Real world tales of repair

THE LAST PICKLE

CASSANDRA SUMMIT - SEPTEMBER 2016

Alexander Dejanovski
@alexanderdeja

Consultant
www.thelastpickle.com

Datastax MVP for Apache Cassandra

About The Last Pickle

We help people deliver and improve Apache Cassandra based solutions.

With staff in 5 countries and over 50 years combined experience in Apache Cassandra.


FINEST MENSWEAR
SPRINGFIELD
UNIQUE AND ORIGINAL
1988
BORN IN EUROPE

What and why ?

Full repair

Incremental repair

How to make it work

Automated repairs

What is repair ?

A maintenance operation that (briefly) restores strong consistency throughout the cluster

Why do we need repair ?

- Eventual consistency
- Downtime / failure recovery
- Safe deletes

Tombstones need repair too

Missing tombstones can lead to zombie data
(repair within gc_grace_seconds)

What and why ?

Full repair

Incremental repair

How to make it work

Automated repairs

How does anti-entropy repair works ?

Reads all data

How does anti-entropy repair works ?

Reads all data

Calculates hashes

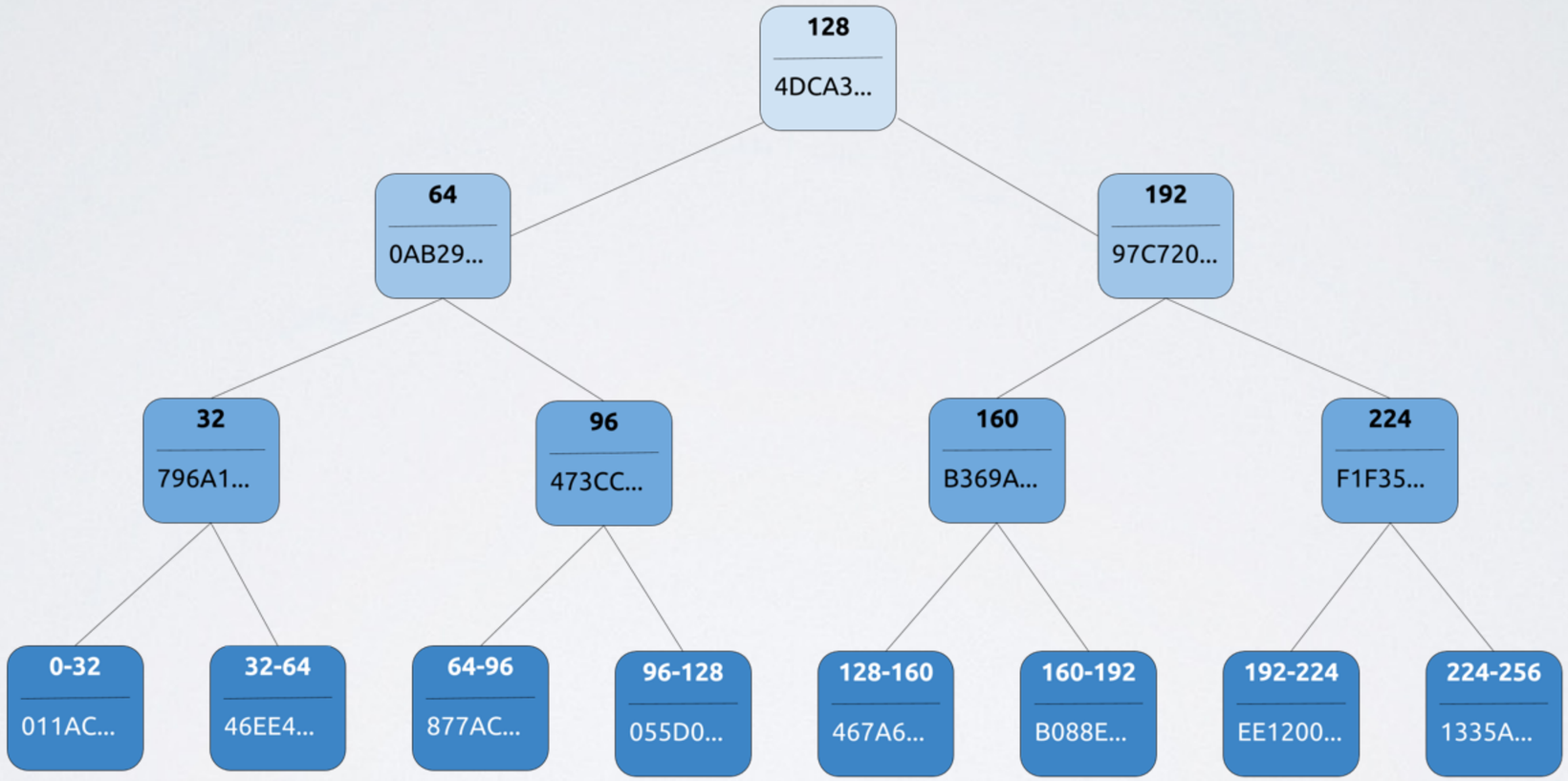
How does anti-entropy repair works ?

Reads all data
Calculates hashes
Compares hashes

How does anti-entropy repair works ?

