

АНАЛІЗ НОВОГО ПОКОЛІННЯ БАЗ ДАНИХ NEWSQL НА ПРИКЛАДІ GOOGLE SPANNER

Із зростанням кількості користувачів з'являються проблеми з базою даних. Кількість запитів зростає, і настає момент, коли погіршується ефективність.

Актуальним вирішенням цих проблем є база даних NewSQL.

NewSQL – це тип мови бази даних, який ґрунтується на концепціях і принципах мов структурованих запитів (SQL) і NoSQL. Об'єднуючи надійність SQL зі швидкістю і продуктивністю NoSQL, NewSQL забезпечує поліпшені функціональні можливості.

Одночасно забезпечується додаткова продуктивність, масштабованість та розподіл. Це робиться за рахунок використання сучасних апаратних засобів та розгортання вдосконалених алгоритмів, які ще не були доступні, коли були розроблені існуючі СУБД.

NewSQL використовує сучасну багатоядерну архітектуру та підтримує кластеризацію, на відміну від попередніх СУБД, у яких відсутні ці функції або вони були додані на більш пізньому етапі життєвого циклу продукту.

NewSQL був розроблений для систем обробки транзакцій в режимі онлайн (OLTP), водночас задовольняючи умовам узгодженості, ізоляції та довговічності (ACID).

Всі NewSQL бази даних мають приблизно однакову архітектуру, яку розглянемо на прикладі Google Spanner:

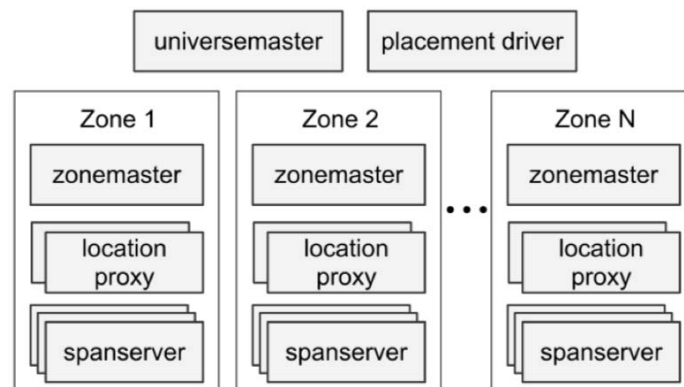
Spanner – горизонтально масштабована послідовна служба реляційних баз даних:

- Повністю керована DBaaS з горизонтальним масштабом та автоматизованим шарінгом даних.

- Традиційна модель реляційної семантики та послідовності: схеми, ACID транзакції, SQL.

- Автоматична синхронна реплікація в межах та між регіонами.
- Стандартний SQL.
- Моніторинг, логування, IAM.
- Клієнтські бібліотеки популярних мов (Java, Python, Go, C#, Node.js тощо).
- JDBC драйвер для підключення до популярних сторонніх інструментів.

Spanner використовує Colossus (GFS нового покоління) як шар зберігання (storage) і алгоритм вирішення колізій Paxos. Дані в Spanner зберігаються в таблицях, які мають схему. Всі дані мають часову мітку (timestamp), що важливо для реалізації підтримки мультиверсійності даних



Кожен розгорнутий екземпляр (*deployment*) Spanner називається *Universe* і складається з:

- *Universe master* – мастер-процес, який координує роботу зон (в термінології Spanner – *Zone*);

- *Множина Zone* – географічно розподілені зонив Spanner. Вони можуть бути як логічні так і фізичні.

Кожна із *Zone*, в свою чергу складається з:

- *ZoneMaster* – мастер процесів зони (синглтон);

- *Множина* – від сотні до декількох тисяч *Spanservers*;

- *Location proxy* – розкривають клієнтам розположення *Spanservers*, відповідних за конкретні дані;

- *Placement driver* – процес, що управляє переміщенням даних між різними *Zone*.

Реалізація транзакцій стала можлива завдяки програмному інтерфейсу TrueTime API. TrueTime API надає глобальний час та деяку невизначеність – TTinterval. Для розподілених систем дуже складно гарантувати миттєвість відгуку вузлів, що важливо для забезпечення узгодженості даних у розподіленому сховищі.

Коли замість точного часу повертається деяких часовий інтервал виконання двох конкурентних транзакцій, то порівнюються TTinterval цих транзакцій. Якщо TTinterval транзакцій не перетинаються, то ми можемо точно знати, яка з транзакцій повинна бути виконана перша. Якщо TTinterval перетинається, то визначити, яка транзакція має виконуватись перша, можна з певним ступенем ймовірності.

NewSQL взяла кращі ідеї двох світових - реляційні СУБД і NoSQL. Підтримка розподілених транзакцій між дата-центрами на велику кількість даних, що йдуть на петабайти, з такою здатністю до масштабування - безумовно вражаюча властивість для будь-якої системи зберігання структурованих та напівструктурованих даних. Дана можливість багато в чому стала наслідком симбіозу двох підходів: підхід до зберігання даних - дані, які містять часовий коментар - і інноваційна концепція отримання глобального часу - TrueTime.

Але гострою залишається проблема міграції даних зі старих баз даних до нової. Це є причиною, чому лише нові клієнти вибирають нову базу даних на відміну від старої.

Її застосування було б актуальним для банків, мобільних операторів, але відсутність вдалих рішень для міграції даних є ключевим блокером в цьому напрямі.

Існує 2 рішення цієї проблеми: найняти команду спеціалістів, котрі будуть писати нову конкретну програму для міграції даних; оновити програмне забезпечення, котре б підтримувало класичну СУБД та NewSQL. Але обидва рішення є досить затратними, тому більшість організацій обрали паузу в очікуванні вдалих міграційних рішень для NewSQL.

Список використаних джерел

1. https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2017-05-1/NET-2017-051_06.pdf
2. <http://static.googleusercontent.com/media/research.google.com/en/us/archive/spanner-osdi2012.pdf>
3. <http://db.cs.cmu.edu/papers/2016/pavlo-newsql-sigmodrec2016.pdf>

Науковий керівник: професор Шарапов О.Д.