

How Netflix manages petabyte scale Apache Cassandra in the cloud

Joey Lynch, Vinay Chella
Netflix's Distributed Database Engineers



Who are we?



Vinay Chella

Distributed Systems Engineer

Focusing on Apache Cassandra and Data Abstractions

Cloud Data Engineering
Netflix



Joey Lynch

Distributed Systems Engineer

Distributed system addict and data wrangler



Cloud Data Engineering
Netflix

Agenda

Why use Cassandra?

Scale of Cassandra

Life of Cassandra Cluster

- Where does it start?
- Provisioning
- Keep it running
- Migration / Retiring

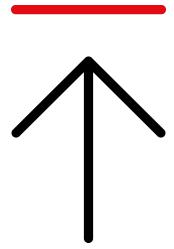
Murphy's law applied

- Millions of operations per sec
- Global data replication
- Failure isolation at rack level
- Chaos ready database
- Tunable consistency
- Log structured storage engine

Scale

- 10's of thousands instances
- 100's of global C* clusters
- >6 PB of data
- Millions of requests / second
- Replicating several GiB/sec data across the globe

Story of Apache Cassandra and Netflix



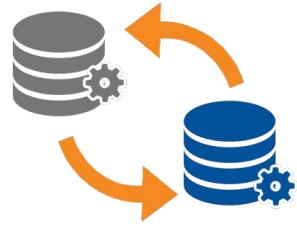
Inception



Provision



Keep it running



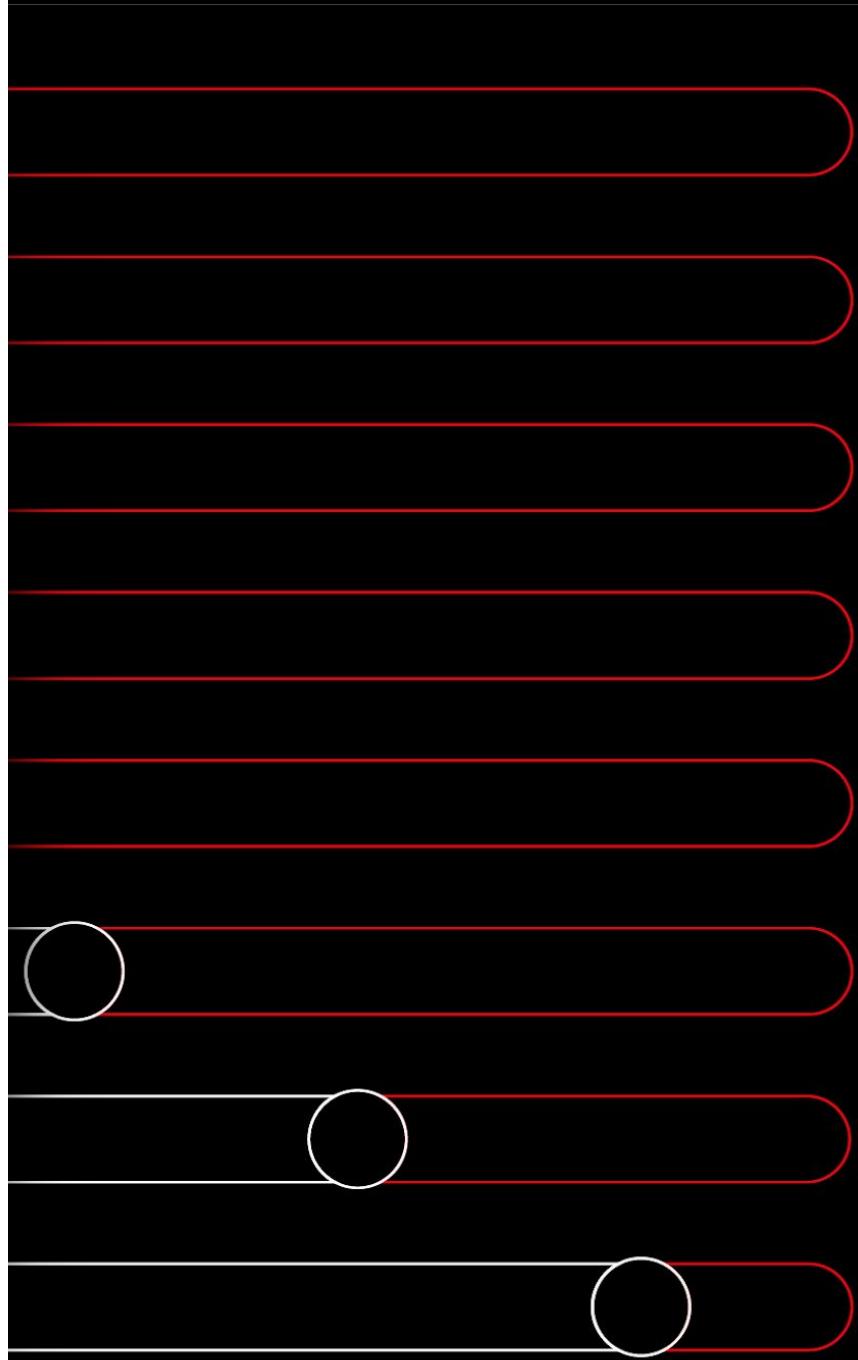
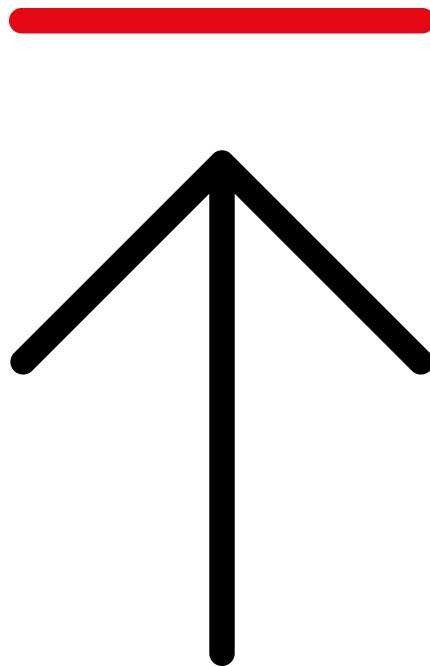
Migrations

ANYTHING
THAT CAN GO WRONG
WILL GO WRONG

Murphy's Law

Inception

Where does it all start?

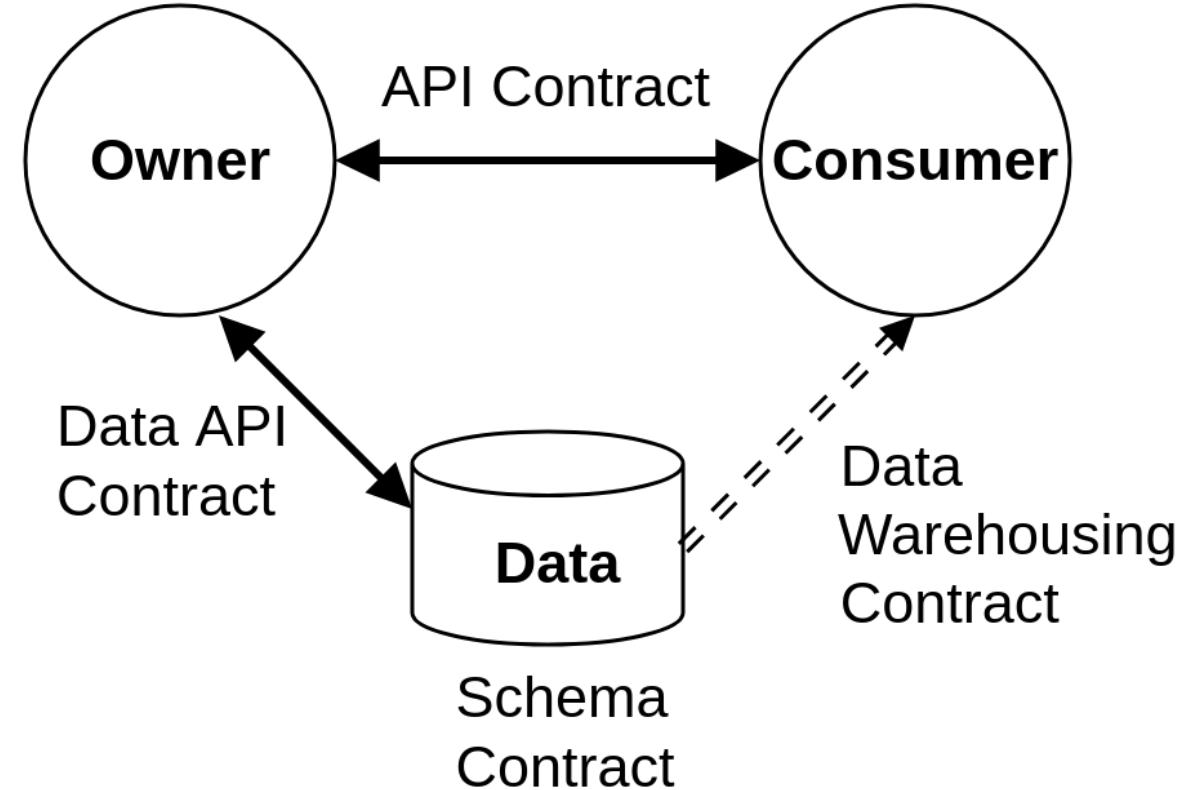


Inception

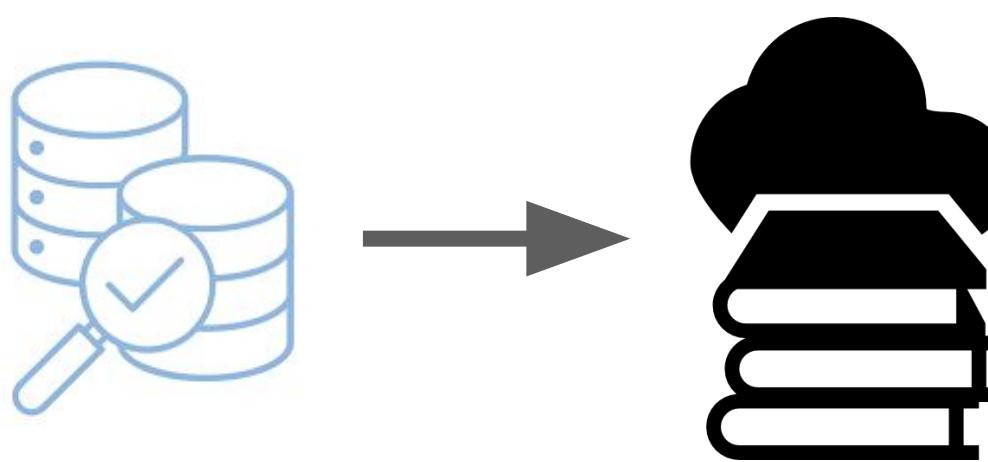


Service philosophy

- Context not *control*
 - ◆ Education
 - ◆ Tooling
- SLOs are key
 - ◆ Size, rate, latency, availability
- Every party must be responsible



Inception



Invest in DevEd

Picking a Datastore

storage density, meaning that storing an incremental piece of data is relatively cheap to other datastores (think GiB / \$\$).

- Scaling duration: Approximate order of magnitude time it takes to scale this datastore up or down. Obviously dataset size matters here, but we're talking orders of magnitude *on average*.

Datastore	Horizontally Scalable	Operation Efficiency	Storage Efficiency	Scaling duration
Cassandra	Yes	Medium	High	0(days)
EVCache	Yes	High	High	0(hours)
Dynomite	Yes	High	Low	0(days)
Elasticsearch	Yes	Medium	High	0(TTL)
ZooKeeper	No	Low	Low	NA
RDS	Partially*	High	High	0(days)
S3	Yes	High	Super High	0(seconds)

*: RDS can scale reads easily, but writes are harder to scale. Aurora is much better but can still struggle with write scaling.

Table of contents

Main Question: Can I use it for Streaming?

Streaming: Multi-region full-active applications

Streaming: Island Model (single active region)

Context: Scalability

Context: Performance

Context: Durability

Context: Consistency

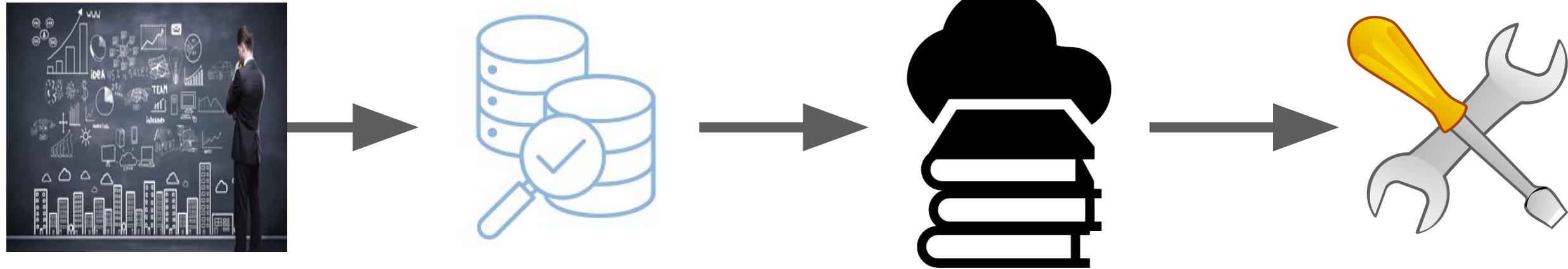
Context: Netflix Ecosystem

Context: Performance

Performance is a complex topic, and generally we can pay more money to get better performance, but generally we



Inception



Better Tooling

CDE Self Service Cassandra Elasticsearch Dynomite EVCache ZooKeeper Admin

Cassandra Clusters » cass_foobar

Summary Costs Nodes Keyspaces Backups Repairs SLOs Maintenance Windows Notes Attributes Ingress

Owner Emails [Update](#)
subdevteam dev1 dev2 dev3 dev4

PagerDuty Service [subeng](#) Slack Channel [subscriberteam](#)

Annualized Cost **\$\$\$\$**
 Has Customer Impact?
 Contains Critical Data?

Maintenance: None Backup: Finished Repair: Started - 97%

Last 1 hour [▼](#) US-EAST-1 EU-WEST-1 US-WEST-2

Coordinator Reads Last 1h AVG 944.1 K/s MIN 920.1 K/s MAX 985.3 K/s TOTAL 3.39 M	Coordinator Writes Last 1h AVG 1.5 M/s MIN 1.4 M/s MAX 1.5 M/s TOTAL 88.5 M	Data Explorer Metrics Cluster Health
Coordinator Read Latency 99th Last 1h AVG 5.5 ms MIN 4.6 ms MAX 7.2 ms	Coordinator Write Latency 99th Last 1h AVG 1.2 ms MIN 1.2 ms MAX 1.3 ms	Data Size TOTAL 258 TB

N

Cost insights

CDE Self Service Cassandra Dynomite Elasticsearch Help

Cassandra Clusters » cass_demo_cluster PROD TEST

Summary \$ Costs Nodes Keyspaces Backups Repairs SLAs Maintenance Windows Notes Attributes Ingress

PROD

Annualized Cost: \$77,844

Instances (EC2)

A donut chart illustrating the distribution of annualized costs for Instances (EC2) across different regions. The chart is divided into three segments: a large yellow segment representing the majority of the cost, a smaller teal segment, and a very small orange segment. A callout box highlights the EU-WEST-1 region, which is the largest segment at \$26,390.