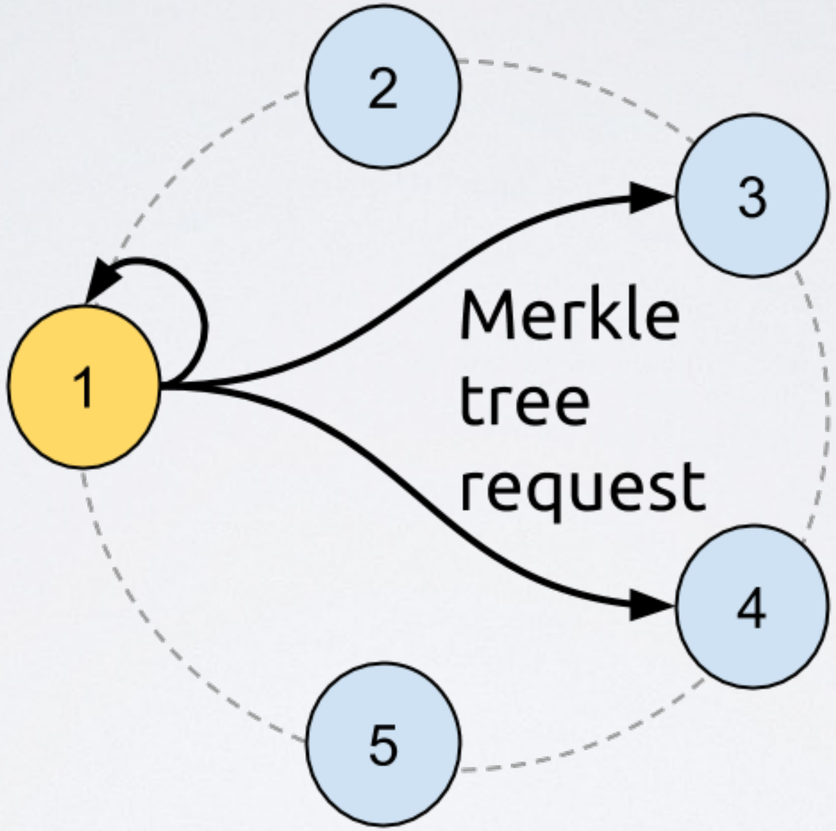
Reads all data
Calculates hashes
Compares hashes
Streams mismatching partitions

How does anti-entropy repair works ?

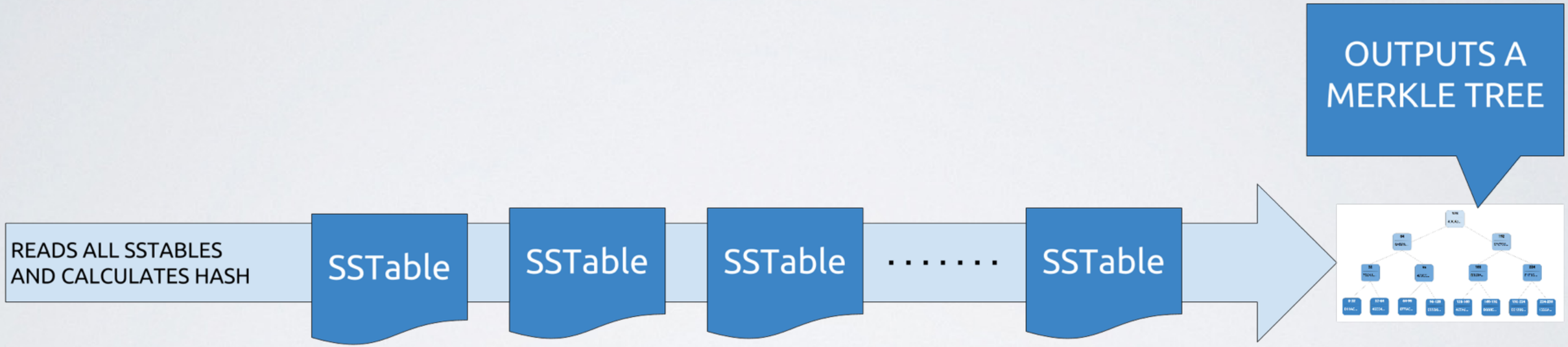


www.thelastpickle.com

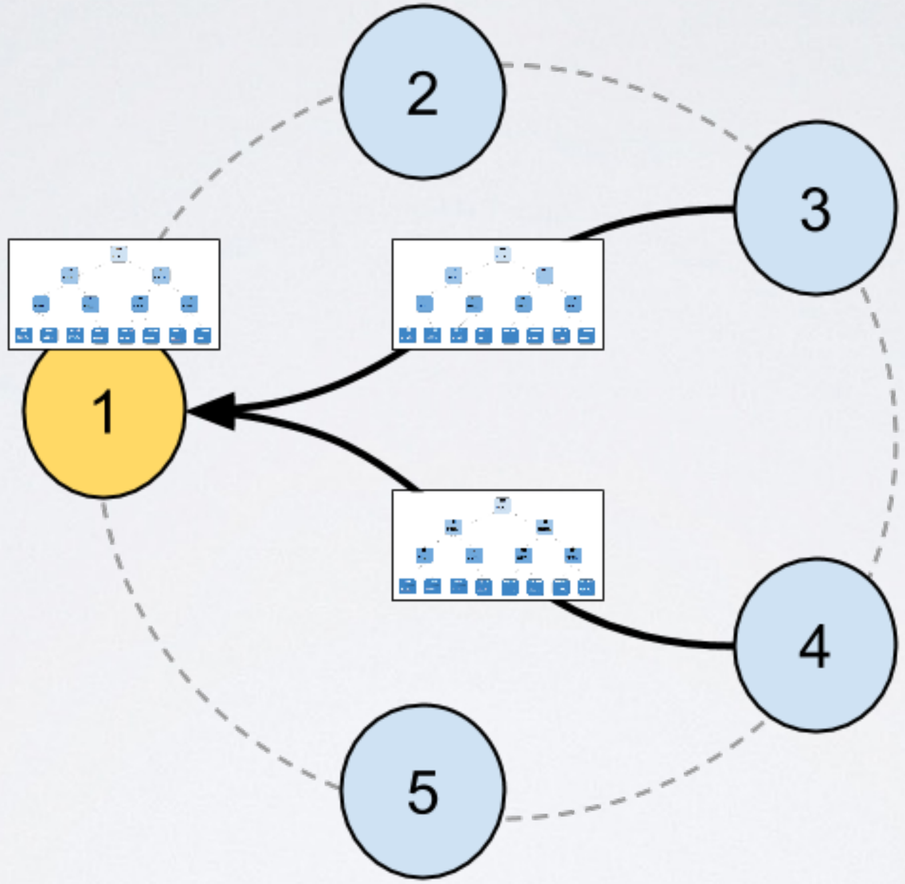
Merkle tree is requested to all replicas



