Automating Cassandra Repairs

Radovan Zvoncek

zvo@spotify.com

github.com/spotify/cassandra-reaper

About zvo

About zvo

Likes pancakes

About zvo

Likes pancakes

Does this for the first time

About zvo

Likes pancakes

Does this for the first time

Works at Spotify

Working at Spotify

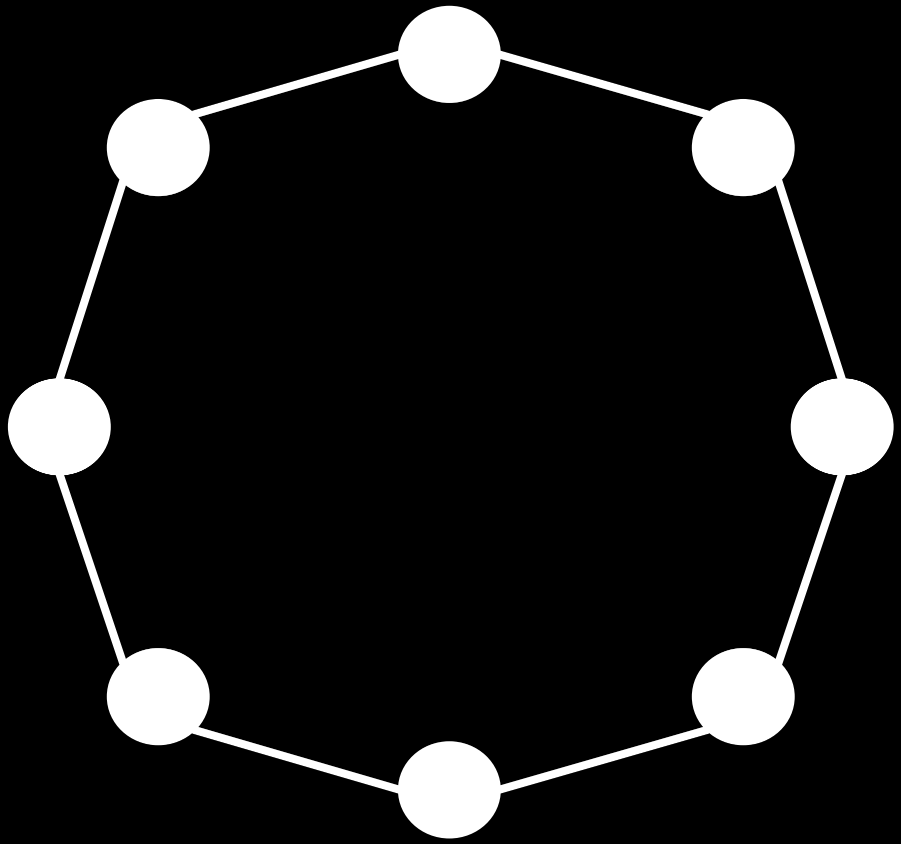
Is autonomous

Squads run their own stuff

...

