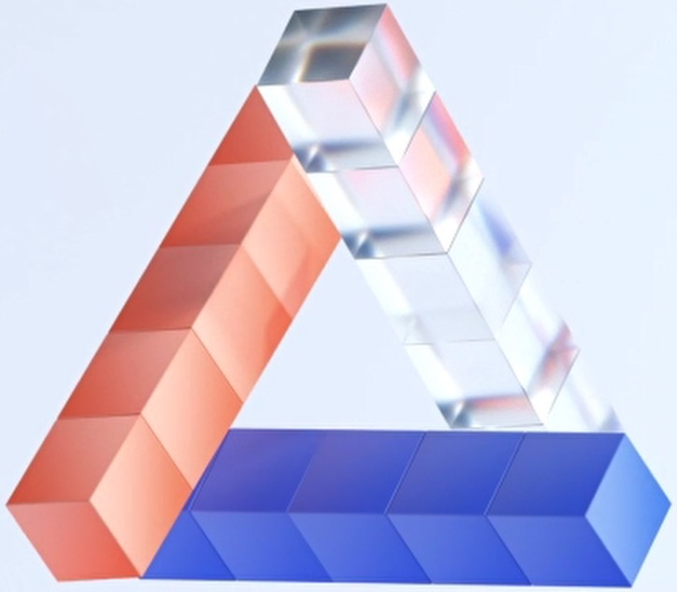
DEVOCEAN

Make Use of Pub/Sub Functionality to Synchronize Local Cache

Hanseo Jo

DEVOCEAN



- Kafka KRU Introduction
- About Cache
- Usefulness of Local Cache
- Problem of Local Cache @Distributed System
- Local Cache Synchronization

Kafka KRU Introduction

- KAFKA KRU FACEBOOK GROUP

https://www.facebook.com/groups/kafka.kru

About Cache

Cache (computing)

Article Talk

From Wikipedia, the free encyclopedia

In computing, a **cache** (/kæʃ/ (listen) *KASH*)[1] is a hardware or software component that stores data so that future requests for that data can be served faster; the data stored in a cache might be the result of an earlier computation or a copy of data stored elsewhere. A *cache hit* occurs when the requested data can be found in a cache, while a *cache miss* occurs when it cannot. Cache hits are served by reading data from the cache, which is faster than recomputing a result or reading from a slower data store; thus, the more requests that can be served from the cache, the faster the system performs.[2]

