Analýza a výber DBS

TP20/21, SmartSpacers, Sprint 1

1. Úvod

V rámci tejto analyzy dôležité je pochopiť že IoT nie je o samotných veciach a od údajoch ktoré tieto inteligentné veci vytvárajú a uchovávajú. Organizácie sa spoliehajú na tieto údaje, aby poskytli lepšiu používateľskú skúsenosť, urobili inteligentnejšie rozhodnutia o dizajne a nakoniec podporili rast databázy.

Však, nič z nie je možne bez spoľahlivého databazoveho systemu , ktorý bude schopný spracovať obrovské množstvo dát generovaných zariadeniami IoT.

2. Dôležité vlastnosti IoT databázového systému

- **škalovatelnost**

  Pre nove IoT systemy, ktore su v stave navrhu, použitie databázy s možnosťou ľahkého rozšírenia bude viac praktickým riešením. Toto umožní rozširovanie zdrojov podľa potreby IoT aplikácie a vysoké počiatočné investície nebudu vyžadované.

  SQL databázy podporuju vertikálnu škálovateľnosť, zatiaľ čo NoSQL podporuju horizontálnu škálovateľnosť. Vertikálna škálovateľnosť vyjadruje schopnosť zvýšiť výkon jedného uzla s pridaním zdrojov ako napr pamäte alebo procesorov do toho istého uzla. Pri horizontálnej škálovateľnosti, počet uzlov (serverov) sa zvyšuje tak, aby distribuovať zaťaženie systému.

- **rychlost získavania a zápisu údajov**

  Táto charakteristika je veľmi dôležitá pre DBS, preto že v rámci IoT projektu sensory a device produkuju velky pocet udajov, ktore potrebujeme rychlo zapisovať a získavať. V SQL, všetky tabulky sú prepojene medzi sebou. Na vyhľadávanie dát z rôznych tabuliek musí byť použity prikaz JOIN, ktory nasledne vytvori pohľad (VIEW), čo je časovo náročným procesom. Na inej strane, NoSQL databázy, údaje sú uložené vo forme objektov, ktoré budú obsahovať všetky súvisiace údaje. To vylučuje proces kombinovania a potom zobrazovanie údajov, čím sa šetrí čas odozvy.
- **podpora technologie**

  SQL je skúsená technológia a teda o väčšinu problémov bolo postarané. Bezpečnosť funkcie ako autentifikácia, dôvernosť údajov a integrita sú začlenené do SQL. Na druhej strane NoSQL databázy este nemaju taky uroven bezpečnosti a podpory, preto že boli vyvinuté ne až tak davno. V niektorých IoT aplikáciách na prenos citlivých údajov sa vyžaduje zabezpečený komunikačný kanál. Pre takéto aplikácie je lepšie použiť a zabezpečený systém ukladania dát spolu so zabezpečeným komunikačný kanálom.

3. SQL databases

MySQL

MySQL bol navrhnutý pre rýchlosť a spoľahlivosť, na úkor úplného dodržiavania štandardného SQL. Vývojári MySQL neustále pracujú na bližšom dodržiavaní štandardných SQL, stále však zaostávajú za ostatnými implementáciami SQL.

Advantages of MySQL

- *Popularita a jednoduché použitie* - jeden z najpopulárnejších databázových systémov na svete, v tlači a online je veľa dokumentácie o tom, ako nainštalovať a spravovať databázu MySQL, ako aj množstvo nástrojov tretích strán - napríklad phpMyAdmin
- *Security* - MySQL je dodávaný s nainštalovaným skriptom, ktorý ä pomôže zlepšiť bezpečnosť databázy nastavením password security level, definovaním hesla pre root používateľa, odstránením anonymných účtov a odstránením testovacích databáz, ktoré sú prístupné všetkým používateľom.
- *Rýchlosť* - MySQL je zatial jednou z najrýchlejších open-source databáz, hoci aj ine verzie RDBMSs, ako je PosgreSQL, sa približujú z hľadiska rychlosti

Disadvantages of MySQL

- *SQL limitacia* - Pretože MySQL bol navrhnutý skôr na rýchlosť a jednoduché použitie ako na úplnú zhodu s SQL, prichádza s určitými funkčnými obmedzeniami. Napríklad mu chýba podpora klauzúl FULL JOIN.
- *Licencovanie* - MySQL je softvér s dvojitou licenciou, s bezplatnou a open-source edíciou licencovanou pod GPLv2 a niekoľkými platenými komerčnými vydaniami vydanými na základe autorských licencií.

