Evaluation of TPC-H like workload for Apache Ignite, VoltDB and PostgreSQL

Badalyan David ISP RAS,
Borisenko Oleg ISP RAS

Plan

- Motivation
- What is Apache Ignite and VoltDB?
- Goal
- TPC-H
- Testing
- Results

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

Motivation

Problems:

- Need to handle large amounts of data
- Traditionally used RDBMS supports only vertical scaling

As a solution - distributed in-memory DBMS, that’s give us:

- Horizontal scaling
- Systems performance increase because of storage in RAM
- The cost of RAM reduced

Apache Ignite

Distributed in-memory DBMS, data processing and caching platform designed for managing large amounts of data using large number of compute nodes.

Despite original key-value nature of the system, the developers declare:

- ACID compliance
- Full support for the SQL:1999 standard

VoltDB

Distributed relational in-memory DBMS representing the NewSQL DBMS class. VoltDB research prototype is H-Store, designed by Michael Stonebraker.

VoltDB also declared to be ACID compliant

Goal

Research capabilities of Apache Ignite and VoltDB from the point of view of relational operations and compare them to PostgreSQL using our implementation of TPC-H like workload

Investigated problems:

- DBMS performance comparison
- correctness of the benchmark query results in the distributed operation mode
- performance impact of increasing the number of nodes in a cluster

TPC-H benchmark

One of the most usable benchmarks for testing DBMS performance. It consists of a suite of business oriented ad-hoc queries. The queries and the data populating the database have been chosen to have broad industry-wide relevance.

Test configurations

- Cluster configurations is 1, 2, 4 and 8 computing nodes

  (Each node in the cluster is a virtual machine with 92GB of RAM, 3 processor cores at 2.1 GHz (Intel Xeon Gold 6152) running under Xen hypervisor and each virtual machine has dedicated physical server-class SSDs)

- Dataset volume is 1, 10, 50 and 100 Gb
- In configurations with multiple cluster nodes data will be partitioned without replication

Test implementation

TPC-H benchmark contains:

- 8 tables, 22 “select” query types and 2 refresh functions
- data and query generators

Our implementation of tpc-h workload:

- We use the same datasets. The reason is that temporal results with running the same queries against different versions of data getting from data generator is close
- We use default queries of each query type
- We don’t use refresh functions. Evaluation of performance changes after updating data isn’t our goal

Test implementation

Testing details:

- Scripts in (Python 3) which:
  - Create database cluster with database containing TPC-H data
  - Run queries against the cluster
- The cluster was re-initialized at every start
- The queries were run sequentially 3 times each. As a result we take average execution time, in seconds

Errors and restrictions discovered, Apache Ignite

- 3 requests contain operations not supported by Apache Ignite
- The interval data type, as well as SQL operations CREATE TYPE and CREATE VIEW are not supported in Apache Ignite

With multiple node cluster configuration:

- 2 requests end with an errors
  - `java.sql.SQLException: General error: "java.lang.ArrayIndexOutOfBoundsException" [50000-195]`
  - `java.sql.SQLException: javax.cache.CacheException: Failed distributed join query: join condition does not use index [joinedCache = SQL_PUBLIC_PART, plan = ... (query code)`
- 8 requests end with incorrect results

Errors and restrictions discovered, VoltDB

- Two requests contain operations not supported by VoltDB
- The char data type as well as the SQL operations CREATE TYPE and CREATE VIEW are not supported by VoltDB

With multiple node cluster configuration:

