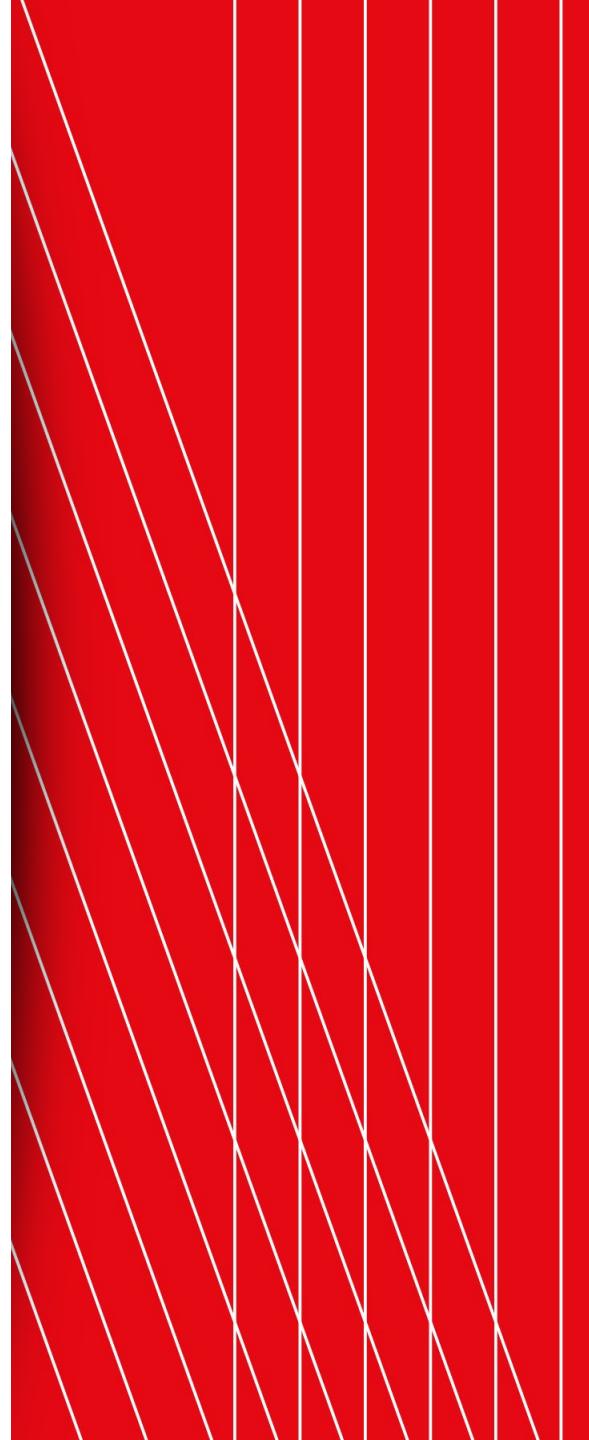


Apache C*

Sidecar

let's make C* attractive and easy to operate

Vinay Chella, Dinesh Joshi



Agenda

- Operating C*
- Operating C* with a sidecar
- State of community sidecars
- Lessons learned from operating
with sidecars
- Goals of C* management sidecar

Operating C*

- Bootstrap and data movement
- Configuration (files, jmx)
- Maintenance
- Monitoring/Metrics
- Backup/Restore
- Repair



Operating C*: Bootstrapping

Create a New Cluster

- Seeds
- Token assignment



Add/Remove/Replace

- Serial or parallel?
- Streaming?

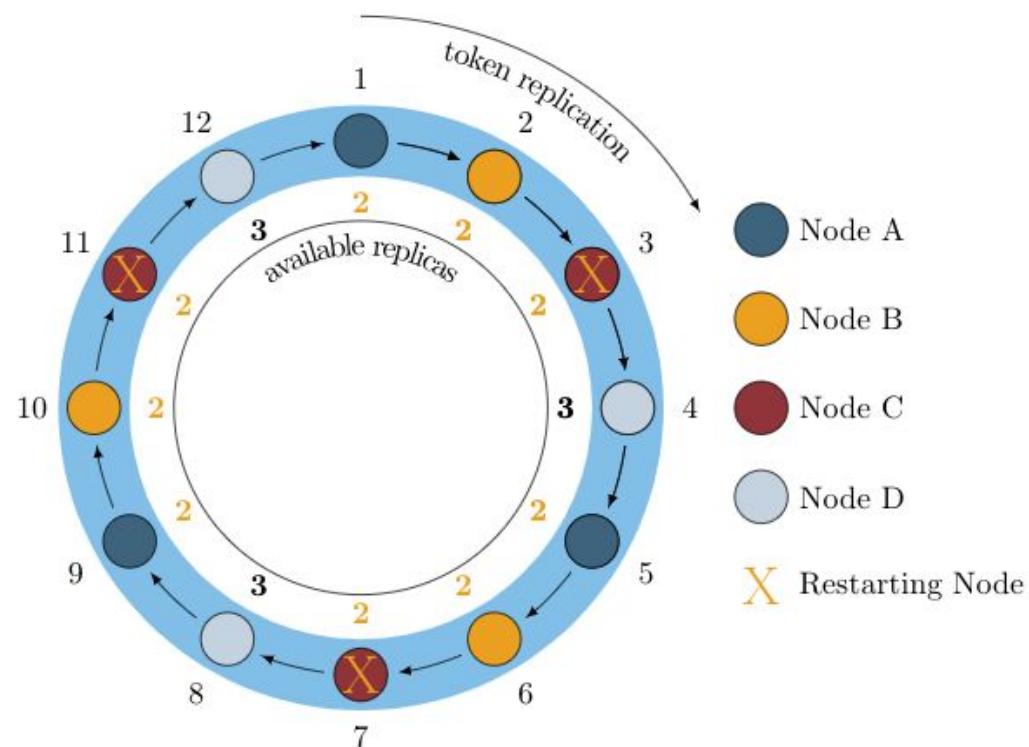


Operating C*: Configuration

```
..  
cassandra-env.ps1  
cassandra-env.sh  
cassandra-rackdc.properties  
cassandra-topology.properties  
cassandra.yaml  
commitlog_archiving.properties  
cqlshrc.sample  
hotspot compiler  
jvm.options  
logback-tools.xml  
logback.xml  
metrics-reporter-config-sample.yaml  
README.txt  
triggers
```

- Probably Have to Tune
 - a. cassandra.yaml
 - b. topology props
 - c. JVM options
- May Have to Tune
 - a. Logging
 - b. Incremental Backup
 - c. More JVM options

Operating C*: Lifecycle



Rolling Restarts (Upgrades)

- Semi-complex single node procedure
- One at a time is too slow
- Token range aware restarts?

What happens when Cassandra dies?

Operating C*: Maintenance

```
~ $ jmxterm  
Welcome to JMX terminal. Type "h"  
$>open localhost:  
#Connection to localhost: [REDACTED] is  
$>domains  
#following domains are available  
JMImplementation  
ch.qos.logback.classic  
com.sun.management  
java.lang  
java.nio  
java.util.logging  
org.apache.cassandra.auth  
org.apache.cassandra.db  
org.apache.cassandra.hints  
org.apache.cassandra.internal  
org.apache.cassandra.metrics  
org.apache.cassandra.net  
org.apache.cassandra.request  
org.apache.cassandra.service
```

```
$>bean [REDACTED] Display all 4258 possibilities? (y or n)
```

- All the Power of JMX
- ... So many possibilities
 - a. Many work with jmxterm/jmxsh
 - b. Many only work with Java code
- What if you want to do it on all nodes?