Kedy je vhodné použiť MySQL

- *Distribuované operácie* - Vďaka podpore replikácie MySQL je skvelá voľba pre nastavenia distribuovanej databázy, ako sú architektúry primárne-sekundárne alebo primárne-primárne.
- *Webové stránky a webové aplikácie*
- *Očakávaný budúci rast* - Podpora replikácie MySQL môže pomôcť uľahčiť horizontálne škálovanie

Kedy nie je vhodné použiť MySQL

- *Úplná podpora SQL je nevyhnutná* - Pretože sa MySQL nepokúša implementovať celý štandard SQL, tento nástroj nie je úplne v súlade s SQL
- *Súbežnosť a veľké objemy údajov* - Aj keď MySQL vo všeobecnosti funguje dobre pri operáciách náročných na čítanie, súbežné čítanie a zápis môže byť problematické.

PostgreSQL

PostgreSQL, tiež známy ako Postgres, sa označuje ako „najpokročilejšia open-source relačná databáza na svete“. Bol vytvorený s cieľom byť vysoko rozšíriteľným a vyhovujúcim štandardom. PostgreSQL je objektovo-relačná databáza, čo znamená, že je primárne relačnou databázou,avšak obsahuje funkcie - ako je dedenie tabuliek a preťaženie funkcií - ktoré sú častejšie spojené s objektovými databázami.

PostgreSQL podporuje číselné, reťazcové a dátové typy dátumu a času, ako aj MySQL. Ďalej podporuje dátové typy pre geometrické tvary, sieťové adresy, bitové reťazce, textové vyhľadávanie a položky JSON, ako aj niekoľko jedinečný dátových typov

Advantages of PostgreSQL

- *Súlad s SQL* - PostgreSQL podporuje 160 zo 179 funkcií vyžadovaných pre úplnú zhodu so štandardom SQL: 2011 a navyše obsahuje dlhý zoznam voliteľných funkcií.
- *Open-source a riadený komunitou*
- *Rozšíriteľný*

Disadvantages of PostgreSQL

- *Výkon pamäte* - Pre každé pripojenie nového klienta, PostgreSQL vytvára nový proces. Každému novému procesu je pridelených asi 10 MB pamäte, čo môže spôsobiť značné zaťaženie na pamäťový systém databáz s veľkým počtom pripojení.

Kedy je vhodné použiť PostgreSQL

- *Integrita údajov je dôležitá* - PostgreSQL je plne kompatibilný s ACID od roku 2001 a implementuje multivérznú kontrolu, ktorá zaisťuje konzistentnosť údajov.
- *Integrácia s inými nástrojmi* - PostgreSQL je kompatibilný so širokou škálou programovacích jazykov a platforiem.To znamená, že ak bude potrebné migrovať databázu do iného operačného systému alebo integrovať s konkrétnym nástrojom, s PostgreSQL to bude pravdepodobne jednoduchšie ako s iným DBMS.
- *Komplexné operacie* - Postgres podporuje dotazy, ktoré môžu využívať viac CPU na rýchlejšie odpovedanie na dotazy. Toto, spolu so silnou podporou viacerých súbežných spisovateľov, z neho robí skvelú voľbu pre zložité operácie, ako je skladovanie dát a online spracovanie transakcií.

