Analysis of Consistency for In Memory Data Grid Apache Ignite

Ivannikov Memorial Workshop

Tapekhin Andrey
a.tapekhin@ispras.ru

Velikanov Oleg
oleg.velikanov@gmail.com

Bogomolov Igor
bogomolov.ispras.ru

13.09.2019

This work is funded by the Minobrnauki Russia (grant id RFMEFI60417X0199, grant number 14.604.21.0199)

Motivation

In some fields (e.g. banking) there are need for fast and consistent systems. Traditional solution - mainframes. Alternative solutions - consistent distributed systems.

Cap Theorem

- Consistency
- Availability
- Partition Tolerance

Apache Ignite

In Memory Data Grid

Can work as a cache to traditional relational DBMS

Can be configured as either CP or AP

Can work as either key-value or SQL-like

Apache Ignite transactions

Atomic - no transactional locks, guarantees data atomicity and consistency for each single operation.

Transactional - ACID compliant transactions. Highest consistency level. Supports only key-value transactions.

Transactional Snapshot - ACID compliant transactions. Can have write skew anomaly. Supports both key-value and SQL transactions.

Jepsen

Testing framework

Allows to check transaction history against consistency models

Customizable workload through customizable workers

Workers can send requests to data storage or cause network failures

Consistency models

Serializable

Parallel execution of a set of transactions is equivalent to the serial execution of single transactions

Linearizable

The writes are instantaneous and all the reads after a write return the value of that write or a later one

Strict serializable

Combines both serializability and linearizability

Experiment design

Key-value transactions

Two Apache Ignite client types: client nodes and thin client

Client nodes configuration:

- serializable isolation level
- pessimistic locks

On thin client isolation level and locks type could not be configured

Experiment design

Configuration variables:

- number of nodes N between 2 and 5
- number of replicas between 1 and N
- number of workers K - multiple of N

One test run:

- random set of operations
- one operation - single read or write of a single value to a single key
- key and value are integers in range from 0 to 4.

Results

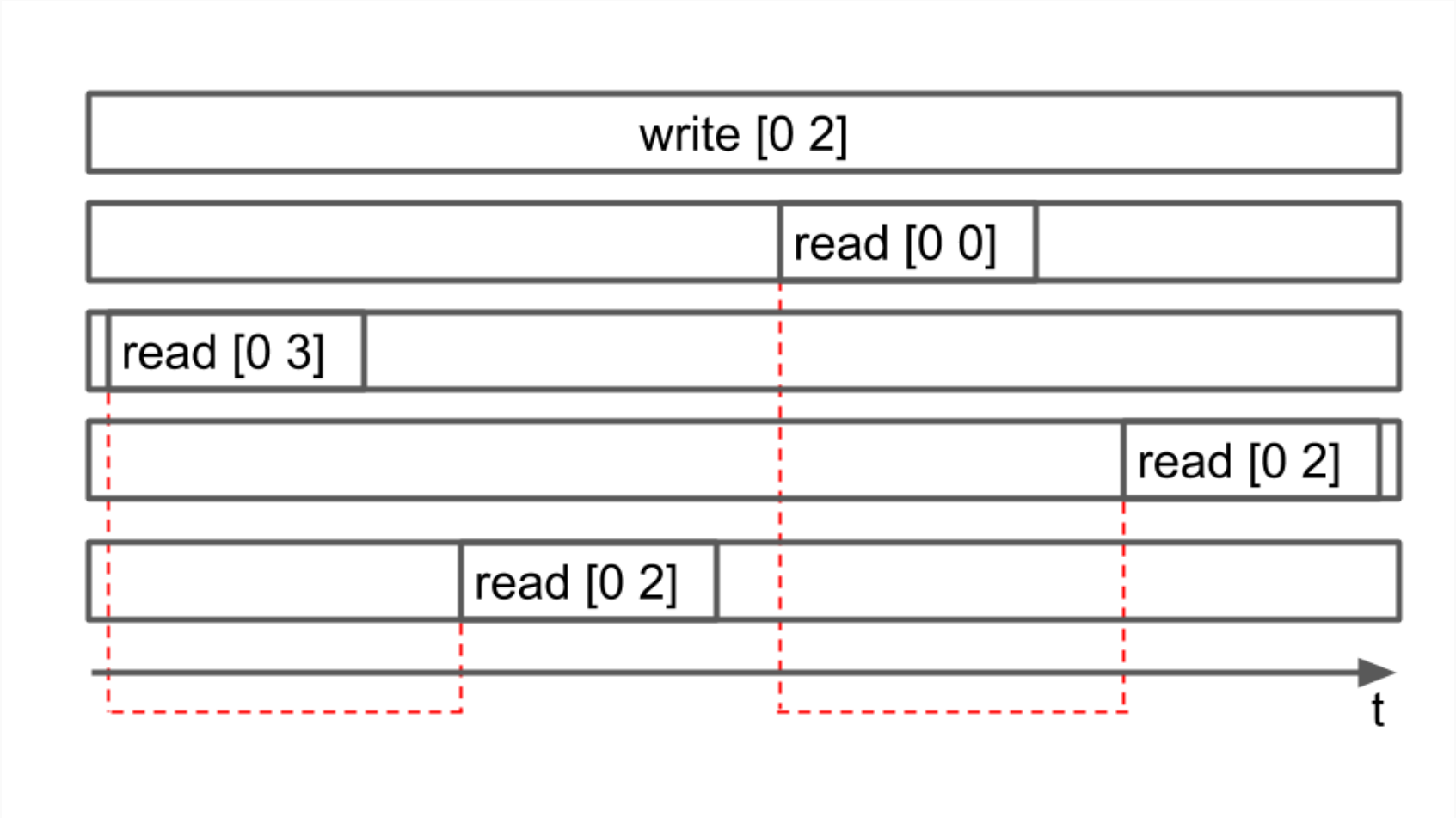
Using client nodes:

- no violations were found

Using thin client:

- with one worker for one node no violations were found
- with 5 nodes, 10 workers and 3 or more replicas **violations were found**

Results. Example of a violation



Example of consistency violation

Conclusions

Cannot achieve linearizable consistency with thin client, most likely due to absence of isolation level control.

Couldn't find consistency violations using client nodes.

As a future work Apache Ignite could be tested against strict serializable model, Jepsen workers that create network problems also could be included.