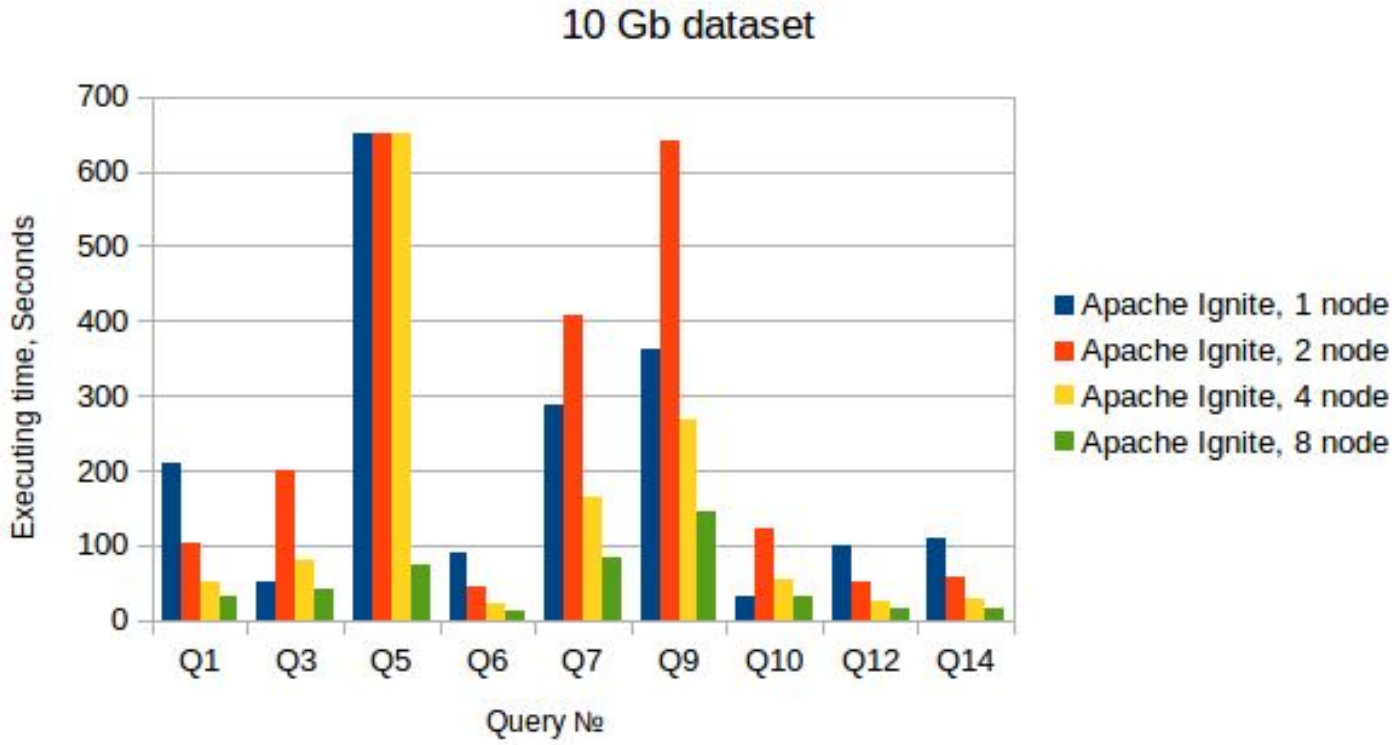
- 12 requests end with an error from VoltDB scheduler containing the following warning:

  “Subquery expressions are only supported for single partition procedures and AdHoc queries referencing only replicated tables”