Kedy nie je vhodné použiť PostgreSQL

- Rýchlosť je nevyhnutná (Na úkor rýchlosti bol PostgreSQL navrhnutý s ohľadom na rozšíriteľnosť a kompatibilitu.)
- Jednoduché nastavenia (Vďaka svojej veľkej množine funkcií a silnému dodržiavaniu štandardného SQL môže byť Postgres pre jednoduché nastavenie databázy prehnaný)

TimeScaleDB

TimescaleDB - prvá open-source relačná databáza pre time-series dáta. TimescaleDB ponúka spoľahlivosť, flexibilitu, jednoduché použitie a škálovateľnosť, ktorú vyžadujú aplikácie, infraštruktúra pre analýzu údajov a zložité systémy.

TimescaleDB je účelovo zostavený na škálovanie a zvládanie time-series dátové zaťaženie a je zámerne navrhnutý ako rozšírenie PostgreSQL.

Advantages of TimescaleDB

- *Much Higher Ingest Rates*

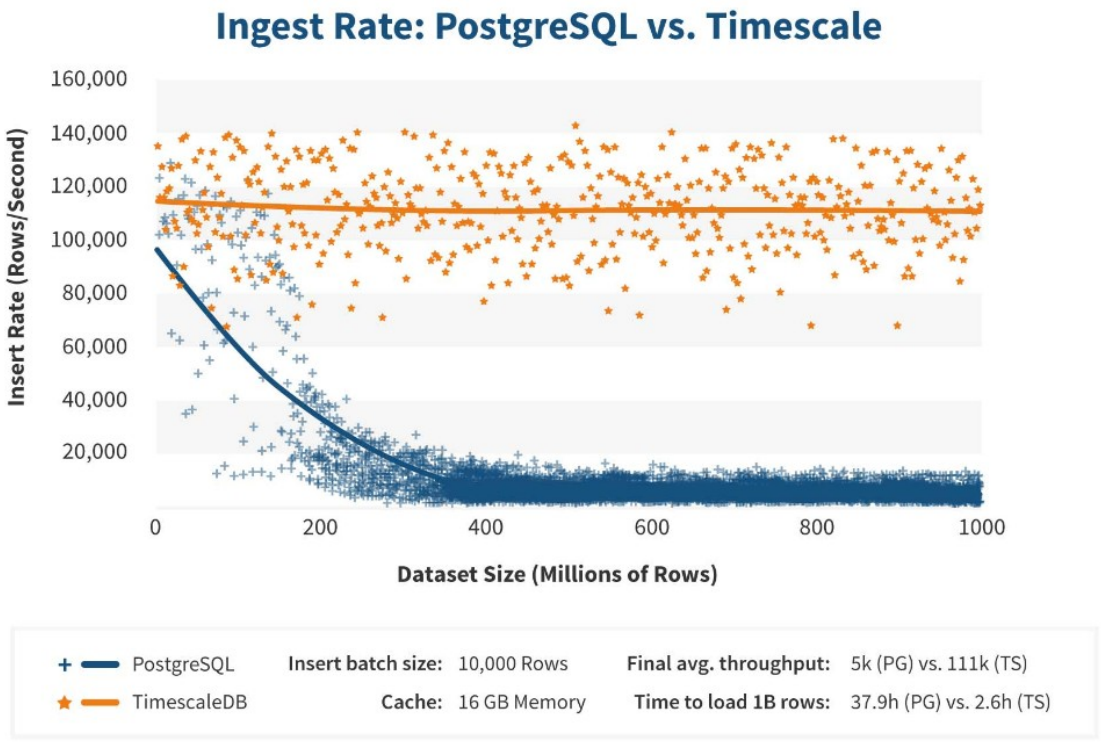
  Aplikácia TimescaleDB dosahuje pre časové rady oveľa vyššiu a stabilnejšiu rýchlosť prijímania ako PostgreSQL. Výkon PostgreSQL začne výrazne klesať, akonáhle sa indexované tabuľky už nezmestia do pamäte.

