DRAFT - Not yet in discussion

EXPLAIN CEP (Draft)

Status

Current state: *Draft*

Discussion thread: *TBD*

JIRA: *TBD*

Released: Not released

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Scope

Add the following functionality to CQL:

- EXPLAIN - Estimate how long a query is likely to take.
- EXPLAIN ANALYZE - Measure how long a query took.

Where possible we will provide recommendations or observations that could help the user, possibilities include:

- DDL to create an appropriate index for a query
- Slow or uncontactable node warning
- Inefficient query warning.
- Tomestone warnings.

Wherever a warning is given the user will be given basic advice on how to resolve the problem.

This is a foundational feature addition that can be improved upon over time.

Not in scope:

- Measuring latency between the server and the client.
- Remote latencies.
- Exposing low level information that the user has no ability to modify.

Goals

- Make it easier for users to capacity plan by providing estimated and actual query times.
- Hide all information that is likely to be difficult for users to understand without internal knowledge on how C* operates or that is beyond the control of users to modify.
- Syntax should be closer to SQL counterparts than the tracing option.
- EXPLAIN should be safe to run against a live cluster without causing issues that running a very large query would.
- Opportunity to provide advice to users on their query performance.

Risks

- It is not possible to create a cost model that is accurate enough to add value for users.

Approach

- Add EXPLAIN and EXPLAIN ANALYZE to CQL grammar
- Add functionality to SelectStatement that will add metrics collection.
- Add functionality to SelectStatement that will allow sampling to be used to collect metrics.
- Work out a cost model. Possibly this could be trainable with calls to EXPLAIN ANALYZE

Timeline

- Post 4.0

Mailing list / Slack channels

Mailing list:

- *here*

Slack channel:

- *here*

Discussion threads:

- *here*

Related JIRA tickets

JIRA(s):

- *here*

Motivation

The current tracing facility in C* requires that the user executes a query against the cluster to measure performance. It includes low level information that users do not have the ability to influence, such as query preparation, calculating ranges etc.

A new EXPLAIN feature that provides the ability to measure, estimate and possibly provide advice on how to improve queries would be a step forward in usability.

When estimating the explain feature must be safe to run against live clusters no matter what query is being explained. This is critical for users who have got a live system and need to understand how executing a query is likely to behave.

Audience

- Devops
- Developers
- Ops

Proposed Changes

Add the following to CQL:

- EXPLAIN - Returns estimated query metrics and no row data.
- EXPLAIN ANALYZE - Returns actual query metrics and row data.

EXPLAIN data will be sent as sideband data in addition to the main payload. This will require protocol changes and driver updates.

The format for the EXPLAIN data is left open at this stage.

Drivers will not be required to reformat EXPLAIN data in a human readable format.

All of the following examples display rendered information in CQLSH. The EXPLAIN data must be sent to the client in a structured format so that tooling can display the information how they wish.

The CQLSH renderings are for example only and may change to enhance readability.

SELECT statements

Users can prepend EXPLAIN or EXPLAIN ANALYZE to their select statements to obtain estimated or actual execution metrics.

The output from EXPLAIN will be clearly marked as `Estimated`

```
> EXPLAIN SELECT * FROM person;

SELECT * FROM person (Estimated)
      type: full scan
      consistency: QUORUM
      rows:9034
      time: 114ms
      filtered: no
      filtered rows: 0
      aggregate: no
      page size: 10000
```

```
tombstones: 2000
replicas: 3
```

This is a purposely simplified view vs the existing tracing functionality.

EXPLAIN will only measure the times taken by the coordinator to perform the actual query and construct the resultset. Information such as time taken to prepare statements will not be included. Such information is not generally relevant to users and developers can still use the tracing facilities.

The information could include:

- **type** - The type of query, for example `full scan`, `single partition slice`, `multi partition range`, `secondary index`
- **consistency** - the consistency level
- **rows** - the number of rows that are returned
- **time** - the time it took to execute the query
- **filtering** - if the query is filtering or not
- **filtered rows** - the number of filtered rows that did not make it to the resultset.
- **aggregate** - if the query is an aggregate
- **page size** - the size of the page
- **tombstones** - the number of tombstones scanned
- **replicas** - the number of replicas hit by the query.

In addition to the query performance information advice for users can be supplied. This could include:

- The DDL statement to add an index that will allow the query to run without filtering. Each type of index will detail the tradeoffs between performance and storage
- If a node in the cluster is taking significantly longer to respond than others
- If nodes were uncontactable
- Avoiding multi-partition reads
- Avoiding full scans
- Very high or low cardinality index warning
- Large number of tombstones.

Some of these warnings already exist in C*, but the critical difference is that basic advice will be supplied on how to avoid or correct the situation.