Validation compaction

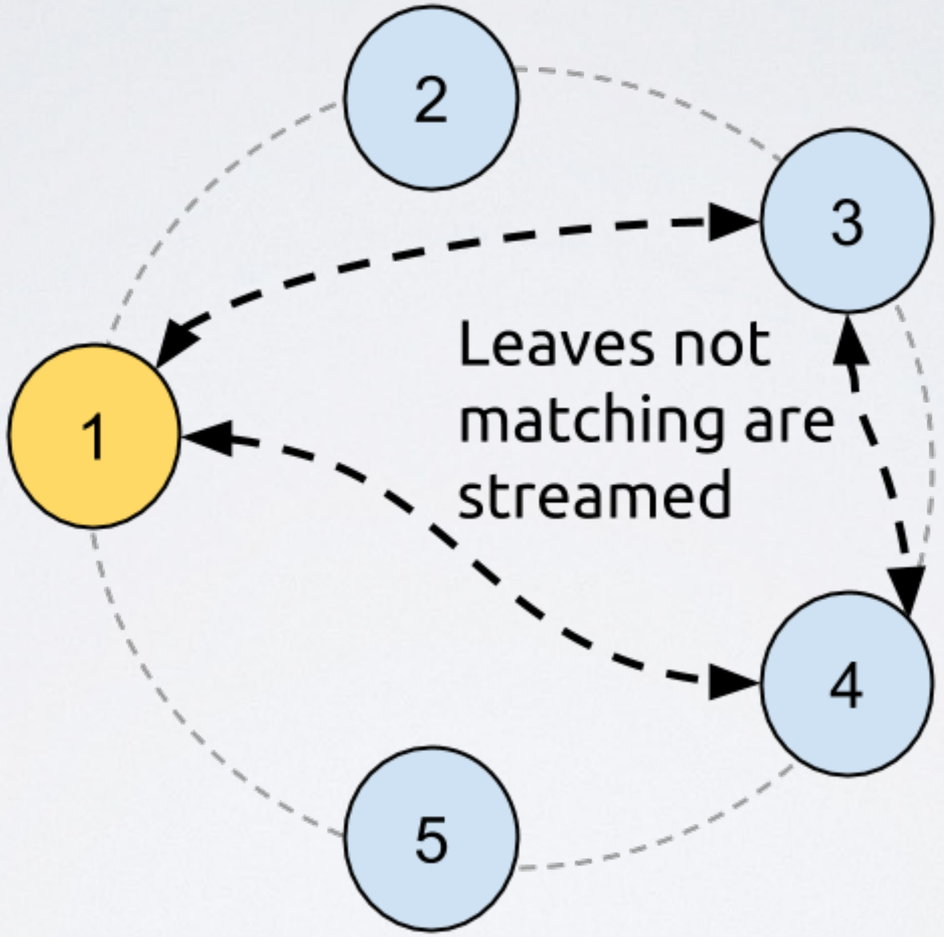


Merkle tree comparison



www.thelastpickle.com

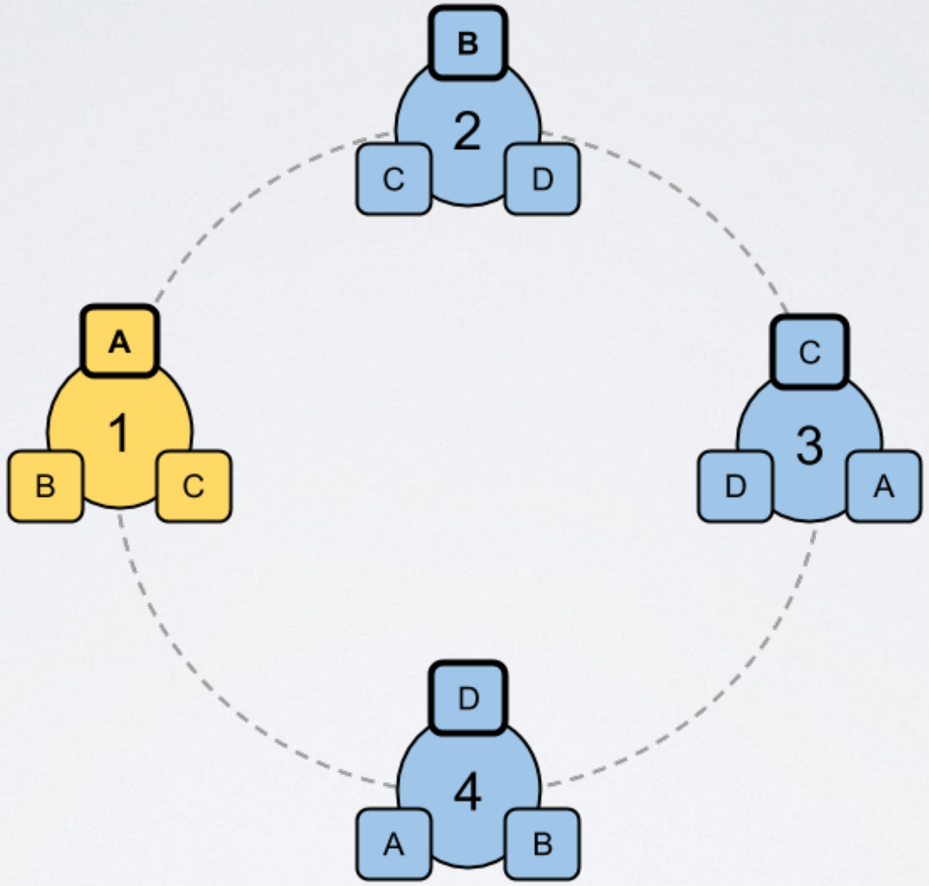
Streaming



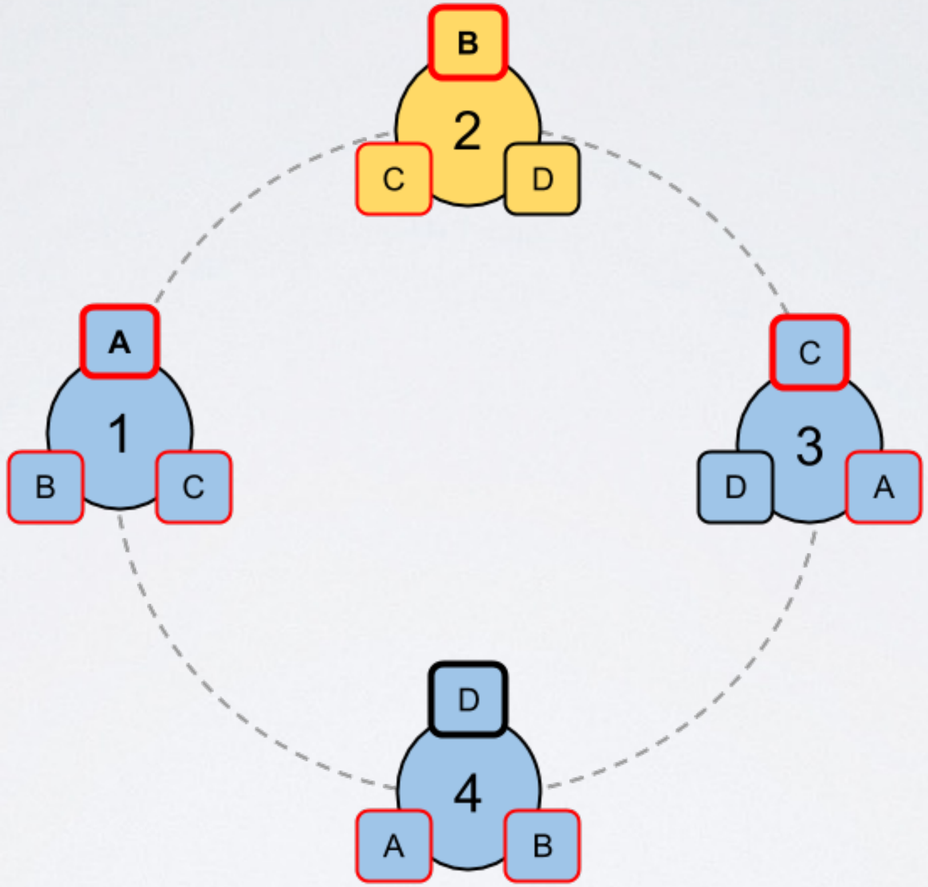
How do we run repair ?

nodetool repair

