Two years in the making: What is new in Apache Cassandra 4.0?

Dinesh A. Joshi

apache cassandra

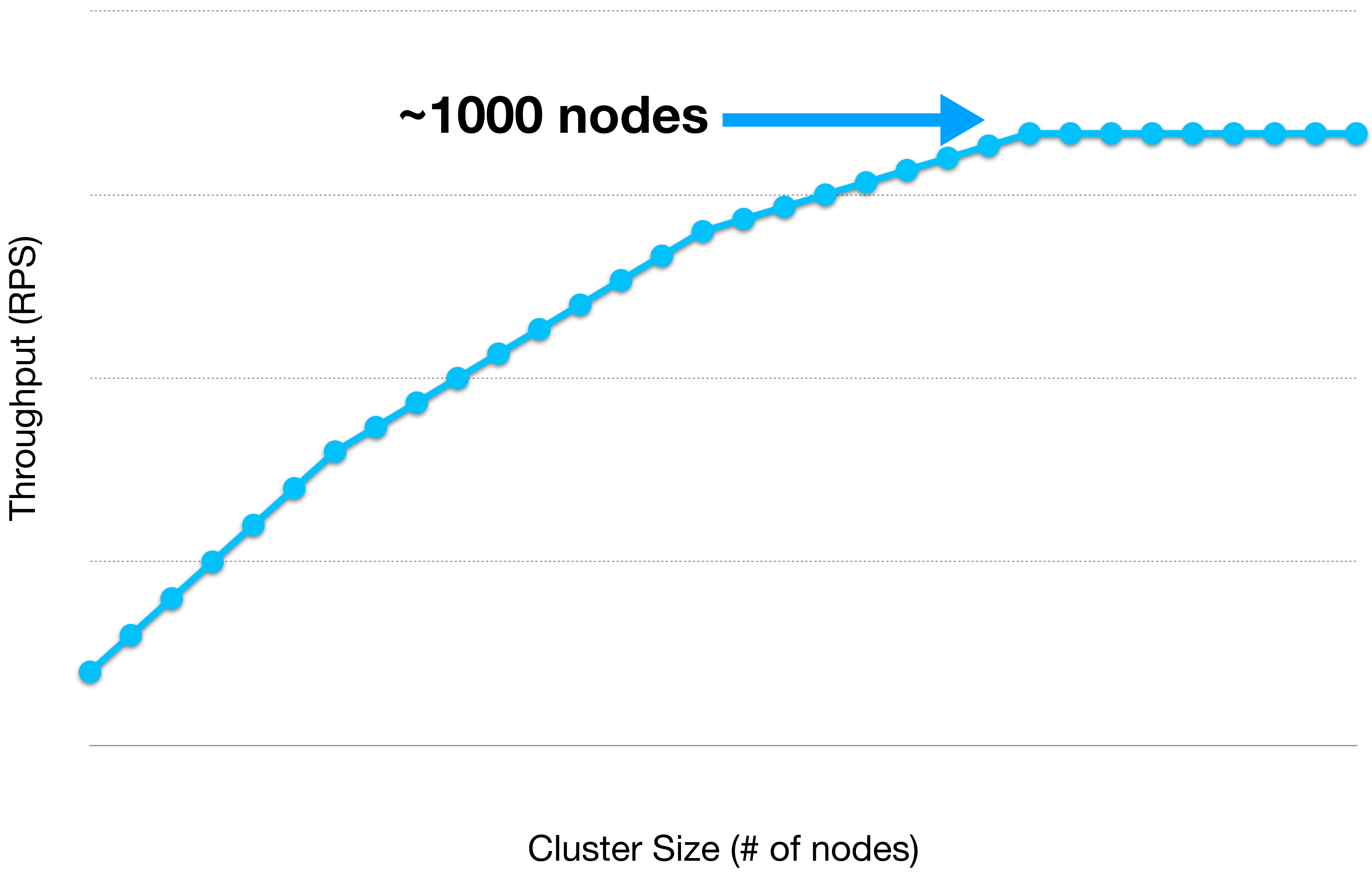
Agenda

- History
- Performance & Efficiency
- Reliability & Stability
- Operability & Observability
- Summary

