

PA036: Projekt z DB systémů

Vlastislav Dohnal

Cíle předmětu

- Praktické používání databázového systému
 - Využití rozšířených vlastností
- Vytvoření týmového projektu a jeho prezentace
 - Tým = 3-4 studenti
- Témata projektů: „Nové“ možnosti DB systémů
 - Představení problematiky
 - Otestování výkonnosti
 - Porovnání s jinými možnostmi

Databázové systémy

- Relační – PostgreSQL
- NoSQL – dokumentové, klíč-hodnota, sloupcové, grafové
- Přístup k DBMS
 - Aplikační frameworky (Nette, Laravel, ...)
 - REST API
 - JDBC/ODBC

Dokumentové NoSQL databázové systémy

- MongoDB
 - Datový model - **JSON dokument**
 - Dynamické schéma
 - Primární přístup k dokumentu pomocí ID
 - Sekundární přístup – index nad vybraným atributem dokumentu
- CouchDB
 - Analogický k MongoDB

Klíč-hodnota NoSQL databázové systémy

- Redis

- Datový model – dvojice klíč-hodnota
- Primitivní, vysoce výkonné operace nad seznamy, množinami a asoc. poli
- Často jako in-memory cache – lze nastavit expiraci záznamů
- Zpracování GEO souřadnic – včetně rozsahových dotazů

- Riak

- Analogický k Redis, navíc zvládá JSON dokumenty jako hodnoty, umí sekundární indexy
 - Indexy se musí udržovat ručně

Sloupcové / grafové NoSQL DB systémy

- Cassandra
 - Relační datový model
 - Variabilní schéma – skupiny sloupců
 - CQL jazyk podobný SQL
- Neo4j
 - Datový model – orientovaný multigraf

Průběh řešení projektu

- Zvolení tématu a specifikace cílů projektu
 - Podrobný plán a způsob práce
 - definice kompetencí členů týmu, časový harmonogram
 - Týdenní reporty členů týmu a strávený čas (sdílený Google doc)
 - stačí jedna věta a číslo -> identifikace nepracujících a jejich hodnocení „X“
 - konzultace s vyučujícím
 - Výsledky projektu do závěrečné zprávy
 - stručně zadání a cíle projektu
 - přístup k řešení, např. popis technologií, dat
 - výsledky experimentů
 - zhodnocení časového plánu
- První prezentace
- Druhá prezentace

Témata projektů pro 2019

- Cizí datové zdroje
- Řízení přístupu k záznamům relace
- Rozšířené statistiky a výkon DB
- Autentizace DB spojení a aplikační framework
- Monitorování databázového systému
- Aplikační framework a kešování DB
- Notifikace DB serveru a aplikační Framework
- High Availability in PostgreSQL
- Master-master replikace
- Výkonnost zpracování JSON dokumentů
- Horizontální dělení relací
- Archivace dat a obnovení
- Temporální databáze
- Analytické dotazy a materializované pohledy
- Optimalizace úložiště DB

Projekt: Cizí datové zdroje v PostgreSQL

- Zpřístupnění datového zdroje v PostgreSQL
 - Tzv. Foreign Data Wrappers - https://wiki.postgresql.org/wiki/Foreign_data_wrappers
- Vyberte si min. dvě implementace (nejlépe 1x NoSQL DBMS, 1x souborový)
 - Představte funkcionalitu a dostupné operace (DML)
 - Otestujte i netriviální SELECT dotazy (joiny, agregace)
 - Prezentujte na modelové aplikaci/datech (dle vlastního výběru)
 - Změřte i vůči nativnímu řešení v NoSQL
- Prezentace projektu:
 - První:
 - Představení principu Foreign Data Wrapper
 - Vybraní reprezentanti, zdůvodnění proč a možnosti využití
 - Návrh vhodné aplikace / experimentů
 - Druhá:
 - Stručně obsah první
 - Popis aplikace / experimentu (nastavení, data, co testuji, ...)
 - Výsledky testů / zkušenost

