

Improving Bad Partition Handling In Apache Cassandra

Cheng Wang
Jordan West

N

Who We Are?

— Jordan West

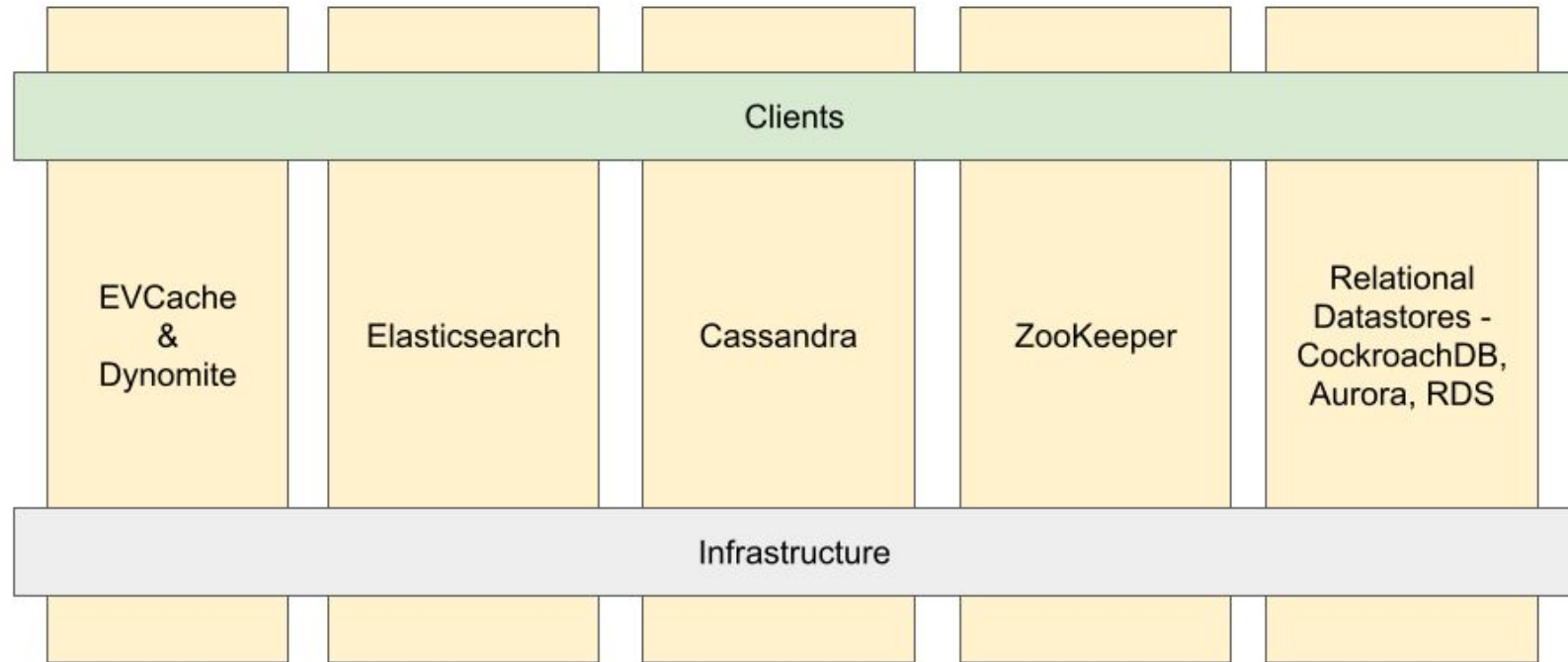
- SWE @ Netflix approx. 2 years
- Cassandra committer
- Been working with Cassandra for approx. 7 years total. Databases 10+.

— Cheng Wang

- SWE @ Netflix approx. 6 months
- Been working with database engine for approx. 6 years total