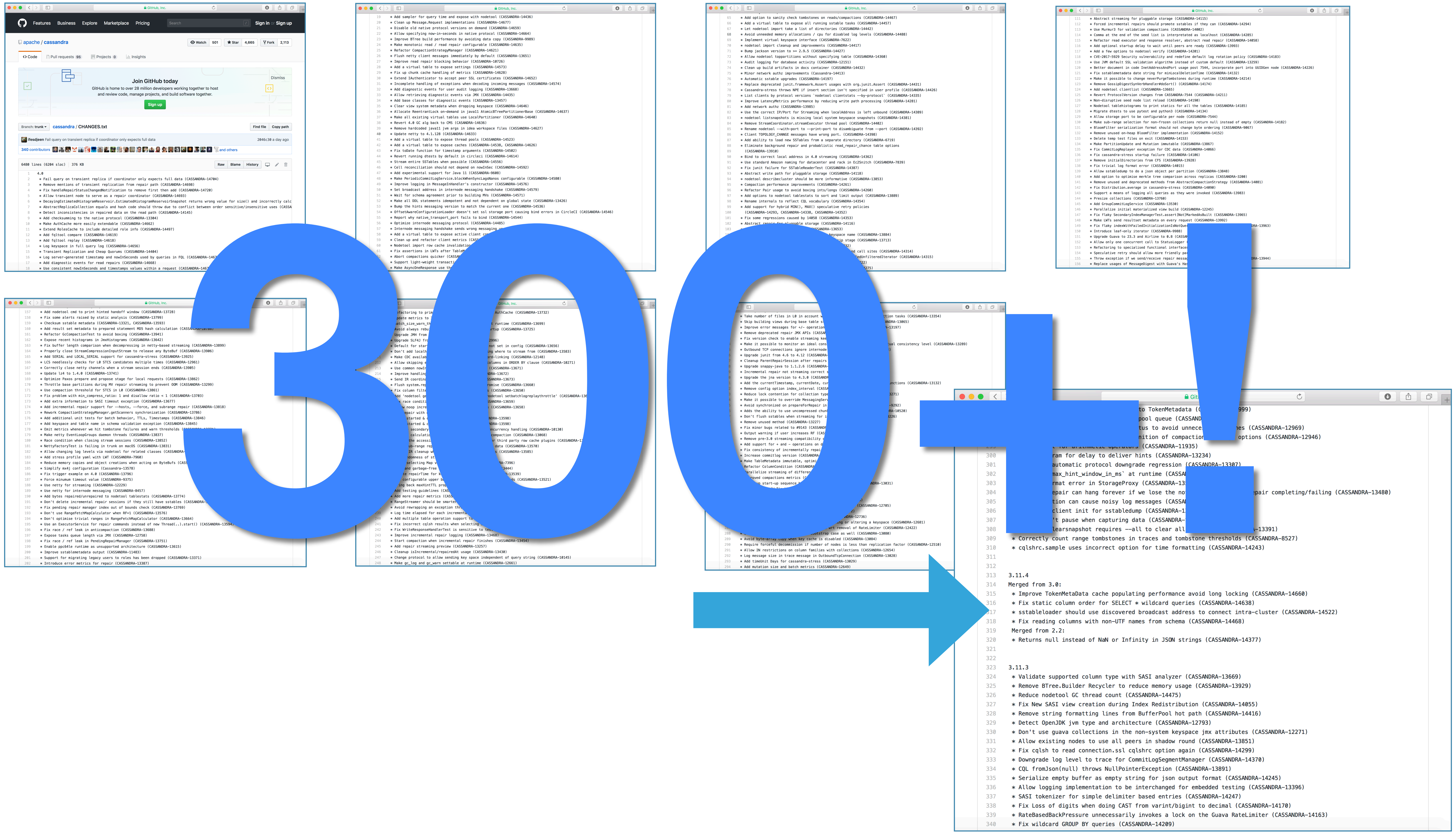
Cassandra in Community

- Cassandra 2.x (stable, widely adopted)
- Cassandra 3.0.x (stable)
- Cassandra 3.11.x (ok, perf improvements over 3.0.x)