Projekt: Řízení přístupu k záznamům relace

- Prozkoumejte možnosti omezení přístupu k záznamům tabulek jednotlivým uživatelům
 - Představte: (i) pohledy a jejich aktualizaci pomocí triggerů; (ii) „politiky“ ([policy](#)) v PostgreSQL; (iii) uložené procedury
 - Porovnejte jejich výkonnost a snadnost implementace (celé DML)
 - Porovnejte možnosti přidávání/rušení/přejmenovávání/změny typu atributů v takto omezených tabulkách (DDL)
 - Popište možnosti využití pro horizontal partitioning
- Prezentace projektu:
 - První:
 - Představení principu omezení záznamů (Views vs. Row-Level Security)
 - Porovnání jejich vlastností (výhody vs. nevýhody), i z pohledu DDL, indexování, možných útoků, ...
 - Návrh vhodných experimentů (DML i DDL)
 - Druhá:
 - Stručně obsah první
 - Zkušenost s řešením
 - Výsledky výkonostních testů

Projekt: Rozšířené statistiky a výkon DB

- Prozkoumejte možnosti rozšířených statistik na vyhodnocování dotazů
 - Představte: CREATE/ALTER/DROP STATISTICS v PostgreSQL
 - Analyzujte vliv na plánovač dotazů a následnou výkonnost zpracování dotazů (SELECT)
 - Např. násobné where podmínky, joiny na více atributech
 - Analyzujte vliv na INSERT/UPDATE/DELETE
 - Kolik místa zabírají na disku, jak dlouho trvá update statistik.
- Prezentace projektu:
 - První:
 - Představení problematiky
 - Návrh vhodných experimentů – typy dotazů (podmínky, joiny); aktualizace záznamů
 - Druhá:
 - Stručně obsah první
 - Zkušenost s řešením
 - Výsledky výkonostních testů

Projekt: Monitorování databázového systému

- Monitorování výkonnosti databáze (např. [PgFouine](#), pgBadger, PGObserver)
 - Sledování „pomalých“ dotazů, četnosti využívání indexů, četnosti „full-scan“ tabulek, vytížení CPU, vytížení IO, obsazenosti úložiště
- Hlášení při překročení definovaných mezí
 - Meze jak absolutně (prům. doby zpracování častých dotazů překročí def. max),
 - tak i relativně (zatížení/prům. doba zpracování se zvýšila o více než x %)
- Testování škálovatelnosti
 - zvyšovat počet dotazů (DML), zvyšovat objem dat.
- Prezentace projektu:
 - První:
 - Představení problému, možnosti sledování (zdroje informací, dodatečné SW) – lze se omezit pouze na Linux platformu
 - Návrh testovací aplikace, ukázka logování v DB (záznamu sledování)
 - Návrh alertingu
 - Druhá:
 - Stručně obsah první
 - Způsob realizace, představení implementace
 - Popis a výsledky testů, praktická ukázka

Projekt: Aplikační framework a kešování DB

- Prostudujte možnosti kešování výsledků databázových dotazů pro některý z aplikačních frameworků, např. Laravel, Nette pro PHP.
- Implementujte příklad, na kterém předvedete funkčnost a výkonnost řešení.
- Prezentace projektu:
 - První:
 - Základní představení zvoleného frameworku
 - Volba příkladu pro implementaci, resp. web aplikace
 - Návrh vhodných experimentů, měření přístupu k datům v DB (přečtených bloků tabulek, disk IOs, memory usage, ...)
 - Druhá:
 - Stručně obsah první
 - Způsob realizace kešování
 - Popis a výsledky experimentu (nastavení, data, ...)

Projekt: Notifikace DB serveru a aplikační framework

- PostgreSQL umožňuje klientům posílat notifikace při libovolné události pomocí příkazu NOTIFY, resp. přijímat pomocí LISTEN.
- Analyzujte možnosti použití v aplikačních frameworkcích (např. Nette) a porovnejte s nativní implementací (např. v JDBC).
- Prezentace projektu:
 - První:
 - Představení problému, vhodného příkladu a jeho implementace
 - Volba frameworku pro možnost implementace do jeho DB abstrakce
 - Představení experimentů (měření doby od vzniku události po aktualizaci dat, zvětšování počtu událostí a porovnání s naivním přístupem)
 - Druhá:
 - Stručně obsah první
 - Způsob realizace ve zvoleném frameworku
 - Popis a výsledky experimentů

Projekt: High Availability in PostgreSQL (A)