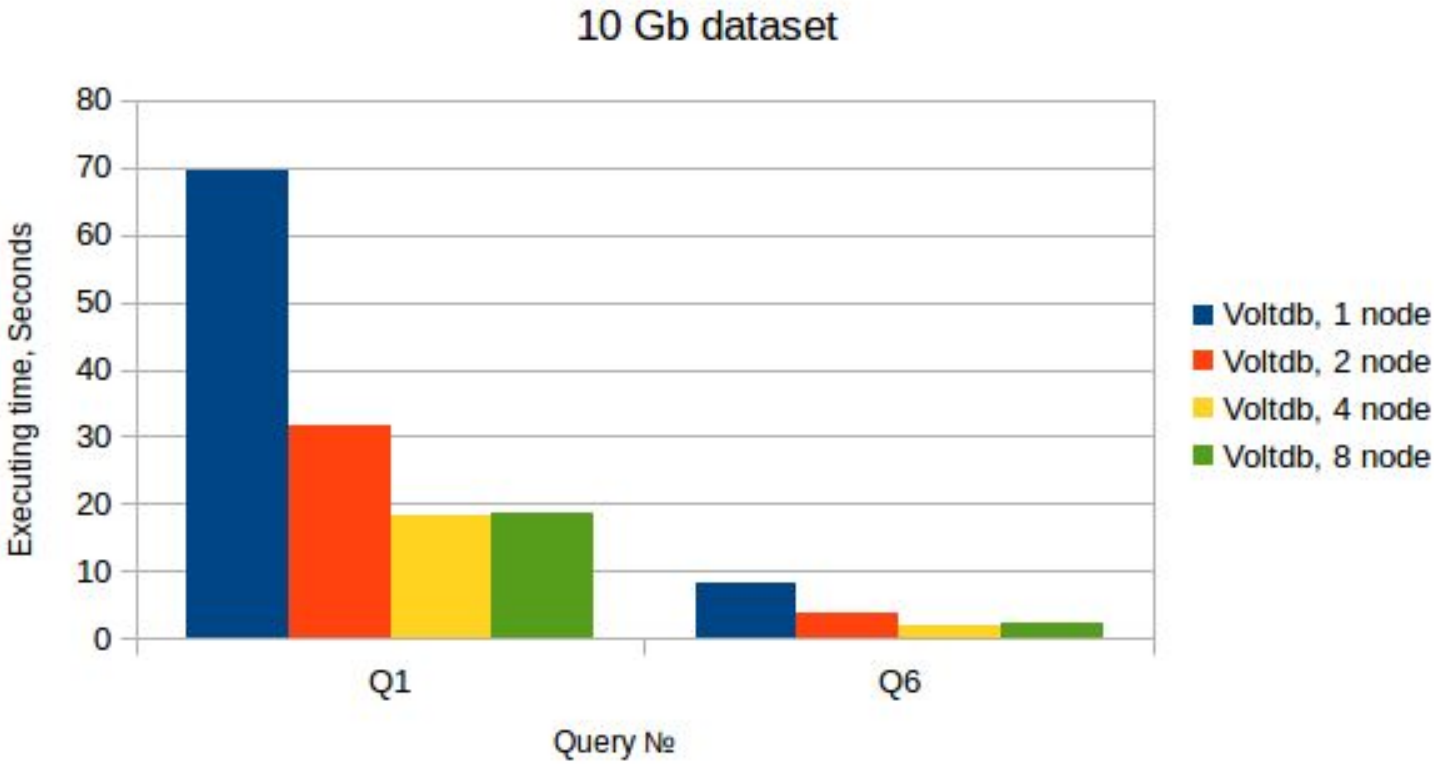
- 6 requests end with error from VoltDB scheduler containing the following warning:

  “This query is not plannable. The planner cannot guarantee that all rows would be in a single partition”

