instaclustr

Apache Cassandra Best Practices

Ben Slater, Chief Product Officer, Instaclustr

March 2017

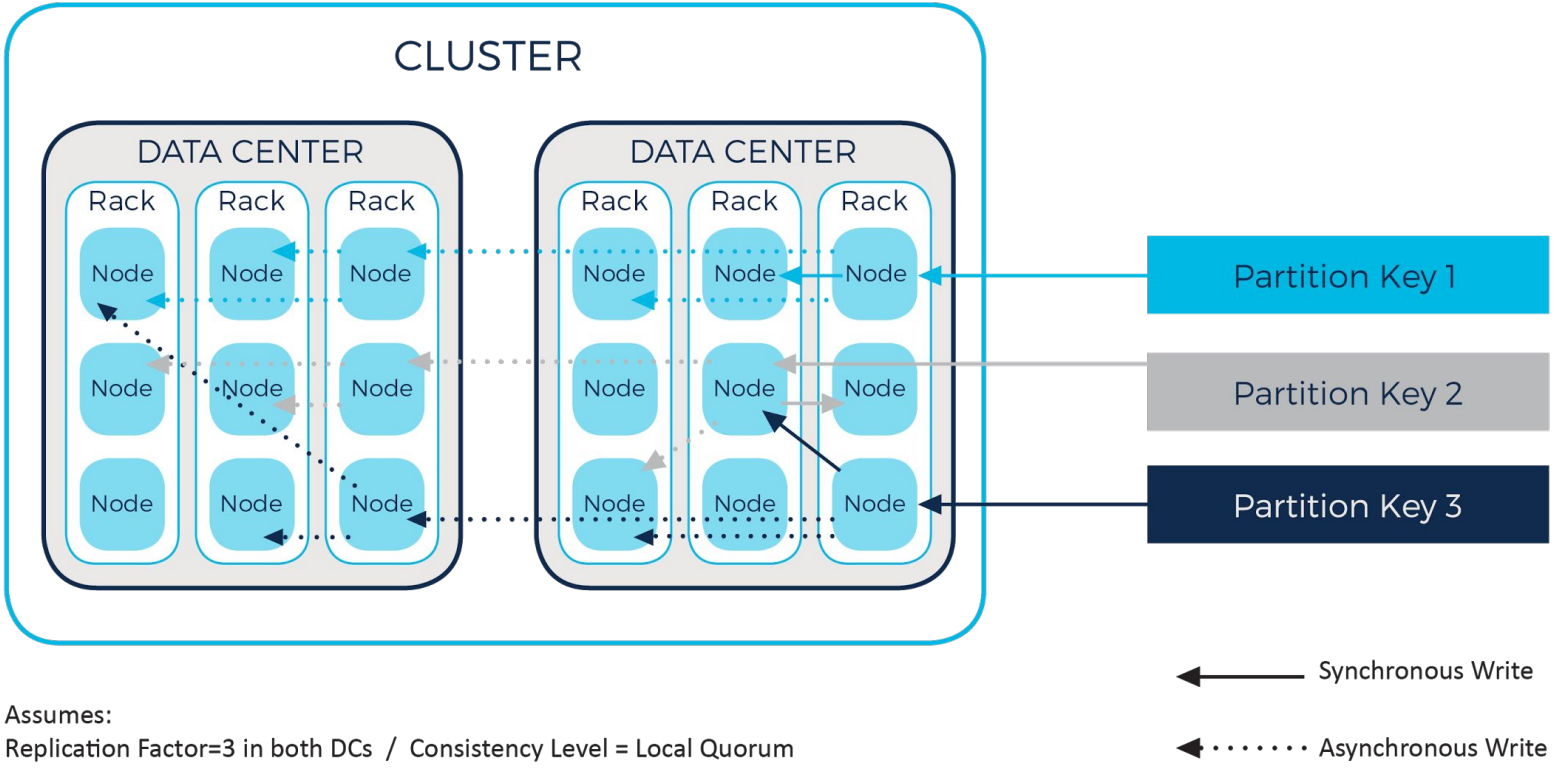
Agenda

instaclustr

- Cassandra - a very brief introduction + glossary
- Best practices
  - Design approach
  - Partitioning
  - Tombstones
  - Compaction Strategies
  - Security
  - Testing
- How Instaclustr can help