Region	Annualized Cost
EU-WEST-1	\$26,390
Other Regions	\$68,300

Primary Backups (S3)

A donut chart showing the annualized cost for Primary Backups stored in S3. The chart is divided into three segments: a large teal segment, a medium yellow segment, and a small orange segment.

Backup Type	Annualized Cost
Primary Backups (S3)	\$6,819

Secondary Backups (S3)

A donut chart showing the annualized cost for Secondary Backups stored in S3. The chart is divided into three segments: a large teal segment, a medium yellow segment, and a small orange segment.

Backup Type	Annualized Cost
Secondary Backups (S3)	\$2,725

Maintenance

CDE Self Service Cassandra ▾ Elasticsearch ▾ Dynomite ▾ EVCache ▾ ZooKeeper ▾

Cassandra Clusters » cass_foobar

Summary Costs Nodes Keyspaces Backups Repairs SLOs Maintenance

Owner Emails [Update](#)
subdevteam dev1 dev2 dev3 dev4

PagerDuty Service
subeng

Slack Channel
subscriberteam

Maintenance: None Backup: Finished Repair: Started - 97%

Last 1 hour

Inception

Coordinator Reads Last 1h Coordinator Writes Last 1h

Maintenance - Repair Insights

Cassandra Clusters » cass_demo_cluster

PROD

TEST

[Summary](#) [\\$ Costs](#) [Nodes](#) [Keyspaces](#) [Backups](#) [Repairs](#) [SLAs](#) [Maintenance Windows](#) [Notes](#) [Attributes](#) [Ingress](#)

History

Settings

i Access to this section is restricted to CDE team members.

PROD

Enable Repairs

Add Table Repair Settings...

Keyspace	Table	Enabled?	Repair Type	Intensity	Min Delay	# Workers	Parallelism	Split Range		
<input type="text"/> Filter by Keyspace	<input type="text"/> Filter by Table									
keyspace_1	table_1	✓	SUBRANGE	0	24	5	SEQUENTIAL	2097152		

1 table repair setting

Cassandra Clusters » cass_demo_cluster

PROD

TEST

[Summary](#) [\\$ Costs](#) [Nodes](#) [Keyspaces](#) [Backups](#) [Repairs](#) [SLAs](#) [Maintenance Windows](#) [Notes](#) [Attributes](#) [Ingress](#)[History](#) [Settings](#)*i* Access to this section is restricted to CDE team members.

PROD

Enable Repairs

Keystpace ▾ Table

 Filter by Keyspace Filter

keyspace_1

table_1

Add Table Repair Settings...

Split Range ▾

2097152



Repair Settings | cass_demo_cluster » keyspace_1 » table_1

Environment

PROD

TEST

* Keyspace

keyspace_1

* Table

table_1

Enable Table Repairs

* Repair Type

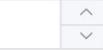
REGULAR

INCREMENTAL

SUBRANGE

* Minimum Delay

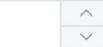
24



hours

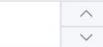
* Intensity

0



* Number of Workers

5



* Parallelism

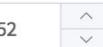
SEQUENTIAL

PARALLEL

DATACENTER AWARE

* Split Range

2097152



Cancel

Save Repair Settings

Maintenance - Backup Insights

Cassandra Clusters » cass_demo_cluster

PROD

TEST

[Summary](#) [\\$ Costs](#) [Nodes](#) [Keystreams](#) [Backups](#) [Repairs](#) [SLAs](#) [Maintenance Windows](#) [Notes](#) [Attributes](#) [Ingress](#)

i Access to this section is restricted to CDE team members.

PROD

Region	Primary Backups				Secondary Backups	
	Enabled?	Schedule	Retention	S3 Location	Enabled?	Location
us-east-1	<input checked="" type="checkbox"/>	0 5 8 * * ?	30 days	useast1-cass-prod-1/prod_backup	<input type="checkbox"/>	
eu-west-1	<input checked="" type="checkbox"/>	0 5 8 * * ?	7 days	euwest1-cass-prod-1/prod_backup	<input type="checkbox"/>	
us-west-2	<input checked="" type="checkbox"/>	0 5 8 * * ?	7 days	uswest2-cass-prod-1/prod_backup	<input type="checkbox"/>	

Cassandra Clusters » cass_demo_cluster

PROD

TEST

[Summary](#) [\\$ Costs](#) [Nodes](#) [Keyspaces](#) [Backups](#) [Repairs](#) [SLAs](#) [Maintenance Windows](#) [Notes](#) [Attributes](#) [Ingress](#)*i* Access to this section is restricted to CDE team members.

PROD

Region	Primary Backups	
	Enabled?	Schedule
us-east-1	<input checked="" type="checkbox"/>	0 5 8 * * ?
eu-west-1	<input checked="" type="checkbox"/>	0 5 8 * * ?
us-west-2	<input checked="" type="checkbox"/>	0 5 8 * * ?

Backup Settings | cass_demo_cluster

Environment

PROD

TEST

US-EAST-1

EU-WEST-1

US-WEST-2

US-WEST-1

Enable Primary Backups



* Cron Schedule

0 5 8 * * ?

* Retention Period

30

days

* S3 Location

useast1-cass-prod-1

/prod_backup

Enable Secondary Backups



Cancel

Save Backup Settings

Maintenance - Node Insights

Cassandra Clusters » cass_foobar

PROD TEST

[Summary](#) [\\$ Costs](#) [Nodes](#) [Keyspaces](#) [Backups](#) [Repairs](#) [SLOs](#) [Maintenance Windows](#) [Notes](#) [Attributes](#) [Ingress](#)

PROD

Last cached: Sep 2, 2019, 12:12:14 PM



Node ID	Zone	Token	C* Version	Sidecar Version	OS	Instance	C* Data Size	Disk Used	Disk Size	Disk Used %
i-0c	f5	us-east-1e				i3.4xlarge	0.0 KB	0.0 KB		
i-0b	d0	us-east-1c	-1056844710747700824	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-09	3a	us-east-1e	-1152921502798271405	3.0.19.1	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-06	f8	us-east-1d	-1248998294848841986	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-0a	51	us-east-1c	-1345075086899412567	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-05	57	us-east-1e	-1441151878949983148	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-0f	72	us-east-1d	-1537228671000553729	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-08	f6	us-east-1c	-1633305463051124310	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-01	073	us-east-1e	-1729382255101694891	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-05	f6	us-east-1d	-1825459047152265472	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.4 TB	3.8 TB	37.00%
i-04	la8	us-east-1c	-192153582292565595	3.0.19.1	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-04	l6	us-east-1c	-1921535839202836053	3.0.19.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.4 TB	3.8 TB	35.00%
i-02	76	us-east-1e	-2017612631253406634	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.4 TB	3.8 TB	37.00%
i-04	141	us-east-1d	-2113689423303977215	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-04	40	us-east-1c	-220976621535457796	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-0d	dd7	us-east-1e	-2305843007405118377	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-02	8d	us-east-1d	-2401919799455688958	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-0e	'88	us-east-1c	-2497996591506259539	3.0.19.1	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-0e	42	us-east-1e	-2594073383556830120	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.4 TB	3.8 TB	35.00%
i-0b	8c	us-east-1d	-2690150175607400701	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-00	58	us-east-1c	-2786226967657971282	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-05	3b	us-east-1e	-288230374343136176	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.4 TB	3.8 TB	36.00%
i-0e	79	us-east-1e	-2882303759708541863	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-0d	df	us-east-1d	-2978380551759112444	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-08	le	us-east-1c	-3074457343809683025	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.4 TB	3.8 TB	37.00%
i-06	7	us-east-1e	-3170534135860253606	3.0.19.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.3 TB	3.8 TB	35.00%
i-08	1	us-east-1d	-3266610927910824187	3.0.17.2	7.0.60-h74.4ef95a1	xenial	1.22 TB	1.4 TB	3.8 TB	36.00%

SLOs are Key

Cassandra Clusters » cass_foobar

PROD TEST

[Summary](#) [Costs](#) [Nodes](#) [Keyspaces](#) [Backups](#) [Repairs](#) [SLOs](#) [Maintenance Windows](#) [Notes](#) [Attributes](#) [Ingress](#)

PROD

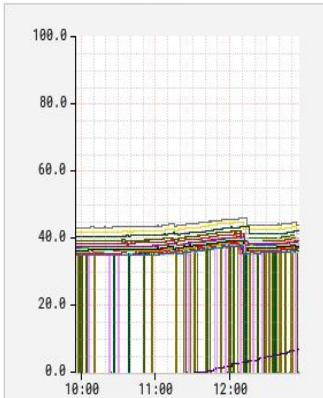
SLOs are visible to all users, but can be modified by CDE team members only.

Enable Monitoring

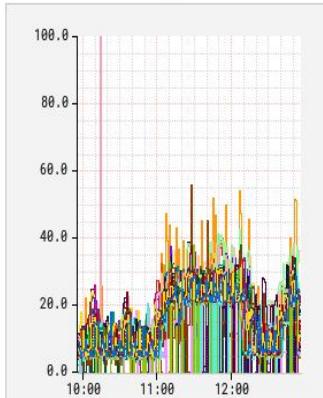
Actions

Update SLOs

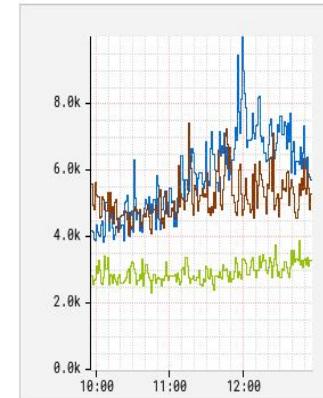
Disk Used (%)



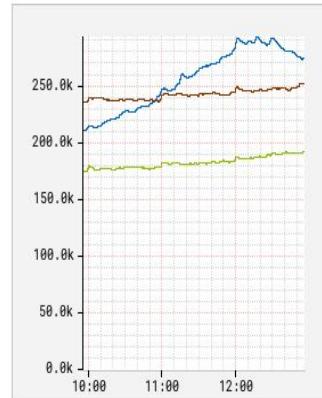
CPU Used (%)



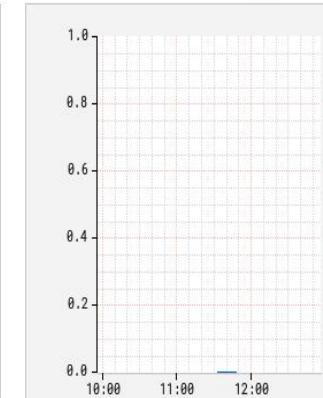
Read Latency 99th (µs)



Read Ops/sec



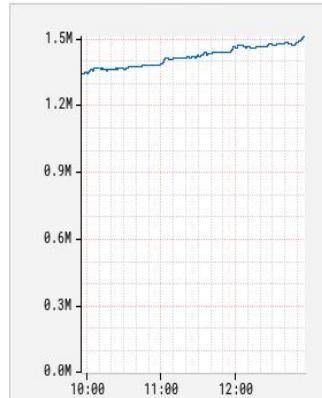
Backup v2 Failures



Write Latency 99th (µs)



Write Ops/sec

 Email Pager SQS Winston Email Pager SQS Winstoncritical period_in_minutes period_in_minutes threshold period_in_minutes threshold threshold_in_percent threshold_in_percent threshold_in_microseconds threshold_in_percent threshold_in_microseconds warn

Whom to page?

The screenshot shows a web-based management interface for a Cassandra cluster named "cass_foobar". The top navigation bar includes links for "CDE Self Service", "Cassandra", "Elasticsearch", "Dynomite", "EVCache", and "ZooKeeper". Below the navigation, the title "Cassandra Clusters » cass_foobar" is displayed. A horizontal menu bar contains icons for "Summary" (selected), "Costs", "Nodes", "Keyspaces", "Backups", "Repairs", "SLOs", and "Mair".

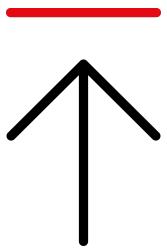
The main content area displays configuration details for the cluster:

- Owner Emails**: A field containing five email addresses: "subdevteam", "dev1", "dev2", "dev3", and "dev4". An "Update" button is located to the right.
- PagerDuty Service**: A field containing the value "subeng".
- Slack Channel**: A field containing the value "subscriberteam".

**Good contracts
make good
partners!**



Story of C*



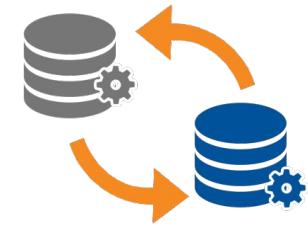
Inception



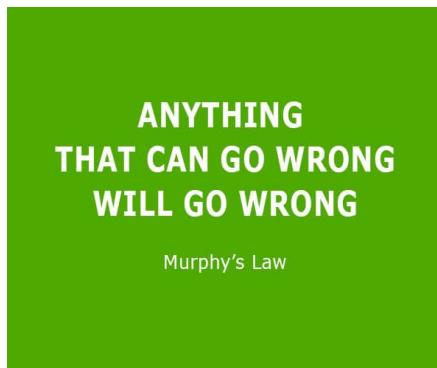
Provision



Keep it running

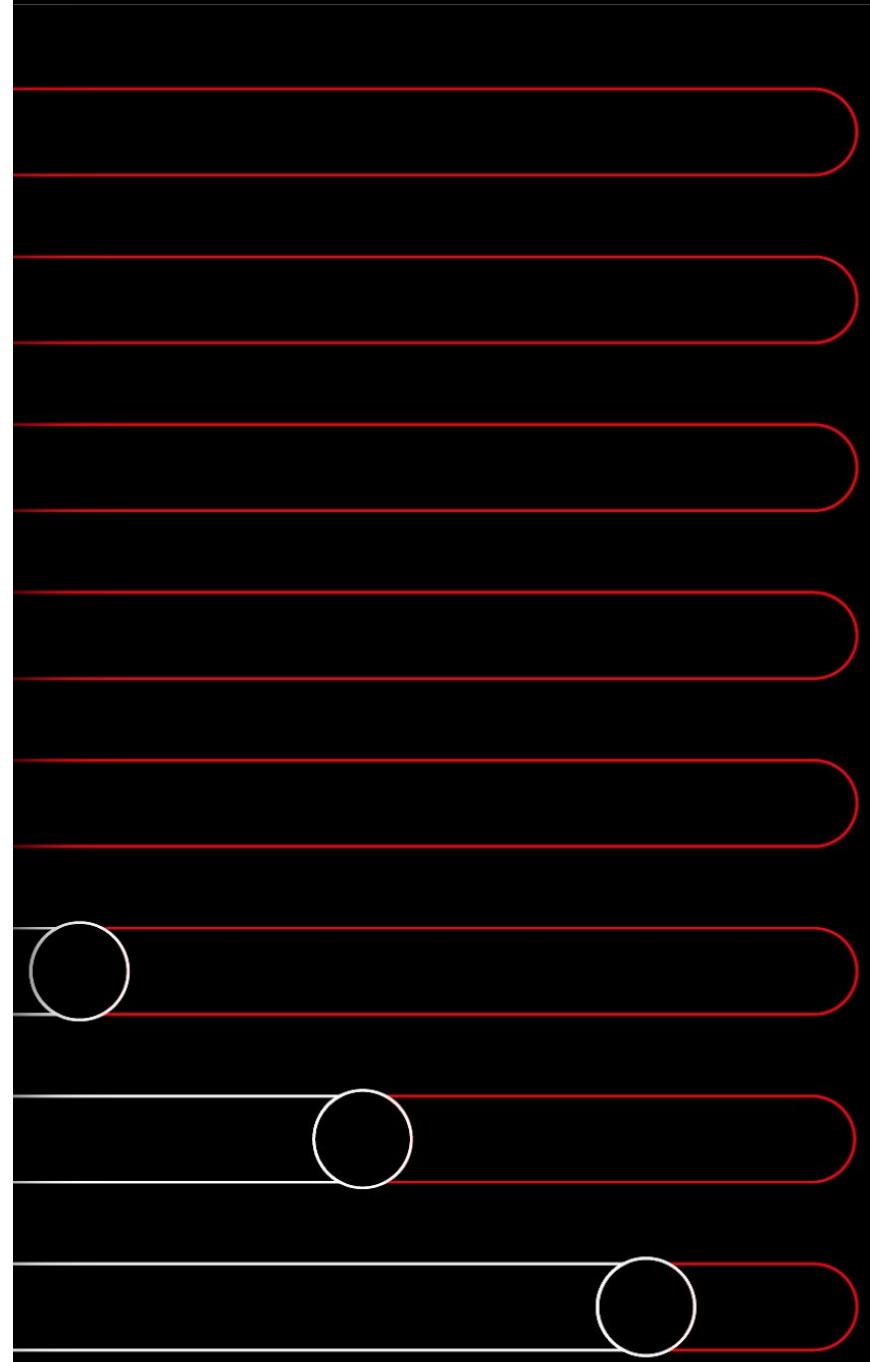


Migrations



Provision

Get up and running fast!