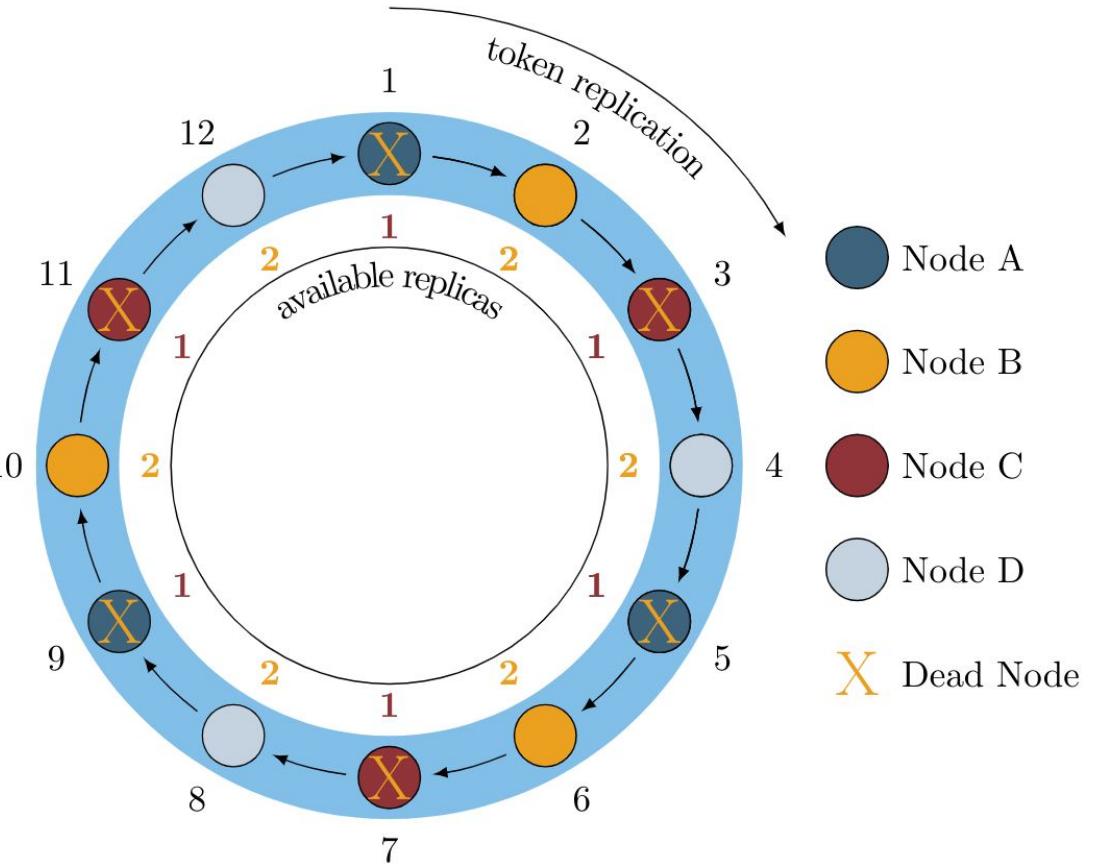
Operating C*: Monitoring

- Many Metrics (good!)
- How to Collect Them?
 - JMX ... no
 - Agent!
- Which agent ...



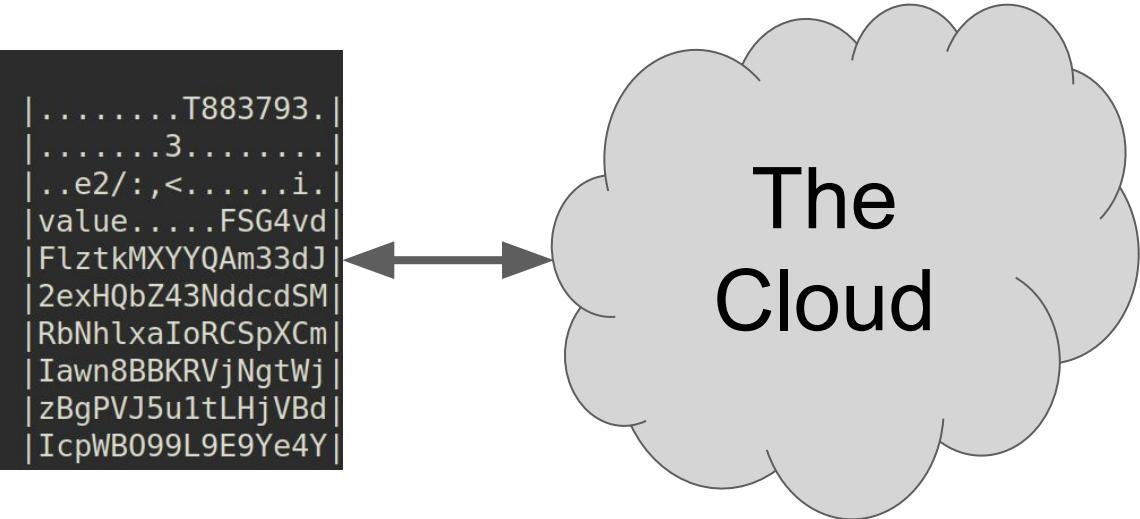
Operating C*: Ring Health

- Cassandra ring health depends on replication
- Strategies
 - Monitor replication of keyspaces
 - **Topology Aware**
 - **Maintenance Aware**



Operating C*: Backup/Restore

```
$> hd pappyperf-test1-ka-3609-Data.db | head  
00000000 00 01 00 f3 00 00 07 54 38 38 33 37 39 33 7f  
00000008 1f ff 80 00 01 00 33 0a 00 04 0a 00 a1 00 00  
00000020 00 05 65 32 2f 3a 2c 3c 0c 00 14 0f 19 00 69 05  
00000030 76 61 6c 75 65 1e 00 f0 c4 c8 46 53 47 34 76 64  
00000040 45 00 7a 74 0b 0d 53 50 51 41 50 33 33 64 4a  
00000050 52 65 78 48 51 62 5a 34 33 4e 61 54 03 64 53 4d  
00000060 52 62 4e 68 6c 78 61 49 6f 52 43 53 70 58 43 6d  
00000070 49 61 77 6e 42 42 4b 52 56 6a 4e 67 74 57 6a  
00000080 7a 42 47 50 46 4a 45 45 81 74 4c 48 6a 56 42 64  
00000090 49 63 70 57 42 4f 39 39 4c 39 45 39 59 65 34 59  
DATA  
... and schemas  
... and tokens
```



- What even do I need to backup!?
- Restore is legitimately tricky, do you practice?

Operating C*: Repair

“Eventually” Consistent

1. Partial Write
2. Read Repair
3. Hints play
... Nope not enough
4. Repair

Datacenter 1			Datacenter 2		
N1	N2	N3	N4	N5	N6
0	1	0	0	0	0
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	0	1	0
1	1	1	1	1	1