Quick introduction to Cassandra

instaclustr



Assumes:
Replication Factor=3 in both DCs / Consistency Level = Local Quorum

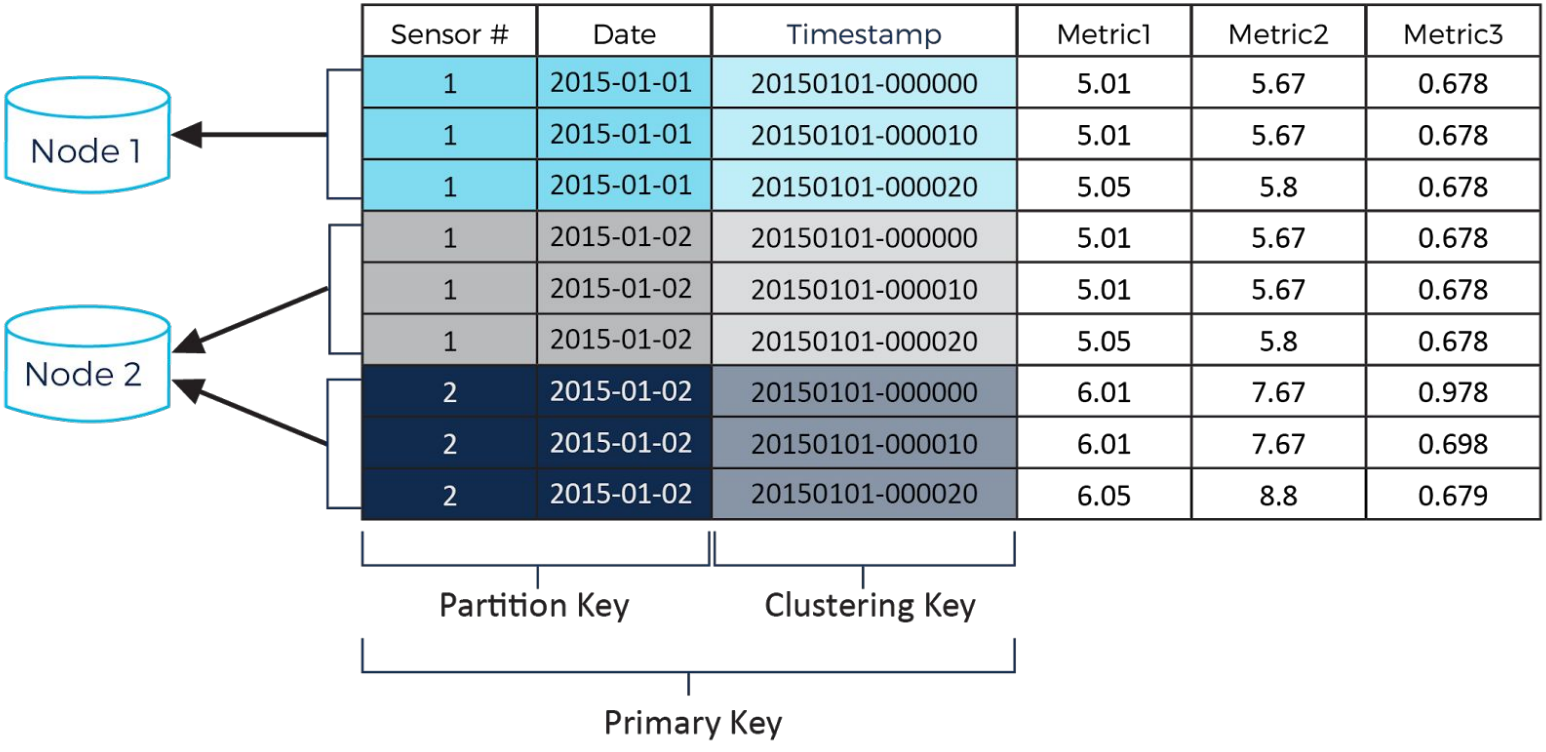
Design Approach

instaclustr

- Phase 1: Understand the data
  - Define the data domain: E-R logical model
  - Define the required access patterns: how will you select an update data?
- Phase 2: Denormalize based on access patterns
  - Identify primary access entities: driven by the access keys
  - Allocate secondary entities: denormalize by pushing up or down to the primary entities
- Phase 3: Review & tune
  - Review partition keys and clusters
    - Do partition keys have sufficient cardinality?
    - Is the number of records in each partition bounded?
    - Does the design consider delete and update impact?
  - Test & tune: check updates, review compaction strategy

Partitioning

instaclustr



PRIMARY KEY ((Sensor,Date),Timestamp)

Partitioning: Diagnosing & Correcting

instaclustr

- Diagnosing
  - Many issues can be identified from data model review
  - nodetool cfstats / tablestats and cfhistograms provide partition size info. <10MB green, <100MB amber
