

Apache Cassandra / ScyllaDB

NHR Monitoring Workshop 2022 | 18.07.22

Michael Ott

- Apache Cassandra and ScyllaDB are API-compatible NoSQL Distributed Databases
 - Clustering is available in the open-source release
 - Horizontal scalability, redundancy
 - Data is distributed across multiple physical nodes
 - Each physical node hosts multiple virtual nodes
 - Any node can be used for queries and inserts
 - Gossip between nodes (e.g. for remote queries, replication)

- Data is organized in partitions
- **(Composite) Partition Keys** determines data locality (also called **row keys**)
- Each virtual nodes owns a set of **tokens** (hashed partition keys)
- Partitions are ordered by **Clustering Keys**
- Partition Key and Clustering Key form the **Primary Key**
- Queries can only be filtered by Primary Key

- For time-series data:
 - Rows must not become too long → use a composite partition key with a time-based component (e.g. year, month, week)
 - Timestamp as clustering key
 - Order ascending or descending depending on use case

Example from DCDB



```
CREATE TABLE dcdb.sensordata (  
    sid text,  
    ws smallint,  
    ts bigint,  
    value bigint,  
    PRIMARY KEY ((sid, ws), ts)  
) WITH CLUSTERING ORDER BY (ts DESC)
```

Cassandra Cons and ScyllaDB Pros



- Cassandra is implemented in Java
 - Most of Cassandra's configuration parameters are really JVM parameters:
 - Stack/heap sizes
 - Threads
 - Garbage Collection
 - Garbage collection regularly causes performance penalties
 - Fine-tuning the garbage collector is tedious
- ScyllaDB is implemented in C++
 - Significantly less configuration parameters
 - Scripts for fine-tuning kernel parameters for disk I/O, IRQ-handling
 - Better performance (>3x compared with Cassandra)
 - Out-of-the-box monitoring