46. PONKRÁC, Miloslav. *PHP a MySQL bez předchozích znalostí*. místo neznámé : Computer press, 2012. 9788025117583.
47. MOMJIAN, Bruce. *PostgreSQL*. místo neznámé : Computer Press, 2003. 80-7226-954-2.
48. HATCH, Paxman. *PostgreSQL Starter Kit*. místo neznámé : Amazon, 2012. B008T02LIC.
49. SCHWARTZ, Baron. *High Performance MySQL: Optimization, Backups, and Replication*. místo neznámé : O'Reilly Media, 2012. 1449314287.
50. MOLINARO, Anthony. *SQL: Kuchařka programátora*. místo neznámé : Computer Press, 2009. 9788025126172.
51. Percona LLC. Benchmarks For Percona Server. *Percona*. [Online] 2014. [Citace: 28. 2 2014.] <http://www.percona.com/software/percona-server/benchmarks>.