Online Datastore Team @Netflix

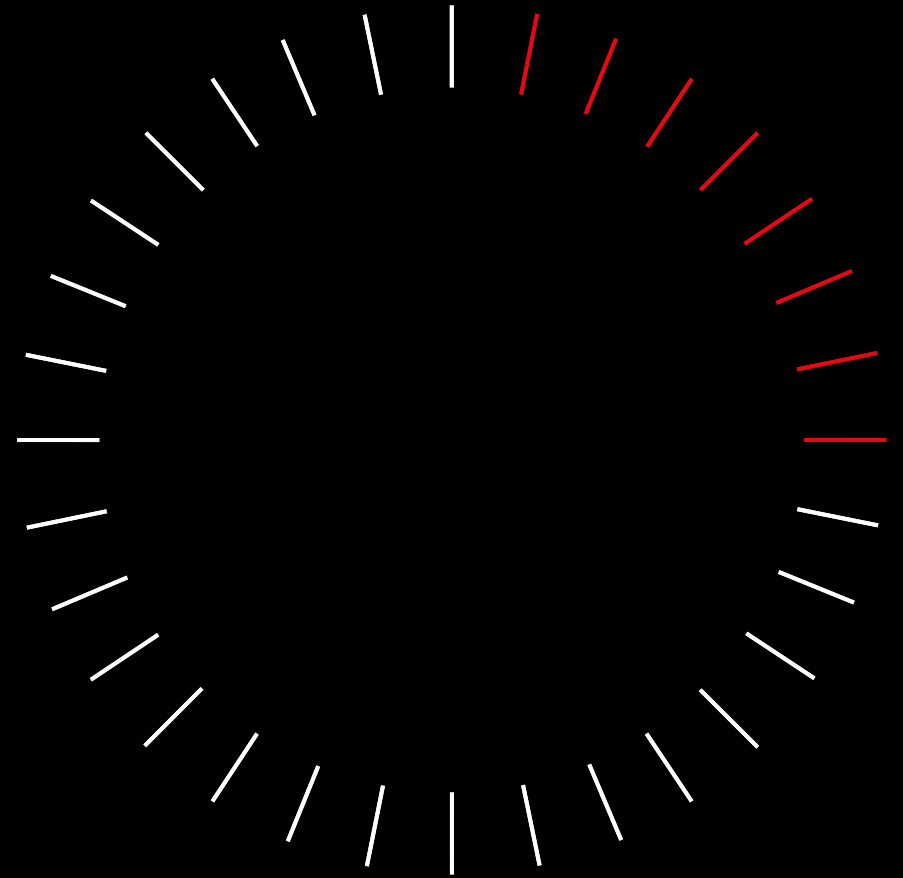
- Provide high leverage datastores as a managed service at scale, to support all operational data needs spanning across Streaming, Studio and Gaming businesses for Netflix.



Cassandra @ Netflix

- Over 22,000 Nodes
- 900+ Clusters
- 12+ PB of Data
- 12M+ req/s (approx. 60-40 read-write)
- Cassandra 3.0.x (for now)
- Moving towards 4.x targeting rollout 2023

Why Are Bad Partitions Bad?





Axis 0

■ percentile_95
 Max : 2.406k Min : 1.867k
 Avg : 1.906k Last : 1.888k
 Tot : 236.375k Cnt : 124.000

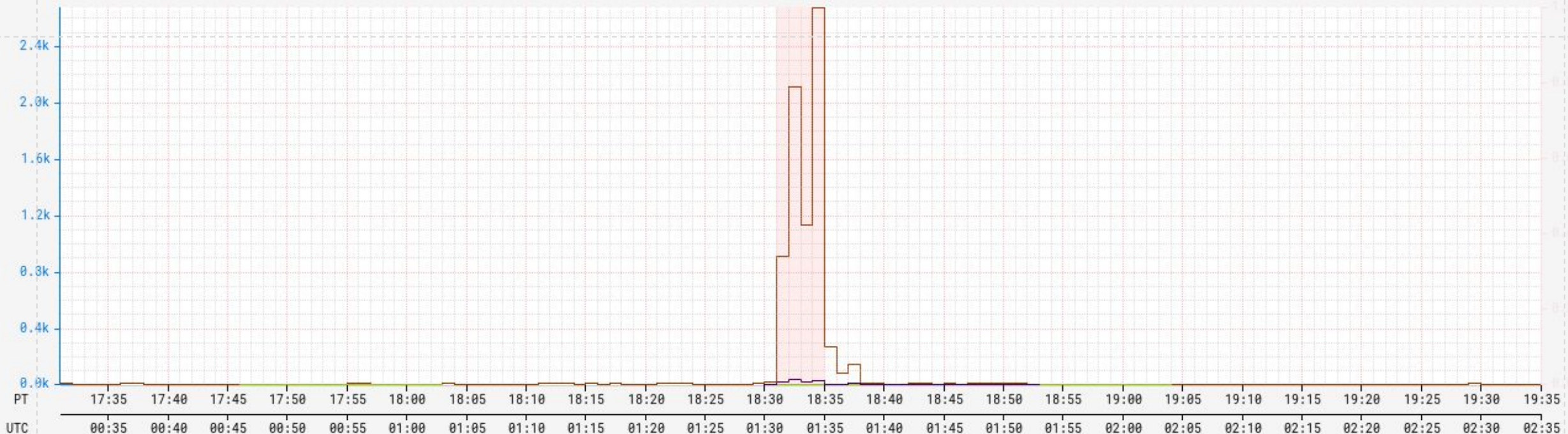
■ percentile_99
 Max : 1.123M Min : 48.421k
 Avg : 92.474k Last : 56.531k
 Tot : 11.467M Cnt : 124.000