EXPLAIN

EXPLAIN will try to estimate query performance.

The output from explain will be clearly marked as `estimated`

```
> EXPLAIN SELECT * FROM person;

SELECT * FROM person (Estimated)
      type: full scan
      consistency: QUORUM
      rows: ~9000
      elapsed time: ~200ms
      filtering: no
      filtered rows: 0
      aggregate: no
      page size: 10000
      tombstones: 2000
      replicas: 3
```

The exact implementation details of the cost model are left open. It is the intent that we start with a simple if inaccurate model, and improve over time.

The output from an estimated EXPLAIN must convey the estimated nature of the results clearly. For instance, by:

- avoiding too many significant figures in results.
- using '~' against anything numeric.

If we are unable to provide a cost model that is sufficiently good to add value for the user then this feature will be dropped.

EXPLAIN ANALYZE

EXPLAIN ANALYZE will perform the real query and return the actual metrics on how long the query took.

The output from explain analyze will be clearly marked as `Actual`

```
> EXPLAIN ANALYZE SELECT * FROM person;

SELECT * FROM person (Actual)
      type: full scan
      consistency: QUORUM
      rows:20034
      elapsed time: 114ms
      filtering: no
      filtered rows: 0
      aggregate: no
      page size: 10000
      tombstones: 2000
      replicas: 3
```

Mutation statements

Mutations are trickier to estimate without modifying the data. For this initial implementation we will only support EXPLAIN ANALYZE.

INSERT, UPDATE, DELETE and BATCH statements are supported.

For example:

```
> EXPLAIN ANALYZE INSERT INTO person (id, name) VALUES (4, "Bif");

INSERT INTO person (id, name) VALUES (4, "Bif")
      consistency: QUORUM
      time: 114ms
      indexes: person_name_idx(sasi)
      materialized views: person_by_name
      creates tombstone: no
```

The information returned will be:

- **consistency -** the consistency level
- **time** - the time it took to execute the query
- **indexes** - the list of indexes that are also updated as part of this mutation
- **materialized views** - the list of materialized views that are updated as part of this mutation.
- **creates tombstones** - if this mutation creates tombstones
- **mutation size** - the total size of the mutation in bytes.

Advice could include:

- Warning if using CAS.
- Warning if using CL_ALL.
- Warning if large numbers of indexes are updated.
- Warning if large numbers of materialized views.
- Warning if large numbers of tombstones will be created
- If a node in the cluster is taking significantly longer to respond than others.
- If nodes were uncontactable.
- Very high or low cardinality index warning

BATCH statements

Batch statement support will be an extension of regular mutation statement support

```
> EXPLAIN ANALYZE BEGIN UNLOGGED BATCH
            INSERT INTO person (id, name) VALUES (4, "Bif")
            INSERT INTO person (id, name) VALUES (5, "Bob")
            APPLY BATCH;

BEGIN UNLOGGED BATCH
|           INSERT INTO person (id, name) VALUES (4, "Bif")
|           INSERT INTO person (id, name) VALUES (5, "Bob")
|           APPLY BATCH; (Actual)
|     elapsed time: 114ms
|
|--INSERT INTO person (id, name) VALUES (4, "Bif")
|       consistency: QUORUM
|       elapsed time: 114ms
|       indexes: person_name_idx(sasi)
|       materialized views: person_by_name
|       creates tombstone: no
|
|--INSERT INTO person (id, name) VALUES (5, "Bob")
        consistency: QUORUM
        elapsed time: 15ms
        indexes: person_name_idx(sasi)
        materialized views: person_by_name
        creates tombstone: no
```

The information returned will be:

- All the information from INSERT
- **elapsed time -** the time taken to execute the entire statement
- subquery information.

Advice could include:

- Multi partition writes warning.
- Logged batch warning.

New or Changed Public Interfaces

The following features will be added to CQL:

- EXPLAIN
- EXPLAIN ANALYZE
- GRANT/REVOKE EXECUTE ON EXPLAIN
- GRANT/REVOKE EXECUTE ON EXPLAIN ANALYZE

In addition the user will not be able to run any statement that they did not already have access to by using EXPLAIN functionality.

Compatibility, Deprecation, and Migration Plan

There should be no breaking changes.

Test Plan

Unit and integration tests for new functionality covering:

- Parsing
- Execution
- Permissions

Prove existing query performance is not affected.
Prove that EXPLAIN can run on clusters under heavy load without causing problems.
Prove that EXPLAIN cannot be used to bypass ACL
Prove that EXPLAIN is accurate enough.
Prove that EXPLAIN ANALYZE cannot be used to bypass ACL
Prove that EXPLAIN ANALYZE is accurate.

All performance tests will be made public so that other parties can verify results.

Rejected Alternatives