Sidecar: Bootstrapping

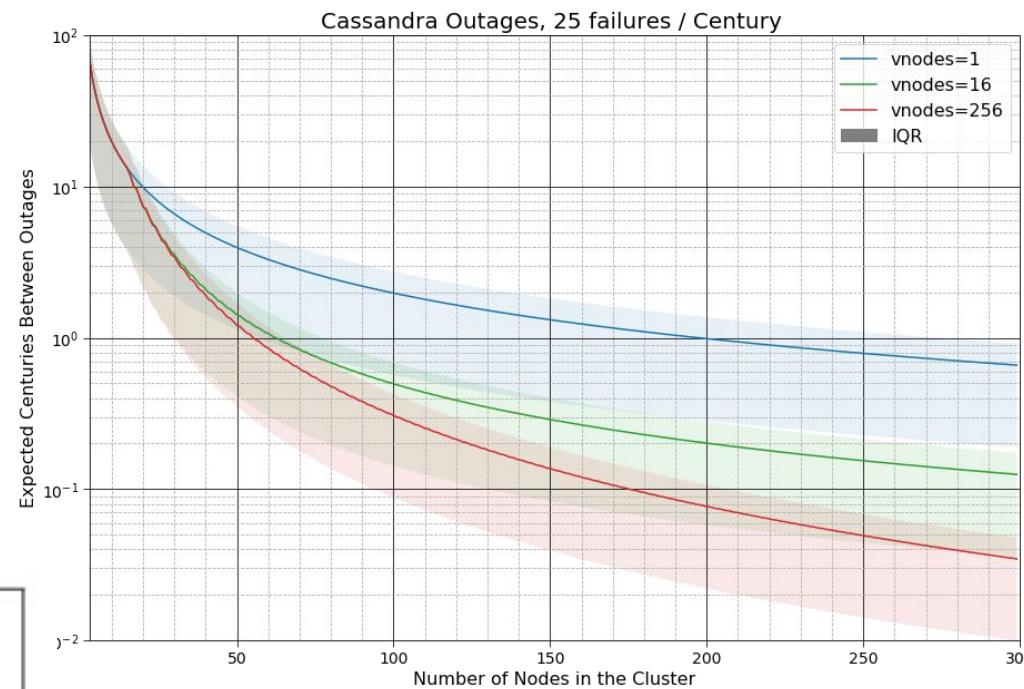
/Priam/REST/v1/cassconfig/get_seeds

Automatic Seed Management using ASGs/db

/Priam/REST/v1/cassconfig/get_replaced_ip

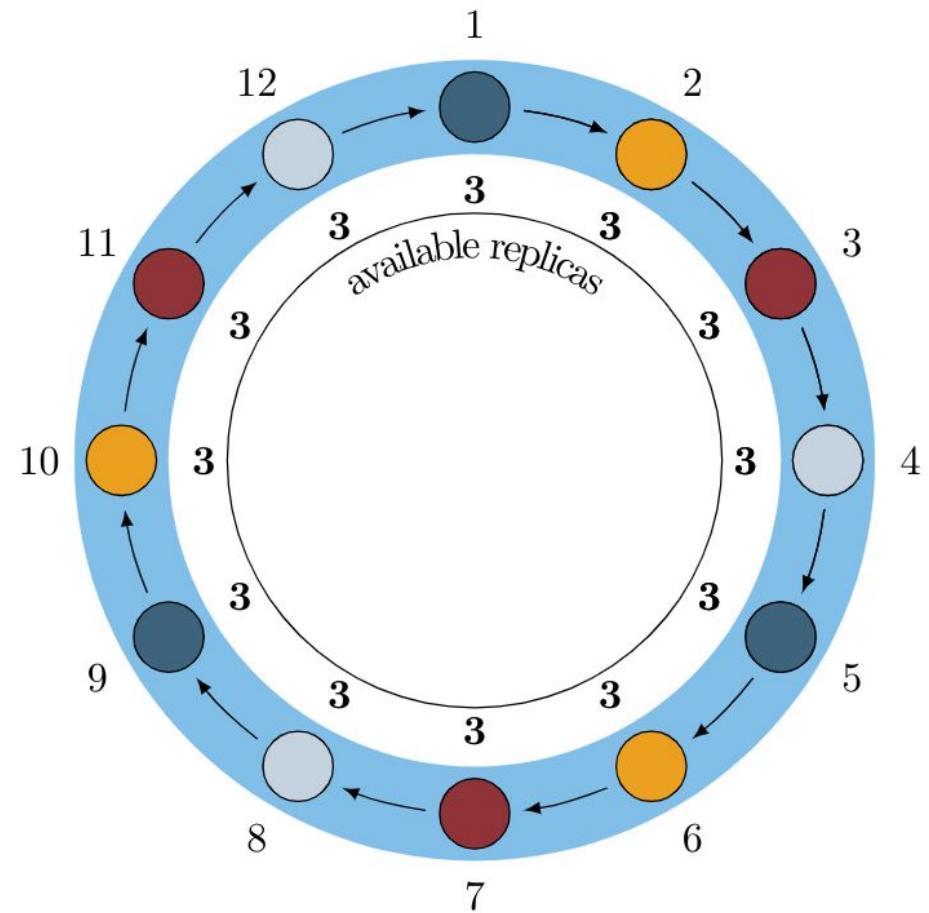
Automatic Instance Replacement

$$\begin{aligned} E[\text{outages}] &= \lambda_{\text{global}} \\ &= (n * \lambda_{\text{split}}) \\ &= (n * (\lambda * P(\text{outage}|\text{failure}))) \\ &= (n * (\lambda * (1 - e^{-E[T_{\text{recovery}}] * E[n_{\text{neighbors}}] * \lambda_s}))) \end{aligned}$$

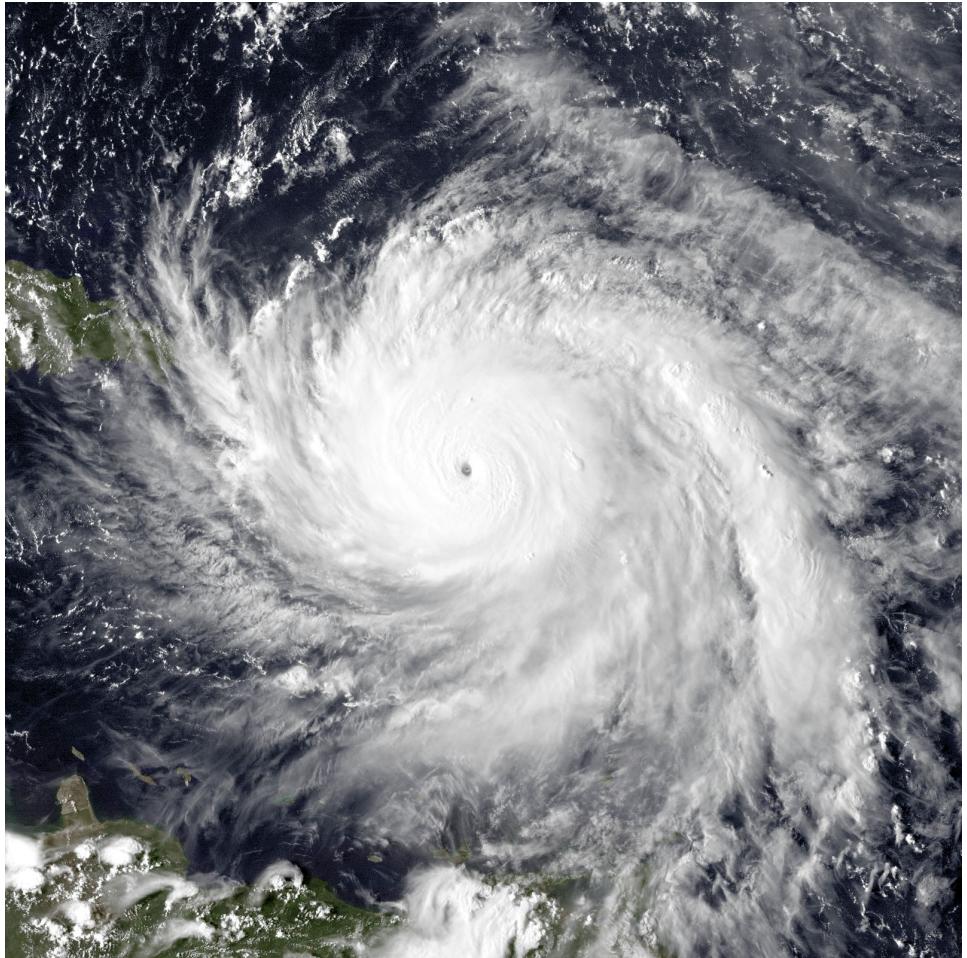
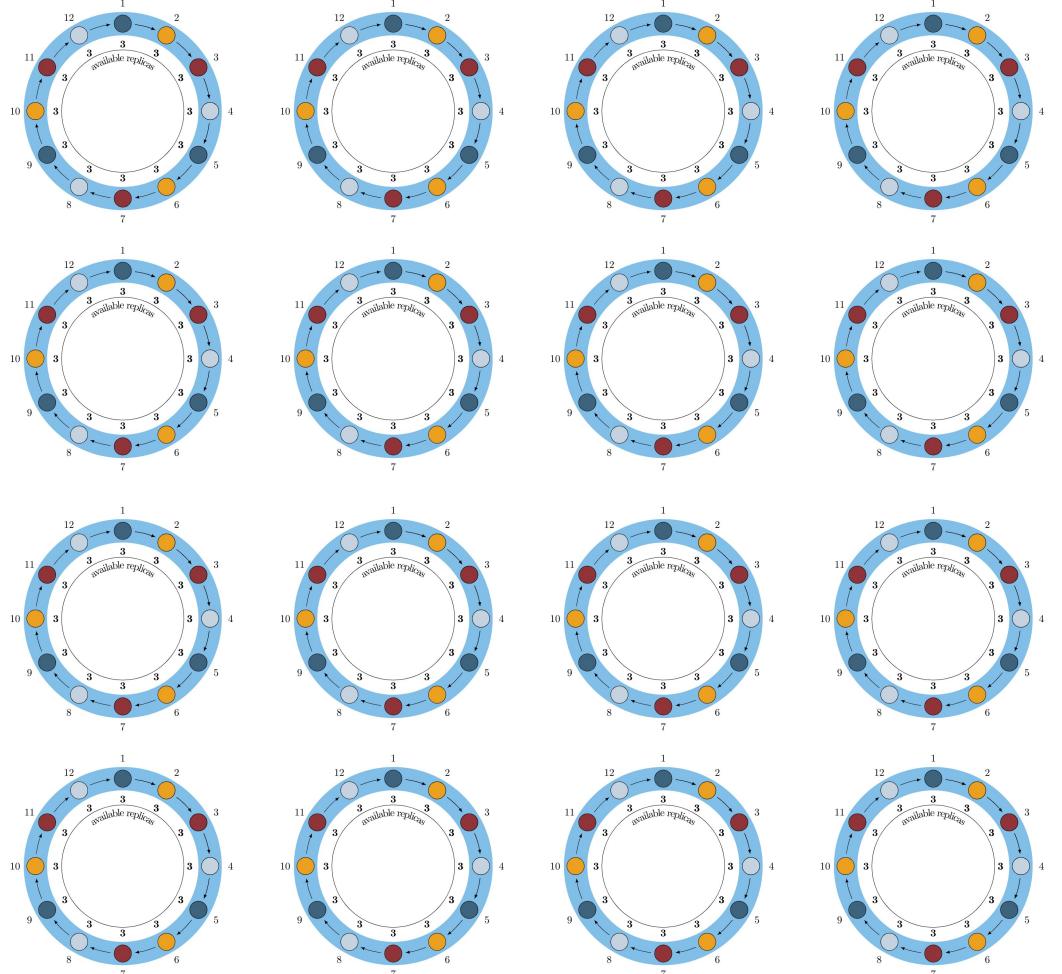


Equation+Graph from "Cassandra Availability with Virtual Nodes" by Joey Lynch and Josh Snyder

Operating C* In General



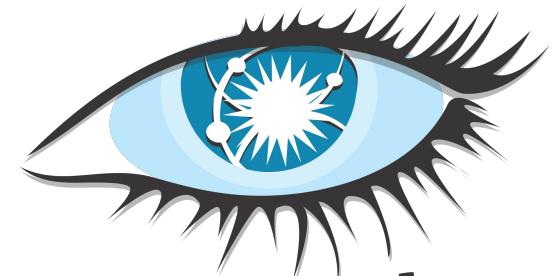
Operating C* In General



What is needed to Operate C*?