Axis 1

□ Impact Window
 Max : 1.000 Min : 0.000
 Avg : 32.258m Last : 0.000
 Tot : 4.000 Cnt : 124.000

Bad Partitions Have Material Business Impact

playapi -> subscriberservice IPC errors by status

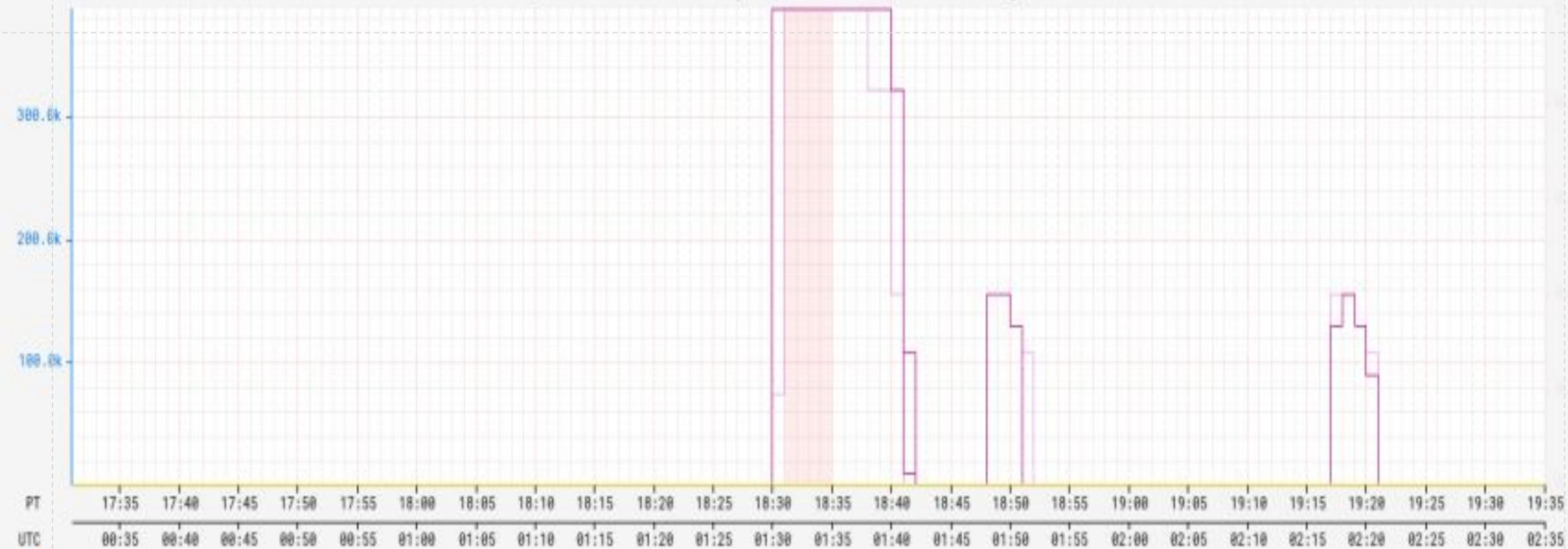


Axis 0

Status	Max	Min	Avg	Last	Tot	Cnt
cancelled	133.333m	0.000	5.031m	0.000	266.667m	53.000
throttled	2.670k	2.033	65.685	2.567	8.145k	124.000
timeout	66.667m	0.000	3.333m	0.000	166.667m	50.000
unexpected_error	42.550	0.000	6.222	0.000	143.100	23.000
Impact Window	1.000	0.000	32.258m	0.000	4.000	124.000



cass_subscriberservice Replica Reads - Maximum Latency99th Per CF



Axis 0

[Blue bar]

Max : 943.127 Min : 943.127
 Avg : 943.127 Last : 943.127
 Tot : 116.948k Cnt : 124.000

[Brown bar]

Max : 943.127 Min : 943.127
 Avg : 943.127 Last : 943.127
 Tot : 116.948k Cnt : 124.000