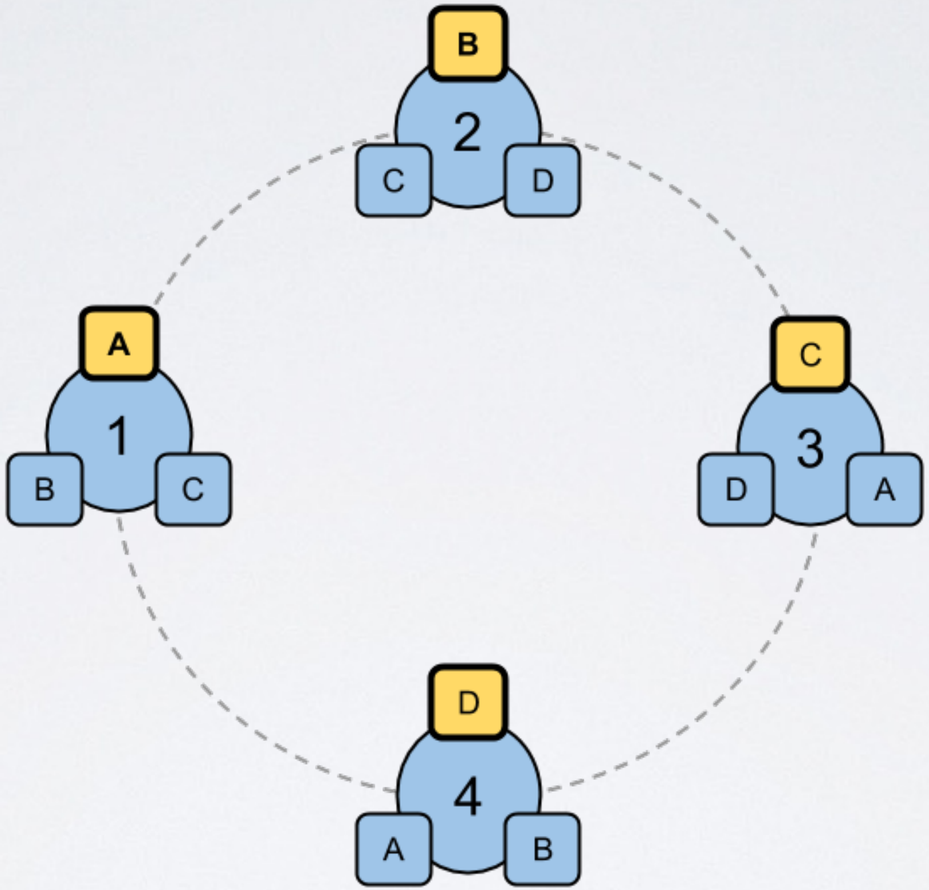
Improving repair



Improving repair



Improving repair



Improving repair

repairing each range **once** is enough

nodetool repair -pr

nodetool repair -pr

not suitable for node recovery

Repair too slow ?