- Cíl: nasazení master-slave replikace na existující databázi
- Nastudovat problematiku master-slave replikace
 - Nakonfigurovat replikaci pro dvojici serverů (Ize využít [Stratus.FI](#))
 - Implementovat migrační skript (deaktivace master, povýšení slave na master)
 - Uvažujte i migraci IP adresy dedikované pro master server
- Navrhnout vhodné experimenty pro ověření funkčnosti a chování řešení
 - při výpadku jednoho ze serverů změřte čas:
 - provedení „fail-over“ na slave,
 - obnovení plné funkčnosti po náběhu chybujícího serveru (jako nový slave)
 - nalezněte případné nekonzistence/chybějící data
 - výkonnostní testy dotazování (celé DML) v TPS, porovnání s jedním strojem
- Použijte vybranou možnost:
 - nativní replikace PostgreSQL
 - SkyTools 3 (<http://skytools.projects.pgfoundry.org/skytools-3.0/>)
 - Slony
 - Pgpool-II
 - pglogical

Projekt: High Availability in PostgreSQL (cont.)

- Prezentace:
 - První
 - zvolená možnost a její představení,
 - plán experimentů (co chci měřit, na jakých datech (velikost), kolik klientů přistupuje, ...)
 - Druhá
 - Stručně obsah první
 - Postup nasazení řešení
 - Prezentace výsledků experimentů

Projekt: High Availability in PostgreSQL (B)

- Cíl: existující master-slave řešení a upgrade SW vybavení DB systému
- Nastudovat problematiku master-slave replikace
 - Nakonfigurovat replikaci pro dvojici serverů (lze využít [Stratus.FI](#))
 - Provést aktualizaci SW DB systému (major version upgrade vs. minor version upgrade)
 - soustředit se na zero downtime řešení
- Navrhnout vhodné experimenty pro ověření funkčnosti a chování řešení
 - výkonnostní testy dotazování (celé DML) v TPS – snížení výkonu v době upgradu
- Použijte vybranou možnost:
 - nativní replikace PostgreSQL
 - SkyTools 3 (<http://skytools.projects.pgfoundry.org/skytools-3.0/>)
 - Slony
 - Pgpool-II
 - pglogical
- Prezentace – analogicky k zadání (A)

Projekt: Master-master replikace (distrib. DB)

- Nastudovat problematiku master-master replikace
 - Nakonfigurovat replikaci pro dvojici serverů (lze využít [Stratus.FI](#))
 - Realizovat pomocí Postgres-BDR / pglogical
 - Vliv na implementaci existující aplikace – např. přesměrování dotazů na stroj s daty
 - Možnosti nastavení – horizontální partitioning (data sharding)
- Navrhnout vhodné experimenty
 - Výkonnostní testy dotazování (celé DML) v TPS, porovnání s jedním strojem
 - Ověření funkčnosti a chování řešení při výpadku jednoho ze serverů
 - provedení „fail-over“ na slave,
 - obnovení plné funkčnosti po náběhu chybného serveru (jako nový slave)
- Prezentace – podle plánu pro Master-Slave projekt

Projekt: Výkonnost zpracování JSON

- Představte práci s JSON v PostgreSQL i MongoDB
 - Na příkladu demonstруйте použití
- Porovnejte MongoDB a PostgreSQL ve zpracování JSON dokumentů
 - Funkcionalita (z pohledu změn v dokumentu, indexování dokumentů)
 - Výkonnost (aktualizace dokumentu, vyhledávání podle obsahu dokumentů)
- Prezentace projektu:
 - První:
 - Představení práce s JSON
 - Základní porovnání
 - Návrh vhodných experimentů, resp. malé web aplikace, která data ukládá v JSON
 - Druhá:
 - Stručně obsah první
 - Popis experimentu (nastavení, data, ...)
 - Výsledky experimentů