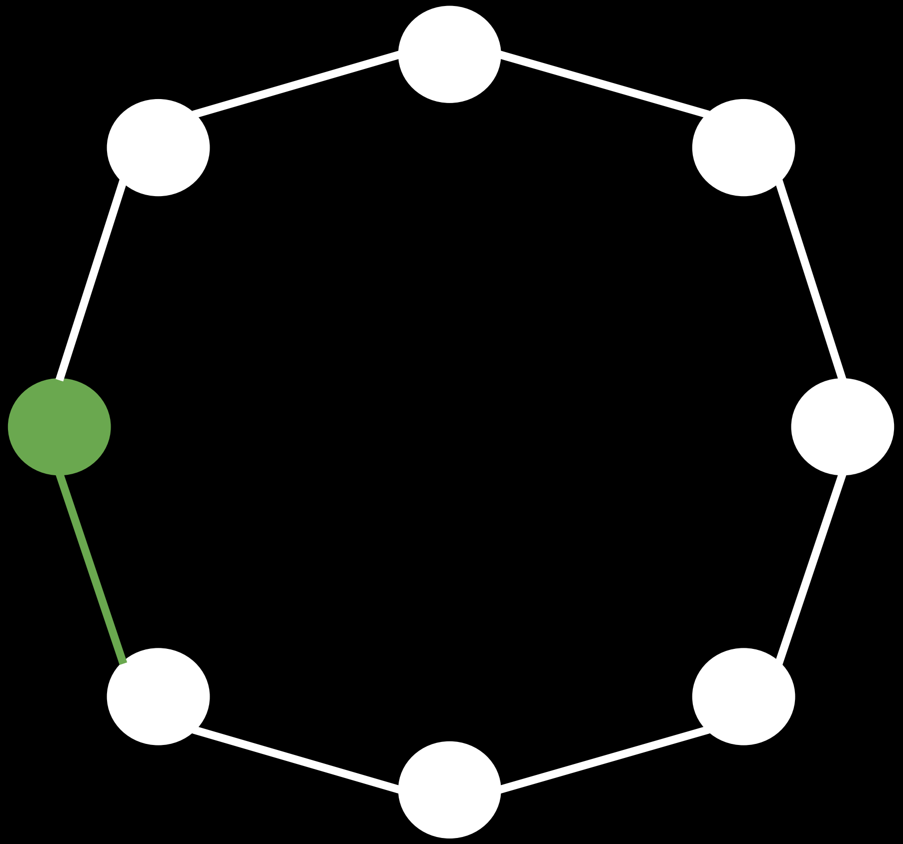
Including Cassandra

Cassandra



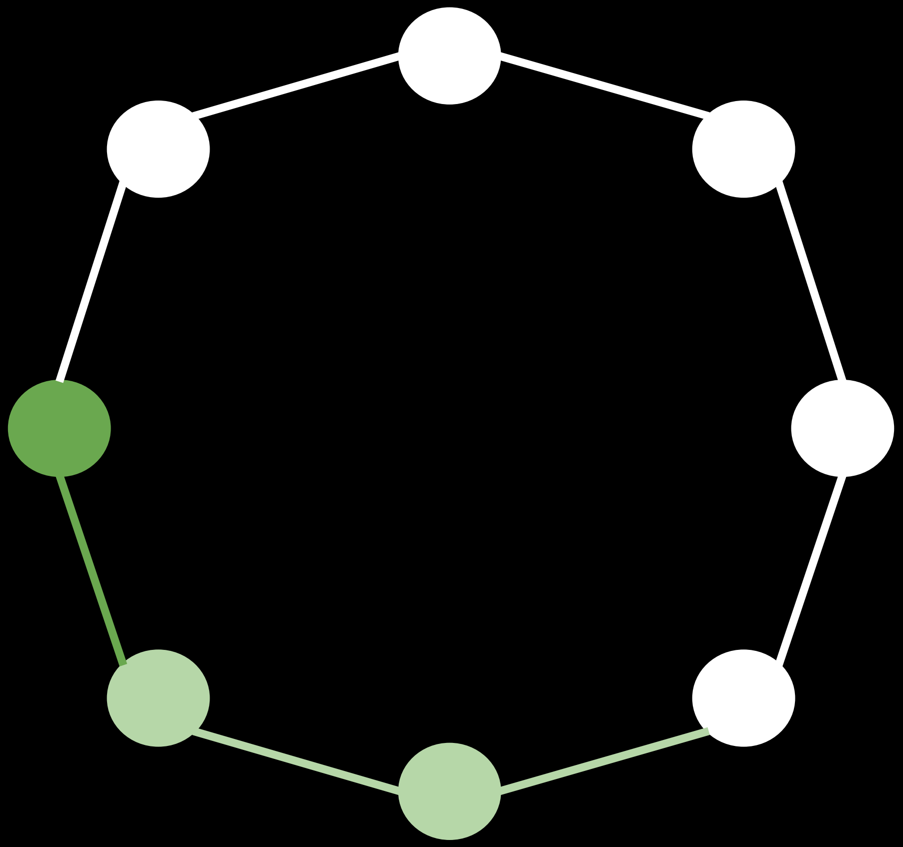
Cassandra

Node’s data



Cassandra

Replication



Running Cassandra

Requires many things

One of them is keeping data consistent…

Running Cassandra