Separate solutions for ...

- Bootstrap and data movement
- Maintenance
- Configuration (files, jmx)
- Monitoring/Metrics
- Backup/Restore
- Repair



We need better tools!

Community needs

⊕ → karll_, Rapture, terraz, kvaster, benton and jlw joined ← kalaolani, kk, Ribeiro79, izmogikan and linkpaper quit ↔ hive-mind nipped out

B benton 24.217.29.219 - http://webchat.freenode.net

new to cassandra, is there a way for a ring to be created without specifying seed nodes in config for each? ex. spin up 3 ec2 instances with the same configs and they find each other?

⊕ ← jlw, ferraz, benton and nighty- quit

**Current state of the
art?**

CockroachDB

OVERVIEW

GRAPH: CLUSTER ▾ DASHBOARD: OVERVIEW ▾ LAST 10 MIN < >

SQL Queries

This chart displays the volume of four types of SQL queries over a 10-minute period. The Y-axis represents the number of queries from 0 to 400. The X-axis shows time from 17:34 to 17:43. Selects (dark blue) are the most frequent, followed by Updates (yellow), Inserts (red), and Deletes (light blue). All series show a general upward trend with some fluctuations.

Service Latency: SQL, 99th percentile

This chart tracks the 99th percentile service latency for SQL requests across multiple nodes. The Y-axis measures latency in seconds from 0 to 6. The X-axis covers the same 10-minute period as the other charts. Latency values fluctuate between approximately 1.5s and 6s, with a notable peak around 17:35.

Replicas per Node

This chart shows the distribution of replicas across nodes. The Y-axis represents the number of replicas from 1 to 4. The X-axis shows time from 17:34 to 17:43. Most nodes have 2 or 3 replicas, with one node consistently having 1 replica.

OVERVIEW DASHBOARD

GRAPH: CLUSTER ▾ DASHBOARD: OVERVIEW ▾ LAST 10 MIN < >

SQL Queries

Summary

Total Nodes [View nodes list](#) 6

Capacity Used 22.43%
You are using 86.6 GiB of 386.1 GiB usable storage capacity across all nodes.

Unavailable ranges 0

Queries per second 1102.1

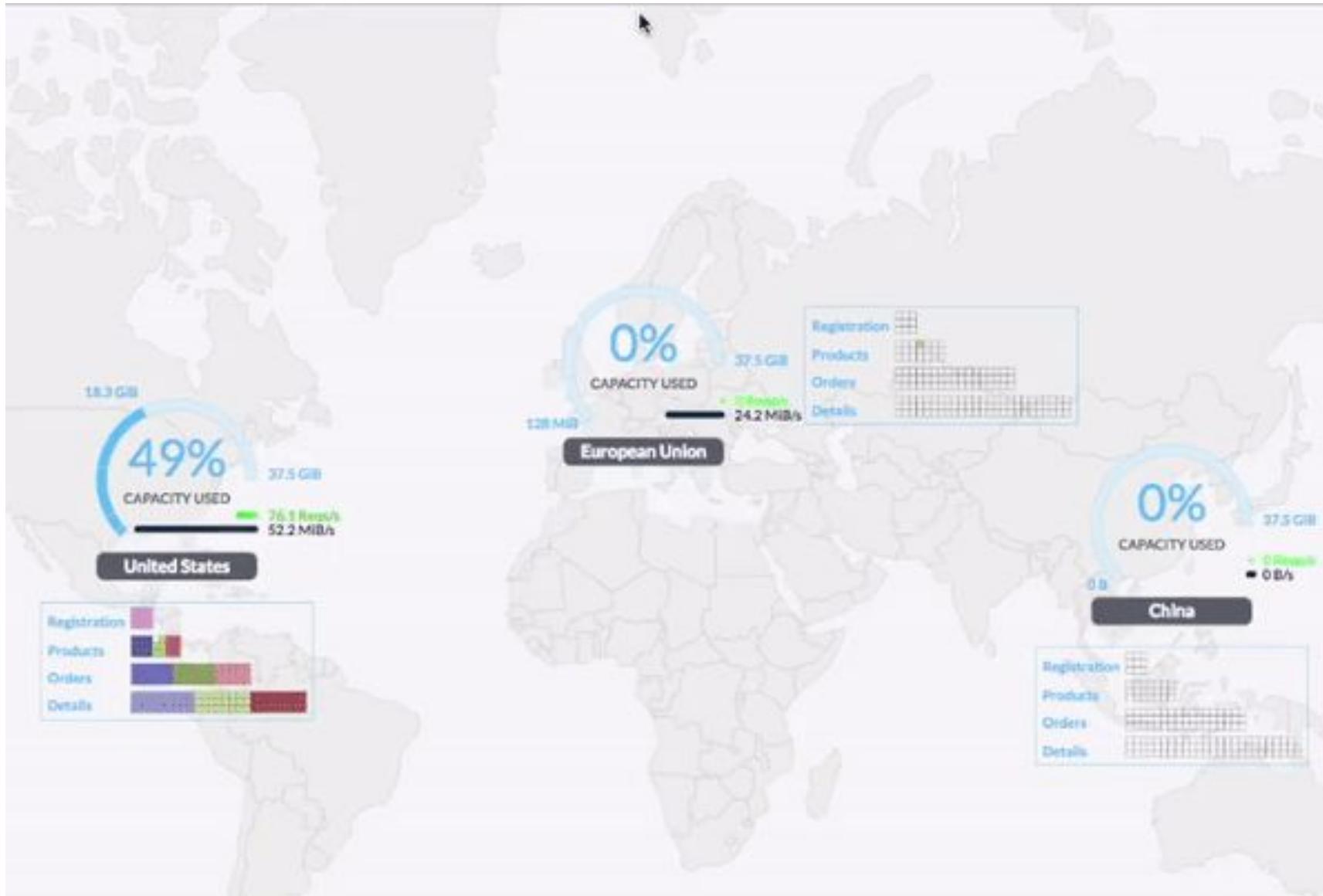
P50 latency 9.4 ms

P99 latency 4026.5 ms

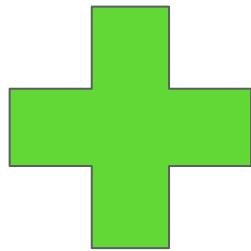
Events

Cluster Setting Changed: U... a day ago

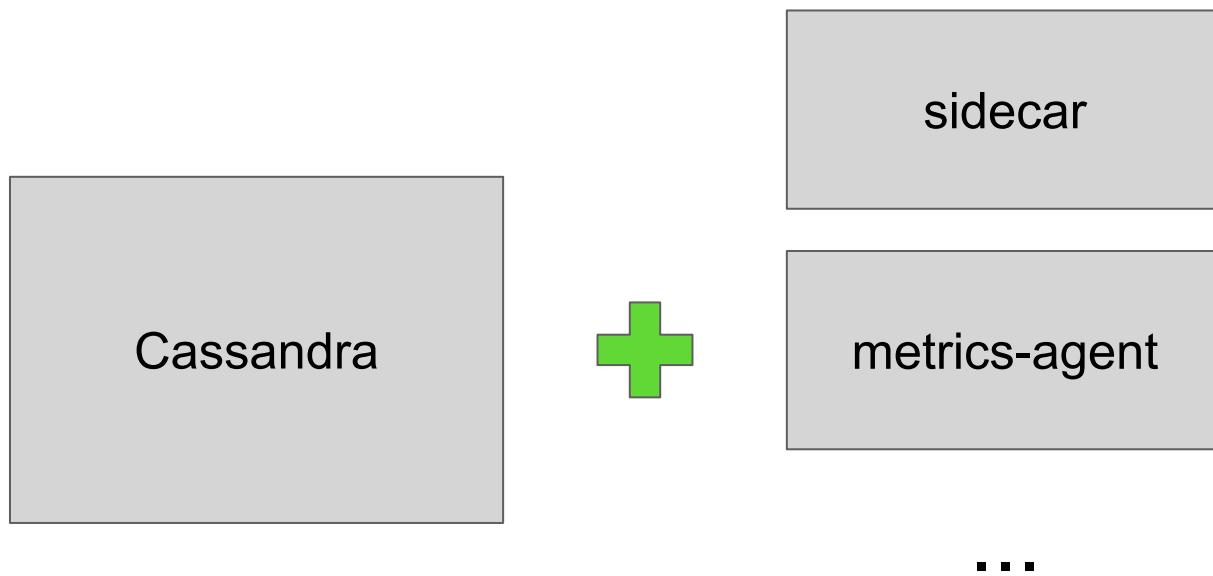
CockroachDB



Operating C* with Sidecar(s)



What's a Sidecar?