Sequential repair is the default since C* 2.0

Repair too slow ?

nodetool repair -par

The problem with dense nodes

Overstreaming

Leaves of the Merkle tree contain several partitions.

The solutions with dense nodes

cassandra_range_repair
(Matt Stump & Brian Gallew)

Breaks the repair sessions in **n** steps

The solutions with dense nodes

vnodes : one repair session per vnode

Drawback : if you have many vnodes, repair takes longer

Repair in…

www.thelastpickle.com

The early days of your cluster

Node density is low, repair works just fine however you run it.

So maybe like I did,
you run « nodetool repair »
on all nodes… at the same
time

The (not so) early days of your cluster

As nodes gets higher in density, repair takes longer… and longer…

The (not so) early days of your cluster

… and latencies rise
as repair is a CPU and
I/O intensive operation

Your cluster is a grown up now

… until it breaks your cluster

How can it break ?

Load gets too high

How can it break ?

Load gets too high

You don’t meet your latency SLA anymore

How can it break ?

Load gets too high

How can it break ?

Load gets too high

Streams get stuck

How can it break ?

Load gets too high
Streams get stuck
and out of nowhere, all nodes start to eat
all your CPU doing nothing

The fun part ?

You need to run repair to recover from the repair outage !

The cluster keeps growing

And you realize orchestration is needed to stop blowing up your cluster

Orchestrating repair

Repair must not run on all nodes at the same time

Tools to orchestrate repairs

OpsCenter repair service (DSE users)
Spotify reaper

Spotify reaper

https://github.com/spotify/cassandra-reaper

Performs subrange repair

Spotify reaper

Performs subrange repair

Limits repair pressure

Spotify reaper

Performs subrange repair

Limits repair pressure

Retries failed sessions

Spotify reaper

Performs subrange repair
Limits repair pressure
Retries failed sessions
Schedules cyclic repairs

Spotify reaper

Performs subrange repair

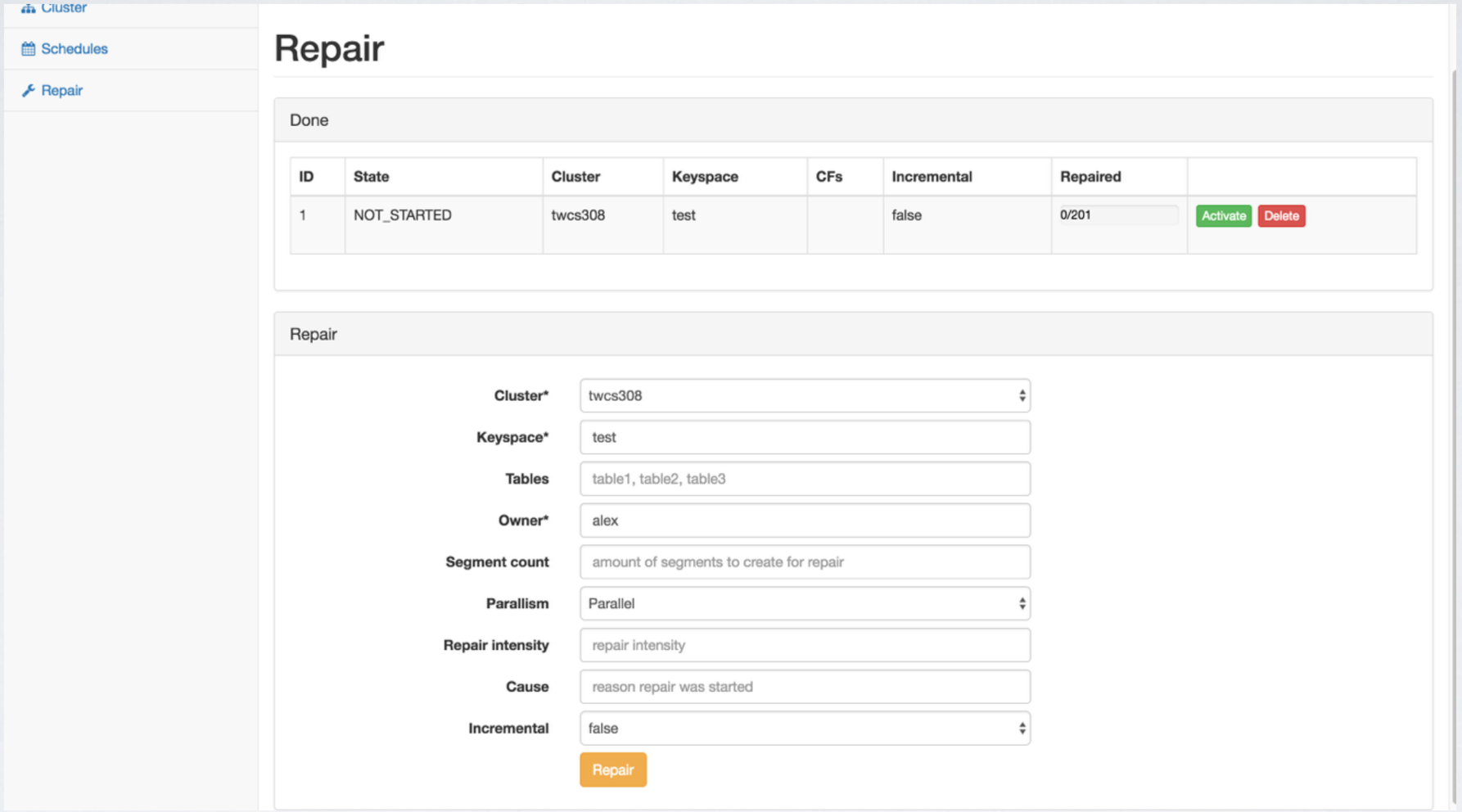
Limits repair pressure

Retries failed sessions

Schedules cyclic repairs

Optimizes cluster load

Spotify reaper - with UI (thx Stefan Podkowinski)



www.thelastpickle.com

What and why ?
Full repair
Incremental repair
How to make it work
Automated repairs

