Cassandra Administration for MySQL DBAs

Emily Slocombe
Who am I?

- on opensource database team within infrastructure engineering at SurveyMonkey
- based in Portland, Oregon
Oh, there are some cassandra databases, they are now yours.
what this talk is about:

- cqlsh>
- consistency
- complaints, user
- replication
what this talk is not about:

- jmx issues
- schema design
- query performance
- monitoring (just use datadog or prometheus)
- ♡ of cassandra
first, docs
first, docs

- we are running:

```
eslocombe@cqlsh> show version
[cqlsh 5.0.1 | Cassandra 3.5 | CQL spec 3.4.0
Native protocol v4]
```
first, docs

- we are running:
  
eslocombe@cqlsh> show version
  [cqlsh 5.0.1 | Cassandra 3.5 | CQL spec 3.4.0
  Native protocol v4]

- Apache Cassandra 3.0
docs

- [https://docs.datastax.com](https://docs.datastax.com)
- [https://marc.info/?l=cassandra-user](https://marc.info/?l=cassandra-user) (best for the biggest problems I've had)
- a very idle freenode #cassandra
definitions:

- keyspace
- node
- seednode
- vnode
definitions:

- keyspace ~ database
- node ~ the server with a cassandra daemon
- seednode ~ host(s) which has the gossip about the cluster topology for startup
- vnode ~ part of hashing strategy
cqlsh>

ok, how do I connect?
cqlsh>

ok, how do I connect?

eslocombe@p9cstar200yolo1:~$ cqlsh
cqlsh>

ok, how do I connect?

eslocombe@p9cstar200yolo1:~$ cqlsh

Connection error: ('Unable to connect to any servers',
{'127.0.0.1': error(111, "Tried connecting to
[('127.0.0.1', 9042)].
Last error: Connection refused")})
our cqlshrc template

# ~/.cassandra/cqlshrc

[authentication]
username = yourcassusername
password = yourcasspassword

[connection]
hostname = thedefaulthost
request_timeout = 40
basic cqlsh> commands:

describe keyspaces;
describe keyspace nameofkeyspace;
list roles;
list all of rolename;

bash$ cqlsh < bunch-of-things.cql > results.txt
basic cqlsh> commands:

eslocombe@cqlsh> describe keyspaces;

system_schema  system                      system_traces
system_auth    emilytest                  system_distributed
basic cqlsh> commands:

eslocombe@cqlsh> describe keyspace system_distributed;

CREATE KEYSPACE system_distributed
  WITH replication =
  {'class': 'SimpleStrategy',
   'replication_factor': '3'}
  AND durable_writes = true;

CREATE TABLE system_distributed.repair_history (  
  keyspace_name text,
  columnfamily_name text,
  id timeuuid,
  ...
  ...
basic cqlsh> commands:

```
eslocombe@cqlsh> list roles;

<table>
<thead>
<tr>
<th>role</th>
<th>super</th>
<th>login</th>
<th>options</th>
</tr>
</thead>
<tbody>
<tr>
<td>blahsvc</td>
<td>False</td>
<td>True</td>
<td>{}</td>
</tr>
<tr>
<td>dba00</td>
<td>True</td>
<td>True</td>
<td>{}</td>
</tr>
<tr>
<td>dba01</td>
<td>True</td>
<td>True</td>
<td>{}</td>
</tr>
<tr>
<td>dba02</td>
<td>True</td>
<td>True</td>
<td>{}</td>
</tr>
<tr>
<td>eslocombe</td>
<td>True</td>
<td>True</td>
<td>{}</td>
</tr>
<tr>
<td>otherrole</td>
<td>False</td>
<td>False</td>
<td>{}</td>
</tr>
</tbody>
</table>

(6 rows)
```
basic cqlsh> commands:

eslocombe@cqlsh> list all of eslocombe;
cqlsh>, but also nodetool

- nodetool, without and with jmx auth (not default to need a pw)

```bash
nodetool status

nodetool -u cassandra -pwf /etc/super-secrets/jmxremote.password 
repair -full -pr
```

- **nodetool** can do a lot
consistency
consistency

(check the cluster status for 'UN')
cqlsh> consistency;
Current consistency level is ONE.
consistency

(set session consistency and run your command)

cqlsh> CONSISTENCY ALL;
cqlsh> use keyspacename;
cqlsh> truncate table tablename;
complaints (user, not mine)
complaints (user, not mine)

same as MySQL!
complaints (user, not mine)

same as MySQL! *timeouts.*
complaints (user, not mine)

timeouts

- session consistency level?
- request_timeout in client (10s)
- indexing / design
complaints (mine, minor)

- grants.
complaints (mine, minor)

- grants.
- one permission per grant statement
complaints (mine, minor)

- grants.
- one permission per grant statement

```sql
CREATE ROLE emily WITH PASSWORD = 'trolololo'
AND LOGIN = true;
GRANT SELECT ON KEYSPACE em TO emily;
GRANT ALTER ON KEYSPACE em TO emily;
...
```
complaints (mine, minor)

- grants.
- but cassandra already uses roles, which is nice
complaints (mine, minor)

- grants.
- but cassandra already uses roles, which is nice

```sql
CREATE ROLE emsteamof1;
GRANT MODIFY ON KEYSPACE em TO emsteamof1;
GRANT emsteamof1 to emily;
```

- not bad, but it can get ugly
replication
replication factors, and consistency
replication

factors, and consistency
Replication is at the *keyspace* level
replication, and consistency

- two choices
  - SimpleStrategy
  - NetworkTopologyStrategy

```sql
CREATE KEYSPACE nameofkeyspace WITH replication = 
  {
    'class': 'NetworkTopologyStrategy',
    'dcname': '3'
  }
AND durable_writes = true;
```
replication, and consistency

friendly suggestion: install Spotify Reaper for your cluster

- automates `nodetool repair` with more control
bonus: complaints

my number one complaint: tooling
bonus: complaints

my number one complaint: tooling

- Spotify github
- TLP github
bonus: complaints

my second complaint: backups and data migration

COPY em.geo_ip_lookup (class_b, startip, city, \ country, endip, latitude, longitude, region) \ TO '/tmp/em_geo_ip_lookup.csv';

COPY em.geo_ip_lookup (class_b, startip, city, \ country, endip, latitude, longitude, region) \ FROM '/tmp/em_geo_ip_lookup.csv';

• doesn't work for certain datatypes
bonus: complaints

my second complaint: backups and data migration

```
sstableloader -cph 4 -u eslocombe \
-pw ${pw} -d $(hostname -s) ${keyspace}
```
tl;dr

- stop trying to make it work like MySQL
- accept your fate & do your best
- standardize everything
- graph something

Slides with notes:
https://eslocombe.github.io/cass-pldub/cassandra-for-mysql-dbas.html

hit 'p' for notes

thank you!