Sidecars Live Outside
Main Daemon Scope

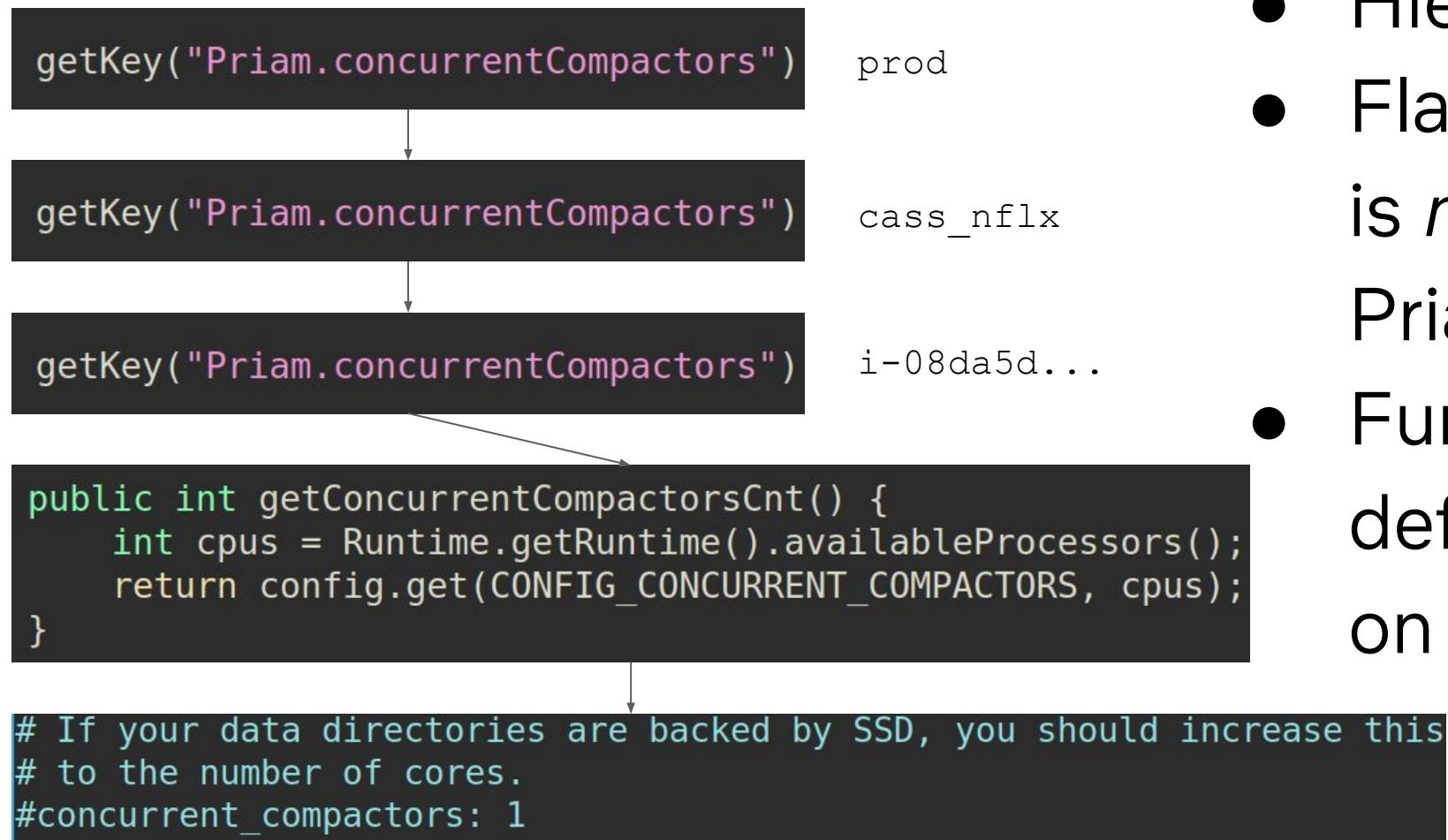
- Often built for a specific purpose
- Typically a different OS process

Sidecar: Configuration

The screenshot shows the GitHub repository page for Netflix/archaius. The repository name is at the top left. At the top right are buttons for Unwatch (460), Star (1,768), and Fork (385). Below the header are navigation links: Code (selected), Issues (53), Pull requests (25), Projects (0), Wiki, and Insights. A summary bar below the navigation shows 541 commits, 9 branches, 124 releases, 33 contributors, and Apache-2.0 license. At the bottom are buttons for Branch: 2.x, View #405, Create new file, Upload files, Find file, and Clone or download.

- Hierarchy: Environment -> Cluster -> Node
- Flat namespace that is *merged* to provide Priam config

Sidecar: Configuration

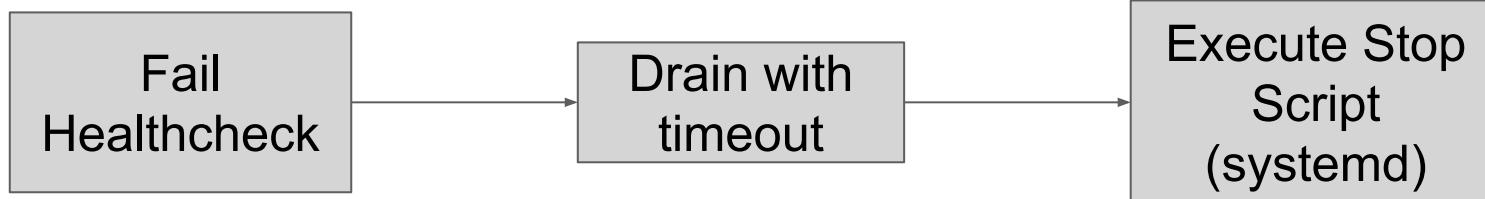


- Hierarchy
- Flat namespace that is *merged* to provide Priam config
- Functions for defaults (e.g. based on cpu)

Sidecar: Lifecycle

```
curl http://127.0.0.1:8080/Priam/REST/healthcheck
```

```
{
  "isBootstrapping": false,
  "isCassandraProcessAlive": true,
  "shouldCassandraBeAlive": true,
  "lastAttemptedStartTime": 1536170471979,
  "isGossipActive": true,
  "isNativeTransportActive": true,
  "isRequiredDirectoriesExist": true,
  "isYmlWritten": true,
  "isHealthy": true,
  "isHealthyOverride": true,
  "restoreStatus": {}
}
```



```
curl http://127.0.0.1:8080/Priam/REST/v1/cassadmin/stop
```

Sidecar: Lifecycle

```
curl http://127.0.0.1:8080/Priam/REST/v1/cassadmin/start
```



Rolling Restarts (Upgrades)

- Cluster automation is now much easier

What happens when Cassandra dies?

- Continuous health monitoring and supervision (OOM)
- Priam + systemd + jvmkill¹ == pretty good

¹ <https://github.com/airlift/jvmkill>

Sidecar: Maintenance

```
public abstract class IClusterManagement<T> extends Task {
```

```
/**  
 * Timer to be used for compaction interval.  
 *  
 * @param config {@link IConfiguration} to get configuration details from priam.  
 * @return the timer to be used for compaction interval from {@link IConfiguration#getCompactionCronExpression()}  
 */  
public static TaskTimer getTimer(IConfiguration config) throws Exception {  
    return CronTimer.getCronTimer(Task.COMPACTION.name(), config.getCompactionCronExpression());  
}
```

- JMX methods on cron
- Can add arbitrary tasks like compactions, flushes, etc

Sidecar: Maintenance

```
$> grep "Path(" CassandraAdmin.java
@Path("/v1/cassadmin")
@Path("/start")
@Path("/stop")
@Path("/refresh")
@Path("/info")
@Path("/partitioner")
@Path("/ring/{id}")
@Path("/flush")
@Path("/compact")
@Path("/cleanup")
@Path("/repair")
@Path("/version")
@Path("/disablegossip")
@Path("/enablegossip")
@Path("/gossipinfo")
@Path("/move")
@Path("/drain")
@Path("/decompress")
```

- Sidecar provides JMX over HTTP
 - Cleanup
 - Invoke complex JMX methods using curl
 - Many of these are better done scheduled (e.g. repair, compaction, flushes, etc)

Sidecar: Monitoring

Netflix / **spectator** Unwatch ▾ 348 Star 347 Fork 84

Code Issues 14 Pull requests 0 Projects 0 Wiki Insights

Branch: master **spectator / spectator-ext-jvm / src / main / resources / cassandra.conf** Find file Copy path