  Najmä vždy, keď sa vloží nový riadok, musí databáza aktualizovať indexy (napr. Stromy B) pre každý z indexovaných stĺpcov tabuľky, čo bude vyžadovať výmenu jednej alebo viacerých stránok z disku. Vkladanie viac pamate na tento problem iba oddiali tento problem a priepustnost v 10k-100k riadkov/s sa moze

klesnuť na stovky riadkov za sekundu, pokial pri time-series tabulke, rychlost zosatne na millionach riadkov.

TimescaleDB to rieši prostredníctvom veľkého využívania časopriestorového rozdelenia, a to aj vtedy, keď je spustená na jednom počítači. Takže všetky zápisy do posledných časových intervalov sú iba do tabuliek, ktoré zostávajú v pamäti, a vďaka tomu je rýchla aj aktualizácia akýchkoľvek sekundárnych indexov.

Benchmarking ukazuje jasnú výhodu tohto prístupu. Nasledujúca referenčná hodnota pre 1 miliardu riadkov (na jednom stroji) emuluje bežný scenár monitorovania.



Poznamenávame, že PostgreSQL aj TimescaleDB začínajú na približne rovnakej priepustnosti (106 kB, respektíve 114 kB) pre prvých 20 miliónov požiadaviek alebo viac ako 1 milión metrík za sekundu. Avšak okolo 50 miliónov riadkov výkon PostgreSQL začína prudko klesať. Jeho priemer za posledných 100 miliónov riadkov je iba 5 000 riadkov / s, zatiaľ čo TimescaleDB si zachováva priepustnosť 111 000 riadkov / s.

Stručne povedané, TimescaleDB načíta miliardovú databázu riadkov v pätnástine celkového času PostgreSQL a pri týchto väčších veľkostiach vidí priepustnosť viac ako 20-násobnú oproti PostgreSQL.

- *Výkon dopytu v rozmedzí od ekvivalentu k rádovo väčším.*

  Na strojoch s jedným diskom,je medzi PostgeSQL a TimescaleDB veľa jednoduchých dotazov, ako vyhladavanie podla indexu alebo skenovanie tabulky, ktore spravujú s rovnakou výkonnosťou.

  Podobné dotazy, ktoré zahŕňajú základné prehľadanie indexu, sú medzi nimi rovnako výkonné:

```
SELECT * FROM cpu
  WHERE usage_user > 90.0
    AND time >= '2017-01-01' AND time < '2017-01-02';
```

Väčšie dotazy týkajúce sa časových Group BY - celkom bežné v časovo orientovanej analýze - dosahujú vynikajúci výkon v TimescaleDB.

Napríklad nasledujúci dotaz, ktorý sa dotkne 33 miliónov riadkov, je v TimescaleDB 5x rýchlejší, keď je celá (hyper) tabuľka 100 miliónov riadkov, a zhruba dvakrát rýchlejší, keď je to 1B riadokov.

```
SELECT date_trunc('hour', time) as hour,
    hostname, avg(usage_user)
  FROM cpu
  WHERE time >= '2017-01-01' AND time < '2017-01-02'
  GROUP BY hour, hostname
  ORDER BY hour;
```

- *Časovo orientované funkcie*
    - Časovo orientovaná analýza
        - Time bucketing - rozsirena funkcionalita date_trunc funkcie
        - Last and first aggregates - Tieto funkcie vám umožňujú získať hodnotu jedného stĺpca podľa poradia druhého
    - Časovo orientovaná správa údajov - TimescaleDB umožňuje prostredníctvom svojej funkcie drop_chunks efektívne mazanie starých údajov na úrovni blokov, a nie na úrovni riadkov. `SELECT drop_chunks(INTERVAL '7 days', 'conditions');`

4. Porovnanie

IoT Database Requirements	PostgreSQL	MongoDB	MySQL
Simultaneous users support (>1000000)	√	√	√
Clustering, management tools	√	√	√
Asynchronous notifications	√	√	
Triggers and Stored procedures	√		√
Transactions and transaction rollbacks	√		√
JSON data types	√	√	
Aggregation functions	√	√	√
Maximum size of data per table	256TB(MyISAM)	128TB	2048PB
Maximum row size	-	Max document size: 16MB	1.6TB
Maximum number of columns	1000	Max document level: 100	1600
Maximum field size	-	-	1GB
Replication strategies	Master to slave(s)	Master to slave(s) Peep-to-peer	Master to slave(s) Circular Master to Master

- PostgreSQL a MySQL podporuje všetky požadované funkcie systému na ukladanie údajov IoT.