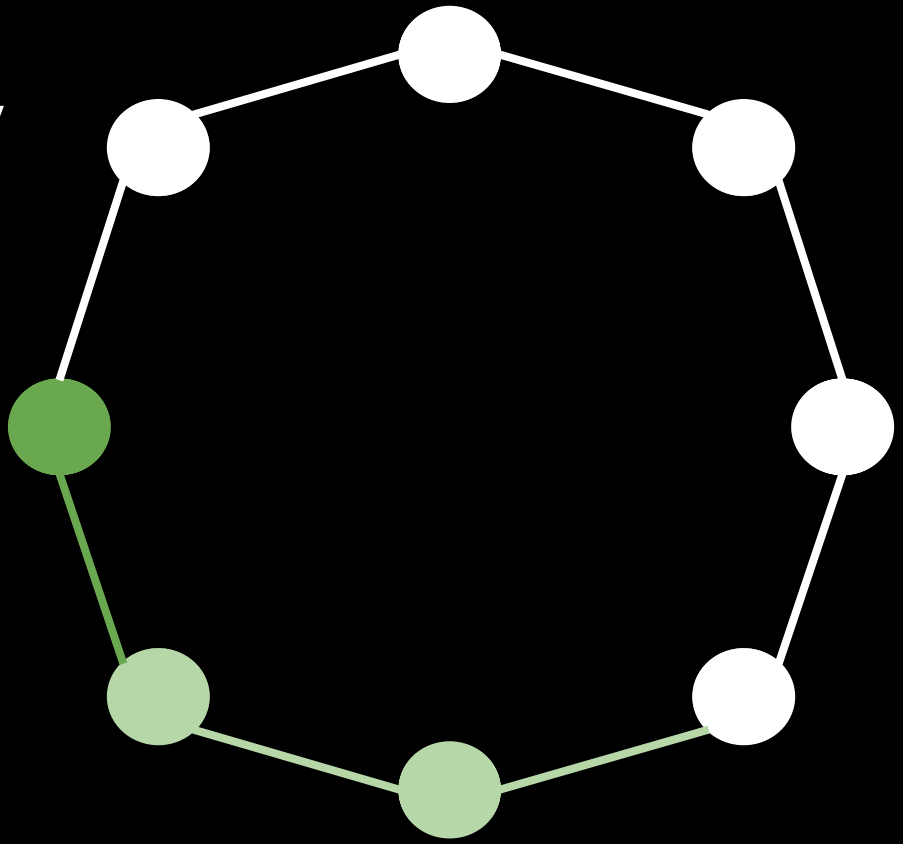
Requires many things

One of them is keeping data consistent…

Eventually

Cassandra

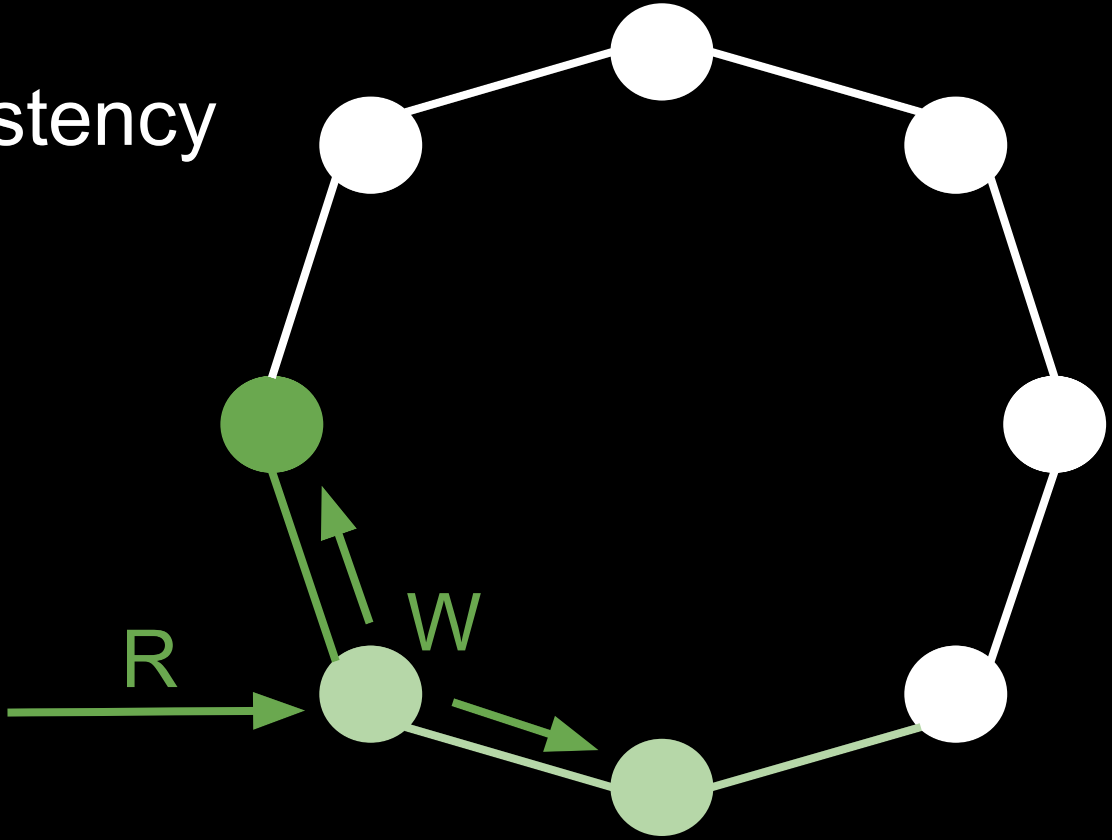
Eventual consistency



Cassandra

Eventual consistency

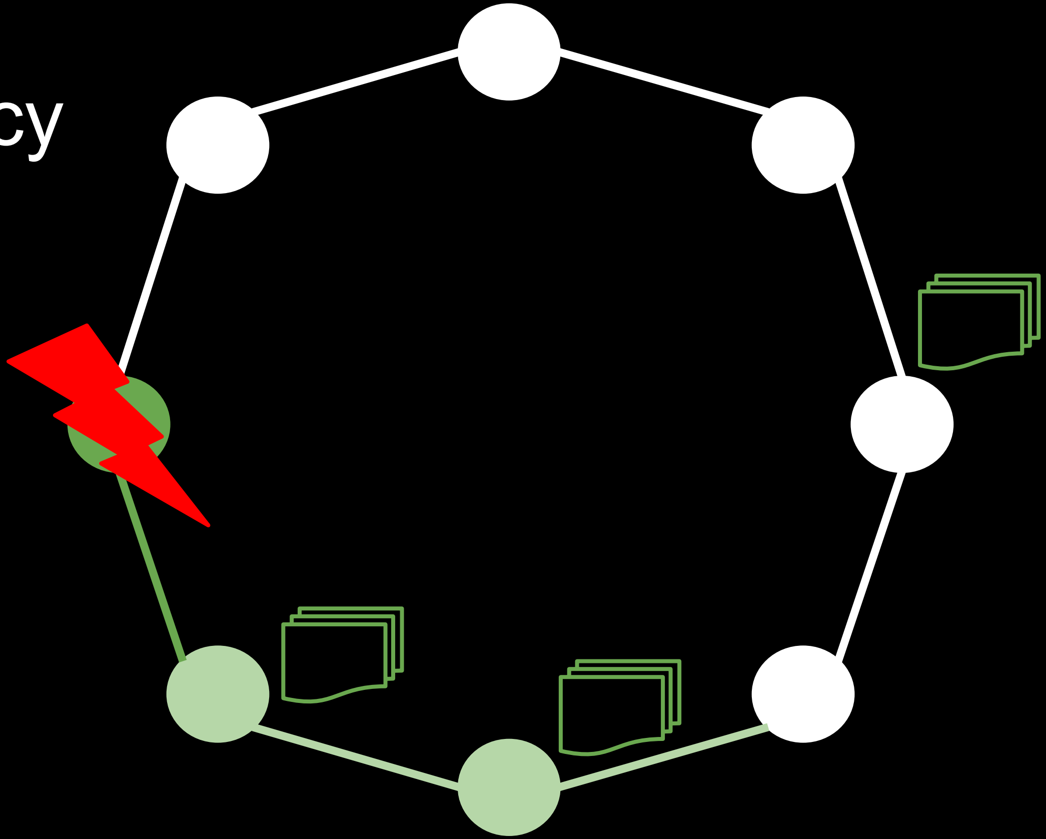
Read Repairs



Cassandra

Eventual consistency