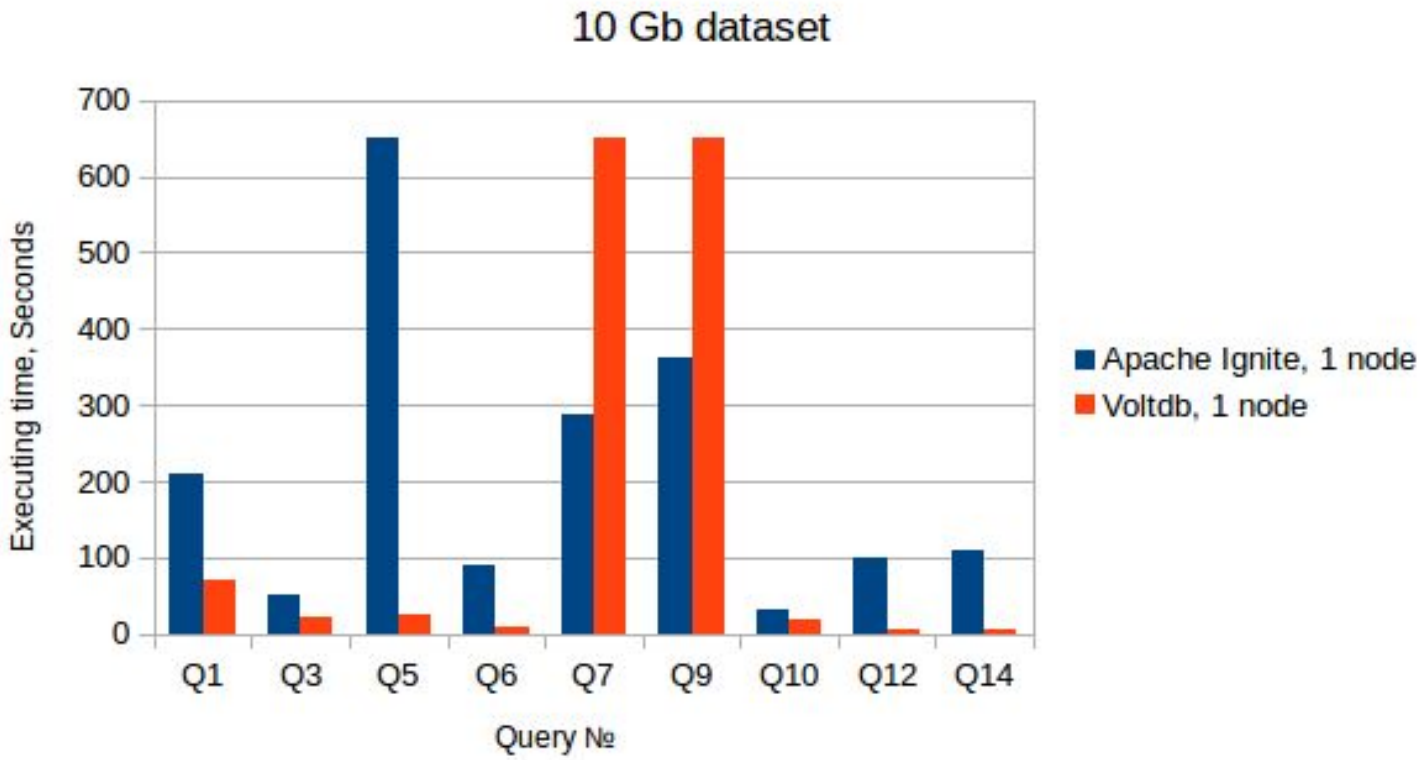
Patterns in temporal results, Apache Ignite