Projekt: Horizontální dělení dat

- Cíl: nasazení horizontálního dělení na existující tabulku
- Nastudovat problematiku horizontálního dělení dat (ručně vs. automaticky)
 - Implementovat pro testovací data
 - Provést „upgrade“ tabulky na dělenou, ale také zpět (zrušení dělení)
 - Provést změnu implementace dělení – z ručního na automatické
- Navrhnout vhodné experimenty pro ověření funkčnosti a chování řešení
 - zaměřit se na zero downtime řešení
 - výkonnostní testy dotazování (celé DML) v TPS, porovnání s jednou tabulkou
- Prezentace:
 - První:
 - Představení vhodné aplikace
 - Návrh řešení horizontálního dělení
 - Funkční požadavky a plán experimentů
 - Druhá:
 - Stručně obsah první
 - Způsob implementace v PostgreSQL a externí skripty
 - Popis a výsledky experimentů

Projekt: Archivace dat a obnovení na žádost

- Realizujte přesuny zastaralých dat do archivu a jejich obnovu na požádání
 - Nejlépe pomocí horizontálního dělení a dědičnosti (tabulka pro každý měsíc)
 - Archiv jsou pak oddělené tabulky, které mohou být i v jiné DB
 - Obnovení z archivu do „živých“ dat
 - Pamatování si, co bylo obnoveno, aby nebylo při archivaci opět zrušeno
 - Výkonnostní testy – jak dlouho trvá archivace, jak dlouho trvá obnovení, jaký je vliv na výkonnost na živých datech, pokud je/není archivace prováděna.
- Prezentace:
 - První:
 - Představení vhodné aplikace
 - Návrh řešení horizontálního dělení
 - Funkční požadavky a plán experimentů
 - Druhá:
 - Stručně obsah první
 - Způsob implementace v PostgreSQL a externí skripty
 - Popis a výsledky experimentů

Projekt: Temporální databáze

- Nastudujte problematiku, najděte možnosti implementace
- Implementujte/instalujte pro PostgreSQL (chronomodel, temporal_tables)
- Otestujte výkonnost
- Prezentace projektu:
 - První:
 - Představení problému, vhodného řešení
 - Návrh aplikace pro předvedení problematiky
 - Představení výkonnostních experimentů (změna výkonu při více „starých“ záznamech, porovnání s „netemporálním“ řešením (tj. nemám historii vůbec))
 - Druhá:
 - Stručně obsah první
 - Způsob implementace v PostgreSQL
 - Popis a výsledky experimentů

Projekt: Analytické dotazy a materializované pohledy

- Představte možnosti materializovaných pohledů v PostgreSQL a analytických dotazů (OLAP)
 - Pro zadaná testovací data navrhňte a implementujte analytické dotazy
 - Porovnejte výkonnost
- Prezentace projektu:
 - První:
 - Představení problematiky, představení testovacích dat
 - Návrh aplikace pro analytiku (příklady dotazů)
 - Návrh výkonnostních experimentů
 - Druhá:
 - Stručně obsah první
 - Implementace dotazů v PostgreSQL, ukázka aplikace
 - Popis a výsledky experimentů

Projekt: Analytické dotazy a materializované pohledy (varianta „A“)

- Testovací data (viz dump vzorku dat ve studijních materiálech)
 - `conn_log(log_key, sim_imsi, time, car_key, pda_imei, gsmnet_id, method, program_ver)`
 - 267108664 | 230024100616400 | 2017-01-01 01:00:00.322+01 | 4033 | 867897023525224 | 26203 | U | A38
 - Vzorek 3,710,626 záznamů; celkově počítejte s objemem 53,559,217 záznamů
 - `service_log(service_key, car_key, time, app_run_time, pda_run_time, device)`
 - 129698573 | 2544 | 2017-01-01 11:27:20+01 | 0.17 | 112.67 | HUAWEI Y600-U20
 - Vzorek 2,323,534 záznamů; celkově počítejte s objemem 30,068,431 záznamů
- Dotazy
 - Pro každou verzi programu zjistit
 - počty různých zařízení; počet jeho restartů programu (`app_run_time ~ 0`)
 - Pro každé zařízení (`pda_imei`) zjistit počet restartů programu
 - Pro každé zařízení zjistit
 - kdy bylo uvedeno do provozu; kolik hodin bylo používáno (“suma” `pda_run_time`); kolik celkem bylo restartů
 - ...
- Prezentovat
 - Výkon dotazů, optimalizovat pomocí materializovaných pohledů
 - Dotazy počítat pro různá časová údobí (měsíčně, ročně)
 - Výsledky zobrazit v GUI (Kibana, Grafana, ...)