Throughput vs Cluster Size



Cassandra 4.0 Changes



300+!

```
to TokenMetadata (C                 999)
pool queue (CASSAND
tus to avoid unnece                  es (CASSANDRA-12969)
nition of compaction                 options (CASSANDRA-12946)
(CASSANDRA-11935)
ram for delay to deliver hints (CASSANDRA-13234)
automatic protocol downgrade regression (CASSANDRA-13307)
max_hint_window_in_ms` at runtime (CASSA
rmat error in StorageProxy (CASSANDRA-13
epair can hang forever if we lose the no                    epair completing/failing (CASSANDRA-13480)
tion can cause noisy log messages (CASSA
client init for sstabledump (CASSANDRA-1
't pause when capturing data (CASSANDRA-
learsnapshot requires --all to clear all                    -13391)
 * Correctly count range tombstones in traces and tombstone thresholds (CASSANDRA-8527)
 * cqlshrc.sample uses incorrect option for time formatting (CASSANDRA-14243)


3.11.4
Merged from 3.0:
 * Improve TokenMetaData cache populating performance avoid long locking (CASSANDRA-14660)
 * Fix static column order for SELECT * wildcard queries (CASSANDRA-14638)
 * sstableloader should use discovered broadcast address to connect intra-cluster (CASSANDRA-14522)
 * Fix reading columns with non-UTF names from schema (CASSANDRA-14468)
 Merged from 2.2:
 * Returns null instead of NaN or Infinity in JSON strings (CASSANDRA-14377)


3.11.3
 * Validate supported column type with SASI analyzer (CASSANDRA-13669)
 * Remove BTree.Builder Recycler to reduce memory usage (CASSANDRA-13929)
 * Reduce nodetool GC thread count (CASSANDRA-14475)
 * Fix New SASI view creation during Index Redistribution (CASSANDRA-14055)
 * Remove string formatting lines from BufferPool hot path (CASSANDRA-14416)
 * Detect OpenJDK jvm type and architecture (CASSANDRA-12793)
 * Don't use guava collections in the non-system keyspace jmx attributes (CASSANDRA-12271)
 * Allow existing nodes to use all peers in shadow round (CASSANDRA-13851)
 * Fix cqlsh to read connection.ssl cqlshrc option again (CASSANDRA-14299)
 * Downgrade log level to trace for CommitLogSegmentManager (CASSANDRA-14370)
 * CQL fromJson(null) throws NullPointerException (CASSANDRA-13891)
 * Serialize empty buffer as empty string for json output format (CASSANDRA-14245)
 * Allow logging implementation to be interchanged for embedded testing (CASSANDRA-13396)
 * SASI tokenizer for simple delimiter based entries (CASSANDRA-14247)
 * Fix Loss of digits when doing CAST from varint/bigint to decimal (CASSANDRA-14170)
 * RateBasedBackPressure unnecessarily invokes a lock on the Guava RateLimiter (CASSANDRA-14163)
 * Fix wildcard GROUP BY queries (CASSANDRA-14209)
```

Repair

What is repair?

- Cassandra is an AP system (CAP theorem)
- Nodes can be out of sync
- Eventually consistent
- Repair ensures nodes are consistent

Incremental Repair

- Pre 4.0 repair took long time
- Run repairs all the time!
- Shorter data reconciliation times
- Enables Transient Replication

