

GridDB

Vít Škrhák

Overview

1. Introduction
2. Structure of GridDB - nodes and clusters
3. Structure of GridDB - Data model
4. Query Language - TQL
5. TQL - Examples
6. Comparison: GridDB vs. Cassandra
7. Pros & Cons

Introduction

- distributed key-value DBMS
- Large capacity by scaling out
- Data replication
- Optimization for IoT data
- ‘Memory first, Storage second’ structure
 - frequently accessed data in memory, the rest is passed on to disks

Structure of GridDB - nodes and clusters

- node vs. cluster
 - node
 - Server process performing data management
 - Only one node can operate in one machine
 - cluster
 - Main component of GridDB
 - Composed of multiple nodes

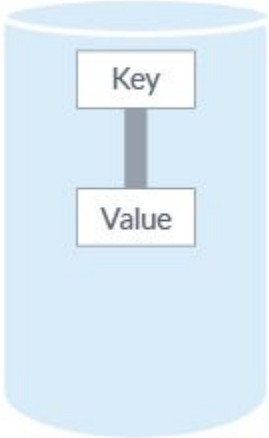
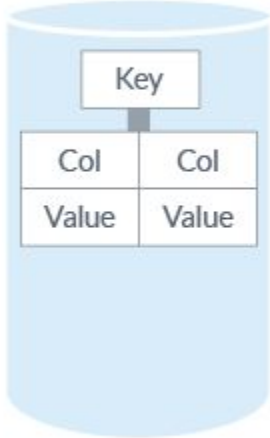
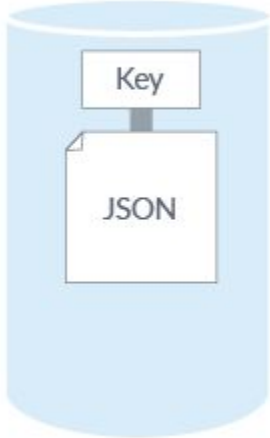
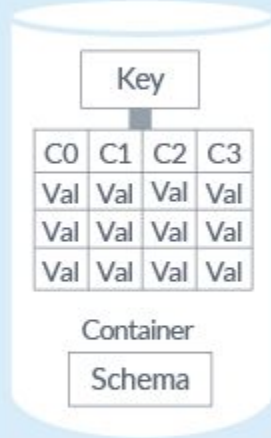
Structure of GridDB - Data model

- Primary model: Key-Container DBMS
- Secondary model: Key-Value store, relational DBMS

- Container - can be of two types:
 - Collection - for managing general data
 - Timeseries container - for managing time series data

- Row
 - a row of data
 - each row is registered to a container
 - fixed schema

Structure of GridDB - Comparison

	Key-Value	Key-Column	Key-Document	Key-Container
Data Model				
Example DB	Riak	Cassandra	MongoDB	GridDB

