Handling Hotspot Accounts

Big accounts, such as Nike, Procter & Gamble & Nintendo, often cause hotspot issues for the payment system.

A hotspot payment account is an account that has a large number of concurrent operations on it.

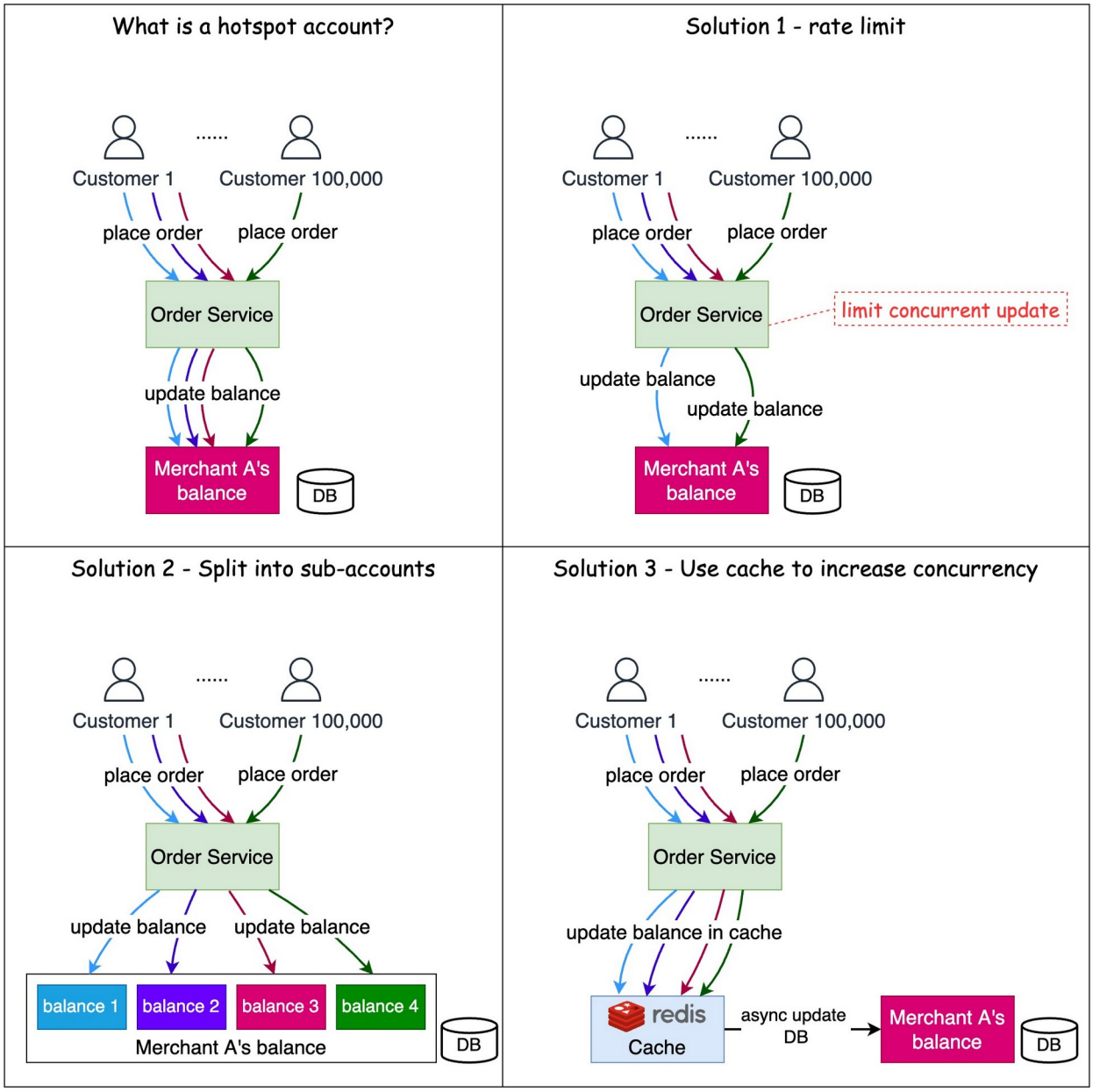
For example, when merchant A starts a promotion on Amazon Prime day, it receives many concurrent purchasing orders. In this case, the merchant’s account in the database becomes a hotspot account due to frequent updates.

In normal operations, we put a row lock on the merchant’s balance when it gets updated. However, this locking mechanism leads to low throughput and becomes a system bottleneck.

The diagram below shows several optimizations.

Handling Hotspot Accounts

blog.bytebytego.com



- Rate limit
  We can limit the number of requests within a certain period. The remaining requests will be rejected or retried at a later time. It is a simple way to increase the system's responsiveness for some users, but this can lead to a bad user experience.
- Split the balance account into sub-accounts
  We can set up sub-accounts for the merchant's account. In this way, one update request only locks one sub-account, and the rest sub-accounts are still available.
- Use cache to update balance first
  We can set up a caching layer to update the merchant's balance. The

detailed statements and balances are updated in the database later asynchronously. The in-memory cache can deal with a much higher throughput than the database.

Quick question: We can also put the requests into a message queue so the requests can be processed at the service’s own pace. Can you think of the limitations of this approach?