
DATASTAX

On Cassandra's evolution

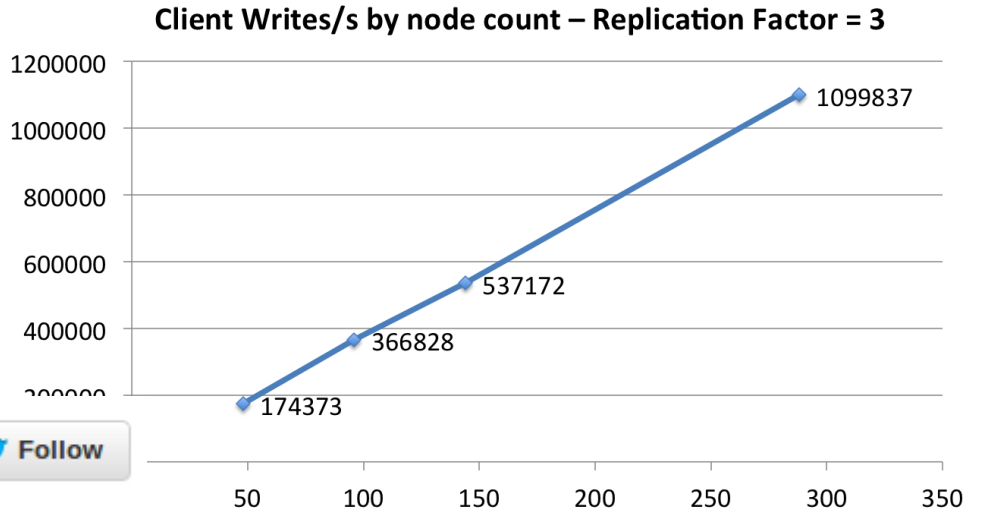
Berlin Buzzwords (June 4th 2013)

Sylvain Lebresne

DATASTAX

Apache Cassandra

- Fully Distributed Database
- Massively Scalable
- High performance
- Highly reliable/available



Nathan Milford
@NathanMilford

Follow

Man, I just love Cassandra. Lost a data center Hurricane Sandy, nodes came up and started working with no pain.

Bill de hÓra
@dehora

Follow

Coming to the conclusion that #cassandra is kind of indestructible. "Robust" doesn't do it justice.

Cassandra: the past

- Cassandra 0.7 (Jan 2011):
  - Dynamic schema creation
  - Expiring columns (TTL)
  - Secondary indexes
- Cassandra 0.8 (Jun 2011):
  - Counters
  - First version of CQL
  - Automatic memtable tuning
- Cassandra 1.0 (Oct 2011):
  - Compression
  - Leveled compaction
- Cassandra 1.1 (Apr 2012):
  - Row level isolation
  - Concurrent schema changes
  - Support for mixed SDD+HDD nodes
  - Self-tuning key/row caches

Cassandra: the present

Cassandra 1.2 (Jan 2013):

- Virtual nodes
- CQL3
- Native protocol
- Tracing
- ...

Data distribution without virtual nodes

Virtual nodes

Repairing without virtual nodes

Virtual nodes: repairing

Virtual nodes

- Not really "virtual nodes", more "multiple tokens per nodes" (but we still call them vnodes).
- Faster rebuilds.
- Allows heterogeneous nodes.
- Simpler load balancing when adding nodes.

The Cassandra Query language

- Initial version introduced in Cassandra 0.8. Version 3 (described here) is a major, more ambitious, revision.
- Goal: provide a much simpler, more abstracted user interface than the legacy thrift one.
- Kind of a "denormalized SQL".
- Strictly real-time oriented:
  - No joins
  - No sub-queries
  - No aggregation/GROUP BY
  - Limited ORDER BY

Storing songs

CQL

```
CREATE TABLE songs (
    id uuid PRIMARY KEY,
    title text,
    artist text,
    album text,
    track int,
    tags set<text>
);

-- Atomic and isolated
INSERT INTO songs (id, title, artist, album, tags, track)
          VALUES (a3e64f8f..., 'La Grange', 'ZTop' 'Tres Hombres', {'blues', '1973'}, 3);

UPDATE songs SET artist='ZZTop' WHERE id=a3e64f8f...;
```

id	title	artist	album	tags	track
a3e64f8f...	La Grange	ZZTop	Tres Hombres	{ blues, 1973 }	3
8a172618...	Moving in Stereo	Fu Manchu	We Must Obey	{ covers, 2003 }	9

Playlists

CQL

```
CREATE TABLE playlists (
    user_id text,
    playlist_name text,
    title text,
    artist text,
    album text,
    song_id uuid,
    PRIMARY KEY ( (user_id, playlist_name) ,  title, album, artist )
);
```

user_id	playlist_name	title	artist	album	song_id
pcmanus	My list	La Grange	ZZTop	Tres Hombres	a3e64f8f...
pcmanus	My list	Moving in Stereo	Fu Manchu	We Must Obey	8a172618...
pcmanus	Other list	La Grange	ZZTop	Tres Hombres	a3e64f8f...
pcmanus	Other list	Outside Woman Blues	Back Door Slame	Roll Away	2b09185b...