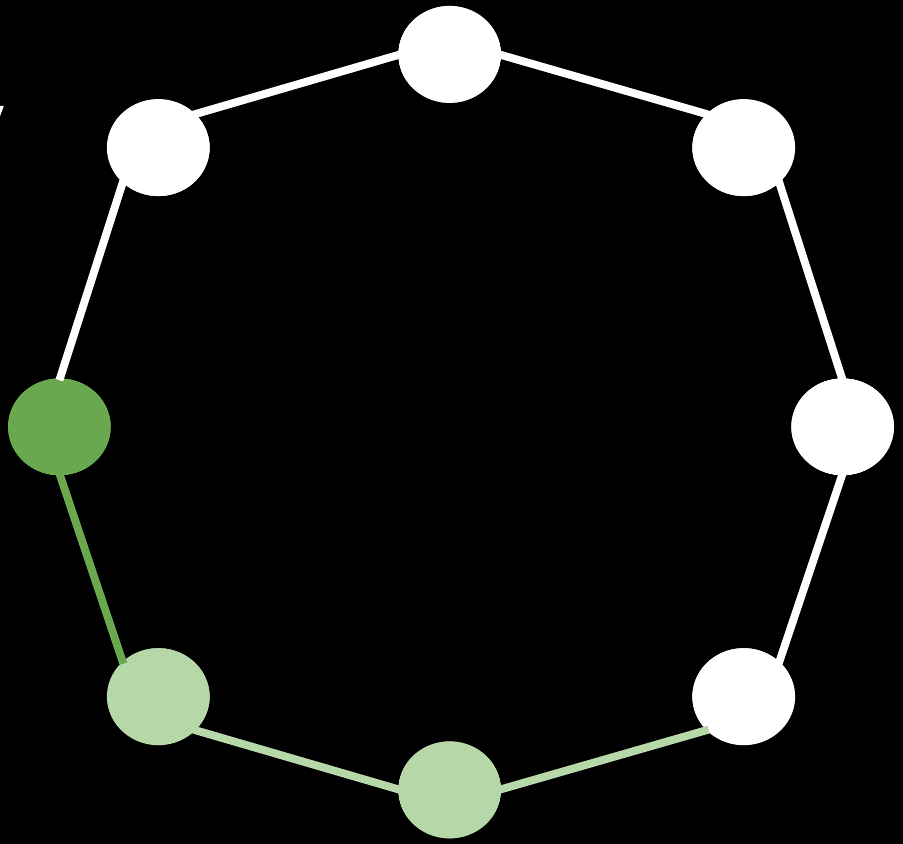
Hinted Handoff



Cassandra

Eventual consistency

Anti-entropy Repair



Anti-entropy Repair

It's a coordinated process of ~~two~~ four steps

- Step 1: compute hashes of data
- Step 2: gather and compare the hashes
- Step 3: stream data around
- Step 3b: deal with the incoming data

Anti-entropy Repair

It’s a coordinated process of ~~two~~ four steps

- Step 1: compute hashes of data
- Step 2: gather and compare the hashes
- Step 3: stream data around
- Step 3b: deal with the incoming data

Repair can go wild...