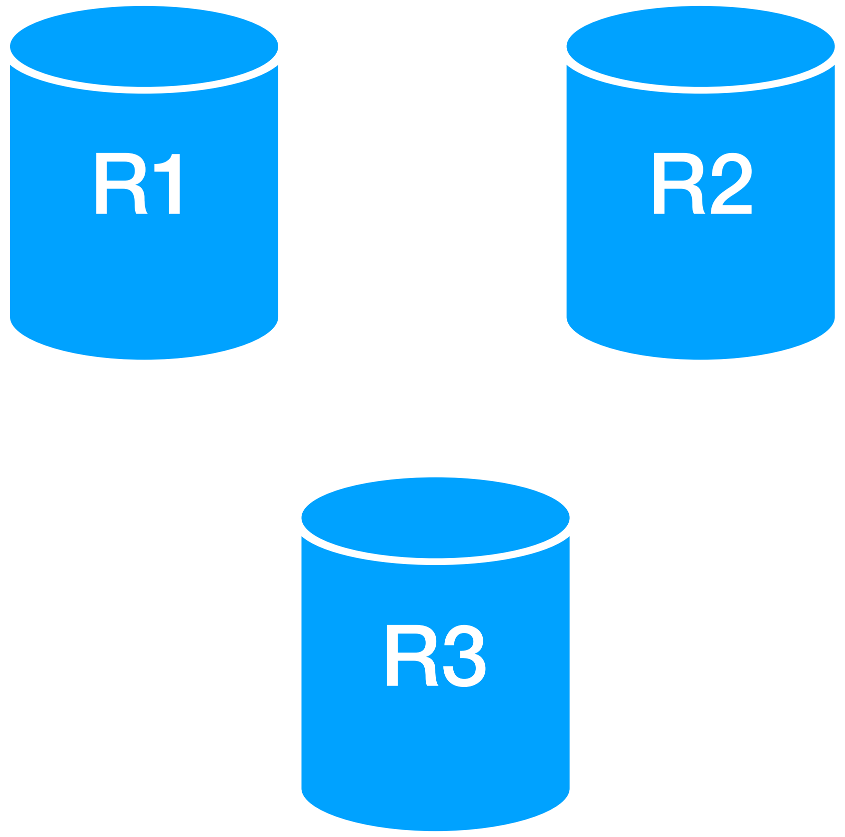
Impact

- Pre 4.0 repair took hours / days / weeks
- Run repair continuously

Transient Replication

Replication

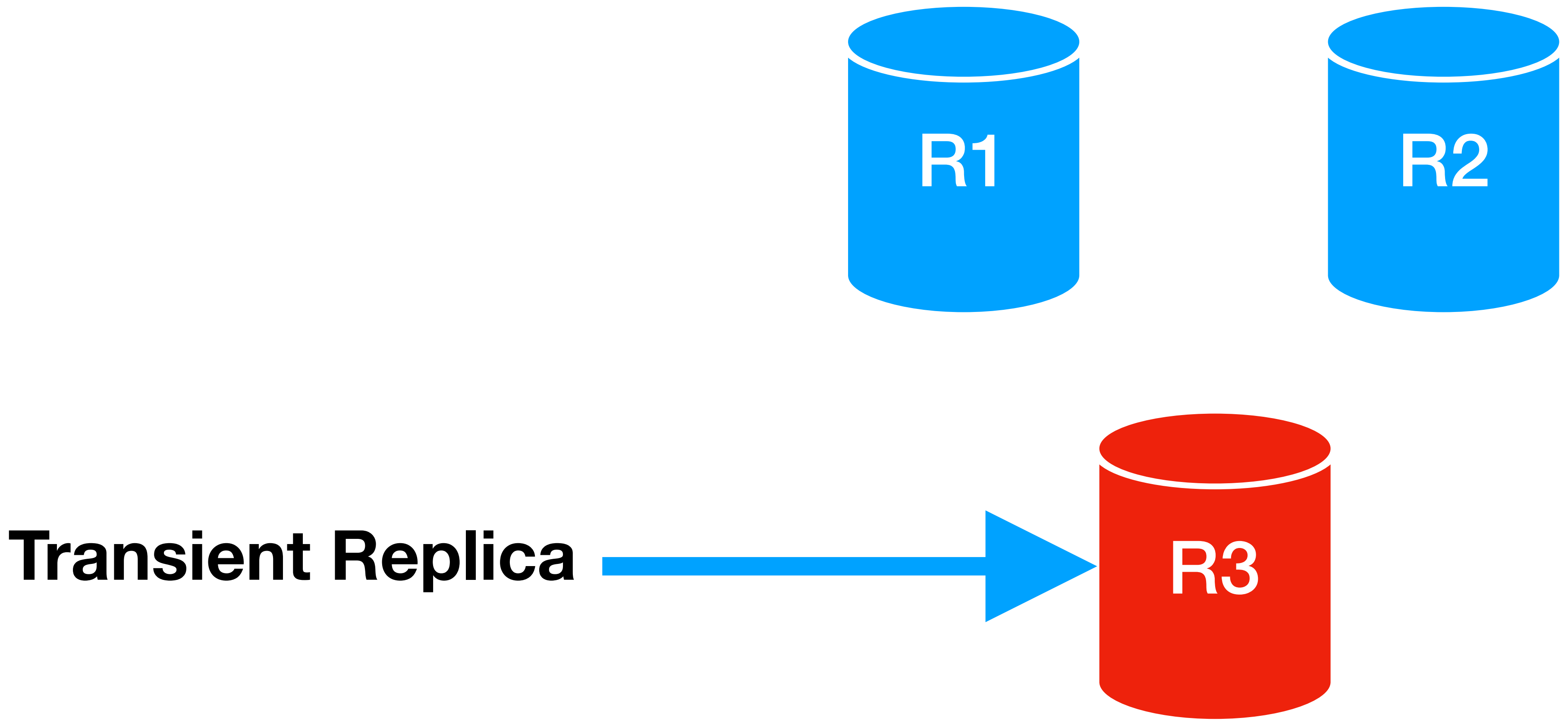
Replication Factor = 3



All nodes are <u>full</u> replicas

Transient Replication

Replication Factor = 3



Transient Replication

- Voting with witnesses
- Cheap quorums
- Brings storage efficiency

Impact

- Up to 33% reduction for RF=3
- Leverages Incremental Repair