Cassandra Clusters » Create new Cluster

* Cluster Name

cass_



* Owner Email(s)

Type to search

PagerDuty Service

Type to filter

Slack Channel

Type to filter



* Cluster Topology

 Island Cluster(s) Global cluster

PROD

 Enable in PROD?

* Cluster regions

 us-east-1 eu-west-1 us-west-2

* Estimated Data Size per region

100 GB

GB

TEST

 Enable in TEST?

* Cluster regions

 us-east-1 eu-west-1 us-west-2

* Estimated Data Size per region

100 GB

GB

 Contains Critical Data?

Prove it before serving it

Robust AMI build process

Pre-flight check

AMI Build process

Filtered by: PIPELINE: Test C* AMI

Test C* AMI

BUILD & BAKE & TEST AMI PIPELINE
anonymous
about 8 hours ago

Status: SUCCEEDED

00:00 18:01 18:32 00:00 14:56 12:41 01:18:40 02:04 01:43 00:11 Duration: 01:53:32

[View All Parameters \(5\)](#) [Execution Details](#)

```
graph LR; EV[Evaluate Vars] --> DOL1[Delete Old NDBench Cluster Build #3601]; EV --> DOL2[Delete Old C* Test Cluster Build #3886]; EV --> UBS[Update Build Status]; DOL1 --> CND[Create NdBench Cluster Build #106]; CND --> FT[Functional Tests Build #658]; CND --> NDC[Create New C* Cluster Build #409]; NDC --> NDT[NdBench Performance Tests Build #1019]; NDT --> TT[Tag tested=true]; TT --> UBS;
```

STAGE DETAILS: EVALUATE VARS
Duration: 00:00

Step	Started	Duration	Status
Evaluate Vars	2019-09-03 14:52:30 PDT	00:00	SUCCEEDED

EVALUATE VARS

Evaluate Variables Config	Task Status
cass_cluster_name ndb_cluster_name load_test_driver sha	cass_perf_ft_30no_sep ndb_ft_30no_sep CassJavaDriverGeneric b6b9c5991f18e27dab79e250e44539 30362e1bea

[Source](#) | [Permalink](#)

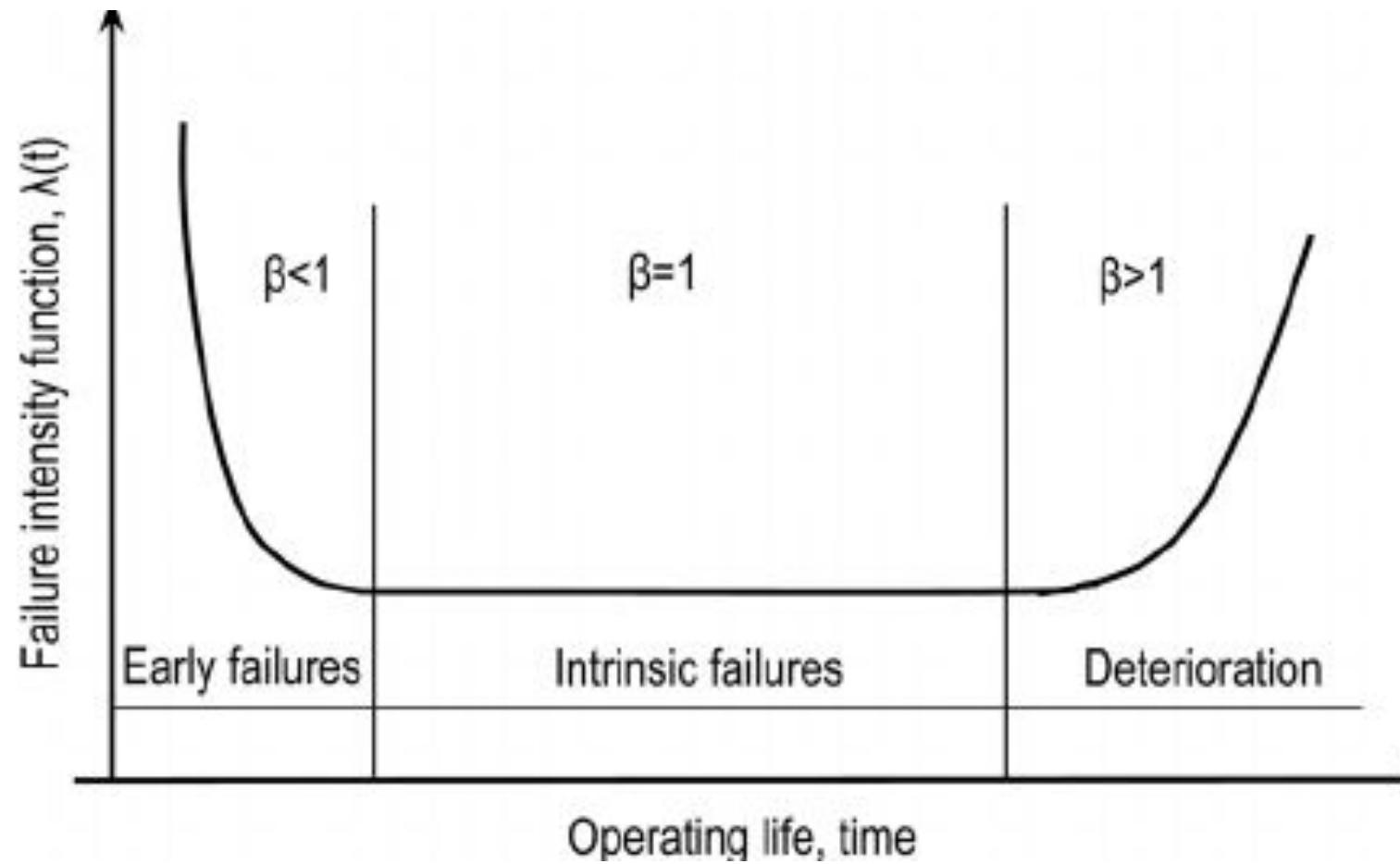
BUILD & BAKE & TEST AMI PIPELINE
anonymous
about 11 hours ago

Status: SUCCEEDED

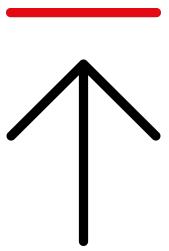
00:00 01:02 00:11 00:00 15:56 12:31 01:20:36 02:04 03:14 00:11 Duration: 01:41:02

[View All Parameters \(5\)](#) [Execution Details](#)

Pre-flight check EC2 Instance lifecycle



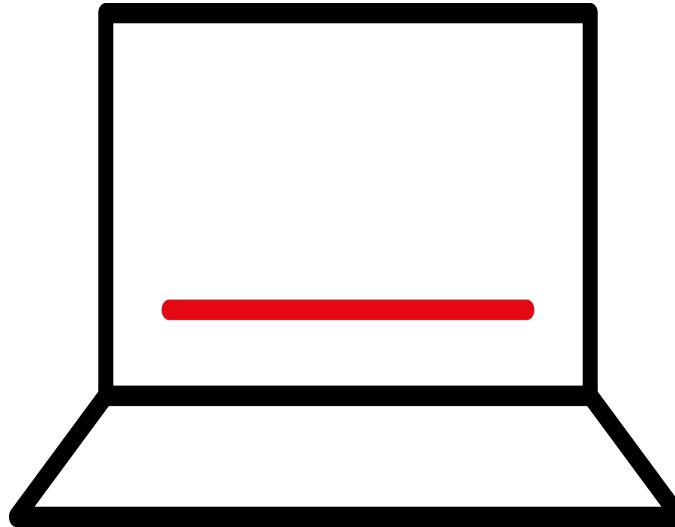
Story of C*



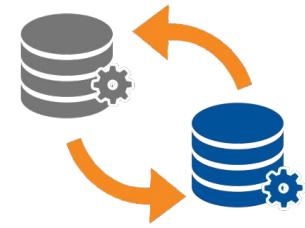
Inception



Provision



Keep it running

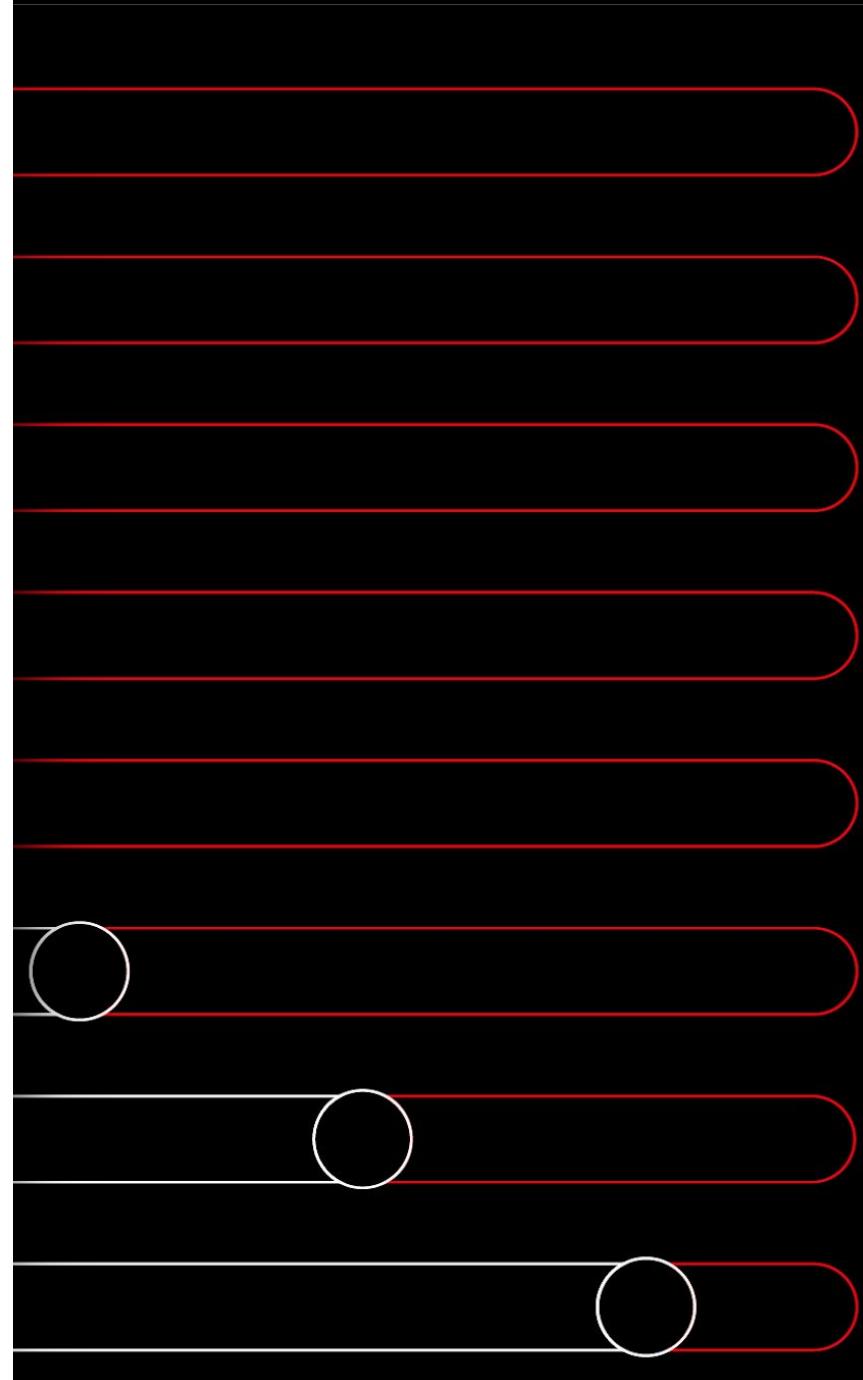
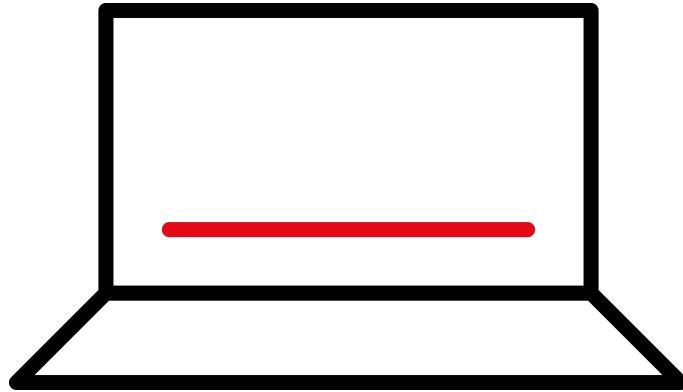