 **brharrington** additional jmx mappings for cassandra 3.x (#621) 1602983 22 hours ago

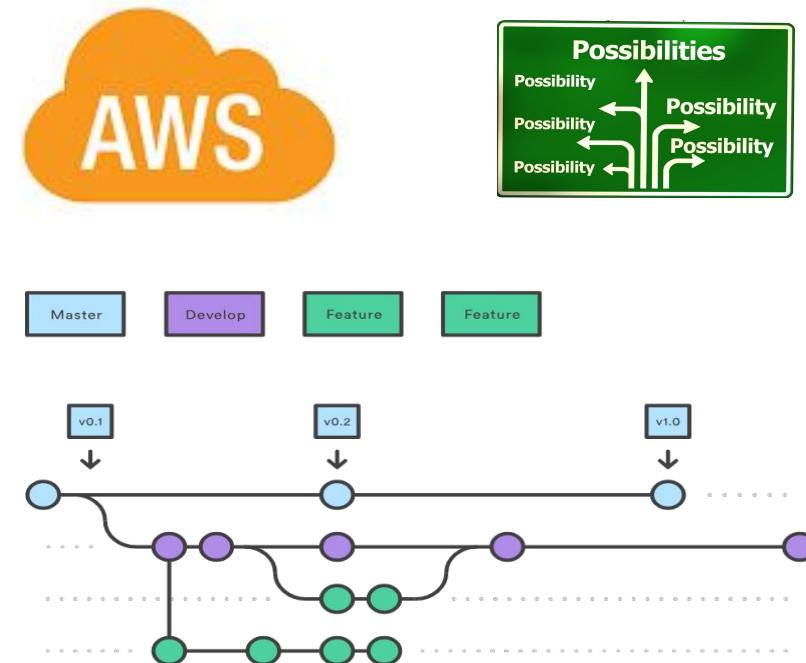
1 contributor

2907 lines (2891 sloc) | 67.7 KB Raw Blame History  

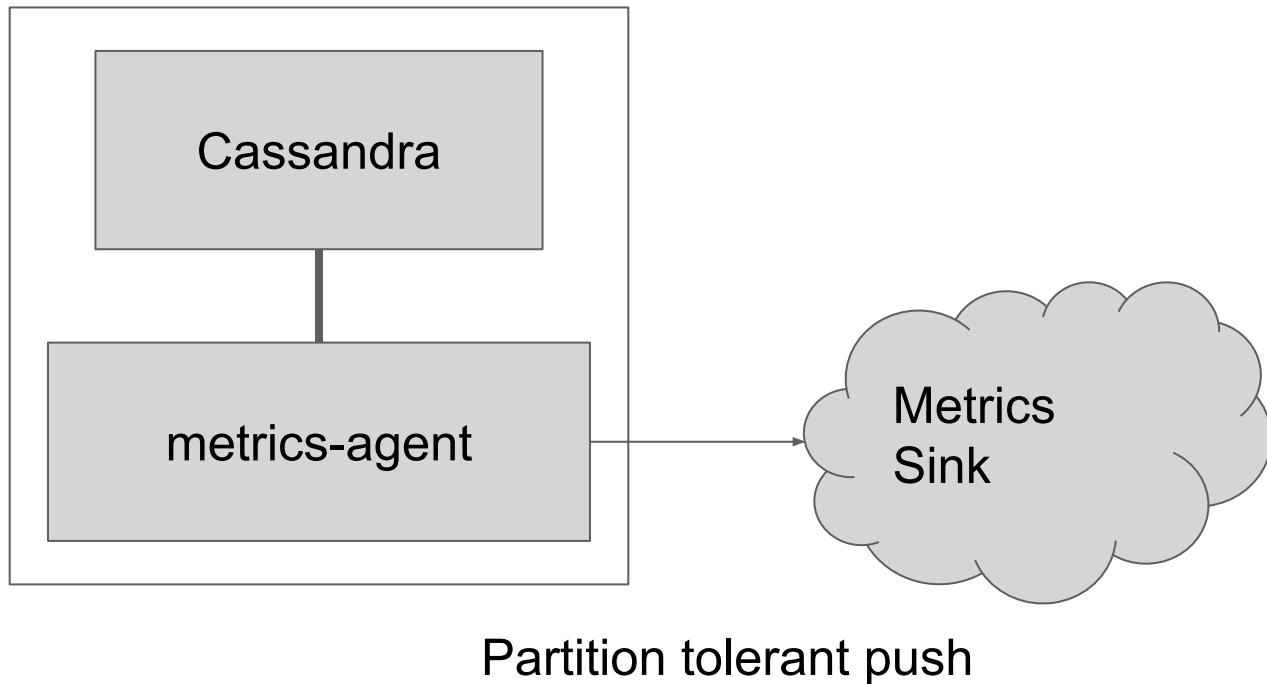
```
1
2 // http://wiki.apache.org/cassandra/Metrics
3 netflix.spectator.agent.jmx {
4     mappings = ${?netflix.spectator.agent.jmx.mappings} [
5         //
6         // type=Cache
7         //
8         {
9             query = "org.apache.cassandra.metrics:type=Cache,name=Hits,*"
10            measurements = [
```

Lessons learned from Existing sidecars

- Specific to deployments
- Not easy to use externally
- Multi-cloud?
- ~3 different implementations of everything
- Version compatibility



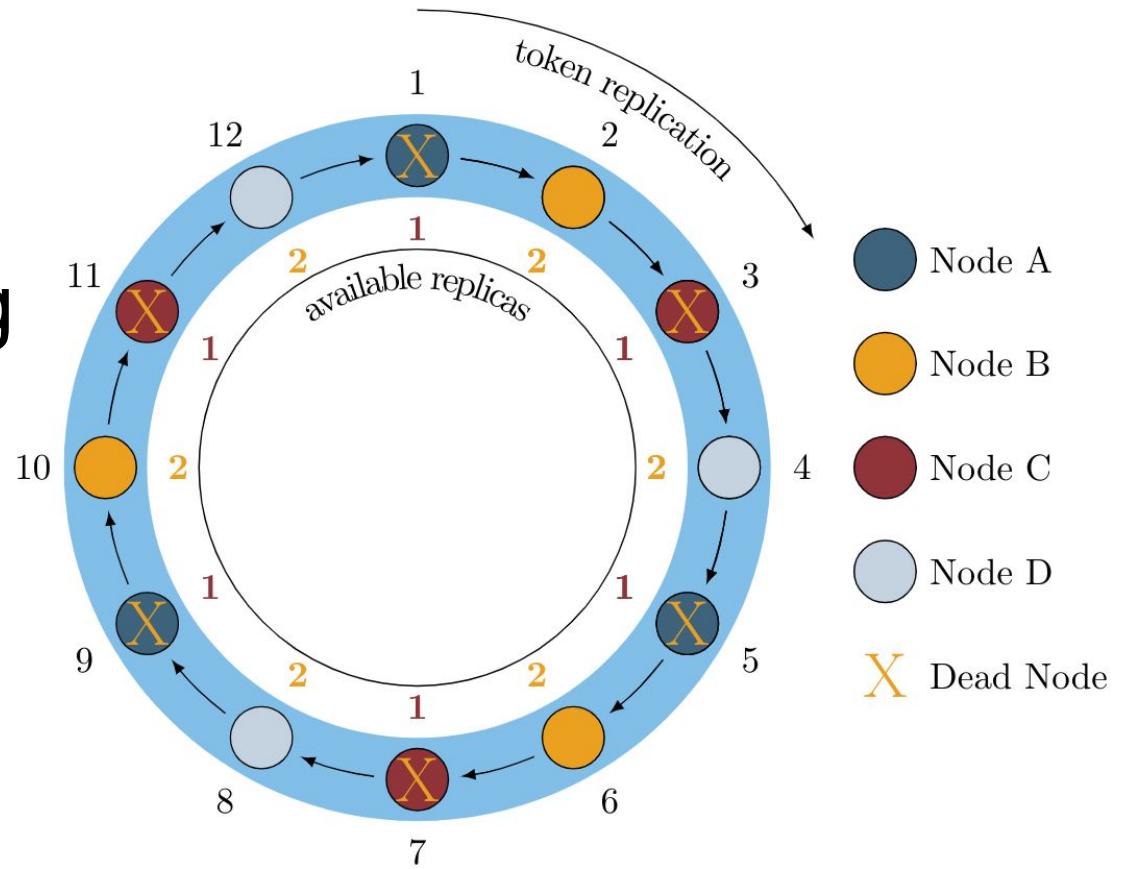
Sidecar: Monitoring



- Sidecar agent = scalable
- Non agent JMX metric export ... not even once
- Still a sidecar, just not Priam

Sidecar: Ring Health

- Sidecar monitoring local node
- Can also look at the ring
 - Gossip state
- Can
 - Ask a few Priams
 - Export info to streaming system



Sidecar: Backup/Restore

```
$> ls backup
AbstractBackup.java          BackupStatusMgr.java      IBackupStatusMgr.java
AbstractBackupPath.java       BackupVerification.java  IFileSystemContext.java
BackupFileSystemAdapter.java  BackupVerificationResult.java IIncrementalBackup.java
BackupFileSystemContext.java  CommitLogBackup.java    IMessageObserver.java
BackupMetadata.java           CommitLogBackupTask.java IncrementalBackup.java
BackupRestoreException.java   FileSnapshotStatusMgr.java IncrementalMetaData.java
BackupRestoreUtil.java        IBackupFileSystem.java   MetaData.java

```

parallel

```
RangeReadInputStream.java
SnapshotBackup.java
Status.java
```

- Backup on cron schedule
 - Snapshot
- Parallel incrementals
- *Verification*
- Pluggable with code (S3, GCS, etc...)

