

ConfigurationNotes

Basics

Per-node options are loaded from yaml and held in DatabaseDescriptor.

Per-KS, per-CF, and per-Column options are loaded from the MigrationsTable at startup and are encapsulated with !KSMetaData, !CFMetaData, and ColumnDefinition objects, which are held by !Schema and !Table. When a migration arrives, it writes to the MigrationsTable, then propagates the changes out to the KS/CFMD objects in the system.

Configuration can be changed at runtime without a restart (excluding the ones that change on-disk format (which cannot be changed without clearing the cluster) and ones that change routing). For per-node options, poke StorageService via JMX (which in turn pokes DatabaseDescriptor). For per-KS options, poke the appropriate !Table. For per-CF and per-Column options, poke the appropriate ColumnFamilyStore. These ephemeral changes are stronger than migrations (they stay set regardless of new config coming in), but do not persist between reboots.

How to add a new CF option post-1906

- update cassandra.thrift and src/avro/internodo.genavro to add the new option **AND UPDATE THE THRIFT API VERSION**
- ant gen-thrift-java, ant avro-generate
- set a static final T DEFAULT_ for it if there should be one
- set the default in init()
- define CFMD foo(T prop) { foo = prop; return this } for the builder
- define T getFoo() {return foo;} since all optional params are private
- update to{Avro/Thrift}() and from{Avro/Thrift}() to handle the new option ->CfDef and CfDef->
- update equals(), hashCode(), and toString() to build with the new prop
- update applyImplicitDefaults()
- update apply() (a.k.a. applyAvroMigrationChangesToCurrentCFMD)
- if desired, add new option to CLI add/update CF
- if desired, create a JMX call to CFS to change it at runtime

Done. Then, whenever you need the option in code, use DD.getCFMD(ksname,cfname).getFoo();

<https://c.statcounter.com/9397521/0/fe557aad/1/> | stats