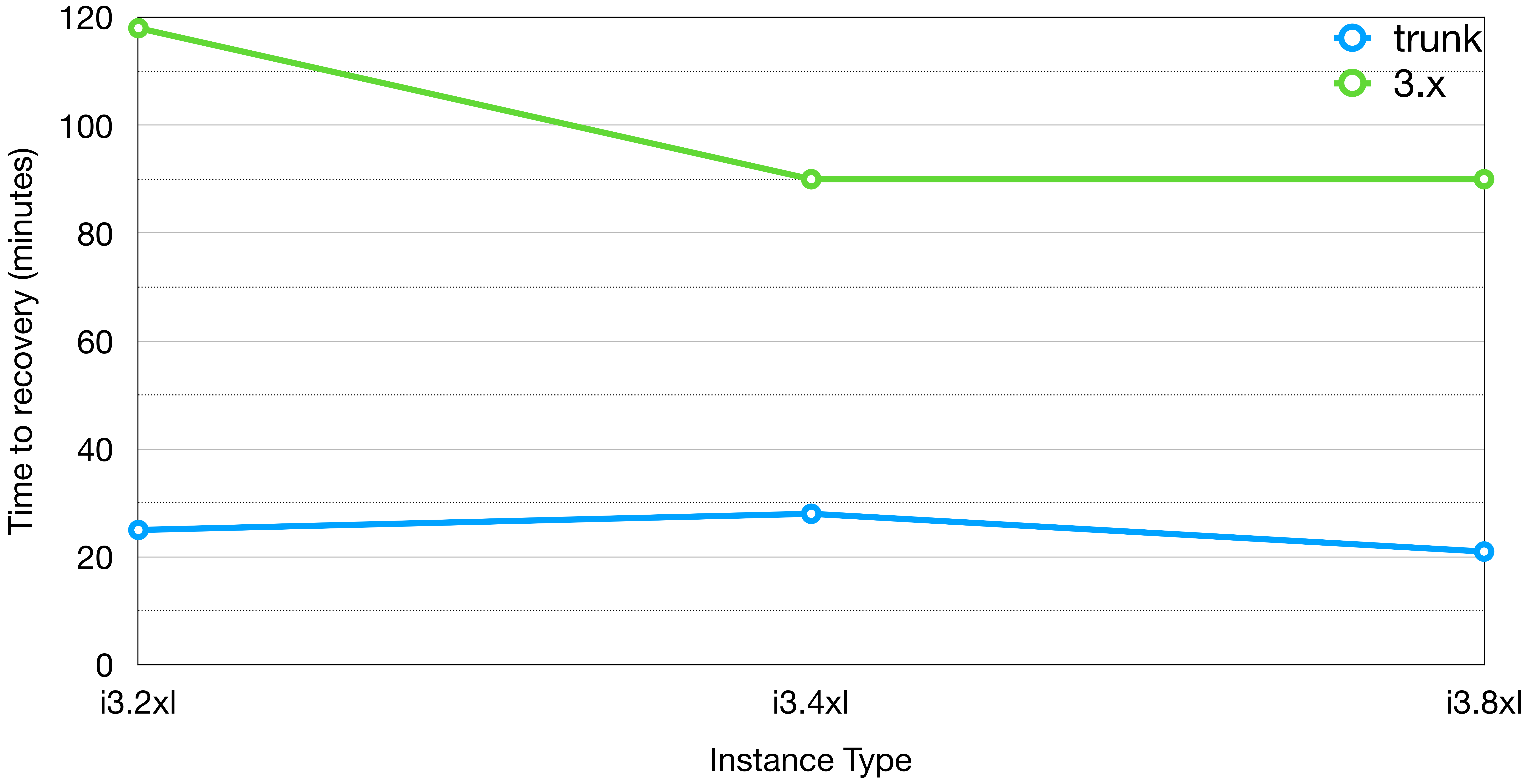
Async Internode Messaging

- Lower Latencies (**40%** lower avg **60%** lower p99)
- Memory Efficiency (~10x reduction)
- Scalable internode encryption (~4x throughput)
- Better throughput & response times (~2x vs 3.0)

Zero Copy Streaming

- Speeds up all Streaming operations (~5x)
- IO Bound (Disk, NIC)
- Dramatically reduces MTTR
- Lowers operational cost

Impact



Source: https://issues.apache.org/jira/browse/CASSANDRA-14765