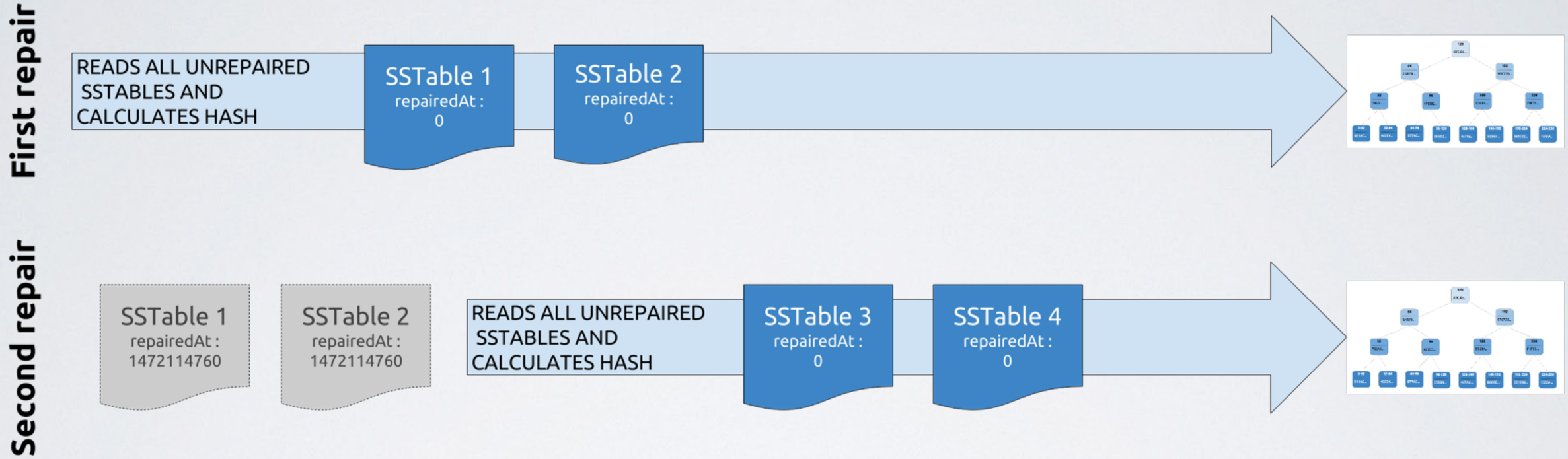
What if we stopped repairing repaired data ?

Here comes the savior !