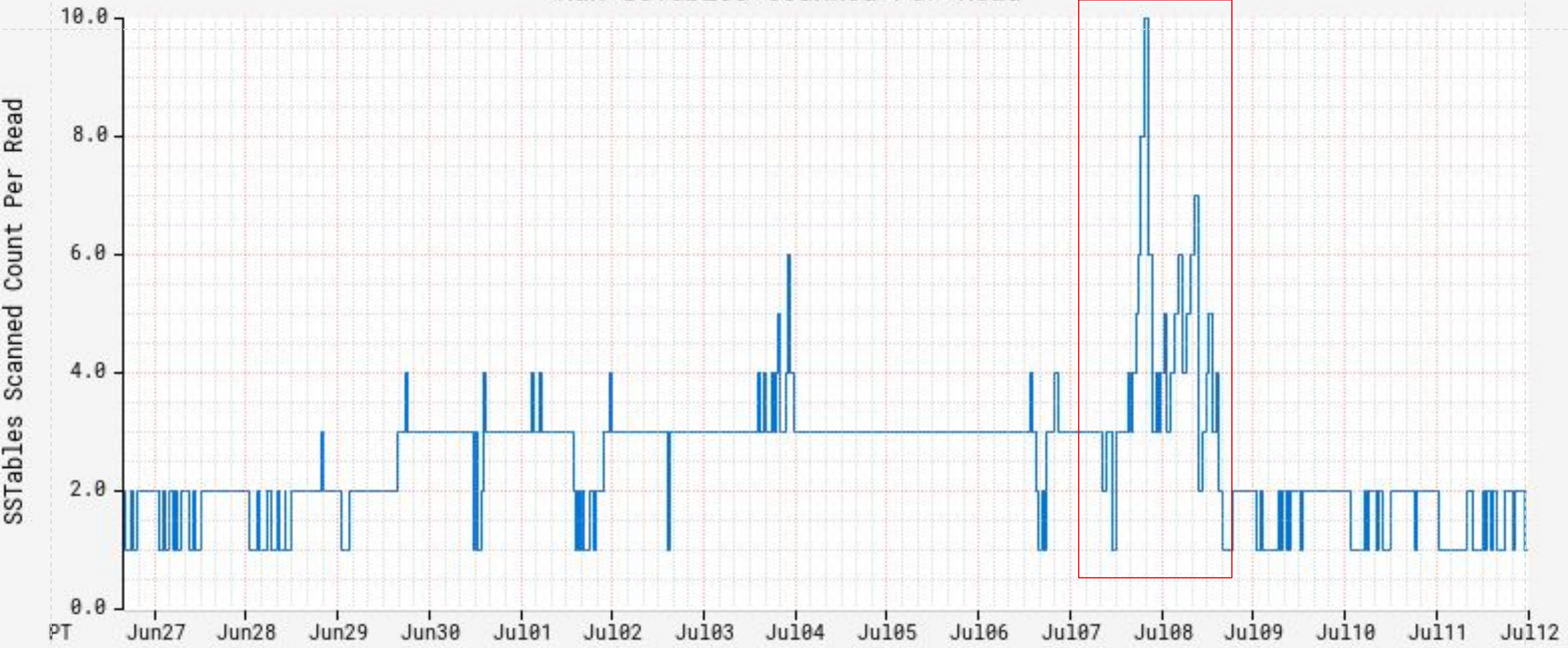
[Green bar]

Max : 1.132k Min : 943.127
 Avg : 946.169 Last : 943.127
 Tot : 117.325k Cnt : 124.000

[Purple bar]

N

Max SSTables Scanned Per Read



S

Max :	10.000	Min :	1.000
Avg :	2.529	Last :	1.000
Tot :	1.859k	Cnt :	735.000

Frame: 368h, End: 2021-07-12T00:30-07:00[US/Pacific], Step: 30m

Fetch: 14ms (L: 35.5k, 102.0, 1.0; D: 2.1M, 75.1k, 736.0k)



Types of Bad Partitions

- **Total Size:** Partitions in the GB+ size range
- **Row Count:** Smaller partitions with Million+ rows
- **Tombstone Count:** Partitions with Million+ tombstones
- **SSTable Count:** A partition spread across many sstables is more expensive to read

What Makes Bad Partitions Bad

- CPU Usage
- Memory Usage
 - Buffer allocations
 - Object overhead
 - GC
- Reading More Files
- Compaction Impacts
- Cascading Read Latency

How We Improve?

