УДК 004.41

Сєріков А.І

SOLVING PROBLEMS WITH DATA UPDATES IN HIGH-LOAD SYSTEMS

Vinnytsia National Technical University

Анотація

У цій статі досліджуються можливі проблеми з оновленням даних в системах високого навантаження та способи їх вирішення.

Ключові слова: Обробка великих об'ємів даних, системи високого навантаження, проблема оновлення даних.

Abstract

This paper explores potential issues with data updating in high-load systems and ways to solve them.

Keywords: Intensive applications, high-load system, problems with data updates

Technical issues

Some IT systems have to deal with many write requests, so let us look at a payment transaction in public transport. Each rider pays the fare with a payment card to a transport company. So, there are two entities for each transfer side: payment transaction and balance. There can be a potential problem for transport company balance with a race condition – you have to lock (a row-level locking a relational database) record or concurrent-safe transaction (like a serializable isolation level in PostgreSQL), otherwise some changes can be lost, and the balance is wrong. [1][2] Explicit lock and safe transaction are heavy solutions (reduces performance) because in most conflict situations, these solutions interrupt one conflict transaction, and the system has to retry the transaction. There is one alternative solution – an optimistic lock when the system explicitly validates a version of the entity before finishing the transaction. This solution preserves performance, but in case of failure, it requires a user to repeat payment, which reduces rider loyalty.

Solution

The basic idea of one solution comes from SST (Sorted string table) structure. [3] To avoid lock, the system can perform a write operation in the background. It allows us to perform update operations in queues with batch processing. Generally, a background job reads transactions for N minutes or hours, calculates the total change by these transactions, and updates the balance with the change. This strategy reduces the number of updates, avoids locks, and increases the response time for a payment transaction request.

Therefore, the balance update will be an asynchronous operation, which means that the complete payment transaction does not refer to the actual balance. This problem is called inconsistency. In terms of the transport company balance, it is an acceptable disadvantage. The number of small write operations is far more than the number of read operations, and the actual balance can only be more than a stored balance. Considering the background job interval, the difference between the real balance and the stored balance will not be large.

Conclusion

If the primary number of operations for an attribute are "write" operations and minor data inconsistency on "read" is acceptable, then a solution with the asynchronous update by merging with the previous result and batch processing in a background job is a pretty good approach to deal with performance and reliability in a high load system.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Lock [Electronic resource] –Access mode: https://www.postgresql.org/docs/15/sql-lock.html
2. Transaction Isolation [Electronic resource]. – Access mode: https://www.postgresql.org/docs/current/transaction-iso.html
3. Martin Kleppmann. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems – Sebastopol, O`Reilly Media Inc. 2017. p.76-79.

Сєріков Антон Ігорович – студент групи 2ПІ-22М, Вінницький національний університет, м.Вінниця
Sierikov Anton Igorovych – student of group 2PE-22m, Vinnytsia National Technical University, Vinnytsia.