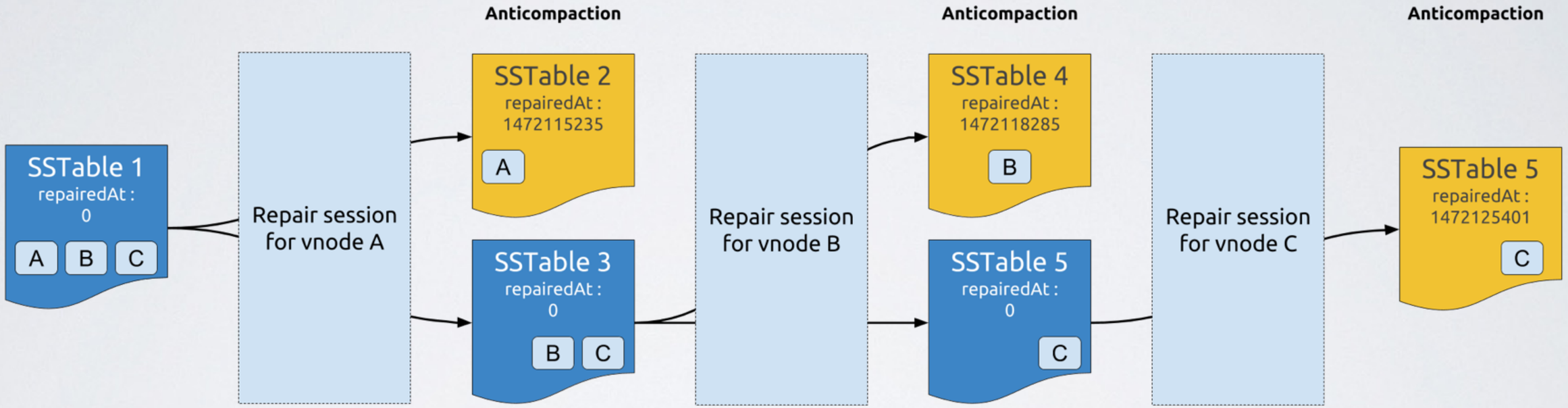
C* 2.1 introduces incremental repair

Default repair mode since C* 2.2

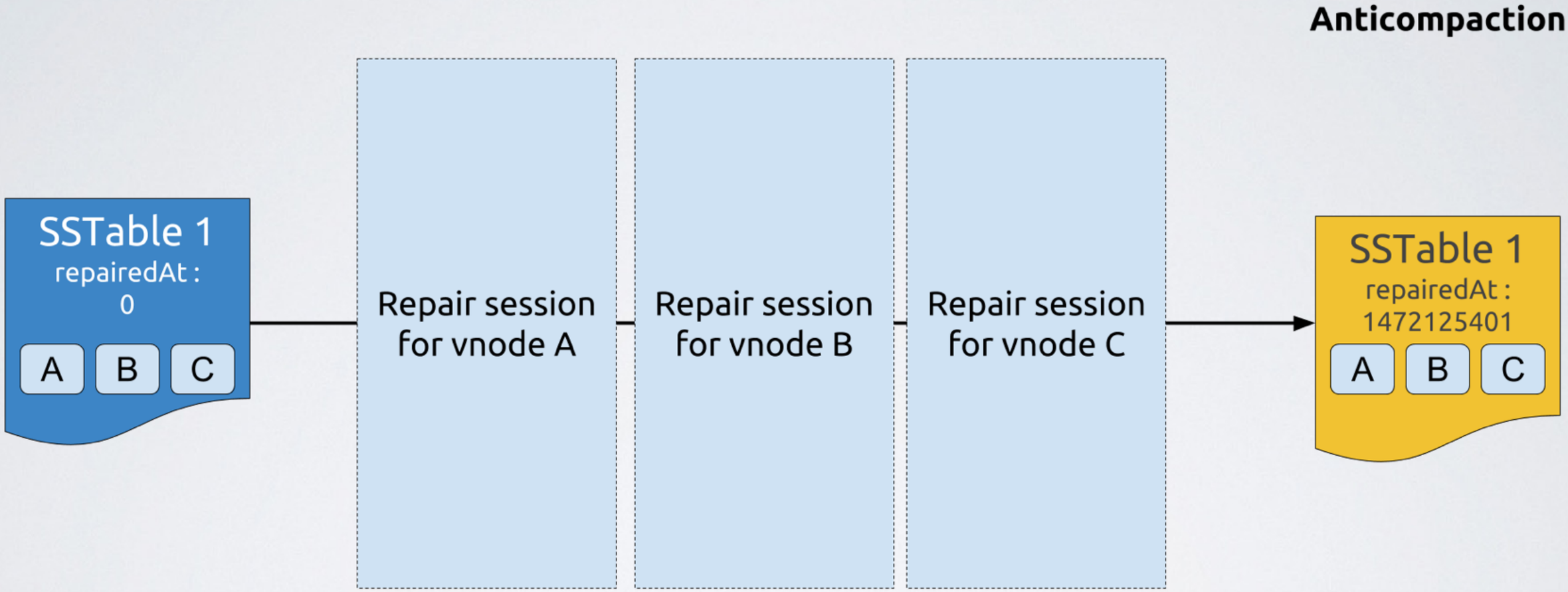
How does incremental repair work ?



Anticompaction



Anticompaction (repair on all ranges on local node)



Incremental repair looks awesome…

…but has flaws and drawbacks

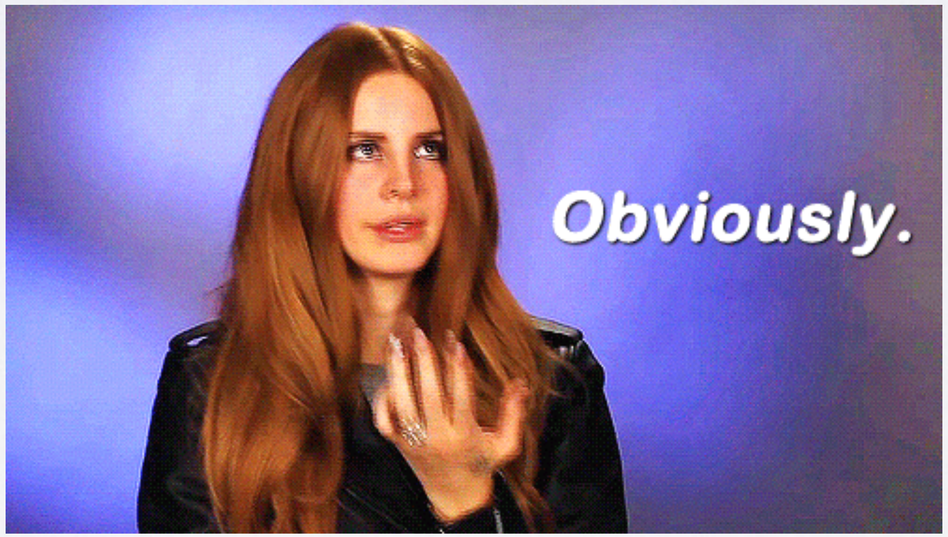
Carefully prepare your switch to incremental repair

Incremental repair caveats

Carefully prepare your switch to incremental repair

i.e. do not run « nodetool repair -inc » straight away…

It doesn't handle missing/corrupted data that was already repaired



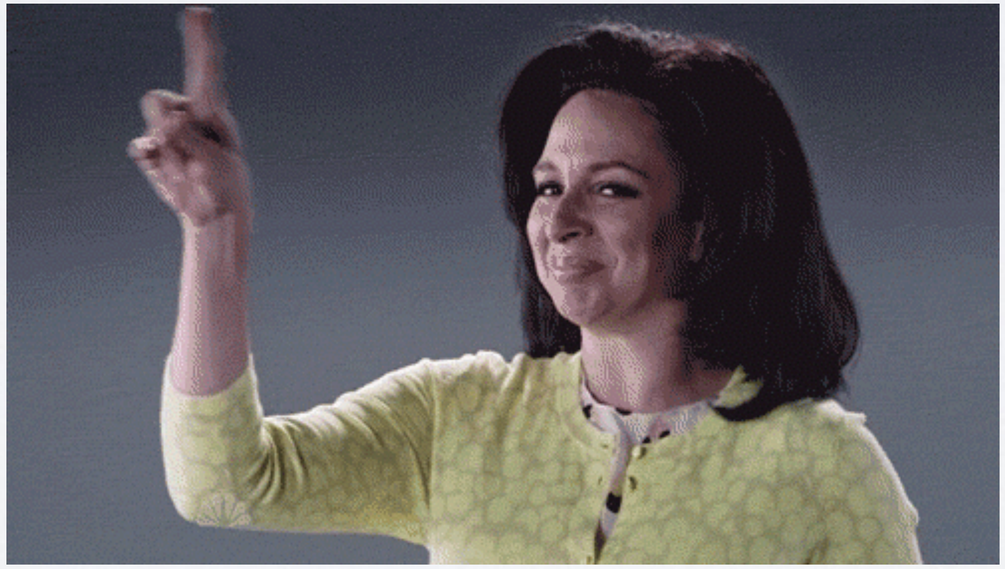
Incremental repair caveats

It splits SSTables in 2 sets that cannot be compacted together

(think tombstone purge)