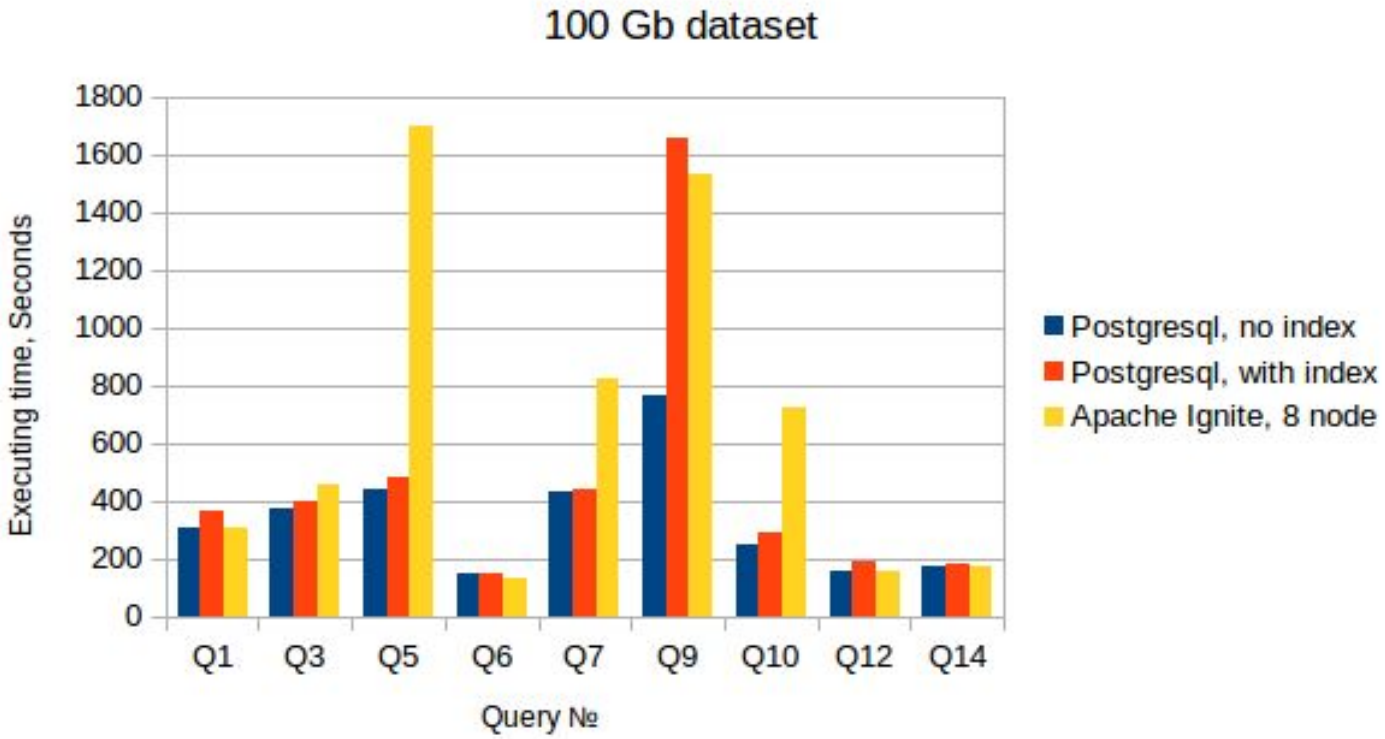
Patterns in temporal results, VoltDB



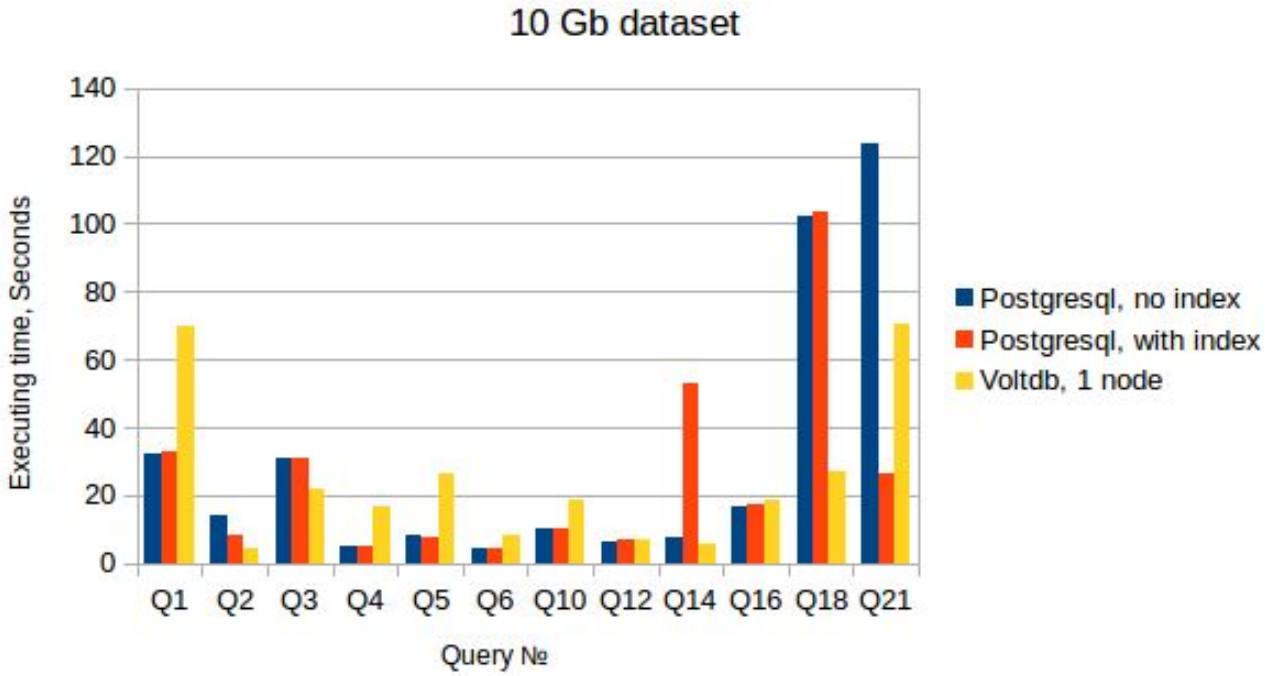