About Cache

Local Cache VS Global Cache

About Cache

Local Cache

- Speed 👍
- Consistency 👎

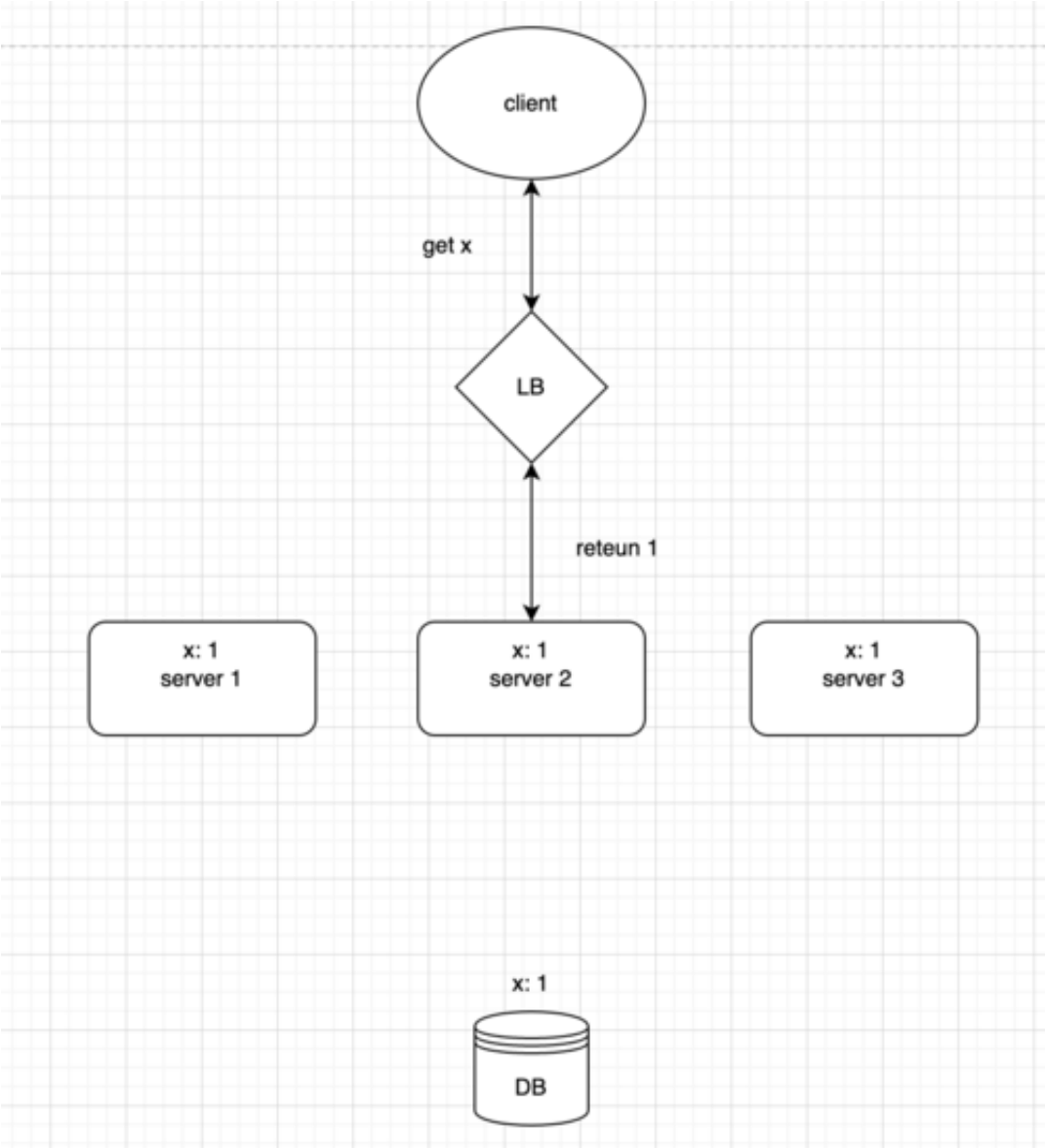
Global Cache

- Speed 👎
- Consistency 👍

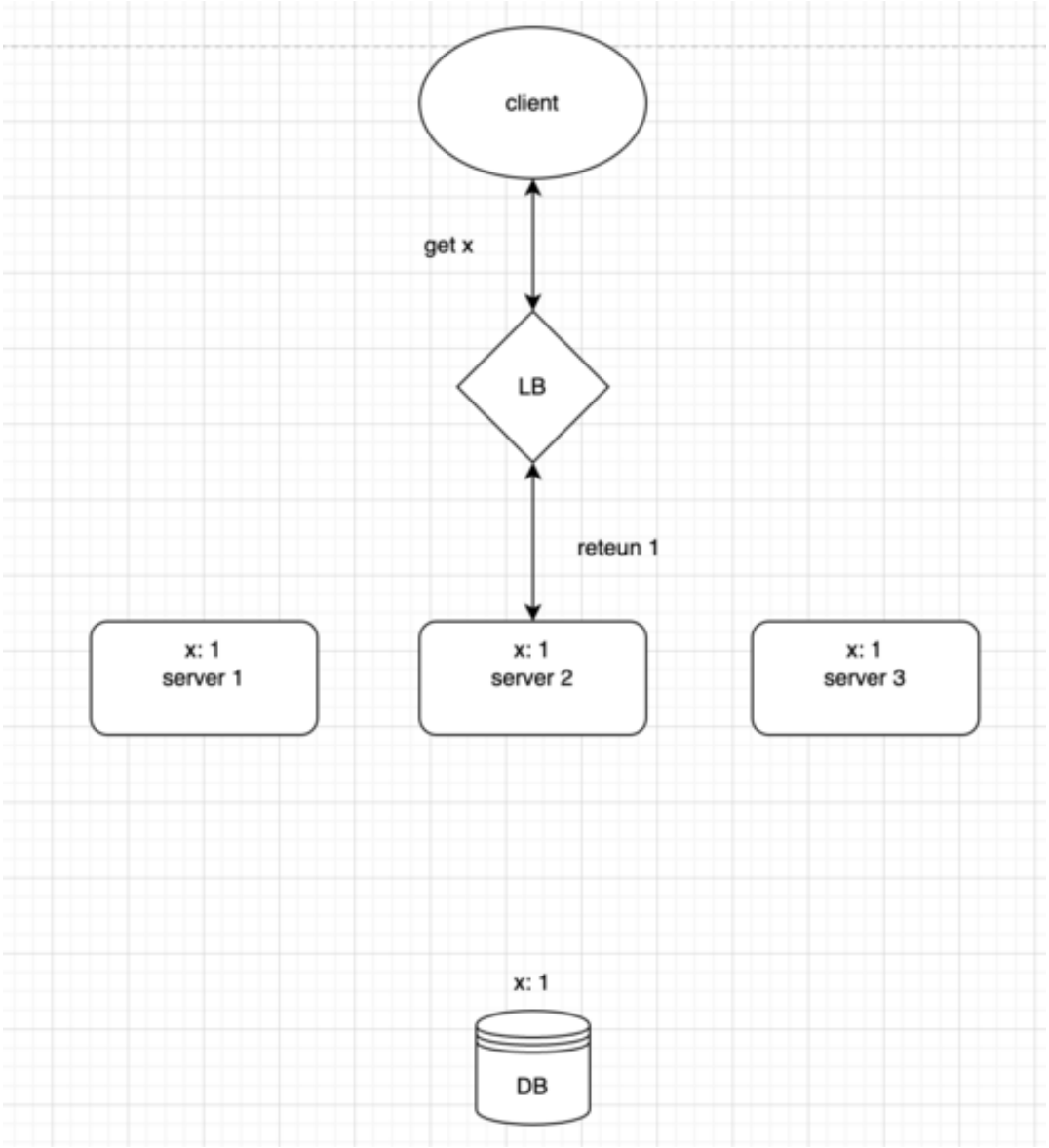
Fit condition for local cache

heavy read request than write operation

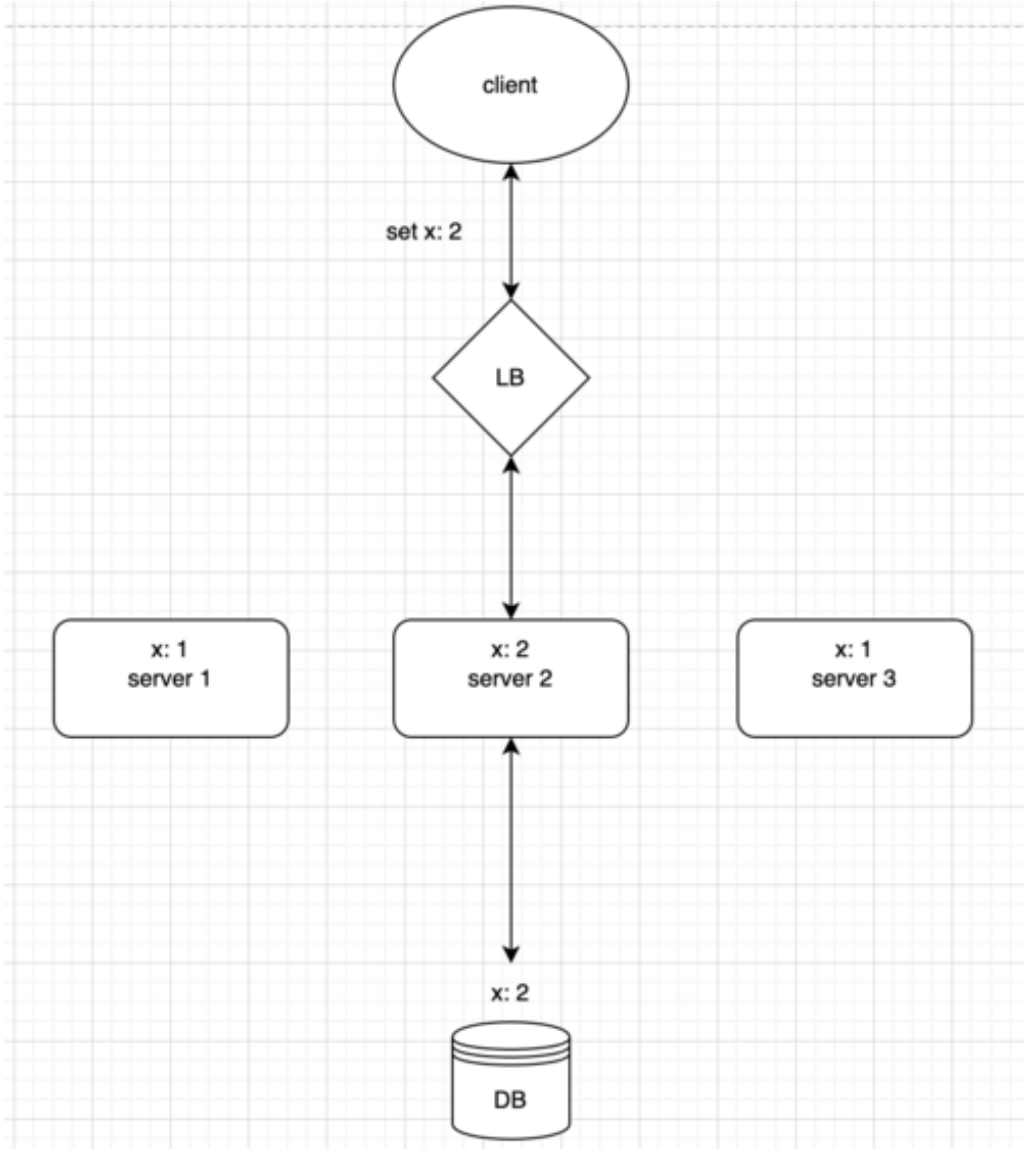
Usefulness of Local Cache