Migrations

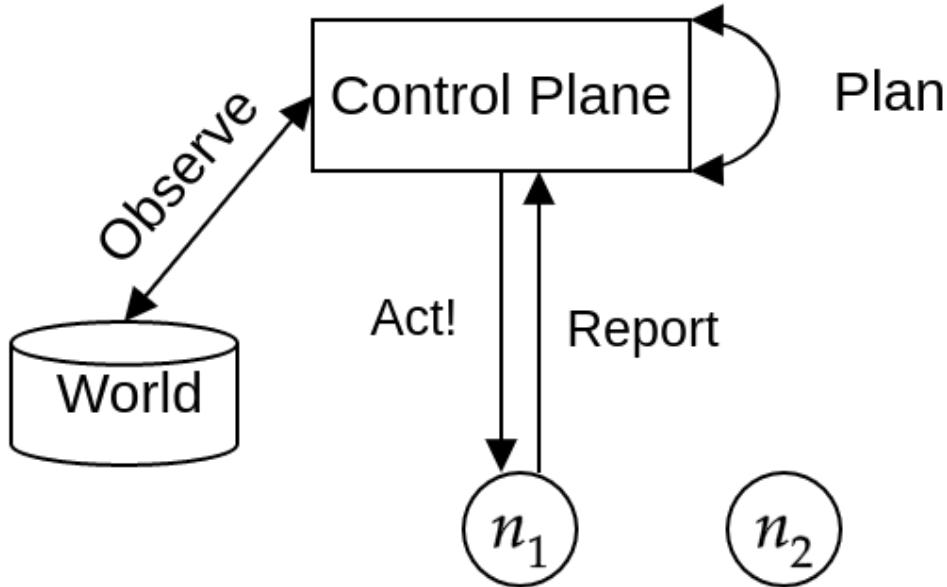
ANYTHING
THAT CAN GO WRONG
WILL GO WRONG

Murphy's Law

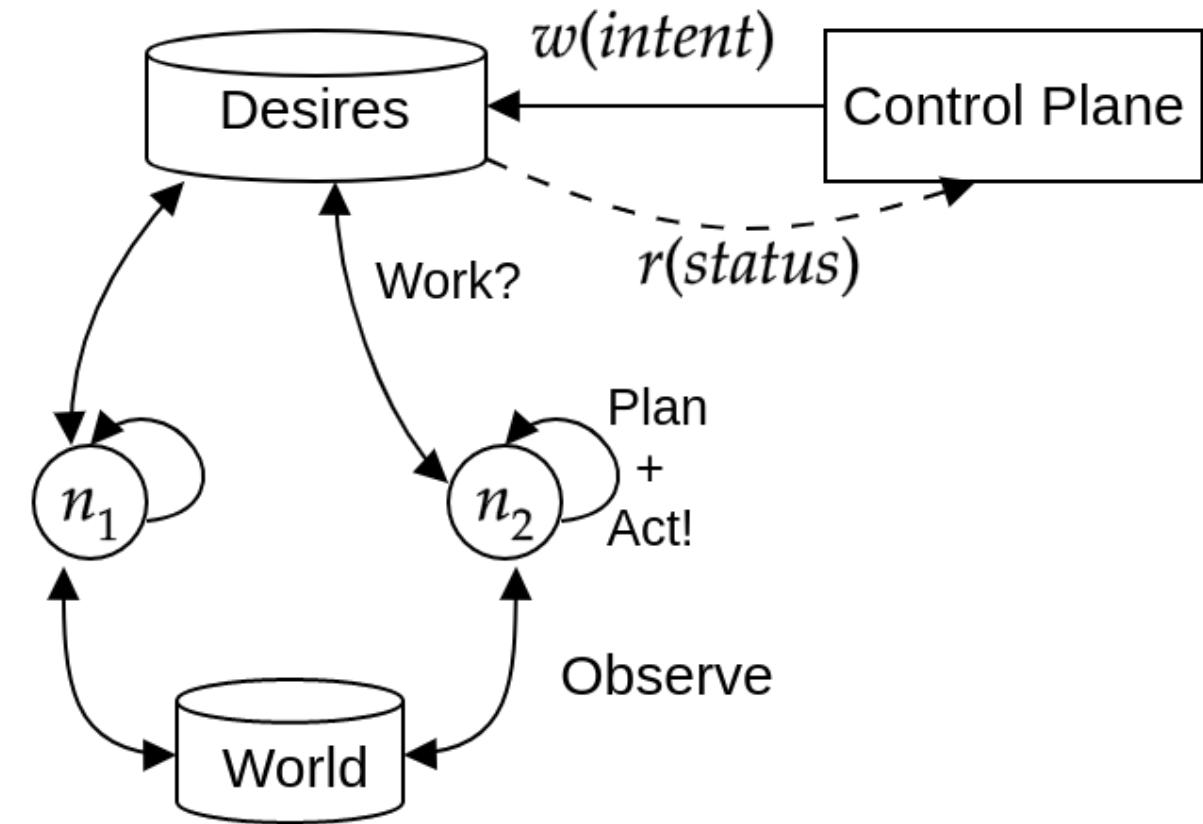
Keep it Running



Imperative Control Plane



Declarative Control Plane



Imperative Control Plane

Pros:

- Easy to implement
- Easy to understand

Cons:

- Fragile to failure
- Scaling issues
- Have to implement parallelism

Declarative Control Plane

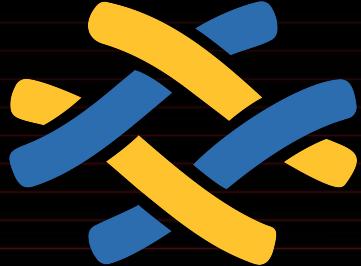
Pros:

- Lower total cost of ownership
- Parallel out of the box

Cons

- Requires different engineering mindset
- Hard to implement
- Have to stop parallelism

Imperative Tools



fabfile.org/



jenkins.io/



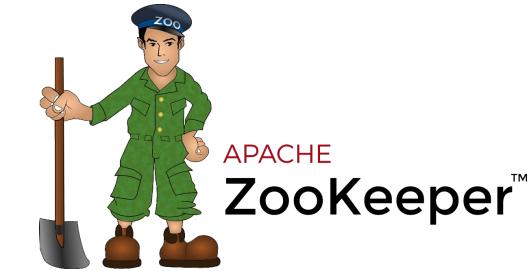
ansible.com/

- “Orchestrators”
 - ◆ Jenkins
 - ◆ Fabric
 - ◆ Ansible
- Imperative operation

Declarative Tools



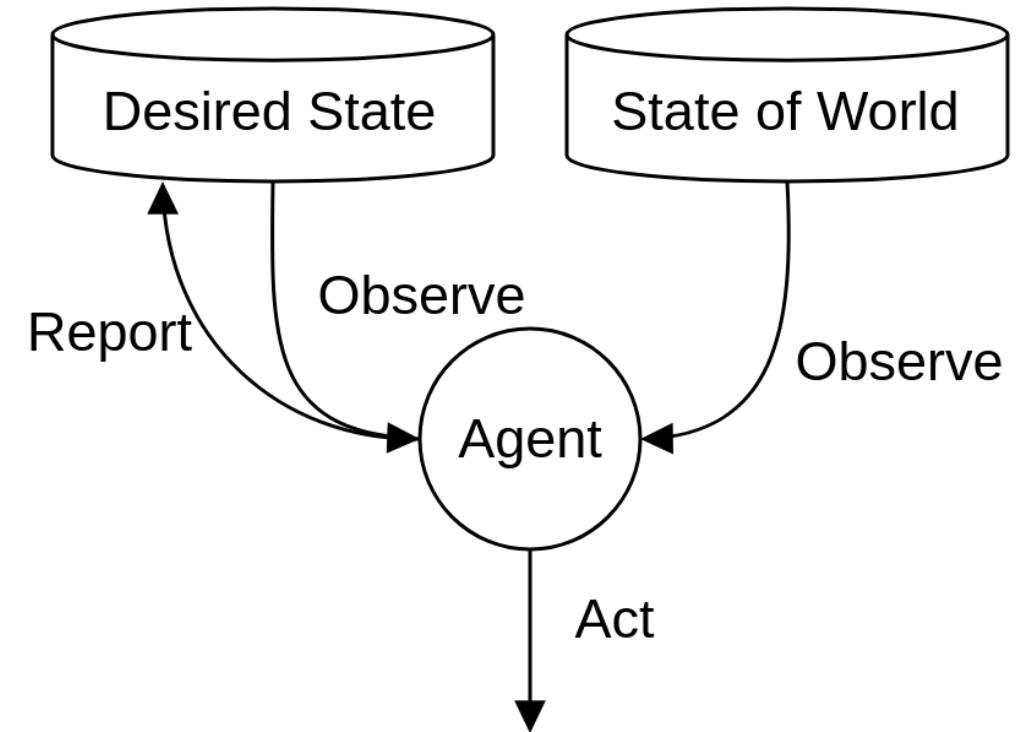
consul.io/



- Distributed Databases
- Distributed operation
- Similar to Actor model

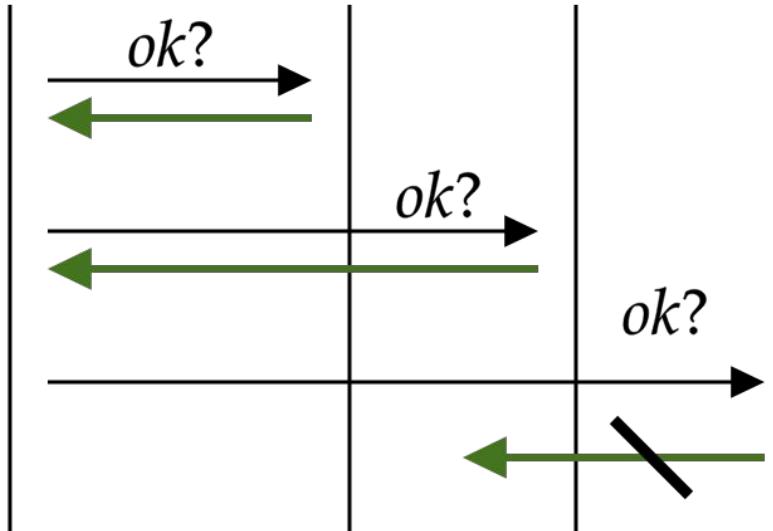
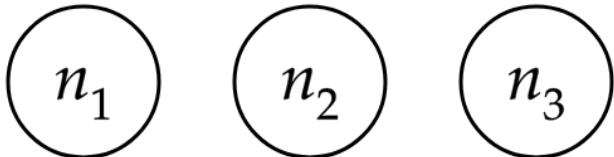
Declarative Control Plane