- MySQL chýba podpora asynchrónnych oznámení (notifications) a nemá podporu JSON.
- PostgreSQL notifikácie možno použiť na prenos asynchrónnych udalostí do iných služieb na úrovni databázy (PaaS).
- Databáza MySQL má na druhej strane podporu rôznych typov replikačných stratégií a ich distribuovany databázový “engine” je robustnejší ako PostgreSQL. MySQL ma vacsiu kapacitu uloziska
- MongoDB využíva uchovacie schopnosti OS, avšak ma obmedzenie vzhladom na rozmer dokumentov pridanych do kazdej zdierky

Spolu s tym uvádzam tu linky na dalsie porovnanie DBSs:

PosgreSQL vs MySQL vs MongoDB - MongoDB vs MySQL vs PostgreSQL
Porovnanie vsetkych DBS - Comparison_of_relational_database_management_systems

a pre porovnanie time-series databáz:

TimescaleDB vs InfluxDB - TimescaleDB vs InfluxDB

5. Zhrnutie

V tejto analyze sme sa zaoberali vyberom databázového systému pre IoT projekt SmartSpace. Je zrejmé že existuje veľa možností implementácie DBS pre IoT projekty, ale vzhľadom na to že väčšina time je oboznámená a vie pracovať s relačnými databázami bolo dohodnuté že bude použitý SQL pristup. Ale na tomto momente sa tiez neda jednoznačne rozhodnut o vhodnom nastroje. Najpouzivanejsie open-source relačnými DBS su MySQL a PostgreSQL. Každý ma svoje vyhody a nevyhody, MySQL - je rýchly a ľahko-použiteľný, PostgreSQL - úplna podpora SQL standardu, možnosť pridania rozšírení a vykonavanie kompleksnych operacii.

Druhá vec je uchovanie time-series dát. Teoretický tuto funkcionalitu je možné spraviť v oboch systemoch, avšak pri veľkom počte operácií s udajmi tohto typu výkon databazy moze vyrazne klesnuť. Na uchovanie tychto udajov je možné použiť bud ešte jednu nerelačnú databázu (ako je MongoDB alebo InfluxDB) alebo rozšírenie pre existujucu databazu (TimescaleDB).

Vzhladom na uvedene vyssie body, pre IoT projekt SmartSpace vhodným výberom je PostgreSQL databázový systém z možnosťou dalšieho rozsirenie pomocou TimescaleDB. Pre navrh prototypu projektu, kedže veľa operacii sa nebude vykonávať na DBS, je vhodné použiť čistý PostgreSQL. Ako riešenie pre budúcnosť, ked time-series data búdu spôsobiť veľký vplyv na výkon DBS, bude vhodné pouziť rozšierenie TimescaleDB.

Zdroje

- Kontogiannis, S. & Asiminidis, Christodoulos & Kokkonis, George. (2019). Comparing Relational and NoSQL Databases for carrying IoT data. The Journal of Scientific and Engineering Research.
- Bell, Charles, MySQL for the Internet of Things,a MySQL Whitepaper, 2016
- Sharvari Rautmare, Dr. D. M. Bhalerao, MySQL and NoSQL database comparison for IoT application,2016 https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7887957&tag=1
- Best open source databases for IoT applications
- Open Source Databases that Work Best for IoT
- 4 Best Time Series Databases To Watch in 2019 | by Antoine Solnichkin | devconnected — DevOps, Sysadmins & Engineering
- 4 Steps to Select the Right Database for Your Internet of Things System
- Use a relational database instead of NoSQL for IoT
- Why bother with NoSQL databases? Choose PostgreSQL for IoT.
- SQL vs. Flux: Choosing the right query language for time-series data