Querying a Playlist

CQL

```
-- Songs in 'My list' with a title starting by 'b' or 'c'
SELECT * FROM playlists
        WHERE user_id = 'pcmanus'
          AND playlist_name = 'My list'
          AND title >= 'b' AND title < 'd';

-- 50 last songs in 'My list'
SELECT * FROM playlists
        WHERE user_id = 'pcmanus'
          AND playlist_name = 'My list'
        ORDER BY title DESC
        LIMIT 50;
```

Native protocol

Binary transport protocol for CQL3 (replace Thrift transport):

- Asynchronous (less connections)
- Server notifications for new nodes, schema changes, etc..
- Optimized for CQL3

See the Datastax Java Driver for a mature driver using this new protocol (https://github.com/datastax/java-driver).

Request tracing

CQL

```
cqlsh:foo> TRACING ON;
cqlsh:foo> INSERT INTO bar (i, j) VALUES (6, 2);
```

```
activity                                  | timestamp    | source    | elapsed
------------------------------------------+--------------+-----------+---------
Determining replicas for mutation         | 00:02:37,015 | 127.0.0.1 |     540
Sending message to /127.0.0.2             | 00:02:37,015 | 127.0.0.1 |     779
Message received from /127.0.0.1          | 00:02:37,016 | 127.0.0.2 |      63
Applying mutation                         | 00:02:37,016 | 127.0.0.2 |     220
Acquiring switchLock                      | 00:02:37,016 | 127.0.0.2 |     250
Appending to commitlog                    | 00:02:37,016 | 127.0.0.2 |     277
Adding to memtable                        | 00:02:37,016 | 127.0.0.2 |     378
Enqueuing response to /127.0.0.1          | 00:02:37,016 | 127.0.0.2 |     710
Sending message to /127.0.0.1             | 00:02:37,016 | 127.0.0.2 |     888
Message received from /127.0.0.2          | 00:02:37,017 | 127.0.0.1 |    2334
Processing response from /127.0.0.2       | 00:02:37,017 | 127.0.0.1 |    2550
```

Tracing an anti-pattern

CQL

```
CREATE TABLE queues (
    id text,
    created_at timestamp,
    value blob,
    PRIMARY KEY (id, created_at)
);
```

id	created_at	value
my_queue	1399121331	0x9b0450d30de9
my_queue	1439051021	0xfc7aee5f6a66
my_queue	1440134565	0x668fdb3a2196
my_queue	1445219887	0xdaf420a01c09
my_queue	1479138491	0x3241ad893ff0

Tracing an anti-pattern

```CQL
cqlsh:foo> TRACING ON;
cqlsh:foo> SELECT FROM queues WHERE id = 'myqueue' ORDER BY created_at LIMIT 1;
```

```
activity                                        | timestamp    | source    | elapsed
------------------------------------------------+--------------+-----------+---------
execute_cql3_query                              | 19:31:05,650 | 127.0.0.1 |       0
Sending message to /127.0.0.3                   | 19:31:05,651 | 127.0.0.1 |     541
Message received from /127.0.0.1                | 19:31:05,651 | 127.0.0.3 |      39
Executing single-partition query                | 19:31:05,652 | 127.0.0.3 |     943
Acquiring sstable references                    | 19:31:05,652 | 127.0.0.3 |     973
Merging memtable contents                       | 19:31:05,652 | 127.0.0.3 |    1020
Merging data from memtables and sstables        | 19:31:05,652 | 127.0.0.3 |    1081
Read 1 live cells and 100000 tombstoned         | 19:31:05,686 | 127.0.0.3 |   35072
Enqueuing response to /127.0.0.1                | 19:31:05,687 | 127.0.0.3 |   35220
Sending message to /127.0.0.1                   | 19:31:05,687 | 127.0.0.3 |   35314
Message received from /127.0.0.3                | 19:31:05,687 | 127.0.0.1 |   36908
Processing response from /127.0.0.3             | 19:31:05,688 | 127.0.0.1 |   37650
Request complete                                | 19:31:05,688 | 127.0.0.1 |   38047
```

But also ...

- Concurrent schema creation
- Improved JBOD support
- Off-heap bloom filters and compression metadata
- Faster (murmur3 based) partitioner
- ...

What's next?

Cassandra 2.0 is scheduled for July:

- Improvements to CQL3 and the native protocol (automatic query paging)
- Compare-and-swap

```CQL
UPDATE users SET login='pcmanus', name='Sylvain Lebresne' IF NOT EXISTS
UPDATE users SET email='sylvain@datastax.com' IF email='slebresne@datastax.com';
```

- Triggers (experimental)
- Eager retries
- Performance improvements (single-pass compaction, more efficient tombstone removal, ...)
- ...

<Thank You!>

Questions?

www http://cassandra.apache.org/

twitter @pcmanus
github github.com/pcmanus


DATASTAX