1. Obtain **desires** from datastore
2. Obtain **state** of world
3. If state != desire, act!



Ring Health *Imperative* Solution

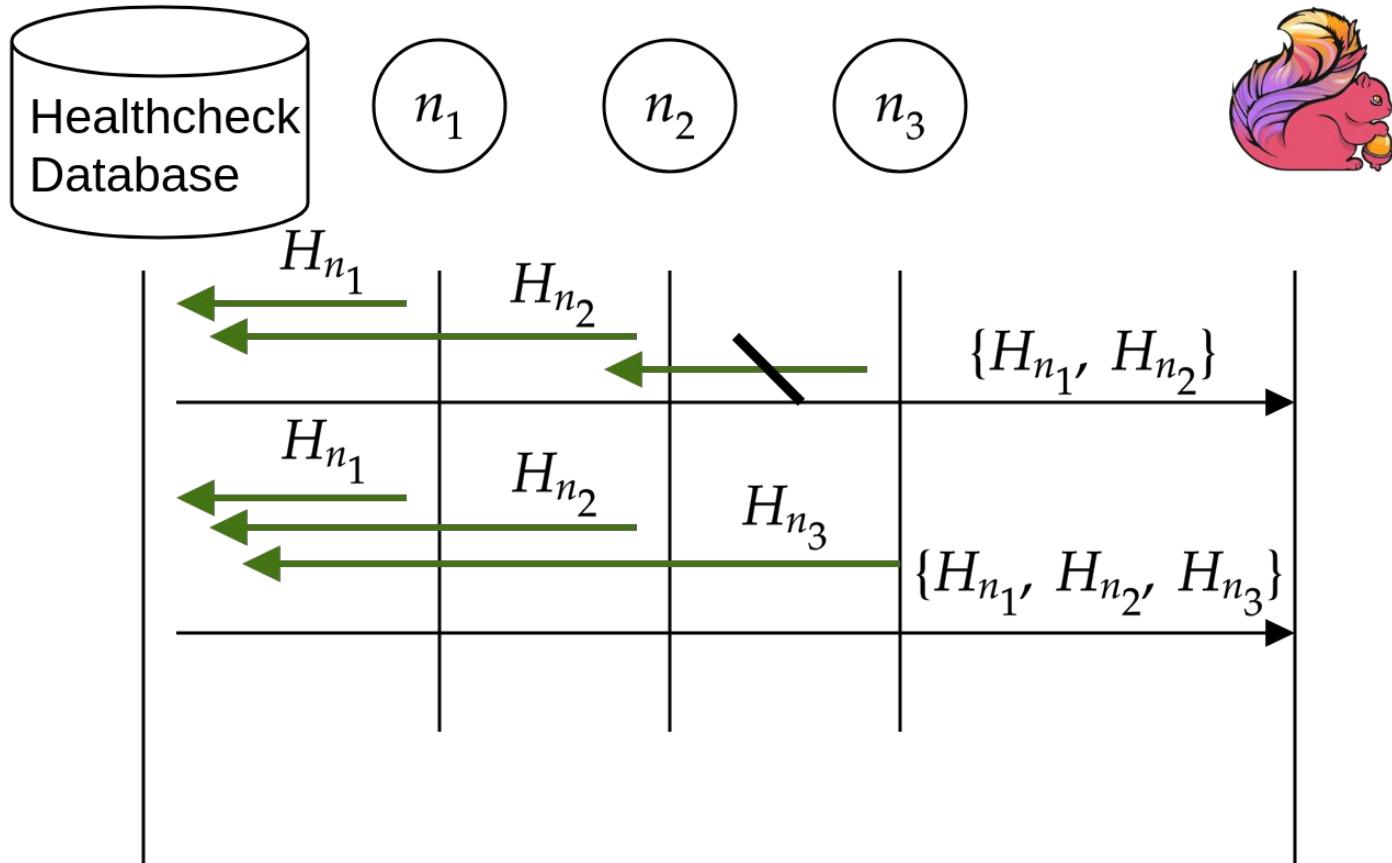
Jenkins



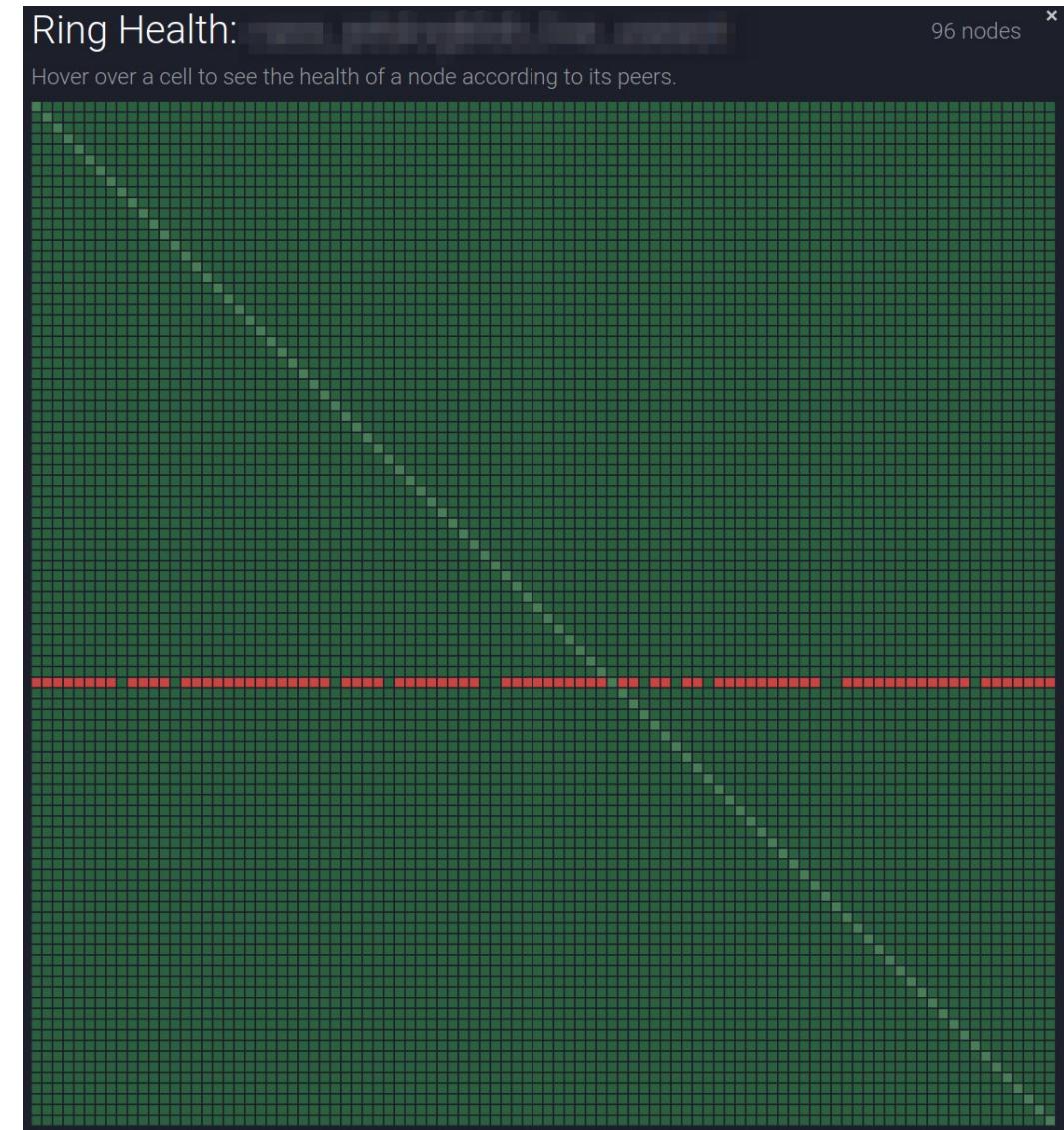
Are we healthy?

- $O(\#nodes)$ commands
- Single Network Path
- Hard to synthesize global view

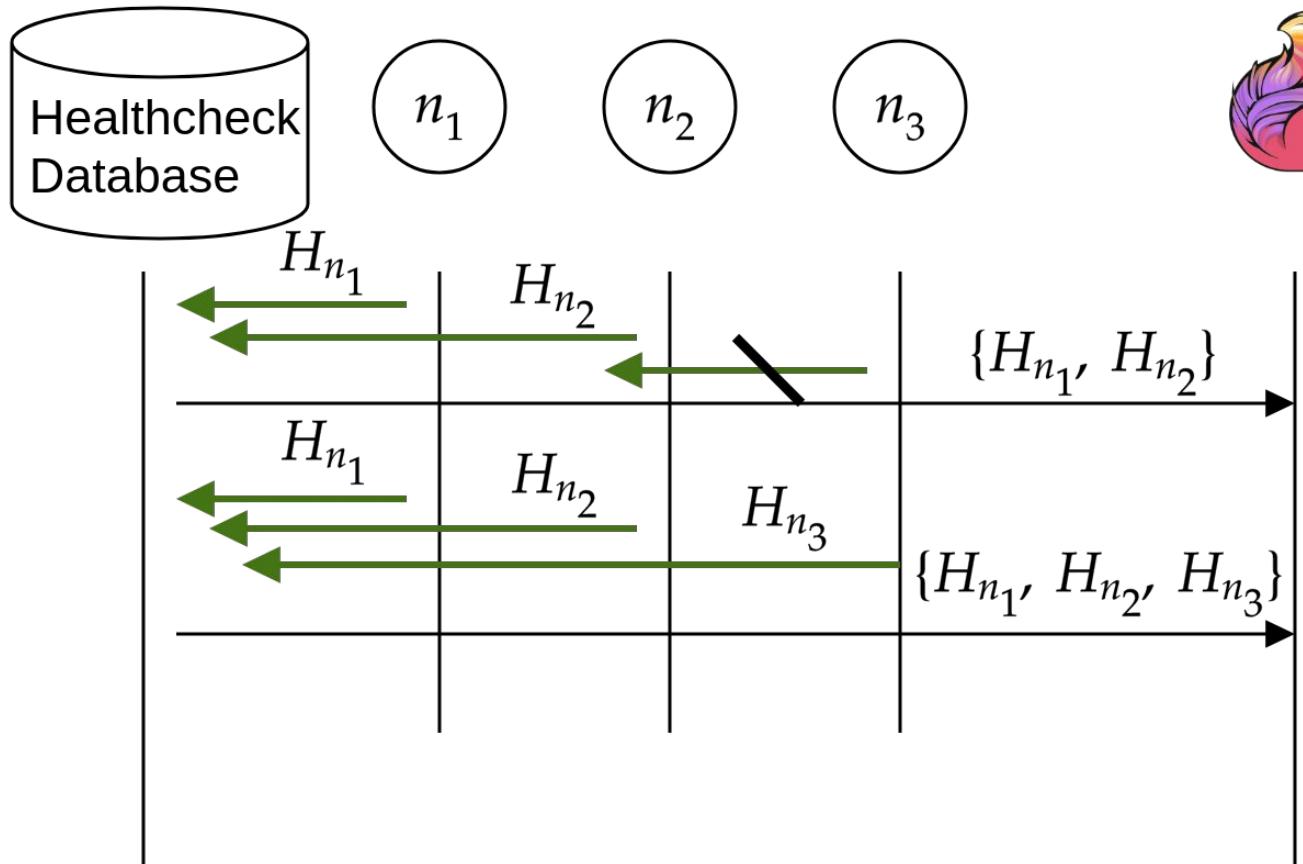
Ring Health *Declarative Solution*



Stream Processing!

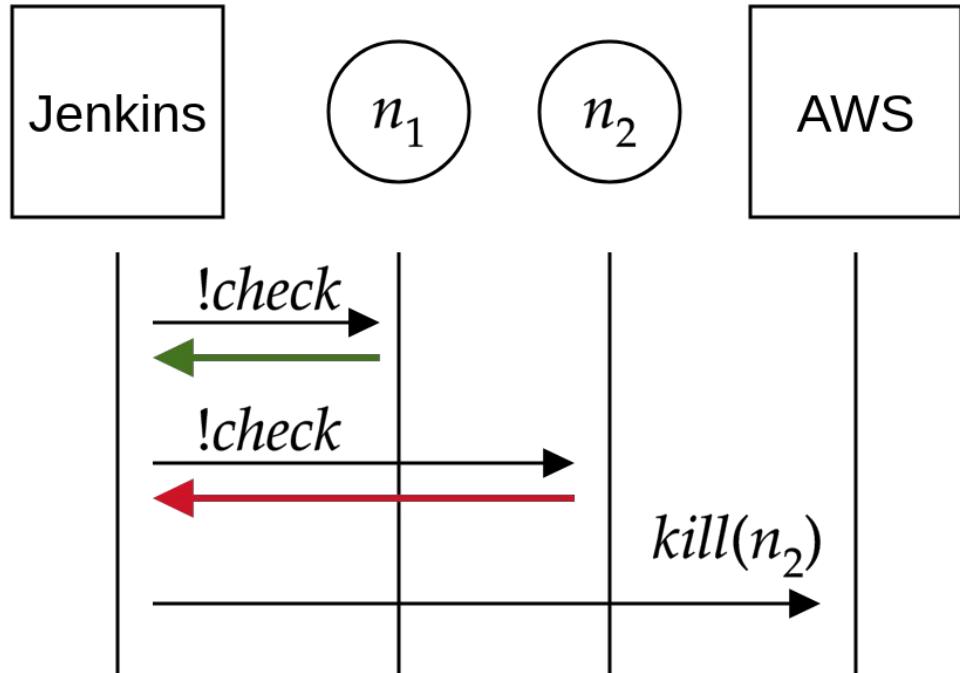


Ring Health *Declarative* Solution



- Desire: Healthy <30m
- Observe: Full cluster state
- Action: Remediate or page

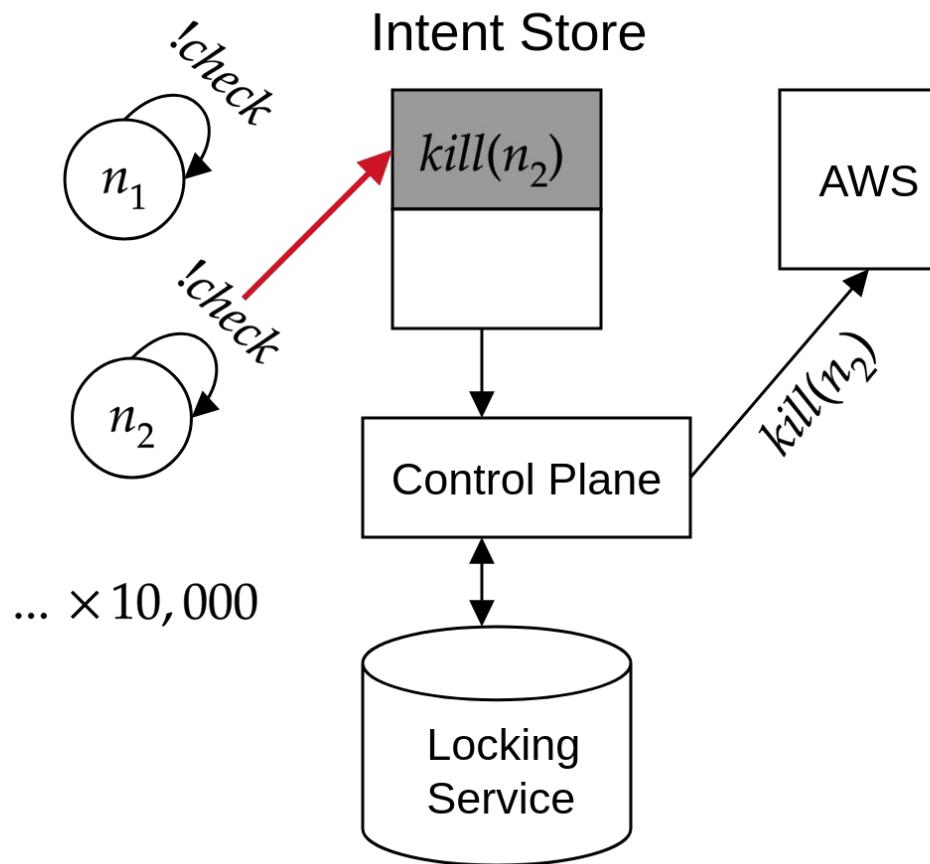
Hardware Health *Imperative* Solution



$\dots \times 10,000$

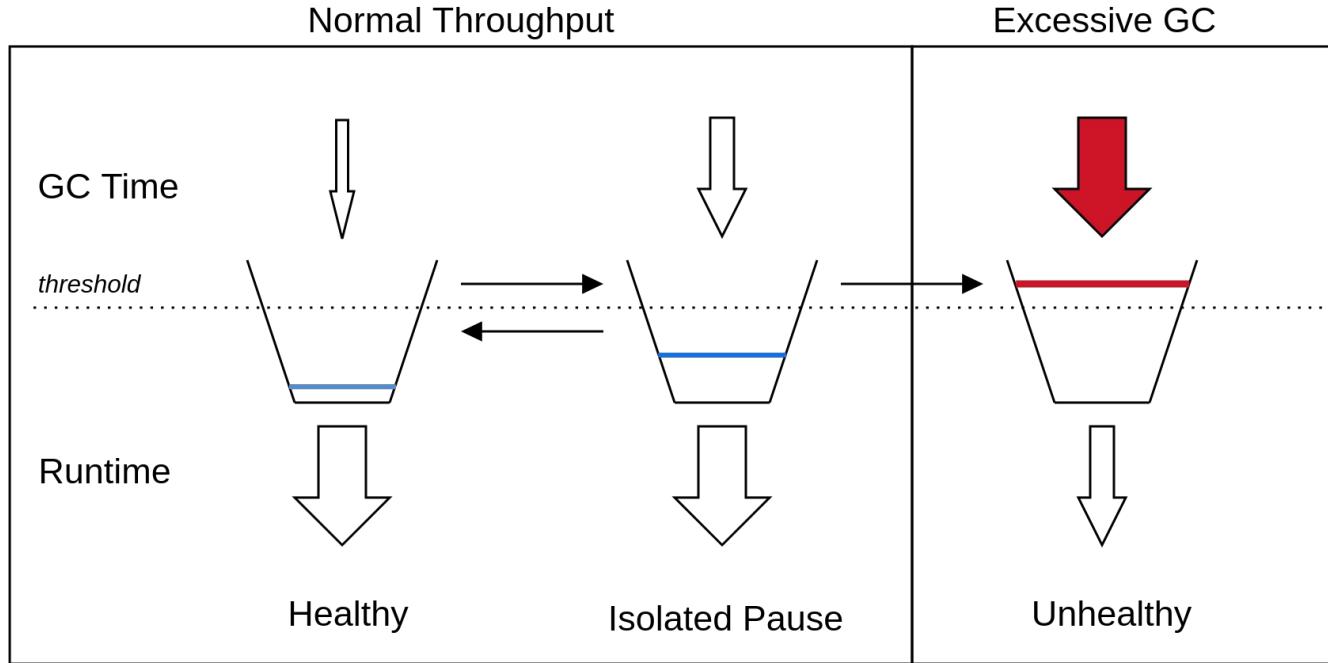
- $O(\#nodes)$ messages
- Degraded hardware is hard to SSH into
 - ◆ Failed ephemerals
 - ◆ Failed network

Hardware Health *Declarative* Solution



- Desire: Degraded hardware should die <10m
- Observe: Local disks, creds, filesystems
- Action: Request termination