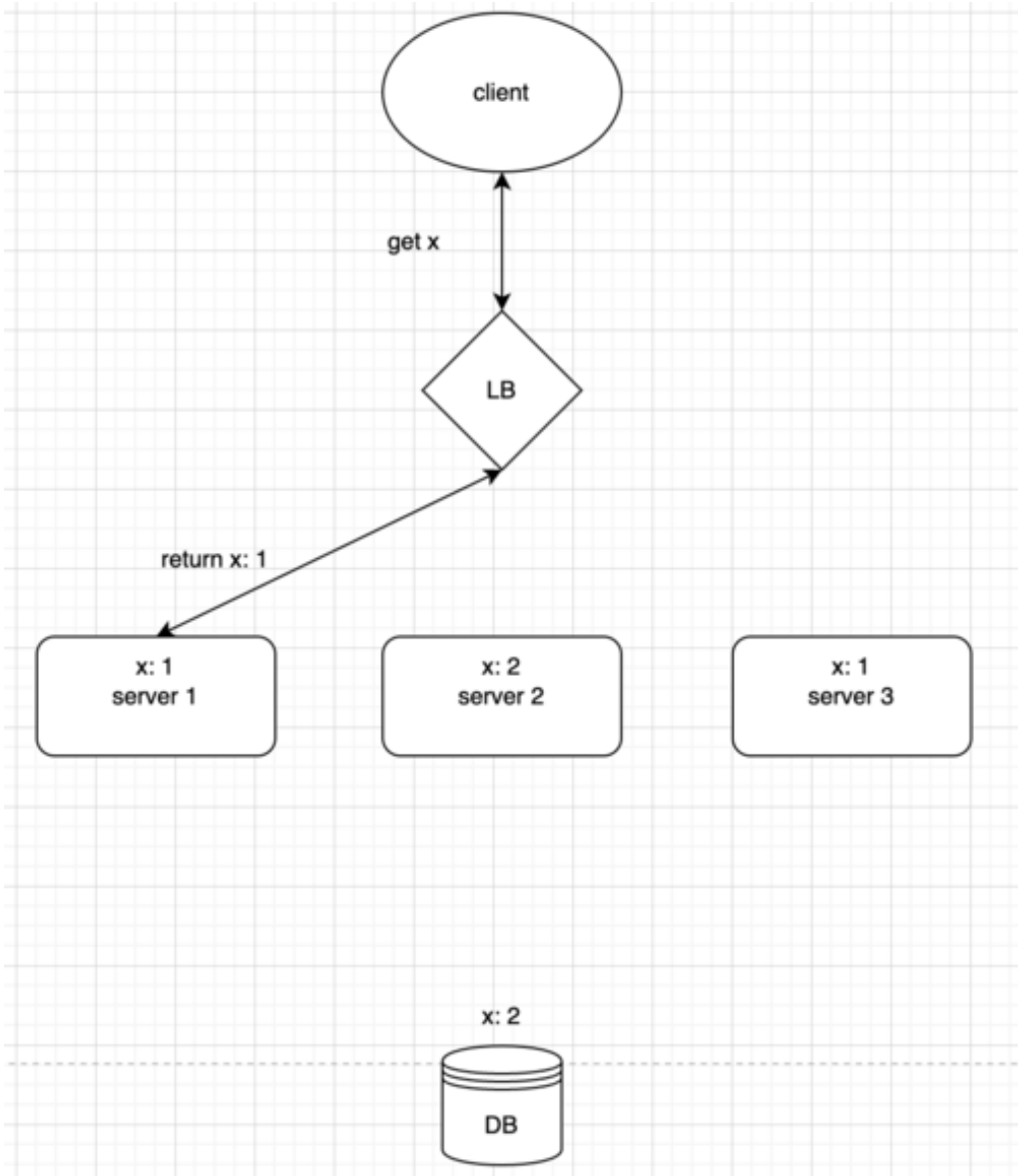
Usefulness of Local Cache



Problem of Local Cache @Distributed System



Problem of Local Cache @Distributed System



Local Cache Synchronization

Publish–subscribe pattern

Article Talk

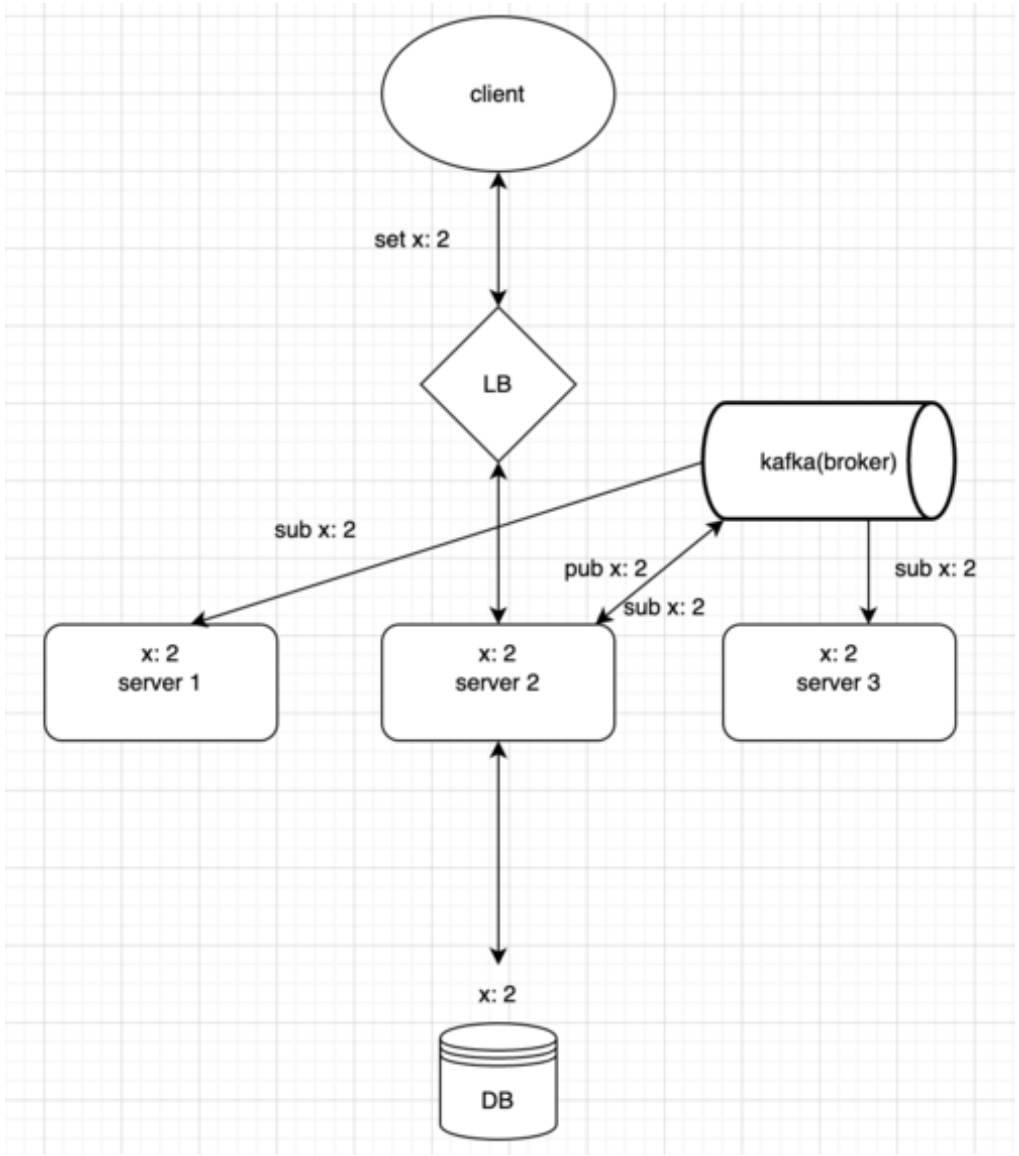
From Wikipedia, the free encyclopedia

In software architecture, **publish–subscribe** is a messaging pattern where senders of messages, called publishers, do not program the messages to be sent directly to specific receivers, called subscribers, but instead categorize published messages into classes without knowledge of which subscribers, if any, there may be. Similarly, subscribers express interest in one or more classes and only receive messages that are of interest, without knowledge of which publishers, if any, there are.

Publish–subscribe is a sibling of the message queue paradigm, and is typically one part of a larger message-oriented middleware system. Most messaging systems support both the pub/sub and message queue models in their API; e.g., Java Message Service (JMS).

This pattern provides greater network scalability and a more dynamic network topology, with a resulting decreased flexibility to modify the publisher and the structure of the published data.

Local Cache Synchronization



Local Cache Synchronization

It does not guarantee perfect consistency.

There is a delay between instances.

Depending on the situation and your circumstance,
it could be right solution or not.

Thank You!

Hanseo Jo
Email: starkensin@gmail.com