Contribution ID: **69** Type: **Sectional reports**

Concept of peer-to-peer caching database for transaction history storage as an alternative to blockchain in digital economy

Tuesday 6 July 2021 13:30 (15 minutes)

The development of the digital economy implies storing the history of a large number of transactions of every citizen involved in business processes based on digital technologies, starting from receiving public and social services in electronic form and ending with consumption of electronic goods and services produced by e-business and e-commerce.

If we look carefully at the data structure within a digital economy system, we can see that transactions are grouped with respect to a natural unique identifier of a citizen, which allows for efficient block distribution based on their hash across the node-segments of a scalable peer-to-peer NoSQL DBMS, eliminating the appearance of "hotspots", and the data itself can be easily represented in the form of "key-value" tuples and stored in a columnar structure, providing a quick search due to the gossip protocol that allows redirecting requests to the node whose responsibility range includes the hash of a specific unique identifier. Well, since we are talking about storing transaction history within the business processes of the digital economy, the key-value relationship is essentially a one-to-many relationship, in the context of a design focused on personalized information output for each specific user. Ensuring communication between users within groups and communities (many-to-many relationship) is also possible and can be implemented by means of secondary indexes, materialized representations or partial data redundancy through denormalization, depending on the power of the set that the data form, to ensure acceptable performance of sampling queries.

If we talk about the task of quickly obtaining summary or aggregated statistical information, it is not difficult to solve it by loading the necessary data in the YARN cluster of the open technology platform Apache Hadoop, for example, in the processing environment of Random Access Memory SPARK, applying the principle of Resilient Distributed Datasets and basic concepts of building a pipeline of operations mapping, moving, sorting and convolution in the framework of functional programming.

However, the relative simplicity of horizontal scaling of disk space, processing power and RAM, does not provide transactional scaling, as simultaneous access of a large number of users to the central database nodes, would make the bandwidth of the data network a bottleneck. Therefore, we need a peer-to-peer caching database that will store all relevant data for a particular user on their device and the closest peer-to-peer servers, based on selected proximity criteria according to a given set of features and attributes.

If we rise to an empirical level, from the perspective of participants in the digital economy, it is a question of storing a set of facts. Facts in a database are immutable; once stored, they do not change. However, old facts may be replaced by new facts over time or due to circumstances. The state of the database is the value determined by the set of facts in effect at a given point in time. So, this analysis allows us to move on to a more detailed consideration of the architecture of the proposed peer-to-peer caching database design solution.

A peer-to-peer client library (a peer-to-peer access library) is embedded into the client application and allows to get data from the peer-to-peer servers, cache data on the client device (to reduce the load on the peer-to-peer servers), while keeping such an important property as "final immutability", and also to exchange the peer-to-peer server lists between the clients.

The peer-to-peer server provides data access by caching the necessary segments of the central database demanded by the connecting clients. Connection to a specific group (farm) of peer-to-peer servers is determined

by specified criteria, which can be geolocation data, type of users, type of processes, type of transactions, etc. Peer-to-peer servers can exchange data segments with each other (peer-to-peer communications), and store as many data segments as the storage system quotas and limitations allow. In certain cases, a client application may also act as a peer-to-peer server, but there are threats of loss of data integrity and validity through the emergence of fake peer-to-peer servers on the network, created by hackers to discredit it.

Records in the central database (if developers wish, in parallel to peer-to-peer servers) can be made by means of transactors, which accept write transactions and process them serially, ensuring guaranteed integrity until successful synchronization with the central database, due to the replication factor of the distributed network file system (odd number of servers greater than 3 (three) is recommended, to ensure a recording quorum), where open technology solutions based on Apache Hadoop HDFS or Apache Cassandra can be selected as the basis. However, HDFS fault tolerance will require the use of additional components such as Zookeeper, Zookeeper Failover Controller and Quorum Journal Manager.

Access to the transactor is recommended as part of a service-oriented architecture, through REST-services that can be scaled by applying standard load-balancing technologies used in web server deployments. This approach allows providing access to the transactor through the usual HTTP protocol, and transactors themselves and the centralized database will be in an isolated network, access to which should be done via routing with the use of modern encryption technologies, and hacker attacks via HTTP protocol can be prevented by modern IPS systems, combining signature and heuristic approaches of malicious activity detection.

According to the principles of organising access to the transactor, access to the central data repository can be easily organised as well. The proposed approach makes it possible to implement staggered isolation of the central database and cascading of network traffic through the use of peer-to-peer server farms and service-oriented architecture.

In conclusion, it would be useful to note that the proposed concept of a distributed horizontally scalable and cascadable peer-to-peer caching database could become the basis for a modern, efficient, as well as easy-to-implement and maintain technological platform for the implementation of digital economy services in the Russian Federation.

Summary

Primary authors: BELOV, Mikhail (Dubna State Univeristy); GRISHKO, Stanislav (University "Dubna"); CHEREMISINA, Evgenia (Dubna International University of Nature, Society and Man. Institute of system analysis and management); TOKAREVA, Nadezhda (Dubna Univeristy)

Presenter: BELOV, Mikhail (Dubna State Univeristy)

Session Classification: Distributed computing, HPC and ML for solving applied tasks

Track Classification: 10. Distributed computing, HPC and ML for solving applied tasks