JVM Health *Declarative Solution*

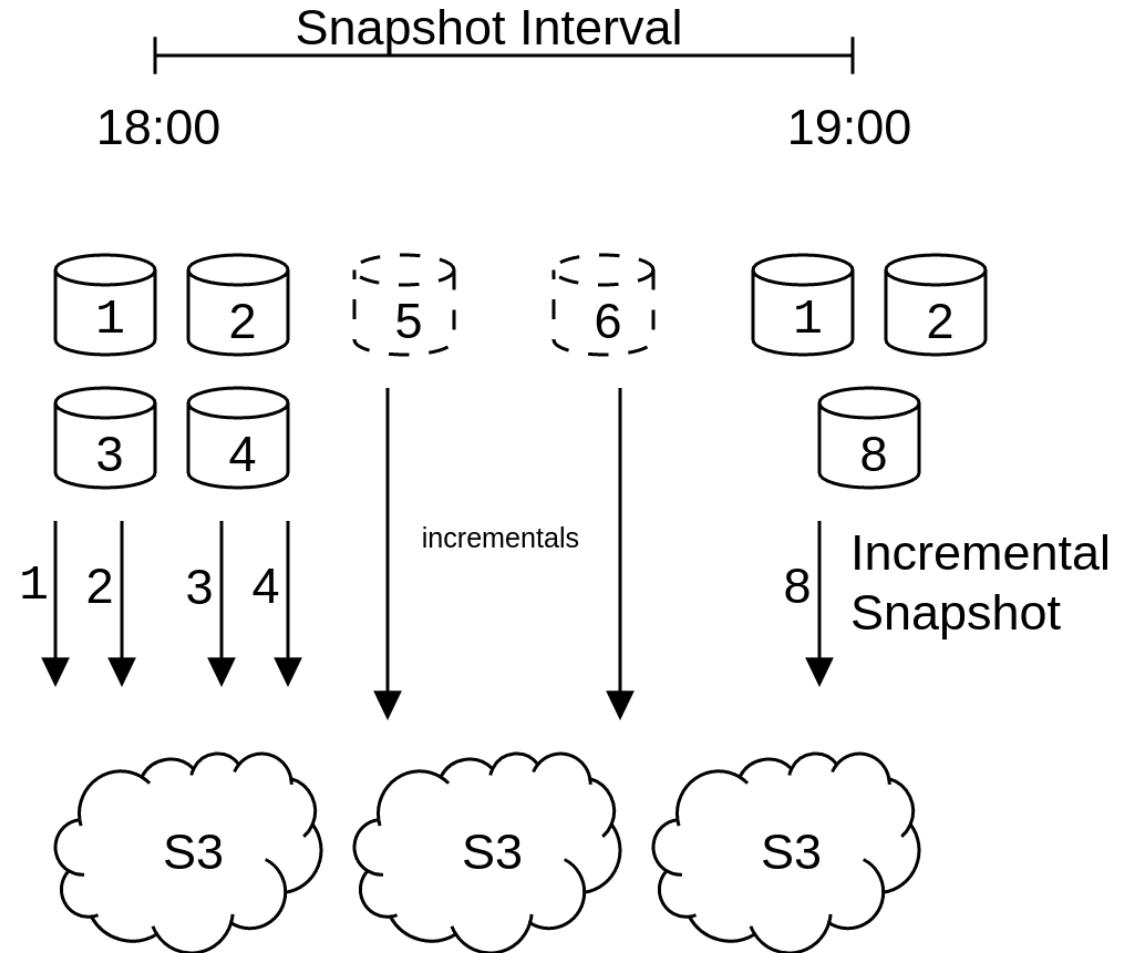


- Desire: JVM Throughput $> 90\%$
- Observe: GC vs Runtime
- Action: core dump + terminate

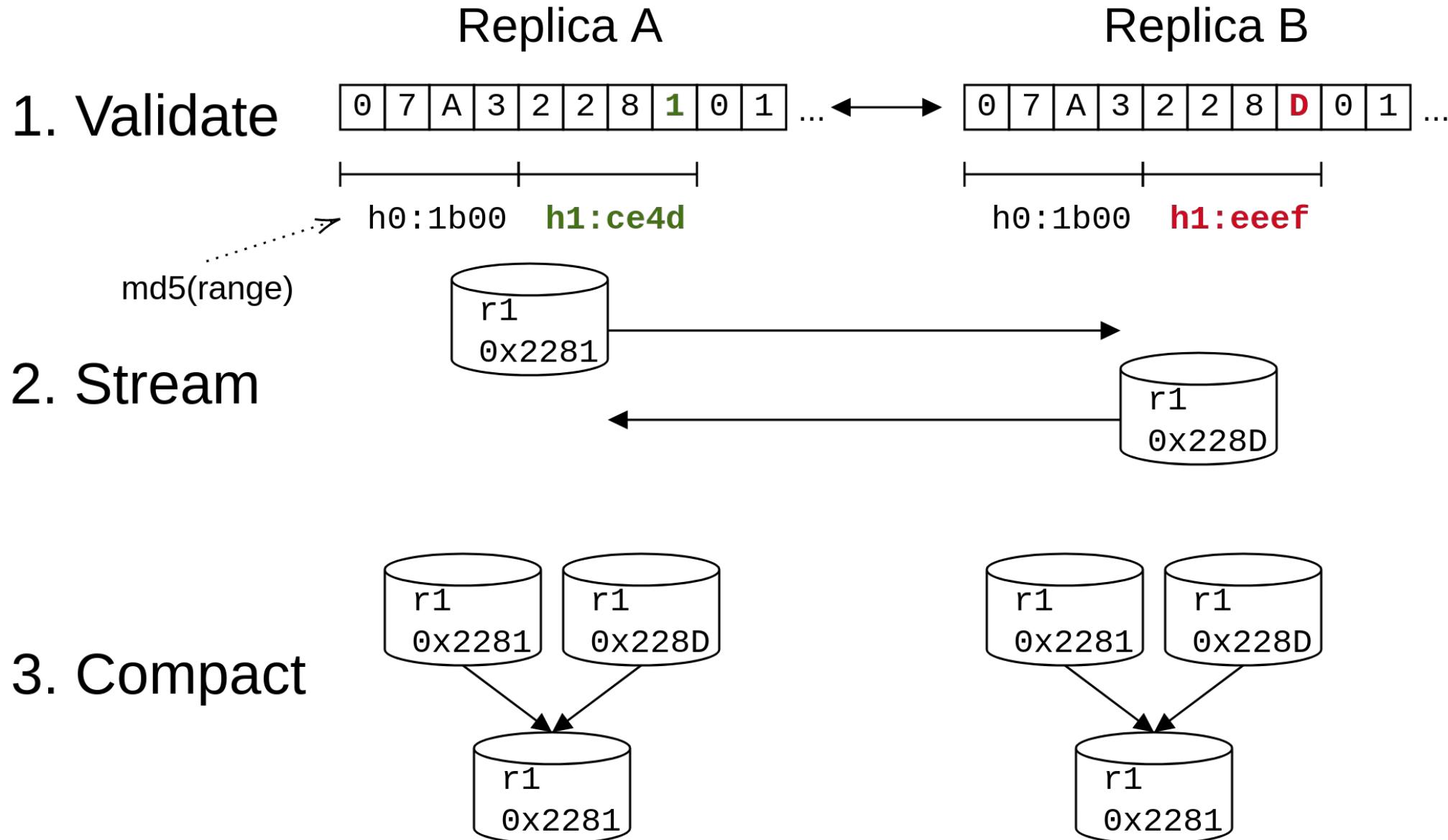
<https://github.com/Netflix-Skunkworks/jvmquake>

Backup *Declarative* Solution

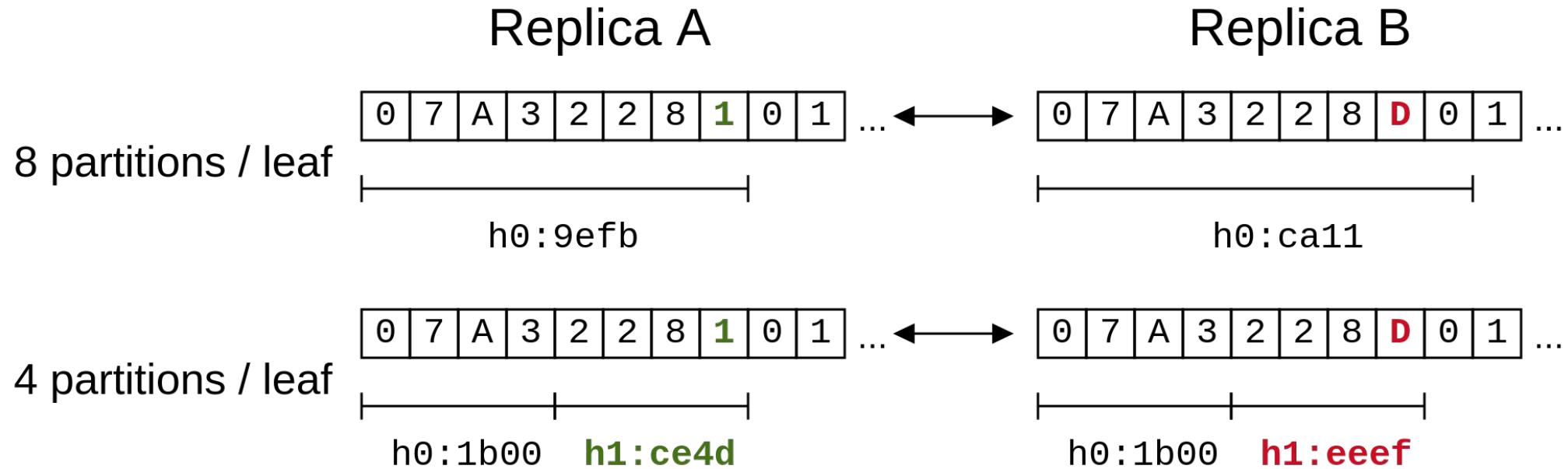
- Desire: Full snapshot every interval
- Observe:
 - ◆ State of data
 - ◆ State of S3
- Action:
 - ◆ Upload diff



Case Study: Cassandra Repair



Case Study: Cassandra Repair



$$\begin{aligned} resolution &= \frac{|partitions|}{|leaf|} \\ cost_{memory} &\approx 192 \text{ bytes} * |leaf| \end{aligned}$$

Due to this Cassandra must limit single tree $|leaf|$ to $\lesssim 2^{18}$

At Scale Subrange is Required

$$192 \frac{\text{bytes}}{\text{partition}} * 2^{27} \frac{\text{partition}}{\text{replica}} \approx 24 \frac{\text{GiB}}{\text{replica}}$$

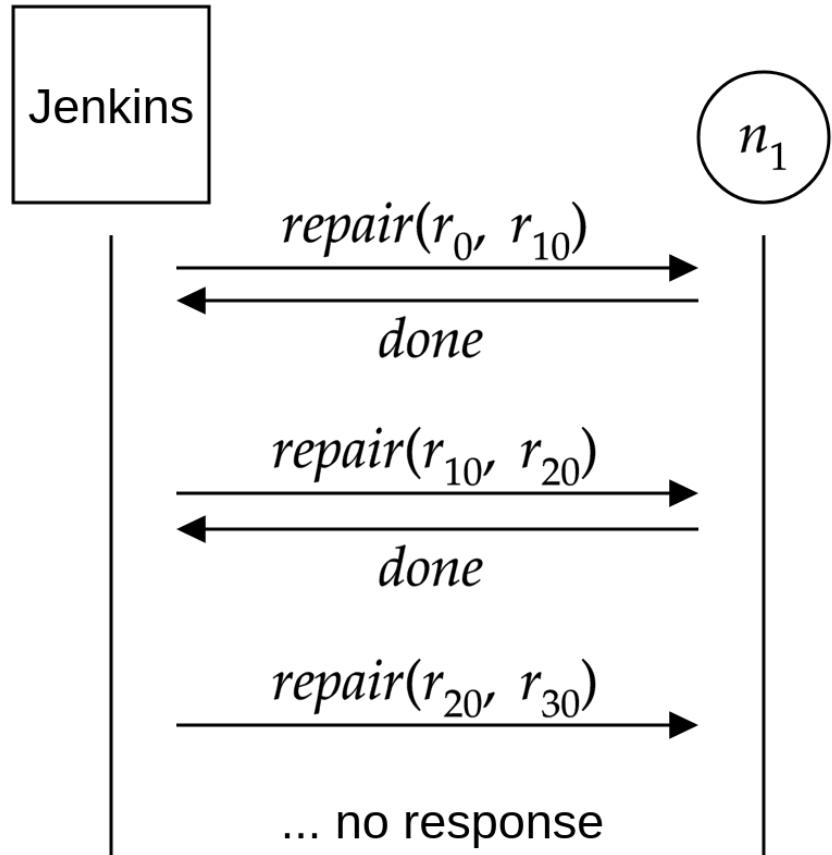
$\rightarrow \frac{24 \text{ GiB}}{\text{replica}} * 9 \text{ replica} \approx 216 \text{ GiB}$ We can OOM

OR

$\text{resolution} \approx \frac{2^{27} \text{ partitions}}{2^{18} \text{ leaves}} \approx 512$ Do 512x work ...

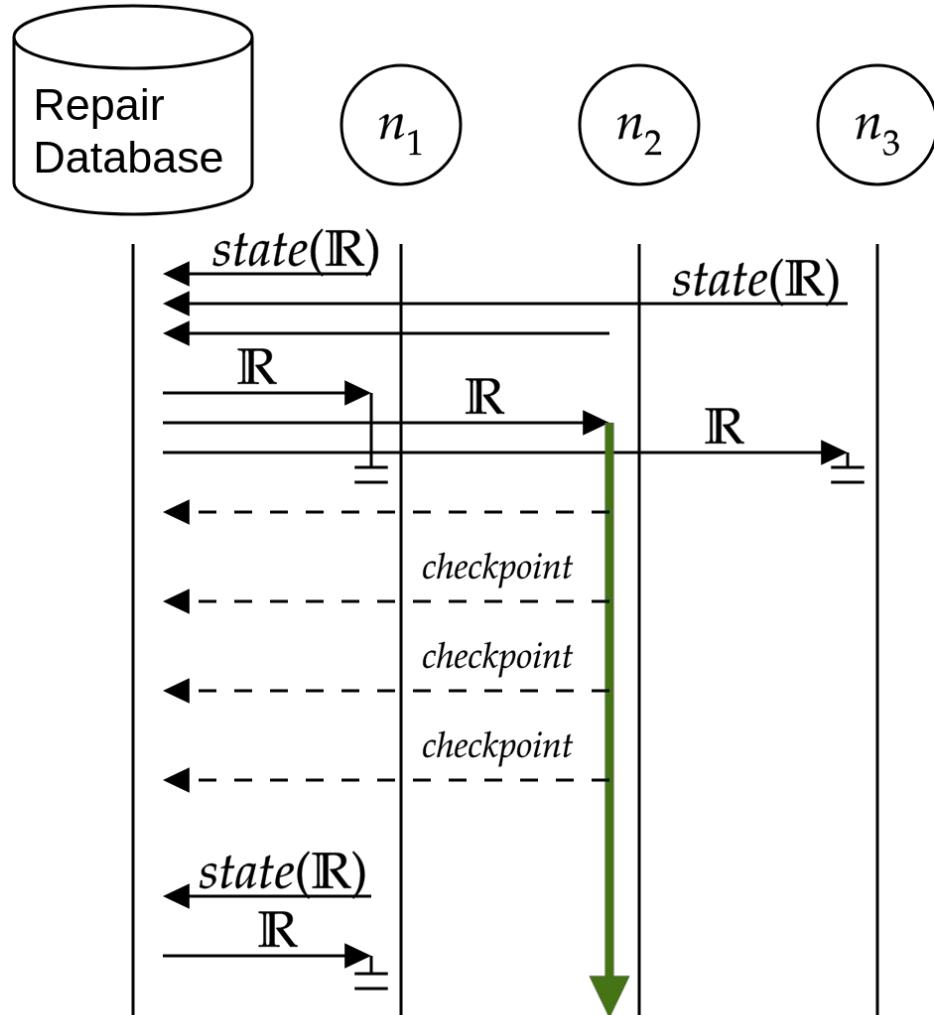
```
cass -useast1c us-east-1
nt cfstats
Number of keys (estimate): 112571545
```