  - Log file warnings - compacting large partition
  - Overlarge partitions will also show up through long GC pauses and difficulty streaming data to new nodes
- Correcting
  - Correcting generally requires data model change although depending on the application, application level change may be possible
  - ic-tools can help by providing info about partition keys of large partitions

Tombstones

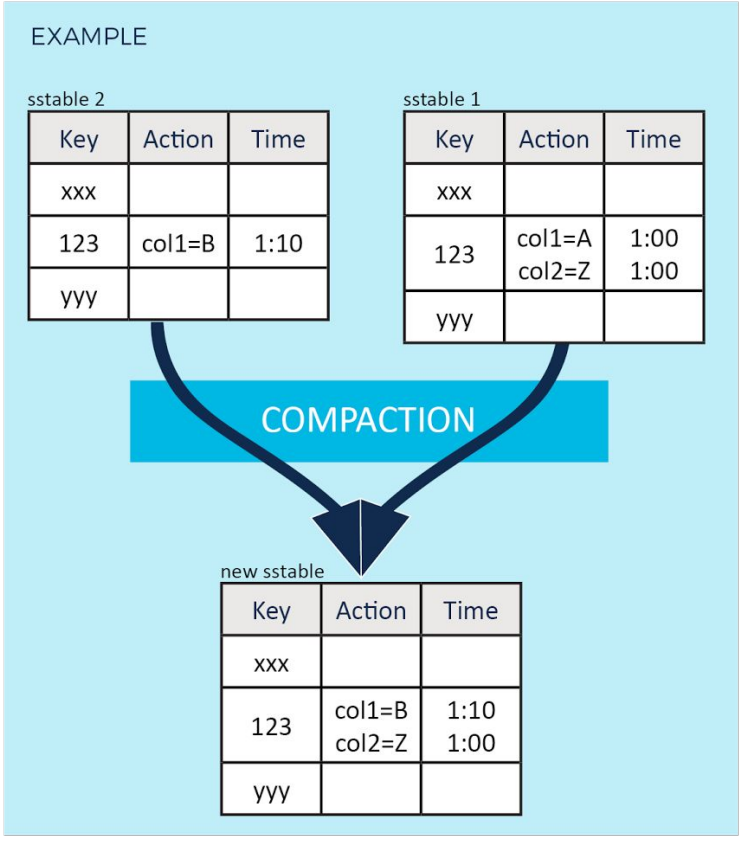
instaclustr

- When a row is deleted in C* it is marked with a tombstone (virtual delete). Tombstones remain in the sstables for at least 10 days by default.
- A high ratio of tombstones to live data can have significant negative performance impacts
- Be wary of tombstones when: deleting data, updating with nulls or updating collection data types.
- Diagnosing
  - nodetool cfstats/cfhistograms and log file warnings
  - slow read queries, sudden performance issues after a bulk delete
- Correcting
  - tune compaction strategy - LCS or TWCS can help in the right circumstances
  - reduce GC grace period & force compaction for emergencies
  - review data model design to reduce deletes within a partition