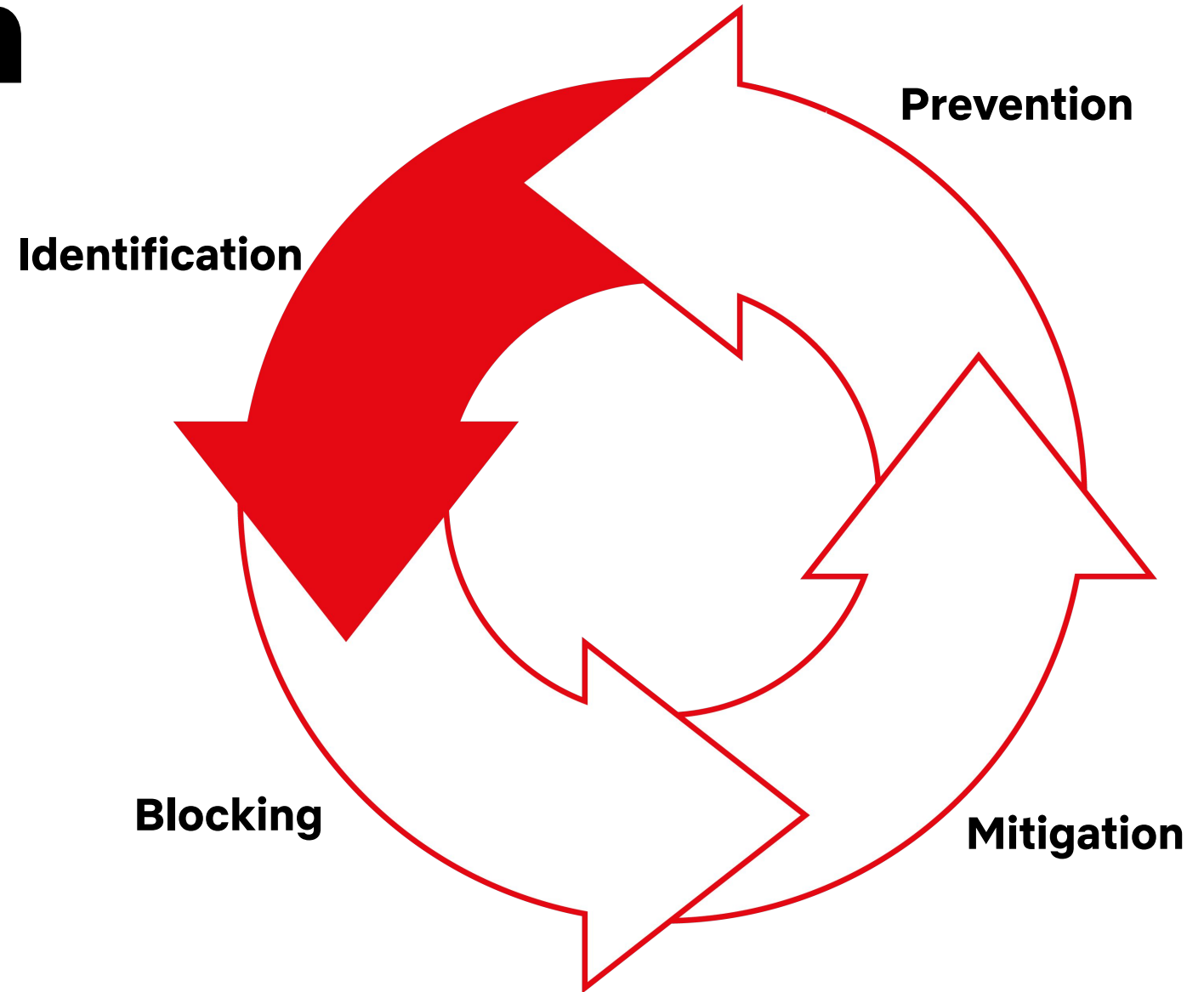
Identification

Prevention

Blocking

Mitigation

Identification



2923s

Dashboard / wide_row_24h_dashboard_default

Share Clone Edit Last 24 hours

Uses lucene query syntax

app: "cass_cde" Add a filter

list_of_apps_with_widerows

C* App	# Wide Keys
cass_cde	3,074

wide_rows_all_apps

C* App	Keyspace	CF/Table	Primary Key	Report Time	Row Size
cass_cde	cassrepair	repair_status	abcassandra:23	September 29th 2022, 01:22:17.342	88.385MB
cass_cde	cassrepair	repair_status	abcassandra:33	September 29th 2022, 01:21:35.892	83.676MB
cass_cde	cassrepair	repair_status	abcassandra:31	September 29th 2022, 01:21:00.447	77.935MB
cass_cde	cassrepair	repair_status	abcassandra:32	September 29th 2022, 01:24:51.747	77.128MB
cass_cde	cassrepair	repair_status	abcassandra:34	September 29th 2022, 01:29:10.678	72.258MB
cass_cde	cassrepair	repair_status	abcassandra:22	September 29th 2022, 01:26:27.380	71.036MB
cass_cde	cassrepair	repair_status	abcassandra:24	September 29th 2022, 01:22:14.190	69.391MB
cass_cde	cassrepair	repair_status	abcassandra:30	September 29th 2022, 01:25:34.719	68.233MB
cass_cde	eunomia	eunomianodesinfo	cass_dgw_ts_tracing	September 29th 2022, 21:00:41.383	64.884MB
cass_cde	cassrepair	repair_status	abcassandra:25	September 29th 2022, 01:39:18.198	62.271MB
cass_cde	cassrepair	repair_status	abcassandra:27	September 29th 2022, 01:22:42.659	61.766MB
cass_cde	cassrepair	repair_status	abcassandra:26	September 29th 2022, 01:22:02.825	60.027MB
cass_cde	cassrepair	repair_status	abcassandra:29	September 29th 2022, 01:25:38.183	59.359MB
cass_cde	cassrepair	repair_sequence	cass_ccs	September 29th 2022, 01:18:14.028	56.088MB