Repair *Imperative* Solution



- O(lots) commands
- Remote JMX = :(
- **Single point of failure** = :(
- 1 month job duration = D:

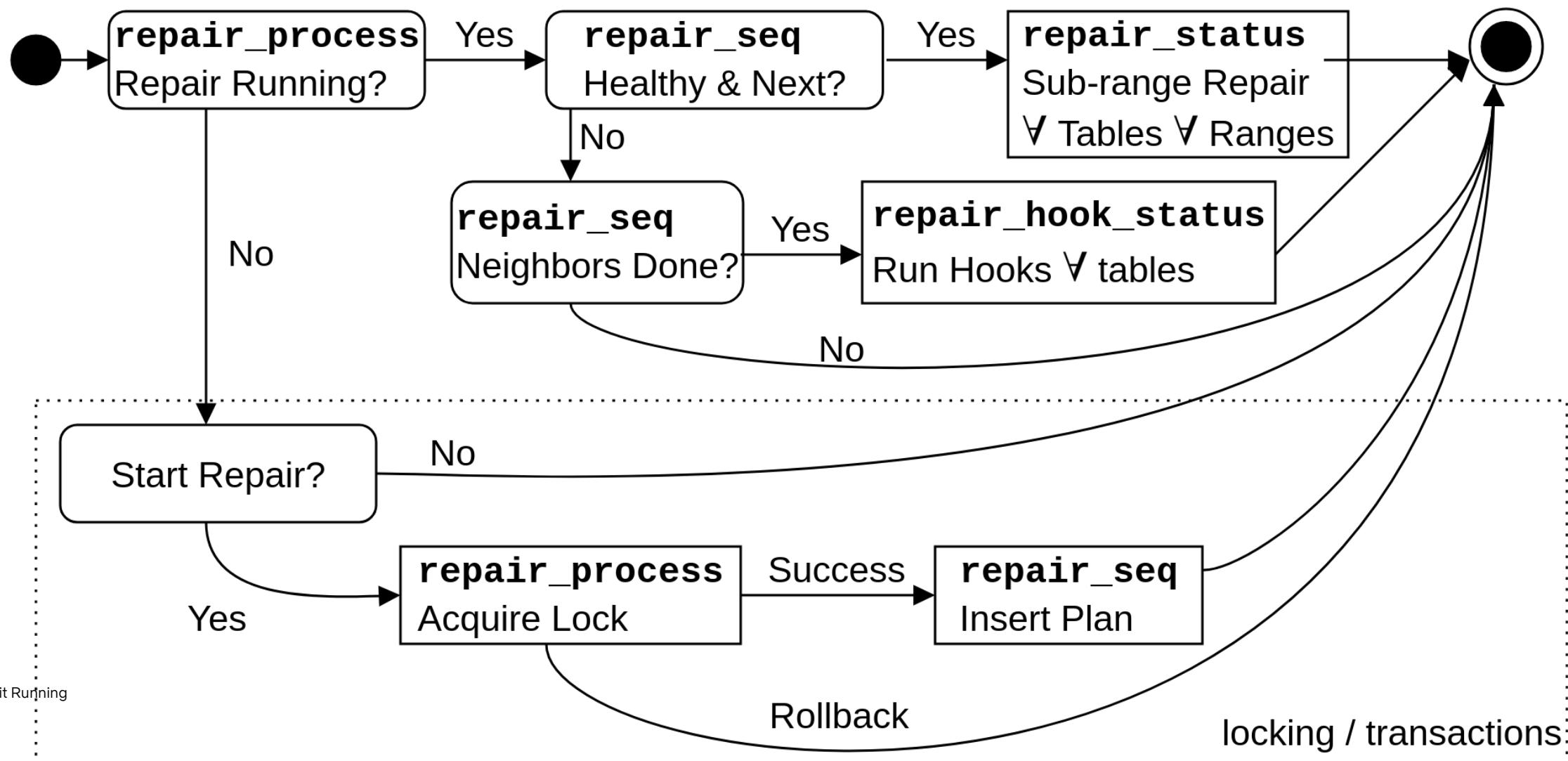
Repair *Declarative* Solution



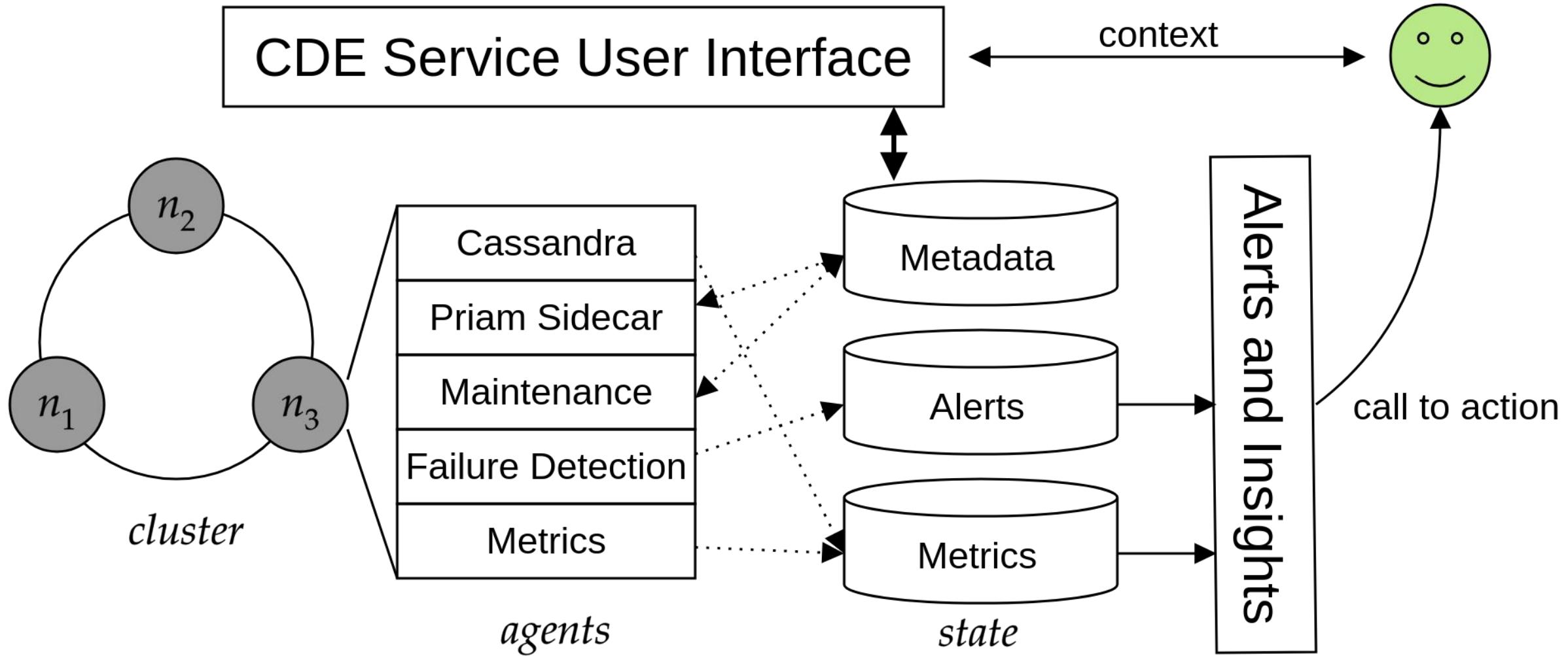
- Distributed Control Plane
- Makes progress when database is available
- Every node follows **repair state machine**

Repair Declarative Solution (#14346)

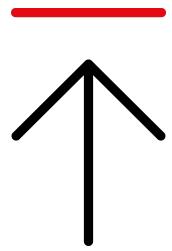
Repair State Machine



Cassandra Ecosystem



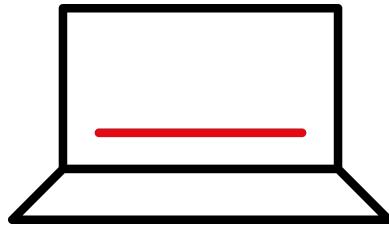
Story of C*



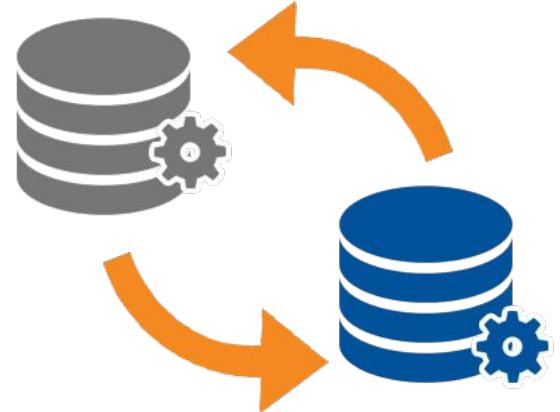
Inception



Provision



Keep it running

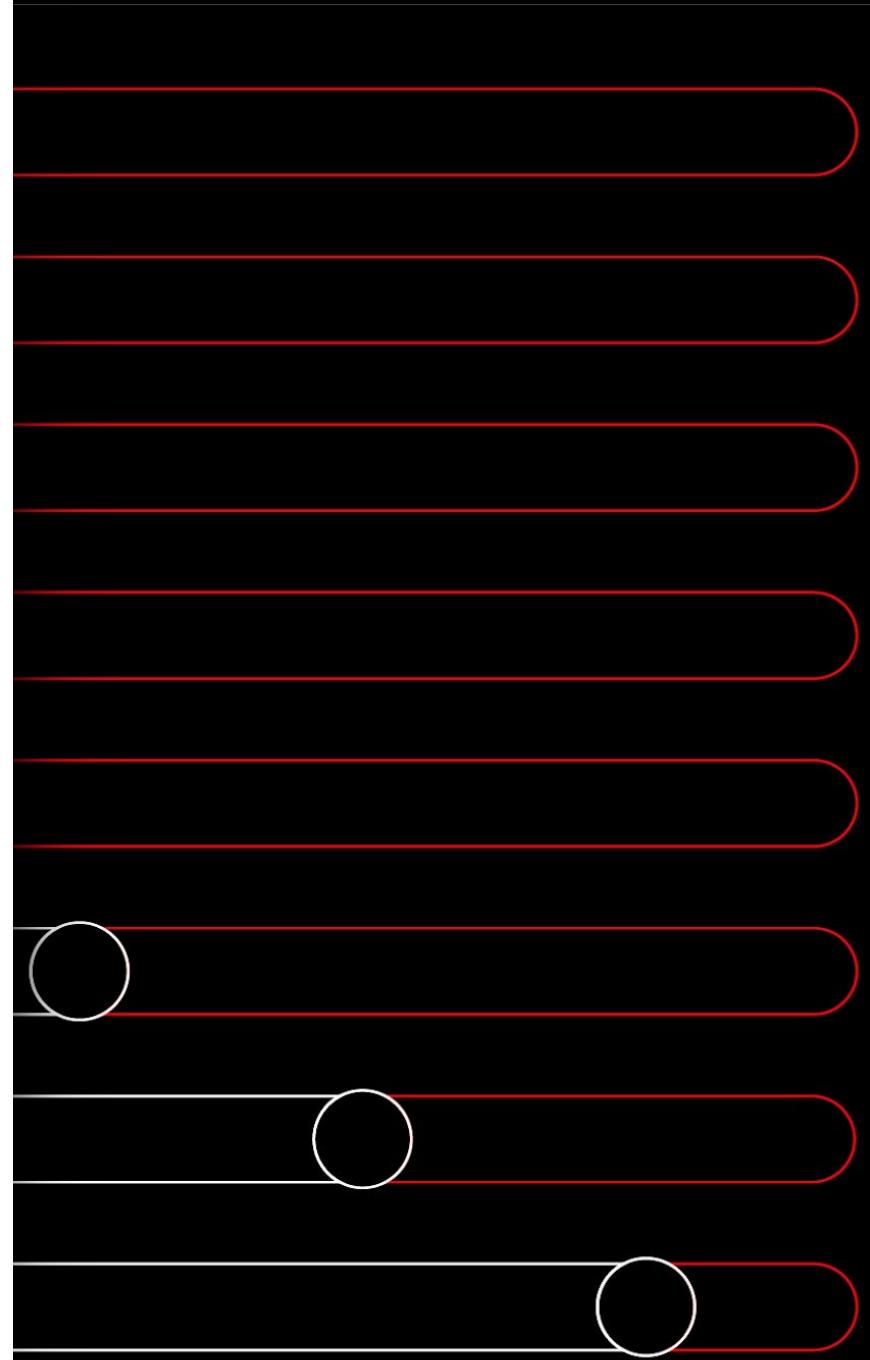
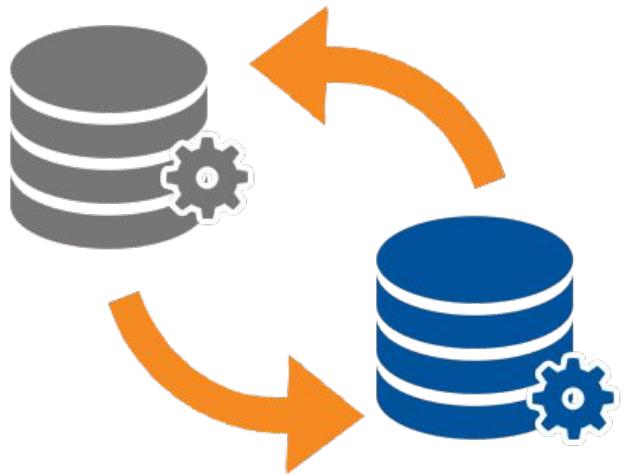


