УДК 004.042

ОГЛЯД КОНЦЕПЦІЇ ПОТОКОВОЇ ОБРОБКИ ДАНИХ

Ковбаса А.О., Лисенко О.І.
Інститут телекомунікаційних систем КПІ ім. Ігоря Сікорського, Україна
E-mail: antonkovbasa@gmail.com, lysenko.a.i.1952@gmail.com

Overview of the Streaming Data Processing concept

The idea of structuring data as a stream of events is nothing new, and it is used in many different fields. Even though the underlying principles are often similar, the terminology is frequently inconsistent across different fields, which can be quite confusing. This article tries to give understanding of the topic and make comparation with other concepts of data processing.

Існує велика плутанина щодо того, що означає потокова обробка даних. Багато визначень поєднують деталі імплементації, вимоги до продуктивності, моделі даних та багато інших аспектів інженерії програмного забезпечення. Світ потокової обробки все ще розвивається, і те, що конкретна популярна реалізація робить специфічні деталі або має певні обмеження, не означає, що ці деталі є невід'ємною частиною обробки потоків даних.

Почнемо спочатку: що таке потік даних (також його називають потоком подій)? Перш за все, потік даних - це абстракція, що представляє необмежений набір даних. Необмежний означає нескінченний і постійно зростаючий. Набір даних необмежений, оскільки з часом нові записи постійно надходять. Це визначення використовується Google, Amazon та майже всіма іншими компаніями. Зауважимо, що ця проста модель (потік подій) може бути використана для представлення майже будь-якої діяльності, яку ми хочемо проаналізувати. Ми можемо переглянути потік транзакцій між кредитними картками, фондові торги, доставку пакетів, мережеві події, що проходять через комутатор, події, про які повідомляють датчики на виробничому обладнанні, електронні листи, що надсилаються, переміщення в грі, тощо. Перелік прикладів нескінченний, тому що досить багато процесів можна розглядати як послідовність подій.

Окрім необмеженості, існує кілька інших атрибутів моделі потокових подій:

***Потоки подій упорядковані.** Існує невід'ємне поняття, які події відбуваються до або після інших подій. Це стає більш зрозумілим якщо дивитися на фінансові події. Послідовність, коли я спочатку кладу гроші на свій рахунок і пізніше витрачаю гроші, сильно відрізняється від послідовності, в якій я спочатку витрачаю гроші, а пізніше покриваю борг, повертаючи гроші банку. Останній буде стягувати плату за овердрафт, а перший - ні. Зауважимо, що це одна з відмінностей між потоком подій та таблицею баз даних: записи в

таблиці завжди вважаються не упорядкованими, а інструкція «*order by*» в SQL не є частиною реляційної моделі.

***Незмінні записи даних.** Події, що відбулися, ніколи не можуть бути змінені. Фінансова операція, яка скасовується, не видаляється. Натомість у потік записується додаткова подія, яка записує скасування попередньої транзакції. Коли клієнт повертає товар у магазин, ми не видаляємо той факт, що товар був проданий йому раніше, а ми фіксуємо повернення як додаткову подію. Це ще одна різниця між потоком даних та таблицею баз даних - ми можемо видалити або оновити записи в таблиці, але це все додаткові транзакції, що відбуваються в базі даних, і як такі можуть бути записані в потоці подій, що реєструє всі транзакції.

***Потоки подій можна відтворити.** Це бажана властивість. Хоча легко уявити потоки, які не відтворюються (пакети TCP, що передаються через сокет, як правило, не підлягають повторенню), для більшості бізнес-програм важливо мати можливість відтворювати необроблений потік подій, що відбувалися місяцями (а іноді й роками) раніше. Це необхідно для того, щоб виправити помилки, спробувати нові методи аналізу, або проведення перевірок.

Варто зазначити, що ні визначення потоку подій, яке ми дали спочатку, ні атрибути, які ми згодом перераховували, нічого не говорять про дані, що містяться в подіях, або про кількість подій в секунду. Дані відрізняються для кожного окремого випадку - події можуть бути крихітними (часом лише кілька байтів) або дуже великими (повідомлення XML з багатьма заголовками); вони також можуть бути повністю неструктурованими, парами ключ-значення, напівструктурованими JSON або структурованими повідомленнями Avro або Protobuf. Хоча часто вважається, що потоки даних є "великими даними" і включають мільйони подій в секунду, ті ж методи, застосовуються однаково добре (і часто краще) для менших потоків подій, що мають лише кілька подій в секунду або хвилину.

Тепер, коли ми знаємо, що таке потоки подій, саме час переконатися, що ми розуміємо потокову обробку. Потокова обробка подій - це парадигма програмування - подібно до запиту-відповіді та пакетної обробки. Давайте розглянемо, як порівнюються різні парадигми програмування, щоб краще зрозуміти, як обробка потоків вписується в архітектуру програмного забезпечення:

***Запит-відповідь.** Це парадигма з найменшою затримкою, з часом відгуку в діапазоні від субмілісекунд до декількох мілісекунд, зазвичай з очікуванням того, що час відгуку буде в високим ступенем узгодження. Режим обробки зазвичай блокується - додаток відправляє запит і чекає відповіді від системи обробки. У світі баз даних ця

парадигма називається обробкою онлайн-транзакцій (OLTP). Системи продажів, обробки кредитних карт і системи відстеження часу зазвичай працюють в цій парадигмі.

Пакетна обробка. Це варіант високої затримки/високої пропускної здатності. Система обробки прокидається у встановлений час - щодня о 2:00, щогодини і т. д. Вона зчитує всі необхідні дані (або всі дані, доступні з моменту останнього виконання, всі дані з початку місяця тощо), записує весь необхідний вихід і чекає до наступного разу, коли її буде запущено. Часи обробки варіюються від хвилин до години, і користувачі розраховують прочитати застарілі дані під час перегляду результатів. У світі баз даних - це сховища даних та системи бізнес-аналізу - дані завантажуються величезними партіями раз на день, створюються звіти, і користувачі переглядають ті самі звіти до наступного завантаження даних. Ця парадигма часто має велику ефективність, але останніми роками підприємства потребують наявних даних у більш короткі терміни, щоб зробити прийняття рішень більш своєчасним та ефективним.

Потокова обробка. Це неблокуючий варіант. Заповнення розриву між світом запитів-відповідей, в якому ми чекаємо подій, для обробки яких потрібно дві мілісекунди, і світом пакетної обробки, в якому дані обробляються один раз в день і для їх завершення потрібно вісім годин. Більшість бізнес-процесів не вимагають негайної відповіді протягом мілісекунд, але також не можуть чекати наступного дня. Більшість бізнес-процесів відбуваються безперервно, і до тих пір, поки бізнес-звіти постійно оновлюються, а лінійка бізнес-додатків може безперервно відповідати, обробка може тривати, і ніхто не буде чекати конкретної відповіді протягом мілісекунд. Бізнес-процеси, такі як оповіщення про підозрілі кредитних операціях або мережеву активність, коригування цін в режимі реального часу на основі попиту і пропозиції або відстеження поставок пакетів - все це природно підходить для безперервної, але неблокуючої обробки.

Література

1. Martin Kleppmann, “Making Sense of Stream Processing” Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol.
2. Neha Narkhede, Gwen Shapira, and Todd Palino, “Kafka: The Definitive Guide” Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol.
3. Клеппман М. Высоконагруженные приложения. Программирование, масштабирование, поддержка. — СПб.: Питер, 2018. — 640 с.: ил. — (Серия «Бестселлеры O’Reilly»).