cass_cde	cassrepair	repair_status	abcassandra:28	September 29th 2022, 01:38:18.814	55.977MB
cass_cde	cassrepair	repair_hook_status	cass_ccs	September 29th 2022, 01:17:40.429	55.631MB
cass_cde	cassrepair	repair_status	abcassandra:23	September 29th 2022, 01:28:45.129	53.336MB
cass_cde	cassrepair	repair_sequence	cass_pdsaccounting	September 29th 2022, 01:18:17.712	49.493MB
cass_cde	cassrepair	repair_hook_status	cass_pdsaccounting	September 29th 2022, 01:17:33.511	48.895MB
cass_cde	cassrepair	repair_status	cass_laseoffline48:20	September 29th 2022, 01:39:07.340	39.989MB
cass_cde	cassrepair	repair_status	abcassandra:20	September 29th 2022, 01:40:23.798	39.339MB

Writing large partition cassrepair/repair_status
abcassandra:23 (11584798 bytes to sstable NNN-Data.db)

ApacheCon 2022

Collapse

```
$ sstabledump me-1176537-big-Data.db -k josephl_test
```

```
[  
  {  
    "partition" : {  
      "key" : [ "josephl_test" ],  
      "position" : 0  
    },  
    "rows" : [  
      {  
        "type" : "row",  
        "position" : 26,  
        "clustering" : [ "0x68656c6c6f0a" ],  
        "liveness_info" : { "tstamp" : "2021-10-23T00:17:11.252097Z" },  
        "cells" : [  
          { "name" : "value", "value" : "0xb068656c6c6f20776f726c64" },  
          { "name" : "value_metadata", "value" : "0x" }  
        ]  
      }  
    ]  
  }  
]
```

```
$ nodetool getendpoints foo bar josephl_test
```

```
$ nodetool getsstables foo bar josephl_test
```

```
$ sstabledump .... | grep row | wc -l
```

```
$ sstablemetadata foo bar ....
```

nodetool getsstables -l

- Small extension to nodetool getsstables
- Only works for LeveledCompactionStrategy
- Helps identify how partitions are spread across sstables