Comparing Apache Ignite and VoltDB



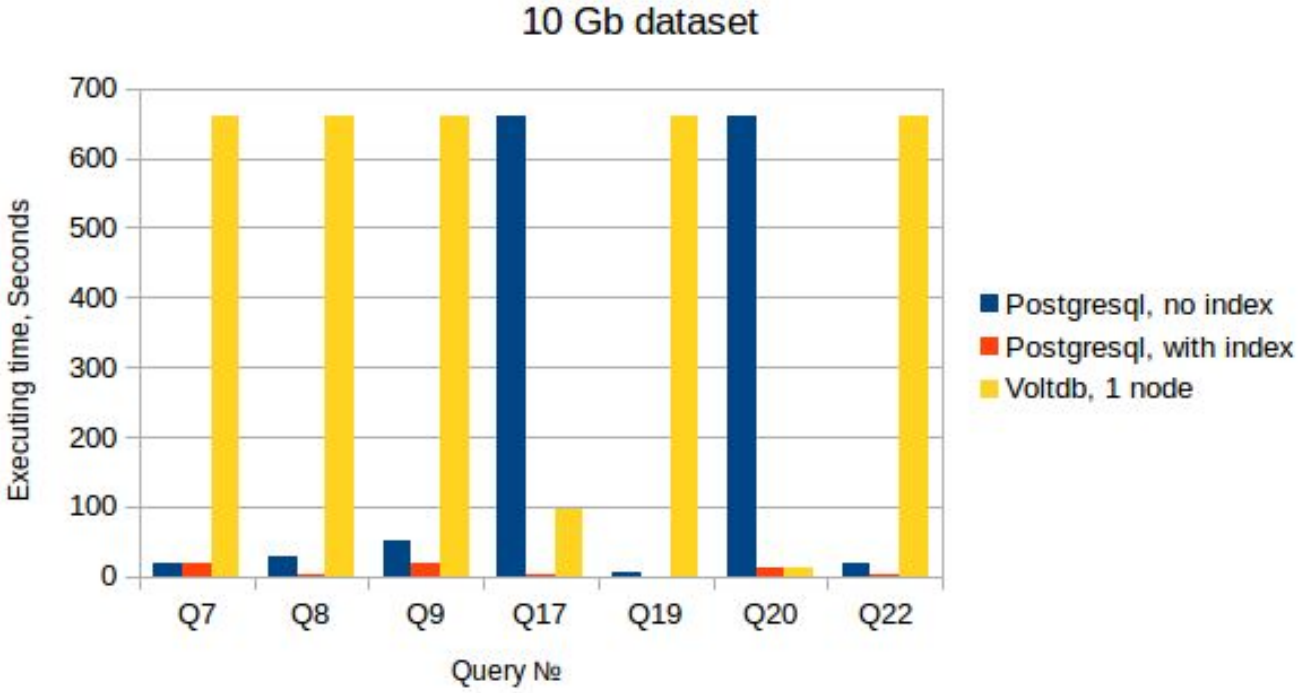
Comparison of Apache Ignite and PostgreSQL