Sidecar: Backup/Restore

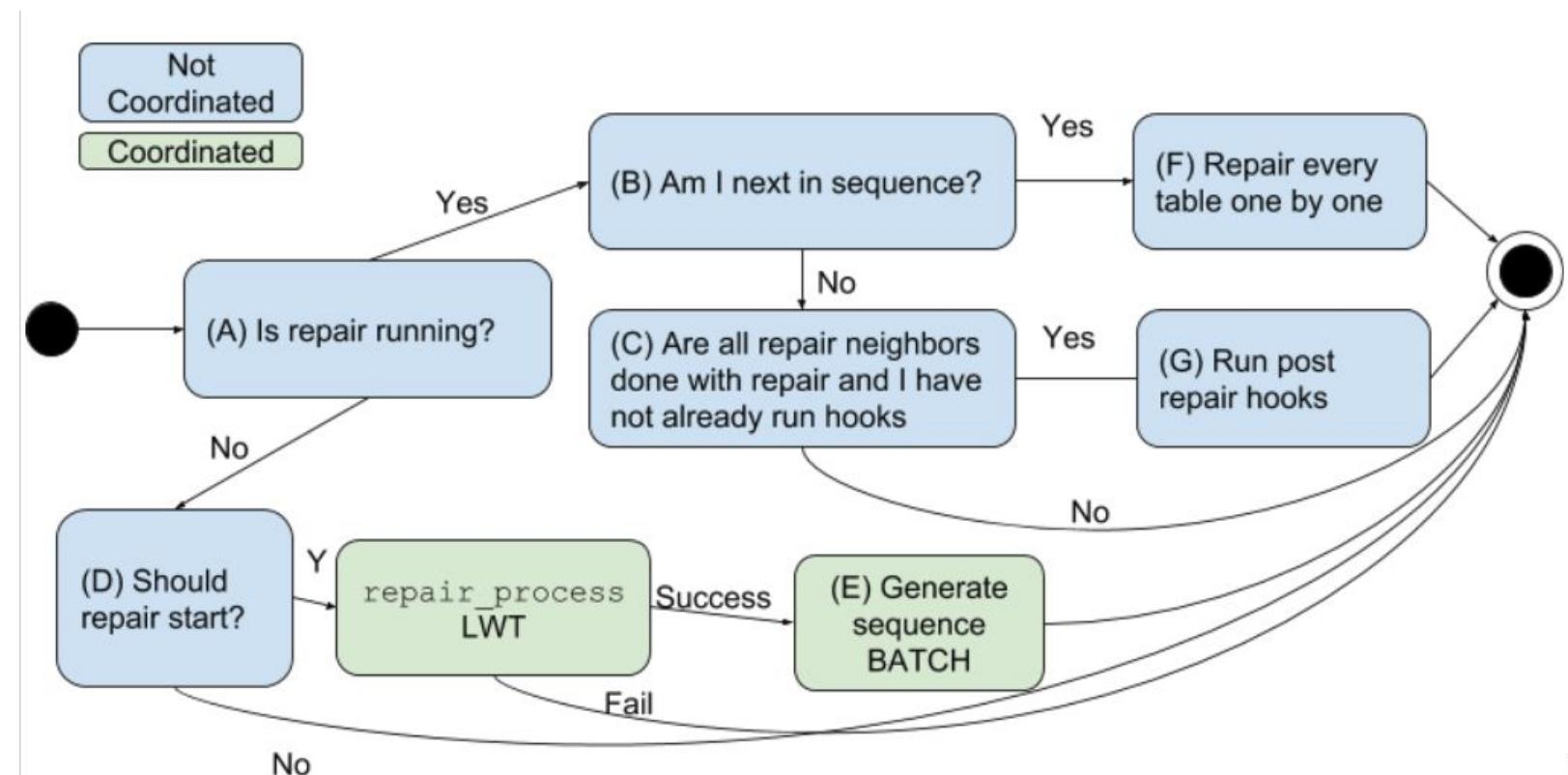
```
$> cd backupv2
$> ls
ColumnfamilyResult.java  MetaFileInfo.java      MetaFileReader.java      PrefixGenerator.java
FileUploadResult.java    MetaFileManager.java   MetaFileWriterBuilder.java
```

- Similar to cassandra-mirror¹
- Keeps track of what files have already been uploaded
 - Unlocks *minute* level snapshot backups
 - Only uploads files once
- Continuous, point in time, self healing, eventually consistent

1: <https://github.com/hashbrowncipher/cassandra-mirror>

Sidecar: Repair

- Always on
 - Just works
 - In Cassandra
itself? ¹



1: <https://issues.apache.org/jira/browse/CASSANDRA-14346>

State of Community Sidecars

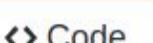
State of Community Sidecars

Everyone builds their own

State of Community Sidecars

 [Netflix / Priam](#)

 [Code](#)  [Issues 20](#)  [Pull requests 9](#)  [Projects 0](#)  [Wiki](#)  [Insights](#)

Co-Process for backup/recovery, Token Management, and Centralized Configuration management for Cassandra.

 [thelastpickle / cassandra-reaper](#)

 [Code](#)  [Issues 83](#)  [Pull requests 5](#)  [Projects 0](#)  [Insights](#)

Automated Repair Awesomeness for Apache Cassandra <http://cassandra-reaper.io/>

[apache-cassandra](#) [cassandra](#) [repairs](#) [cassandra-repairs](#) [repair-schedules](#) [cassandra-clusters](#) [compactions](#) [snapshots](#) [cleanups](#)

State of Community Tooling

 [spotify / cstar](#)

 Watch ▾ 28  Star 85  Fork 9

 Code

 Issues 5

 Pull requests 1

 Projects 0

 Wiki

 Insights

Apache Cassandra cluster orchestration tool for the command line

[python](#) [cassandra](#) [orchestration](#)

 [CrowdStrike / cassandra-tools](#)

 Watch ▾ 38  Star 47  Fork 13

 Code

 Issues 1

 Pull requests 0

 Projects 0

 Wiki

 Insights

Python Fabric scripts to help automate the launching and managing of clusters for testing

State of Community Tooling 2

 [instaclustr / cassandra-operator](#)

 Watch ▾ 17

 Star 46

 Fork 16

 Code

 Issues 21

 Pull requests 5

 Projects 1

 Wiki

 Insights

Kubernetes operator for Apache Cassandra

 [instaclustr / cassandra-sstable-tools](#)

 Watch ▾ 12

 Star 35

 Fork 13

 Code

 Issues 1

 Pull requests 0

 Projects 0

 Wiki

 Insights

 [smartcat-labs / cassandra-diagnostics](#)

 Watch ▾ 13

 Star 49

 Fork 5

 Code

 Issues 22

 Pull requests 1

 Projects 0

 Wiki

 Insights

Cassandra Node Diagnostics Tools

State of Community Tooling 3

 [instaclustr / instarepair](#)

 [Code](#)  [Issues 0](#)  [Pull requests 0](#)  [Projects 0](#)  [Wiki](#)  [Insights](#)

Repair Cassandra cluster using read repairs

 [BrianGallew / cassandra_range_repair](#)

forked from [mstump/cassandra_range_repair](#)

 [Code](#)  [Issues 7](#)  [Pull requests 0](#)  [Projects 0](#)  [Wiki](#)  [Insights](#)

python script to repair the primary range of a node in N discrete steps

State of Community Backup

 [tbarbugli / cassandra_snapshotter](#)

 Watch ▾ 18

 Star 191

 Fork 124

 Code

 Issues 23

 Pull requests 7

 Projects 0

 Wiki

 Insights

A tool to backup cassandra nodes using snapshots and incremental backups on S3

 [instaclustr / cassandra-backup](#)

 Watch ▾ 3

 Star 2

 Fork 1

 Code

 Issues 0

 Pull requests 1

 Projects 0

 Wiki

 Insights

Backup utility and library for Apache Cassandra

 [hashbrowncipher / cassandra-mirror](#)

 Watch ▾ 1

 Star 0

 Fork 1

 Code

 Pull requests 5

 Projects 0

 Wiki

 Insights

Utilities for copying Cassandra SSTables to a durable store

**And that's not even all of 'em
... but do they solve our
problems?**

	Puppet/ Chef	Terraform	CRON	Fabric / SSH in a for loop	Reaper	Priam	Jenkins	Sensu/ Nagios
Bootstrap	No	Yes	No	Maybe	No	Yes	Maybe	No
Maintenance	Yes	Maybe	Maybe	Yes	No	Yes	Yes	No
Configuration	Yes	Maybe	No	Maybe	No	Yes	Maybe	No
Monitoring	Maybe	No	Maybe	Maybe	No	Yes	Maybe	Yes
Backup	Maybe	No	Maybe	Maybe	Maybe	Yes	Yes	No
Repair	No	No	Sorta	Yes	Yes	Soon	Yes	No
Easy to Use	Yes	No*	Yes	Yes	Yes	No	Maybe	Yes
Multi-Cloud	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Does it Scale	Yes	Yes	Yes	Sorta	Yes*	Yes	Sorta*	Yes

Oops.