Projekt: Analytické dotazy a materializované pohledy (varianta „B“)

- Testovací data (viz dump vzorku dat ve studijních materiálech)
 - tracking(pos_key, car_key, time, car_status, pos_gps, speed)
 - 222400281, 2412, '2017-01-01 02:50:37+01', '*', '(49.854599999999978,13.8704000000000001)', 0.00
 - Vzorek 691,990 záznamů; celkově počítejte s objemem 4,546,844 záznamů
 - service(service_key, car_key, time, pos_gps, code, liter, price, currency)
 - 129715637, 4020, '2017-01-02 01:20:23+01', '(48.948099999999966,16.7285000000000004)', 'C\$\$\$', 28.61, NULL, 'CZK'
 - Vzorek 2,927 záznamů; celkově počítejte s objemem 18,010 záznamů
- Problémy k řešení
 - Detekce parkovišť (veřejných vs. firemních) a čerpacích stanic
 - Provést grupování záznamů s daným stavem auta (spánek „*“, stání „_“) nebo události (tankování) a detekovat uvedená místa
 - Grupování musí být flexibilní (odrazy GPS, rozlehlé parkoviště), např. max. 200 metrů
 - Odfiltrovat malé shluky, např. < 5 záznamů
 - Výsledky ukládat do vhodné tabulky
 - Realizovat aktualizaci tabulky (přidávání nově nalezených, rušení nepoužívaných), vývoj v čase (aktualizovat např. 1x týdně)
 - Ukládat i význam detekovaného bodu (kolik aut tam bylo, kolik různých firem tam bylo)
- Prezentovat
 - Řešení dotazů a jejich výkon
 - Zobrazit na mapě, popř. dotazem na Google API zkusit zjistit detaily

Projekt: Optimalizace úložiště DB

- Představte problematiku tablespaces a organizaci úložiště pro DB a logů
 - Navrhněte a otestujte několik scénářů (1 až n discích)
 - Žurnál (log) na samostatném disku vs. společně s daty
 - Různé RAID konfigurace RAID10, RAID5
 - Nastavení souborového systému
 - Provedení adekvátních výkonnostních testů
 - Velký objem dat
 - Prevaletní čtení vs. změna dat (oboje ve všech částech)
- Prezentace
 - První:
 - Představení problematiky a možností v PostgreSQL
 - Testovací prostředí (dodá vyučující nebo vlastní ((-:))
 - Plán experimentů
 - Druhá:
 - Stručně první
 - Prezentace výsledků a doporučení

Poznámka k realizaci

- Experimenty
 - Každý test musí být proveden opakovaně
 - Zachycovat požadované metriky
 - Prezentovat průměrnou hodnotu metriky, její odchylku
- Data
 - lze použít pgbench, který generuje data odvozená z TPC-B benchmarku
 - jiný TPC benchmark (viz <http://www.tpc.org/information/benchmarks.asp>)
- Časové nároky
 - dotace 2 kredity -> 52 člověkohodin na semestr
 - započítat povinnou přítomnost na semináři (4x prezentace)
 - plánovat v Gantt diagramu

Průběh semináře

- 1. fáze
 - Sestavení týmu
 - Volba tématu projektu, volba náhradního tématu projektu; určení kompetencí v týmu
- 2. fáze
 - Bližší specifikace projektu, konzultace s vyučujícím, stanovení finálního rozsahu
 - Včetně Gantt diagramu prací (co, kdy, kdo) včetně odhadů pracnosti (v člověkohodinách), nejlépe s týdenní granularitou termínů
- 3. fáze
 - Odevzdání specifikace projektu (lze i přímo jako prezentaci ppt/pdf)
 - Vypracování posudku na projekt týmu „s náhradním tématem“
- 4. fáze
 - Prezentace projektu ostatním, představení posudku oponenty, diskuze
- 5. fáze
 - Vlastní realizace (doma 😊, konzultace s vyučujícím v rámci hodin semináře, týmové schůzky a týdenní reporty)
- 6. fáze
 - Odevzdání hotového díla včetně Gantt diagramu doplněného o skutečné časové nároky a termíny dokončení (tj. plán vs. skutečnost)
- 7. fáze
 - Finální prezentace výsledků projektu

Dotazy: ?_