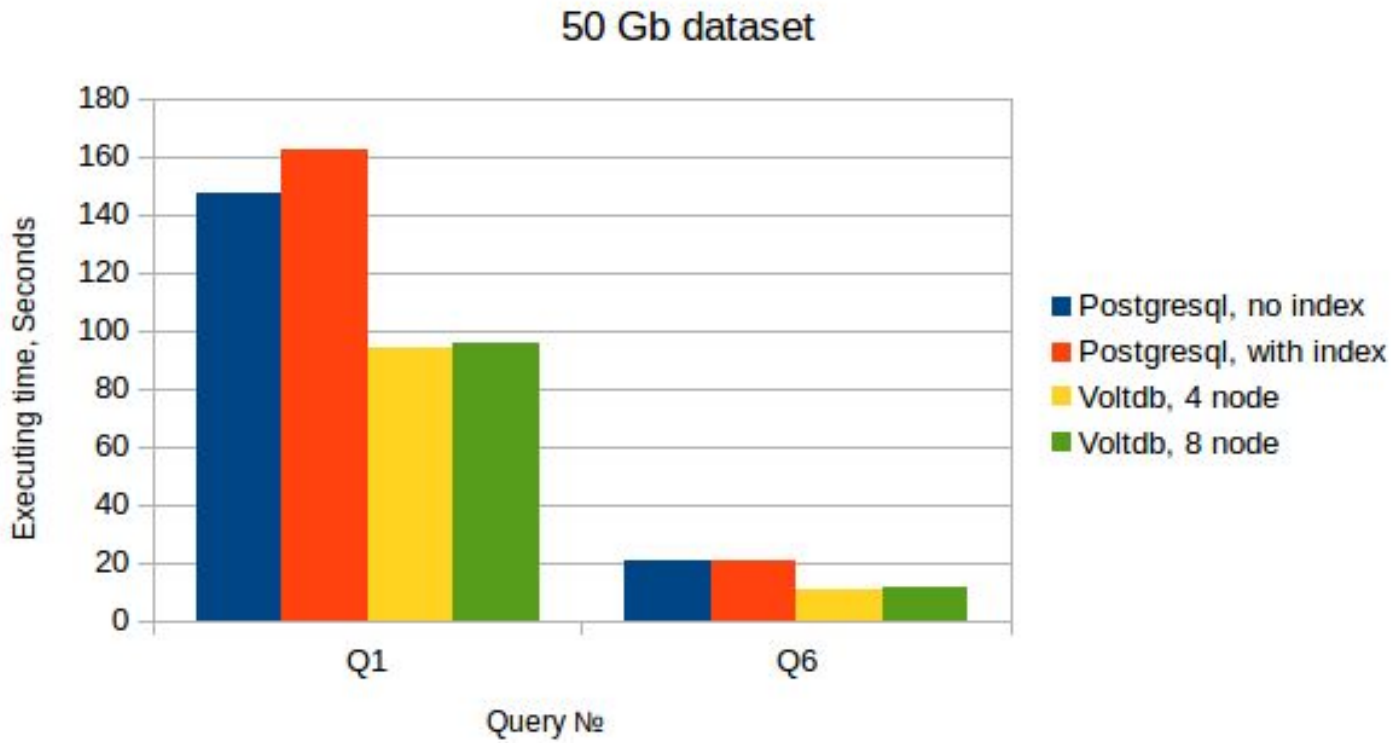
Comparison of VoltDB and PostgreSQL



Comparison of VoltDB and PostgreSQL



Comparison of VoltDB and PostgreSQL



Conclusion

Single-node configuration:

- VoltDB is ahead of Apache Ignite in almost all queries
- Apache Ignite performance less then Postgresql
- VoltDB performance is the same or more then Postgresql in about half of the requests
- No errors

Multiple-node configuration:

- Apache Ignite performance increases with the number of nodes in the cluster
- VoltDB performance stops growing after a certain number of nodes in the cluster, despite further scalability
- VoltDB succeeds to complete only 2 queries
- Apache Ignite has 8 unsupported queries with incorrect results
- Comparing to PostgreSQL, Apache Ignite performance become equal to Postgresql in some queries only at 8-node cluster and large amount of data
- VoltDB is ahead of Postgresql in 2 successfully executed queries