Incremental repair caveats

It is incompatible with subrange repair (anticompaction)



Incremental repair caveats

It doesn't like concurrency very much

even in C* 3.x

Incremental repair caveats

Validator.java:261 -
Failed creating a merkle tree for [repair #e4c782d0-11fc-11e6-b616-51a3849870bb on table_v2/table_attributes, [(8835460833482333317,8838777311566358575], (-7300486781514672850,-7298192396576668423], (-9592984746751672 25,-9591779641060742 09]]], /10.10.10.33 (see log for details)

Incremental repair caveats

CompactionManager.java:1320 - **Cannot start multiple repair sessions over the same sstables**

Incremental repair caveats

CASSANDRA-8316

A running anticompation prevents validation compaction

Incremental repair caveats

Do not use -pr with incremental repair

Incremental repair caveats

Do not use -pr with incremental repair

Useless : data is repaired once only

Incremental repair caveats

Do not use -pr with incremental repair

Useless : data is repaired once only
Expensive : anticompaction overhead

Incremental repair will not…

Fix a poor repair strategy

Incremental repair will not…

Prevent you from having to run full repair

Reaper does not support incremental repair

But this fork does :

https://github.com/adejanovski/cassandra-reaper/tree/inc-repair-that-works

Reaper does not support incremental repair

And this one embeds the modded UI :

https://github.com/adejanovski/cassandra-reaper/tree/inc-repair-support-with-ui

Reaper does not support incremental repair

Not enough time to write those urls ?

github.com/adejanovski

No subrange repair

Reaper inc repair fork

No subrange repair

Single repair thread => no concurrency

What and why ?

Full repair

Incremental repair

How to make it work

Automated repairs

Put your repair strategy in place on day 1

Use appropriate tooling or build your own

Repair best practices

Spread repair over
a gc_grace_seconds cycle

Adjust repair pressure on your cluster (Reaper does that)

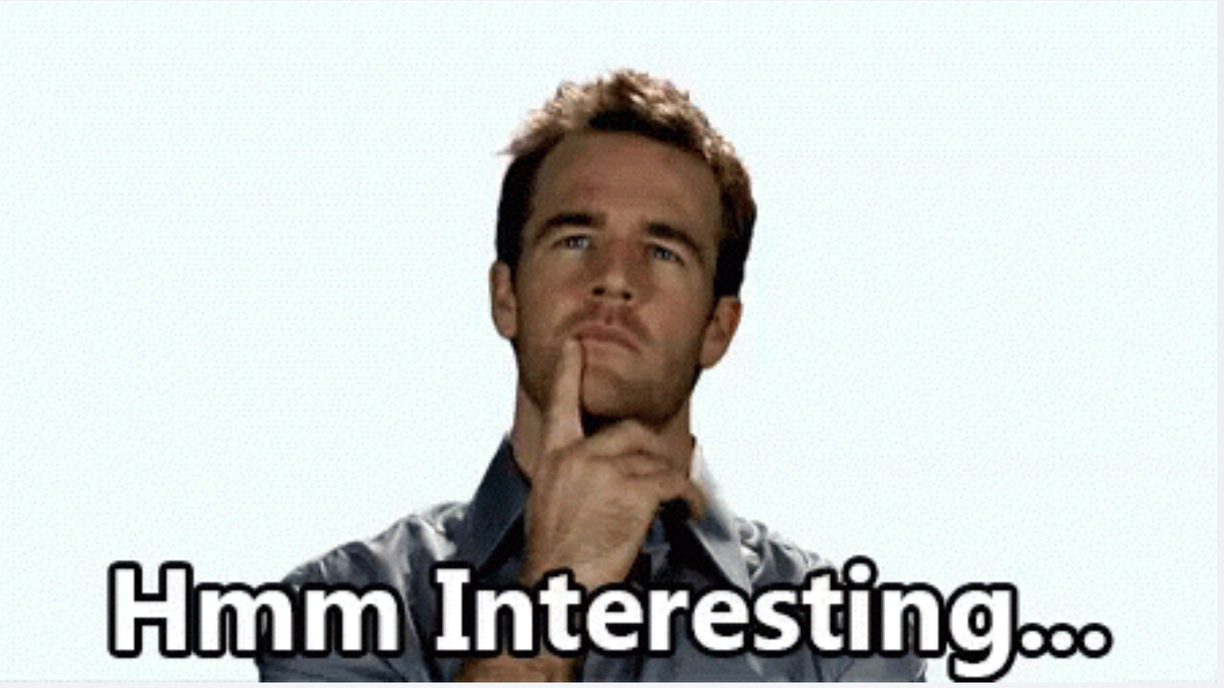
Repair best practices

Don’t repair everything !

Pick tables with deletes and those with critical data

Repair best practices

If every data is critical, then none is ;)



Repair best practices

Be tight on your schedule with inc repair

Tombstones and anticompaction

www.thelastpickle.com

Avoid concurrency with inc repair

One node at a time

What and why ?
Full repair
Incremental repair
How to make it work
Automated repairs

The bright future ?

Not having to think about cyclic maintenance repairs

CASSANDRA-10070

Automatic repair scheduling

www.thelastpickle.com

CASSANDRA-8911

Mutation-based Repairs

Thanks!

@alexanderdeja