BTree Build Performance

- Affects hot path
- Bulk loads BTree
- ~2.6x speed up in throughput

Reliability & Stability

- Checksummed Native Protocol
- Checksummed SSTable Metadata

Audit Logging

- Logs everything
- Performant (Binary Logging)
- Helps in compliance for Enterprises

Virtual Tables

Virtual Tables

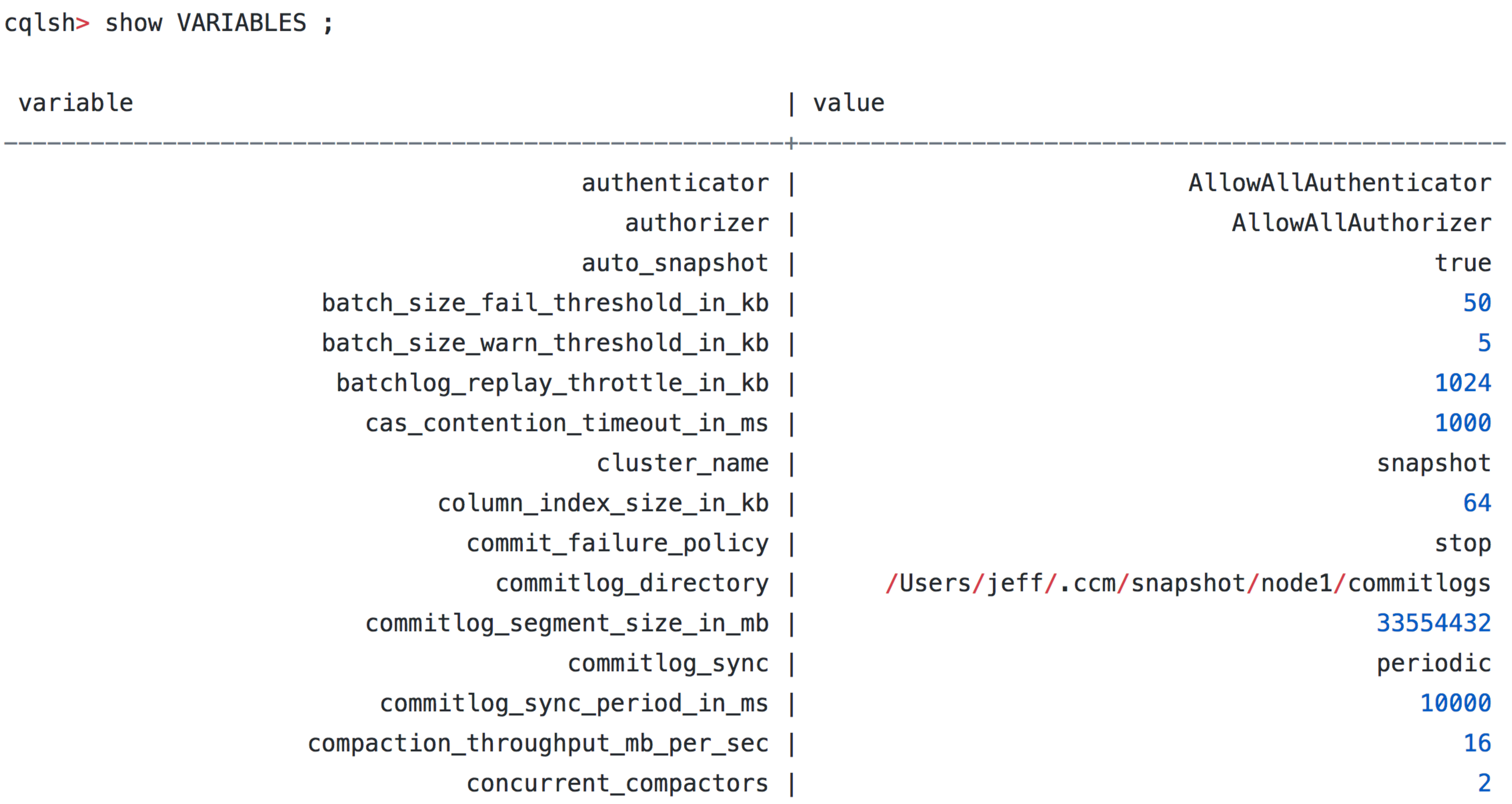
- Table backed by an API
- Queried through CQL
- NO JMX!
- Driver support