Migration

ANYTHING
THAT CAN GO WRONG
WILL GO WRONG

Murphy's Law

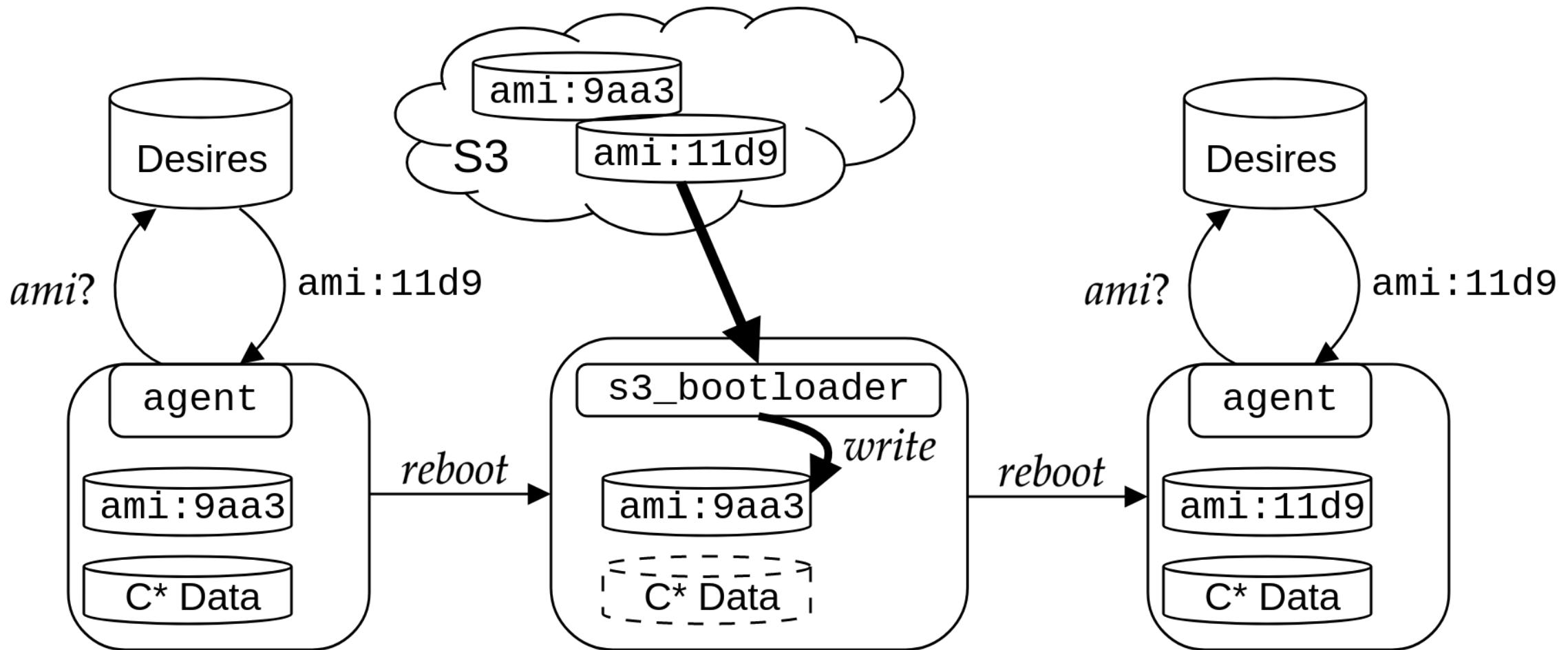
Migration



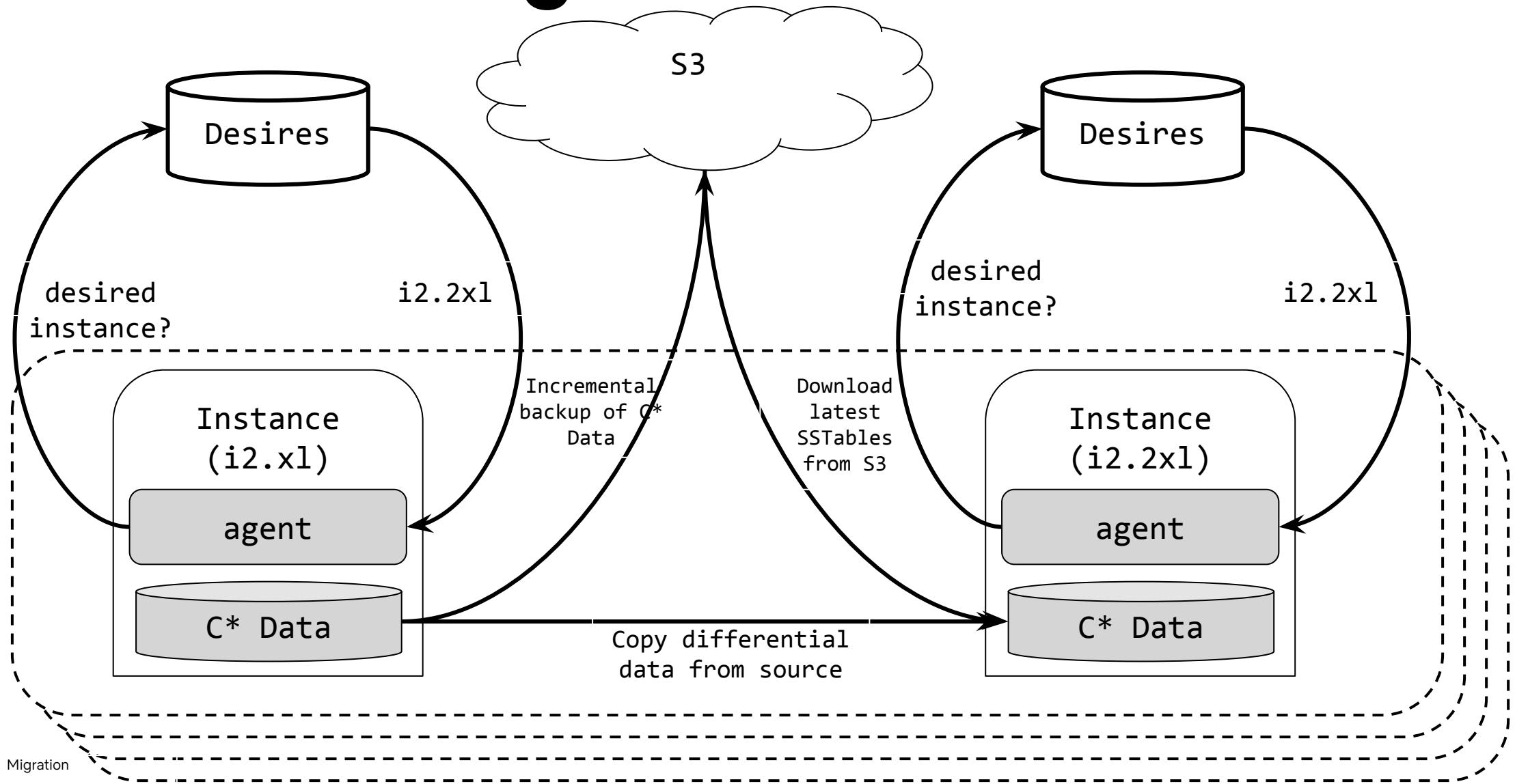
Migration Philosophy

- Immutable infrastructure
- Seamless and transparent migrations

Software Migration



Hardware Migration



cass_foobar

PIPELINES

INFRASTRUCTURE

TASKS

Infrastructure

CLUSTERS

LOAD BALANCERS

SECURITY GROUPS

PROPERTIES

Clear All

SEARCH

ACCOUNT

 [REDACTED]

REGION

 eu-west-1 us-east-1

STACK

 (none)

DETAIL

 (none) euwest1a euwest1b euwest1c useast1c useast1d useast1e

STATUS

 Healthy Unhealthy Disabled Starting Edit multiple Show Instances with details

Filtered by: REGION: eu-west-1



Hardware Migration: The Numbers

$$500 \frac{\text{GiB}}{\text{node}} / 200 \frac{\text{mb}}{\text{second}} = 5.96 \frac{\text{hours}}{\text{node}}$$

$$5.96 \frac{\text{hours}}{\text{node}} * 288 \text{ node} = 2.35 \text{ months}$$

$$500 \text{ GiB} / 2000 \frac{\text{mb}}{\text{second}} = 0.6 \text{ hours}$$

$$0.6 \text{ hours} + 10 \frac{\text{minutes}}{\text{node}} * 288 \text{ node} = 2.02 \text{ days}$$

$$500 \text{ GiB} / 2000 \frac{\text{mb}}{\text{second}} = 0.6 \text{ hours}$$

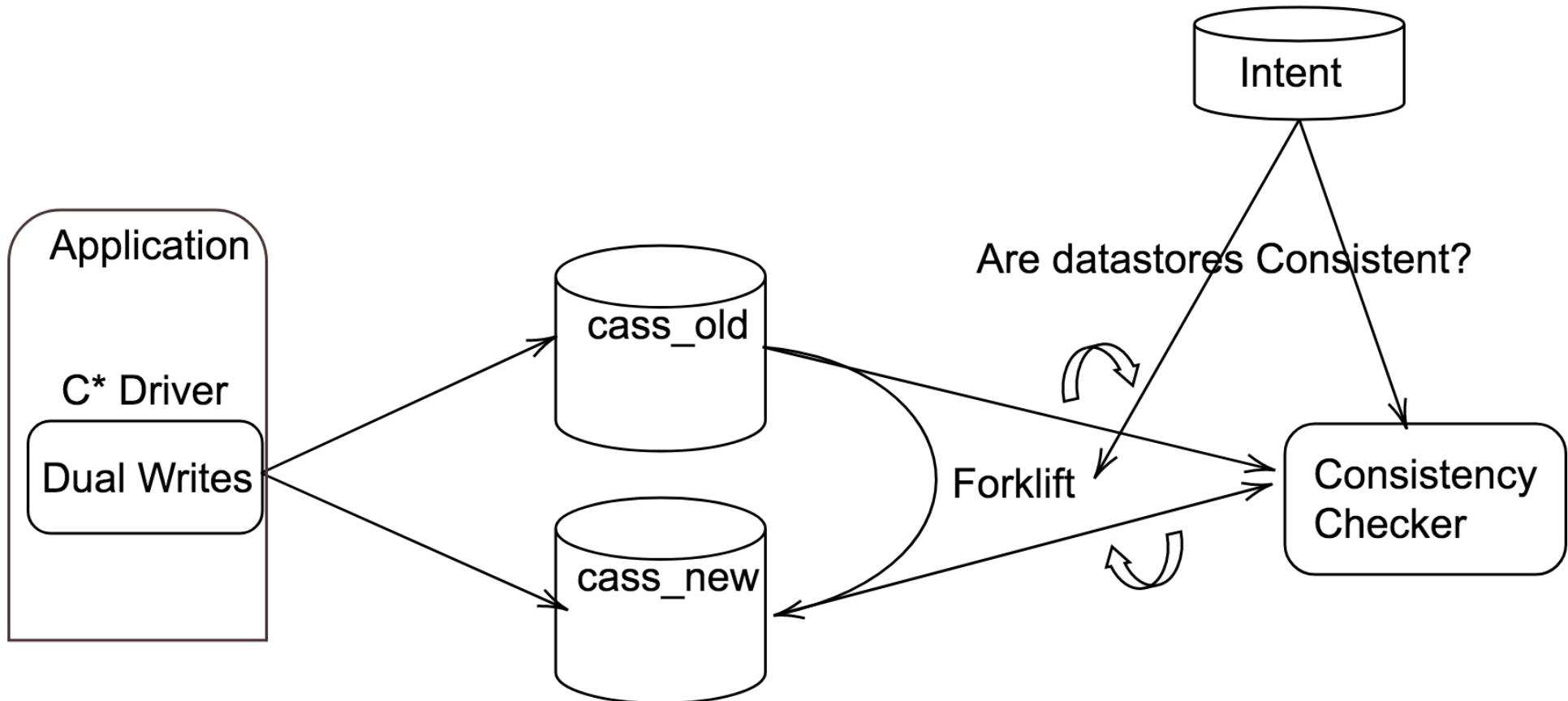
$$0.6 \text{ hours} + 20 \frac{\text{minutes}}{\text{zone}} * 3 \text{ zone} = 1.596 \text{ hours}$$

Naive solution,
sequential transfer

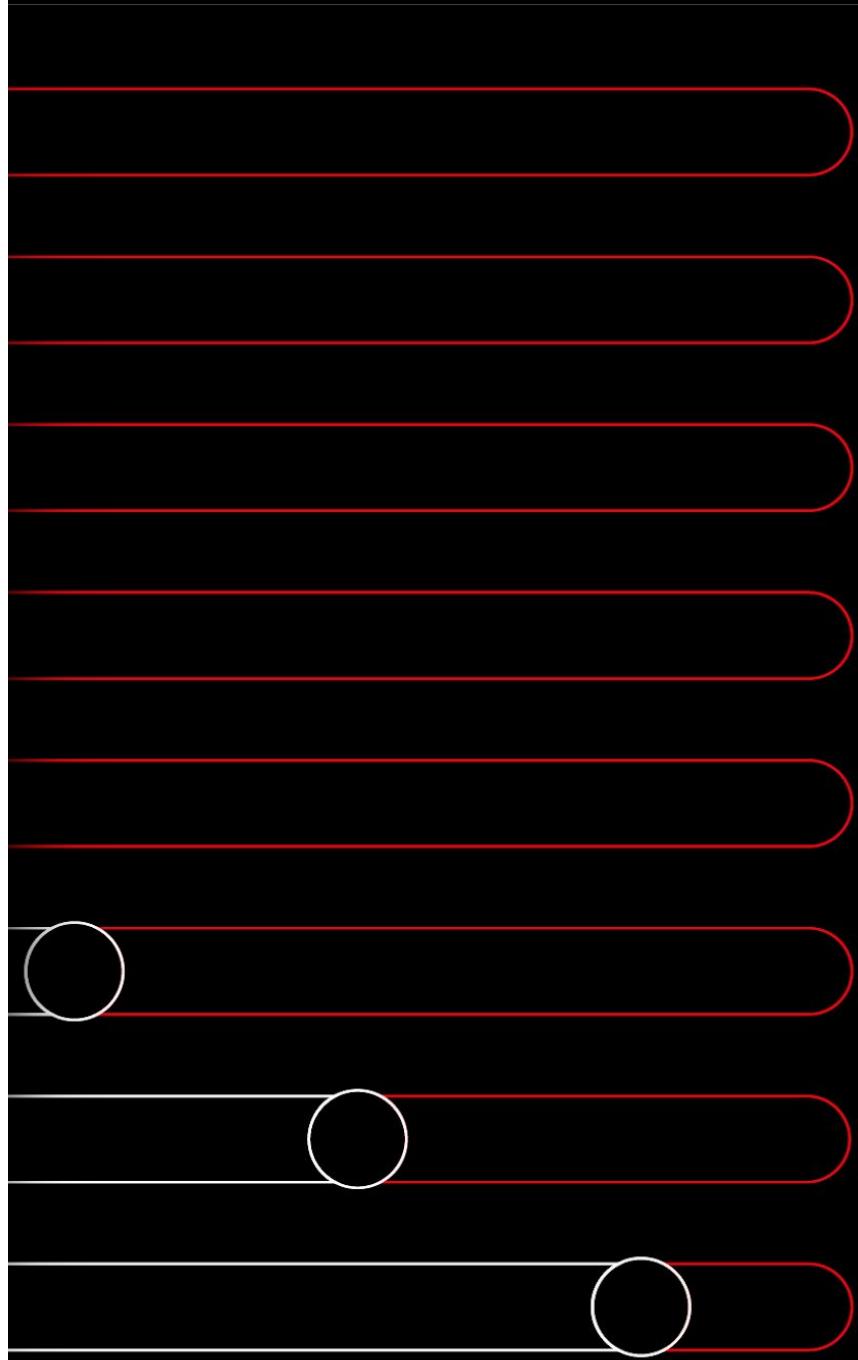
Parallel S3 download
sequential fixup

Parallel download and
zone wide fixup

Data Migration



Murphy's law



Cautionary Tale: Rarely run automation

- Auto remediation to clean up stale backups, repair snapshots
 - ◆ Triggers only above FATAL disk threshold
 - ◆ Assumption mismatch

Cautionary Tale: Rarely run automation

- Auto remediation to clean up stale backups, repair snapshots
 - ◆ Triggers only above FATAL disk threshold
 - ◆ Assumption mismatch
- Result:
 - ◆ Automation ended up deleting real backup data

Do you restore your backups constantly?

Take Away

Invest in context sharing, tooling and declarative(desire) infrastructure to run Apache Cassandra at petabyte scale

Thank You.