Compaction Intro

instaclustr

- Cassandra never updates files once written to disk
- Instead all inserts and updates are essentially written as transaction logs that are reconstituted when read
- Compaction is the process of consolidating transaction logs to simplify reads
- It's an ongoing background process in Cassandra
- Compaction ≠ Compression



Compaction Strategies

instaclustr

- Compaction strategies determine which files Cassandra picks to compact and when
- Three compaction strategies are supported: Size Tiered Compaction Strategy, Levelled Compaction Strategy, Time Windowed Compaction Strategy (technically Data Tiered also exists but deprecated)
- Diagnosing
  - Slower than desired reads, old data being retained on disk
  - nodetool cfstats sstables per read state (ideally 1 or 2 sstables on average)
- Correcting
  - Rule of thumb: if using STCS with issues, look at LCS
  - Compaction strategy can be changed online but can result in significant load due to recompaction
  - Can use jmx to update strategy on one node at a time before changing schema
  - Can use write survey mode to test changes in write overhead

Security

instaclustr

- At a minimum
  - Enable password auth
  - Enable client->server encryption (particularly if using public IPs to connect)
  - Enable internode encryption
  - Don’t use the default Cassandra user
- Best practice
  - Encrypt sensitive data at the client
    - Works well with typical C* access patterns where PK values are hashed anyway
    - Dates are the most common case of range selects and typically are not sensitive if other identifying data is encrypted

Testing Cassandra applications

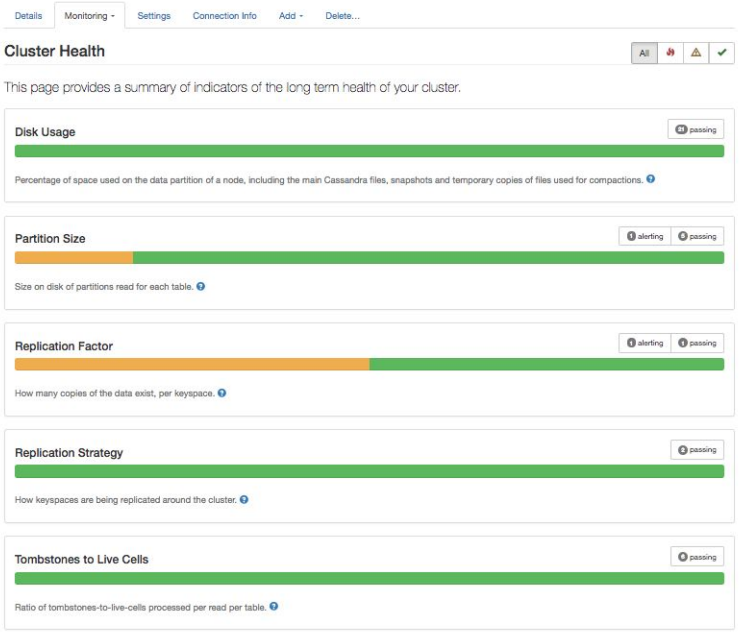
instaclustr

- Long running tests with background load are vital
  - Can run extremely high write loads for an hour or two but might take days to catch up on compactions
  - Don’t forget repairs
- Make sure your data volumes on disk are representative as well as read/write ops - cache hit rates can make a big difference to performance
- Mirror production data demographics as closely as possible (eg partition size)
- Don’t forget to include update/delete workload if applicable
- For core cassandra features, can test on reduce size and rely on scale-up but beware:
  - Secondary indexes
  - MVs

How Instaclustr can help

instaclustr

- Managed Service
  - Gives you a proven, best practice configured Cassandra cluster in < ½ hour
  - AWS, Azure, GCP and SoftLayer
  - Security configuration with tick of a check box
  - Cluster health page for automated best practice checks
  - Customer monitoring UI & ongoing monitoring & response by our Ops Team
- Consulting
  - Cluster health reviews
  - Data model & Cassandra application design assistance
- Enterprise Support
  - Support from our Managed Service tech-ops team where you run your own cluster



instaclustr

Questions?

Ben Slater
Chief Product Officer
ben.slater@instaclustr.com