Virtual Tables

```
cqlsh> select * from virtual.tables5 where keyspace_name = ‘my_ks' and metric > 'memtable' and
metric < 'memtableZ' ALLOW FILTERING;

 keyspace_name | table_name                | metric               | value
---------------+---------------------------+----------------------+-----------------
         my_ks |        monitoring_example |   memtableOnHeapSize |  {"value":95201}
         my_ks |        monitoring_example |  memtableOffHeapSize |  {"value":44811}
         my_ks |        monitoring_example | memtableLiveDataSize |  {"value":42128}
         my_ks |        monitoring_example | memtableColumnsCount |    {"value":248}
         my_ks |        monitoring_example |  memtableSwitchCount |    {"count":4}
...
```

Virtual Tables



```
cqlsh> show VARIABLES ;

 variable                                             | value
------------------------------------------------------+-------------------------------------------------
                                        authenticator |                           AllowAllAuthenticator
                                           authorizer |                              AllowAllAuthorizer
                                        auto_snapshot |                                            true
                      batch_size_fail_threshold_in_kb |                                              50
                      batch_size_warn_threshold_in_kb |                                               5
                       batchlog_replay_throttle_in_kb |                                            1024
                         cas_contention_timeout_in_ms |                                            1000
                                         cluster_name |                                        snapshot
                              column_index_size_in_kb |                                              64
                                commit_failure_policy |                                            stop
                                  commitlog_directory |      /Users/jeff/.ccm/snapshot/node1/commitlogs
                         commitlog_segment_size_in_mb |                                        33554432
                                       commitlog_sync |                                        periodic
                          commitlog_sync_period_in_ms |                                           10000
                     compaction_throughput_mb_per_sec |                                              16
                                concurrent_compactors |                                               2
```

SSL Certificates Hot Reloading

- Certificates hot reload on update
- Optional Manual trigger via nodetool
- No disruption to live traffic
- Operators love it

Full Query Logging (FQL)

- High performance query capture
- Replay tool (fqltool replay)
- Compare tool (fqltool compare)
- Useful for replaying traffic

Summary (4.0 vs 3.0)

- Better operability
- Better scalability
- Better latencies
- Faster recovery