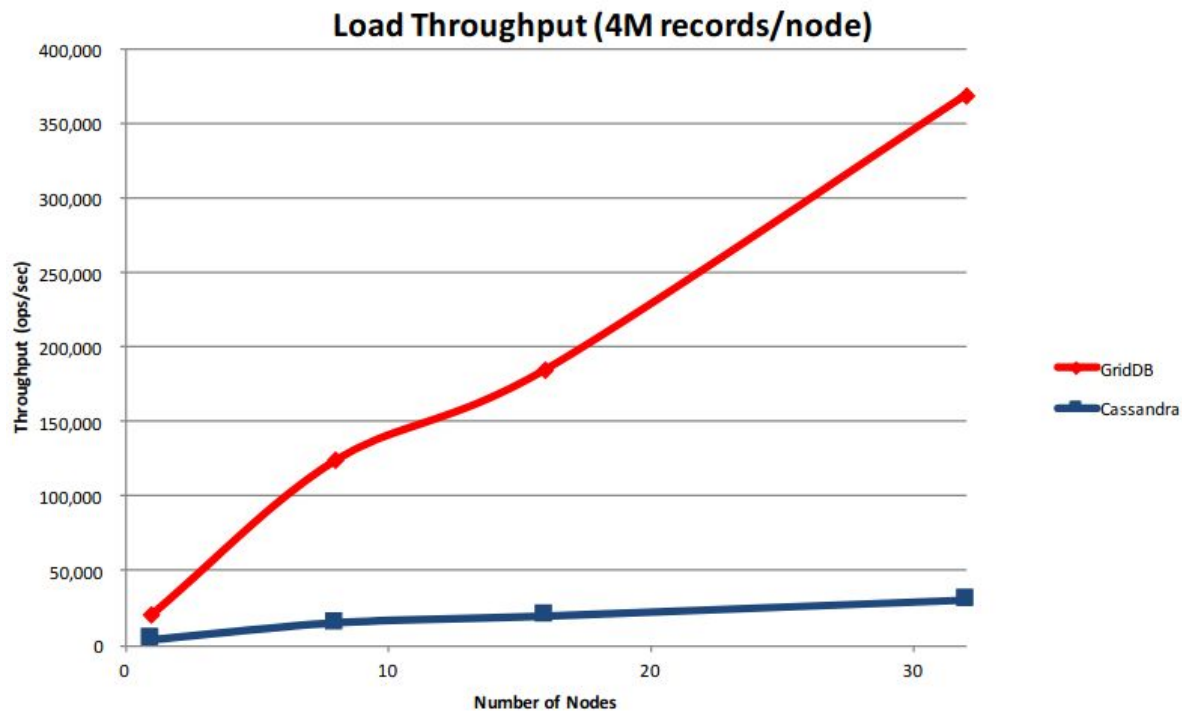
Query Language - TQL

- similar to SQL
- Search rows of a particular container
- SQL-like Syntax: Select, Where, Order By, Limit, Offset
- Support for Time-Type operations: Now(), Timestamp(...), TimestampDiff(...), ToEpochMS(...), etc.
- Aggregation operations: Max, Min, Count, Sum, Avg, Time_Avg, ...
- For Enterprise version, GridDB offers also SQL92

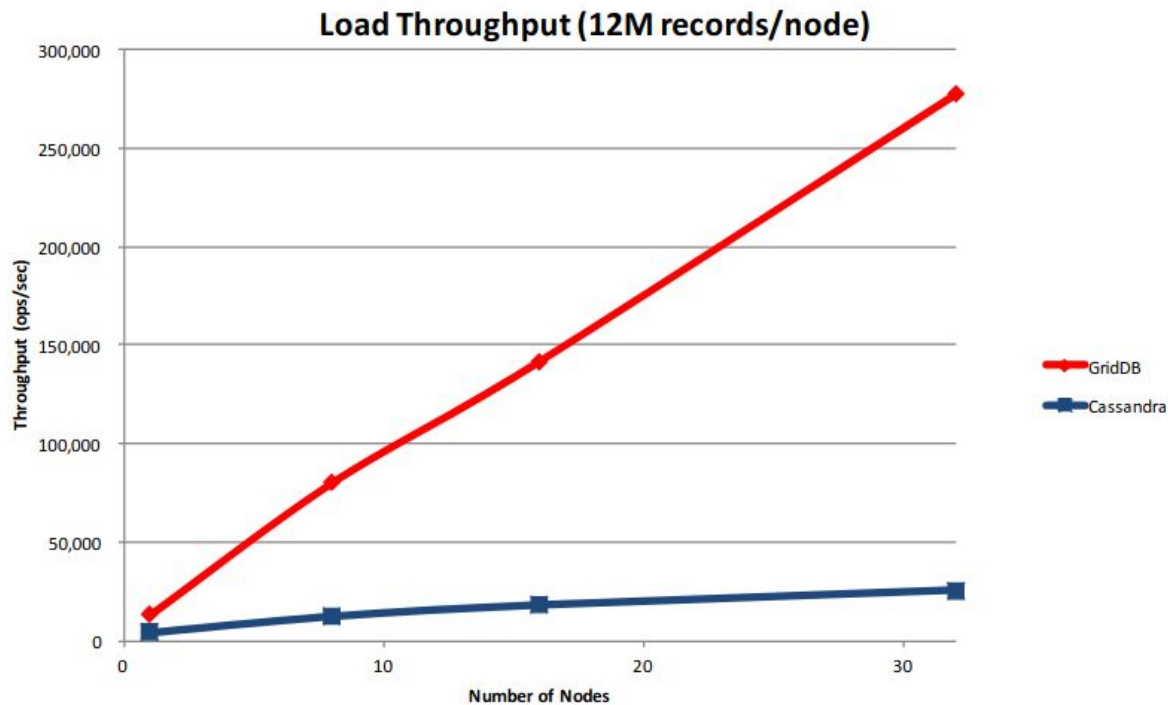
TQL - Examples

1. `SELECT * ORDER BY col1 DESC, col2 ASC`
2. `SELECT * LIMIT 100`
3. `SELECT * WHERE timestampCol > TIMESTAMP('2022-05-02T12:00:00Z')`
`AND (col1 = "Hello" or col2 = "World")`
4. `SELECT * WHERE ELEMENT(1, arrayCol) = "HelloWorld"`
5. `SELECT * WHERE ARRAY_LENGTH(arrayCol) >= 1`
6. `SELECT COUNT(*) WHERE NOT completed`

Comparison: GridDB vs. Cassandra (small dataset)



Comparison: GridDB vs. Cassandra (big dataset)



Pros & Cons

Pros:

- Distributed
- Horizontal Scalability
- Optimized for IoT and Big Data
- High reliability (data replication, data transaction functions)
- fixed schema
- open-source

Cons:

- Difficult deployment (in case you don't have CentOS)
- fixed schema
- Linux-only
- Not that many connectors to other programming languages
- Small community