Repair gone wild

Eats a lot of disk IO

- because of hashing all the data

Repair gone wild

Eats a lot of disk IO

Saturates the network

- because of streaming a lot of the data around

Repair gone wild

Eats a lot of disk IO

Saturates the network

Fills up the disk

- because of receiving all replicas, possibly from all other data centers

Repair gone wild

Eats a lot of disk IO

Saturates the network

Fills up the disk

Causes a ton of compactions

- because of having to merge the received data

Repair gone wild

Eats a lot of disk IO

Saturates the network

Fills up the disk

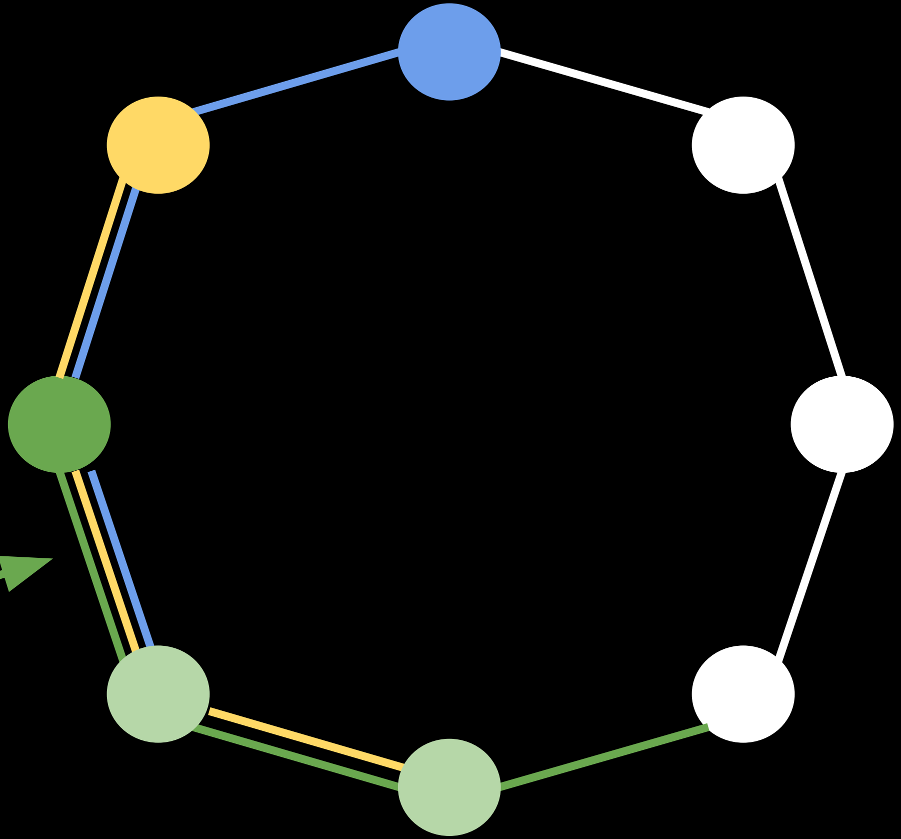
Causes a ton of compactions

… one better is careful

Careful repair

Primary range

- nodetool repair -pr

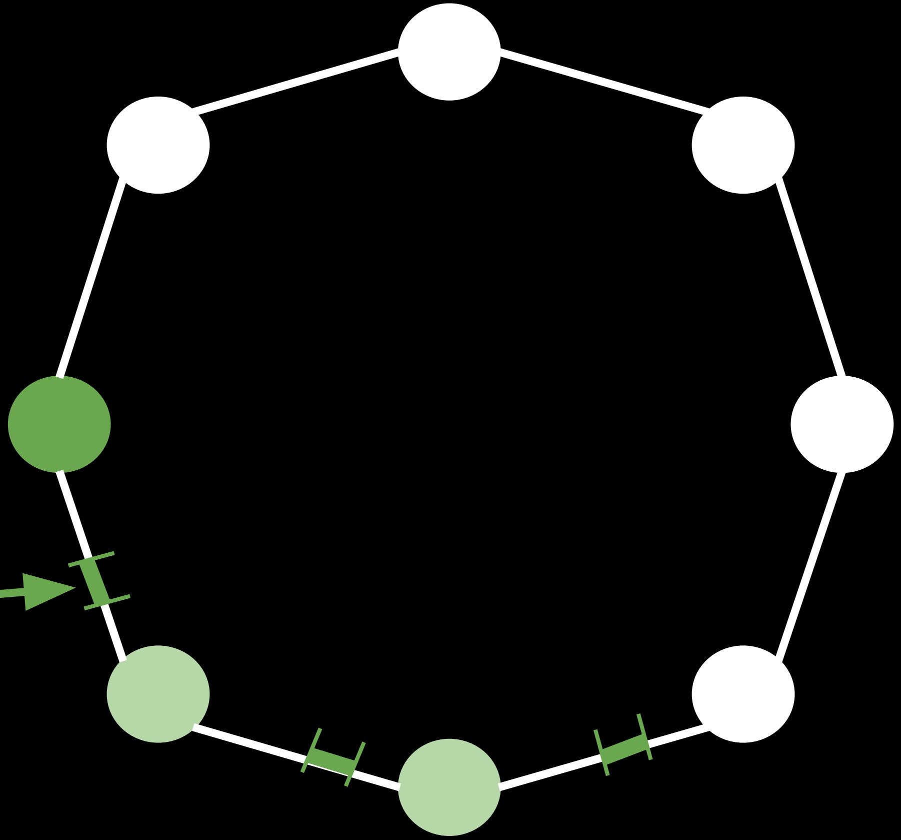


This interval only

Careful repair

Start & end tokens

- nodetool repair -st -et



A part of that interval only

Careful repair

Requires splitting the ring into smaller chunks

Smaller chunks mean less data

Less data means less repairs gone wild

Careful repair

Smaller chunks also mean more chunks

More chunks mean more actual repairs

Repairs need to be babysitted :(

The Spotify way

Feature teams build & run features

Nobody to operate their systems

Cronning repairs is no good

- mostly due to no feedback loop

This all led to creation of the Reaper

The Reaper