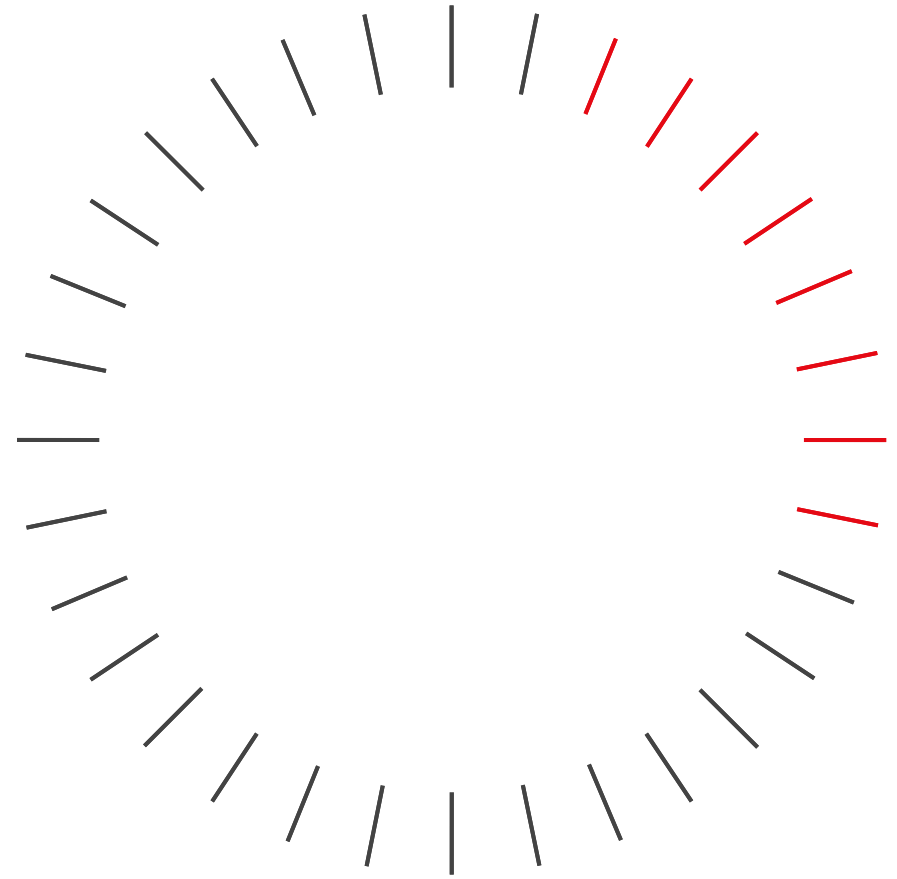
nodetool
toppartitions
-m MAX

- Extension to top partitions to find TopK in addition to counts
- Latency
- SSTables
- Rows
- Tombstones

fix intermittent failure in top partitions

- CASSANDRA-17254
- Incorrect use of ByteBuffers led to intermittent formatting errors when outputting top partitions results
- Would often cause a delay in our ability to identify problematic partitions

Live Demo



```
$ nt toppartitions -m MAX -a TOMBSTONES marken_01 startedannotationoperationid 10000
```

```
TOMBSTONES Max Sampler:
```

```
Top 10 partitions:
```

Partition	Max
STARTED	5911

```
$ cqlsh
```

```
[cqlsh 5.0.1 | Cassandra 3.0.26.1 | CQL spec 3.4.0 | Native protocol v3]
```

```
Use HELP for help.
```

```
cqlsh> ALTER TABLE marken_01.startedannotationoperationid WITH gc_grace_seconds=3600;
```

```
$ nt compact marken_01 startedannotationoperationid
```

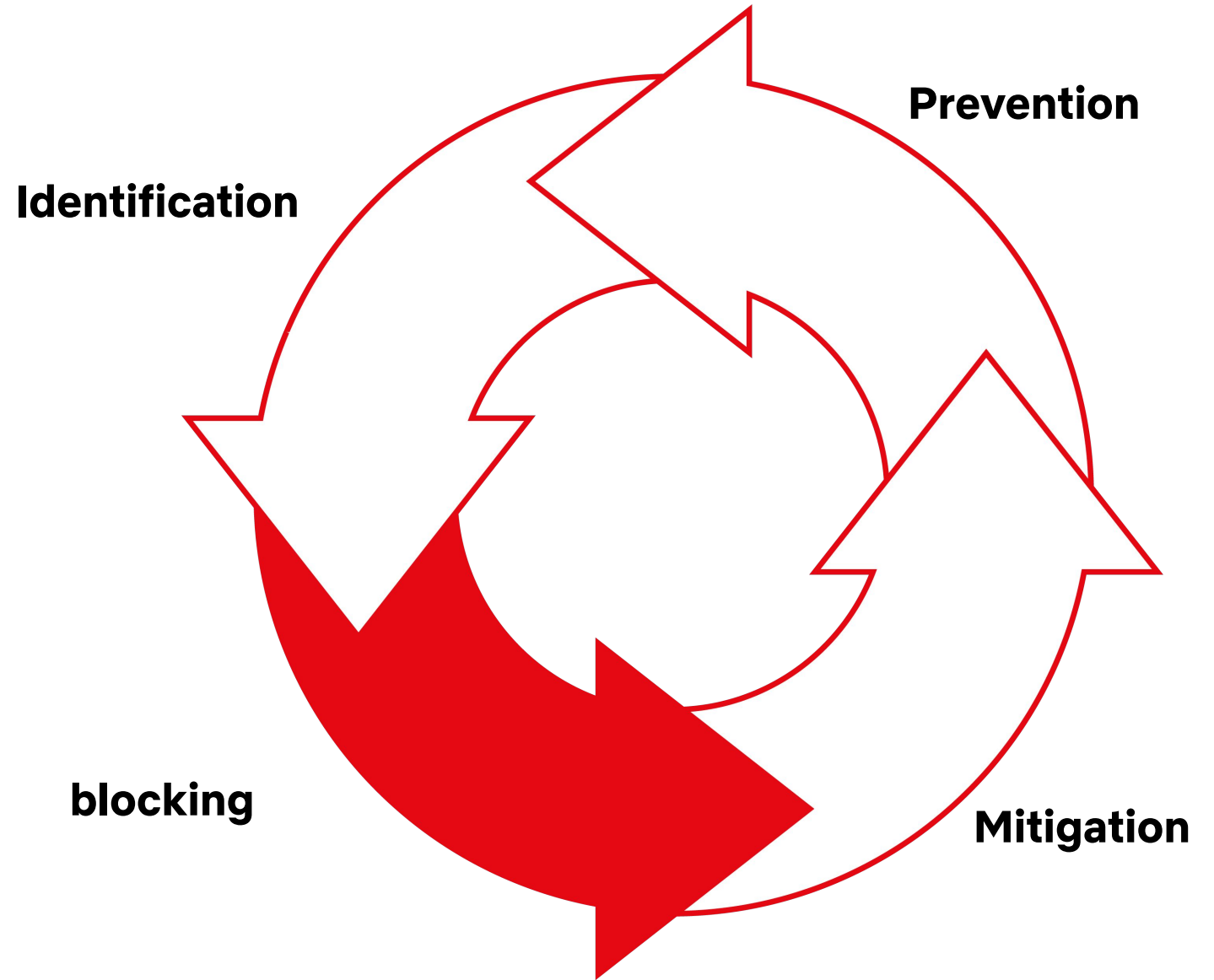
```
$ nt toppartitions -m MAX -a TOMBSTONES marken_01 startedannotationoperationid 10000
```

```
TOMBSTONES Max Sampler:
```

```
Top 10 partitions:
```

