Apache CASSANDRA

Cassandra Day

2022

Cassandra 4.1 Guardrails

Sponsored by DataStax

Apache Cassandra Guardrails

A framework that allows to enforce certain system-wide rules, soft and hard limits.

- First implemented in DSE 6.8
- Merged into Cassandra this year
- Released with C* 4.1
- Minimum-to-zero overhead
- Designed to easily implement new and/or custom rules

→ Ask Aleks to explain why it happens! ←

- Too many tables
- Or secondary indexes
- Queries touching too many partitions
- Queries using replica-side filtering
- Size of non-frozen collections
- Etc.

Configuration

- *cassandra.yaml* ✅
- Dynamically via JMX ✅
- Virtual tables ⚒️

```
tables_warn_threshold: 5
tables_fail_threshold: 10
secondary_indexes_per_table_warn_threshold: 5
secondary_indexes_per_table_fail_threshold: 10
allow_filtering_enabled: false
partition_keys_in_select_warn_threshold: 10
partition_keys_in_select_fail_threshold: 20
collection_size_warn_threshold: 10MiB
collection_size_fail_threshold: 20MiB
```

Soft and Hard Limits

CREATE TABLE table81 (.....);

Warning :
Guardrail tables violated: Creating table t81, current number of tables exceeds warning threshold of 80.

CREATE TABLE k.t101 (.....);

InvalidRequest: **Error** from server: code=2200 [Invalid query] message="Guardrail tables violated: Cannot have more than 100 tables, aborting the creation of table t101"

CQL-Layer Guardrails 1/2

- Implemented on CQL level (quick)
- Don't involve Storage level
- Don't involve additional replicas
- "Foreground"
- Support both soft and hard limits

```
cqlsh> SELECT * FROM table WHERE v=0
ALLOW FILTERING;

InvalidRequest: Error from server:
code=2200 [Invalid query]
message="Guardrail allow_filtering
violated: Querying with ALLOW FILTERING
is not allowed"
```

Background Guardrails 2/2

- Run in background
- Aren't associated with any query
- Used when CQL-specific guardrail would be slow or not related to CQL
- "Background"
- Support mostly soft limits
- **Watch the logs!**

- Disk Space Usage
- Number of items in a non-frozen collections

Already Available

- Number of user keyspaces
- Number of user tables
- Number of columns per table
- Number of secondary indexes per table
- Number of materialized tables per table
- Number of fields per user-defined type
- Number of items in a collection
- Number of partition keys selected by an IN restriction
- Number of partition keys selected by the cartesian product of multiple IN restrictions.
- Allowed table properties
- Allowed read consistency levels
- Allowed write consistency levels
- Allowed write consistency levels
- Collections size
- Query page size
- Minimum replication factor
- Data disk usage, defined either as a percentage or as an absolute size
- Whether user-defined timestamps are allowed
- Whether GROUP BY queries are allowed
- Whether the creation of secondary indexes is allowed
- Whether the creation of uncompressed tables is allowed
- Whether querying with ALLOW FILTERING is allowed
- Whether dropping or truncating a table is allowed

And more to come!

Psssssst. You can add yours, it's easy!

Implementation and Migration

Be careful planning to enable guardrails. Your app may totally fail!

1. First start with soft limits!
2. Watch the logs!
3. Test PRs (new code) against it!

DEMO

Apache CASSANDRA

Thank you!

Sponsored by DataStax