REST(ish) service written in java

Does a lot of JMX

Orchestrates repairs for you

The reaping

You do:

```
curl http://reaper/cluster --data ‘{“seedHost” : “my.cassandra.host.net”}’
curl http://reaper/repair_run --data ‘{“clusterName”: “myCluster”}’
curl -X PUT http://reaper/repair_run/42 -d state=RUNNING
```

The Reaper does:

- Figures out cluster topology
- Splits the ring
- Orchestrates the partial repairs
- Makes you happy

Reaper’s features

Carefulness - doesn’t kill a node

Resilience - retries when things break

Parallelism - no idle nodes

Persistency - state saved somewhere

Scheduling - setup things only once

What we reaped

First repair done 2015-01-28

415 repairs since then, recently ~70 per week

28 repair failures

2,2M segment failures + postpones

Parallelism speed up 12 -> 2 days

Reaper’s Future

Changing cluster topology

Changing seed host

Future Cassandra versions

And few others, most likely

Contributions

About five to the Reaper itself

Standalone UI

https://github.com/spodkowinski/cassandra-reaper-ui

Greatest benefit

Cassandra Reaper automates a very tedious maintenance operation of Cassandra clusters in a rather smart, efficient and careful manner while requiring minimal Cassandra expertise

github.com/spotify/cassandra-reaper

Closing

Thanks for bearing me

Check out the Reaper

github.com/spotify/cassandra-reaper