Partition	Max
STARTED	2970

Blocking



Partition blacklist

- Backport **CASSANDRA-12106**: add ability to blacklist / denylist a CQL partition so all requests are ignored
- Prevent reads, range reads and writes (configurable) to given partition keys
- Write to system table via client or JMX to add or remove denied partitions
- Provides a tool to operators to control undesirable application behavior

Mitigation

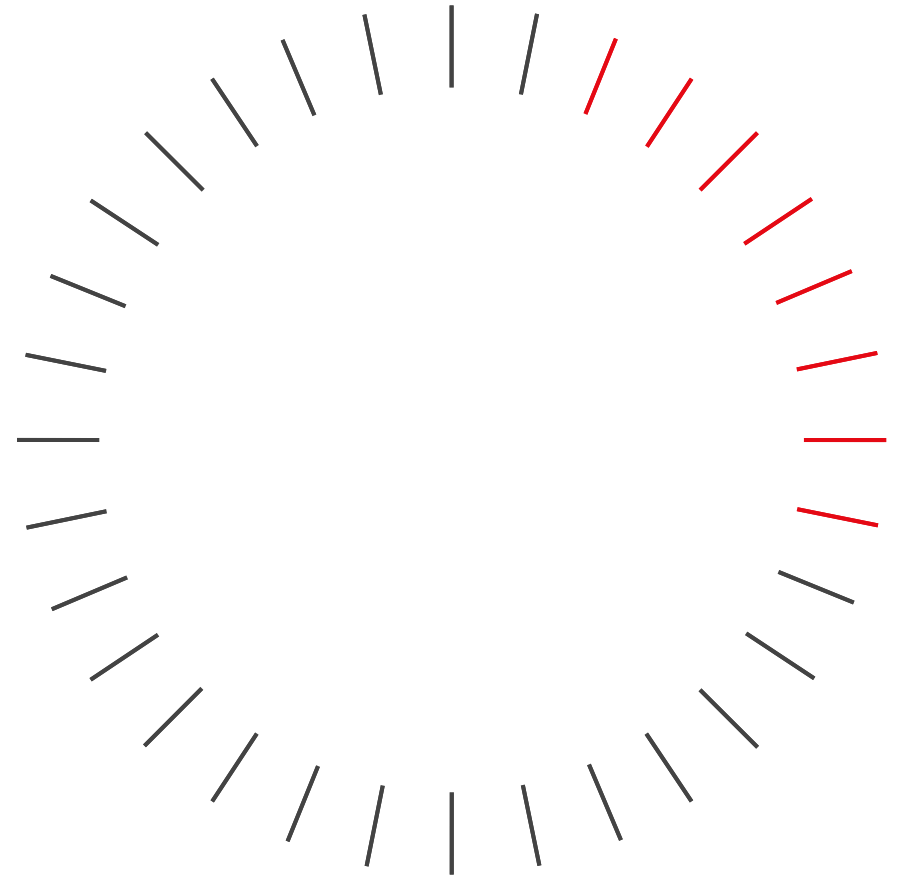


nodetool force compact

```
$ nodetool  
forcecompact  
keyspace table  
list_of_partition_keys
```

- New tool developed internally - Cassandra
- Previously, we reduced the **gc_grace_seconds** and ran on the **nodetool compact**
 - Took hours even days to scan all the sstables
 - Ran at the risk of deleted(tomestone) data reappearing
- Force compact
 - Only scans the sstables for the keys given
 - Help to quickly mitigate the bad partitions and avoid scanning the whole table
 - **Ignore gc_grace_seconds:** For keys where we know it is safe to remove tombstoned or TTL'd data
- Target for OSS 4.x

Live Demo



Thank You.