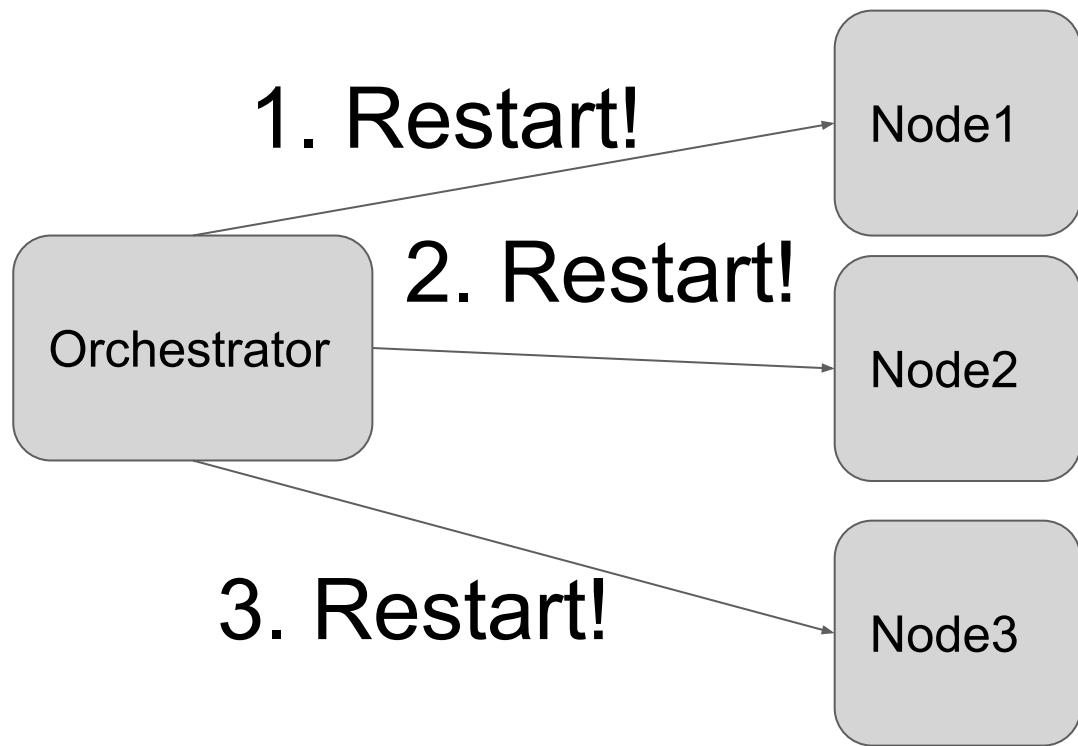
Lessons Learned

Community has operated Cassandra using

aforementioned sidecars for **years**

... We have learned a few things

Lesson Learned: Control Plane

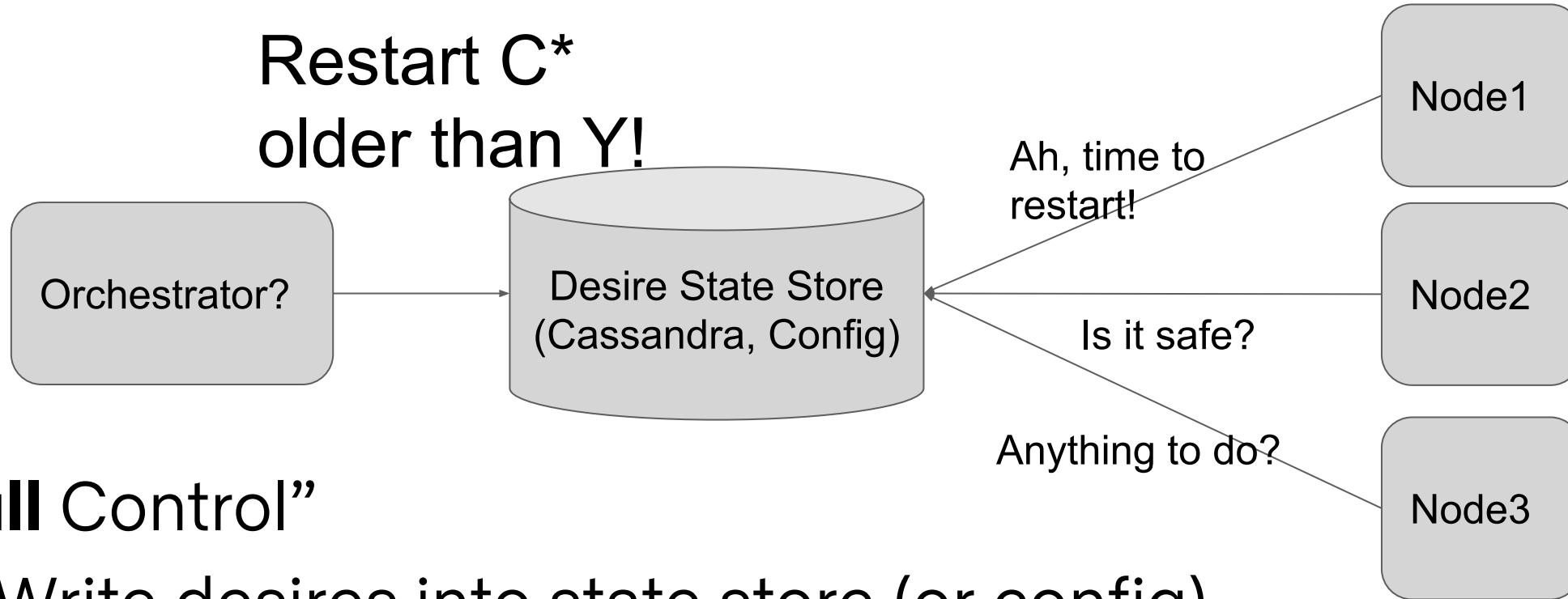


“Push Control”

- Relies on unreliable communication (ssh < http)
- Hard to make eventually consistent
- Hard to guarantee safety
- Very **easy** to implement

Lesson Learned: Control Plane

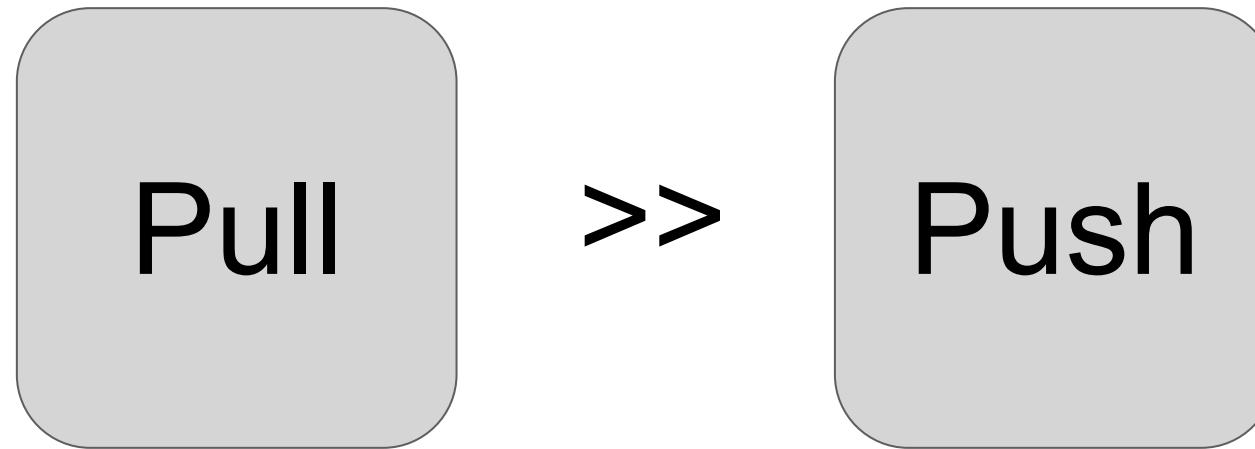
Restart C*
older than Y!



“Pull Control”

- Write desires into state store (or config)
- Nodes coordinate through state store as needed
- Tradeoff: Hard to implement

Lesson Learned: Control Plane



Goals of C* management process

1. Easy to Use

Goals of C* management process

2. Solves or eases most common problems

Goals of C* management process

3. Pluggable



Goals of C* management process

4. Scalable



C* Management Process design

**Unopinionated,
pluggable & extensible**

More info?

Management Process:

<https://issues.apache.org/jira/browse/CASSANDRA-14395>

apache / cassandra-sidecar

Code

Pull requests 1

Actions

Security

Insights

Sidecar for Apache Cassandra <https://cassandra.apache.org/>

database

java

cassandra

7 commits

1 branch

0 releases

Branch: master ▾

New pull request

Create



tolbertam and **dineshjoshi** Read config from sidecar.config System Property instead of classpath

Can it scale?

Currently deployed in Netflix ecosystem on thousands of
 C^* nodes

What else is coming?

- Bulk command tools
- More @ Cassandra [CIP-1](#)
- CDC processor?

Thank You.

