Change Point Detection in Software Performance Testing

David Daly, William Brown, Henrik Ingo, Jim O'Leary, David Bradford

David Daly | Lead Engineer -- Performance | @daviddaly44 | https://daviddaly.me/

Performance Testing Goals (In CI)

Know if and when the performance changes

- If it gets slower, quickly fix it
- If it gets faster, lock in the improvement

Part of our release process

- The quicker the notification, the easier it is to:
  - Isolate the cause of the change
  - Fix or backout the the responsible change

Performance Testing in Continuous Integration

Setup a system under test

Run a workload

Report the results

Decide (and alert) if the performance changed

Visualize the result

Automate everything/Keep noise down

Performance Testing in Continuous Integration (V0)

Setup a system under test

Run a workload

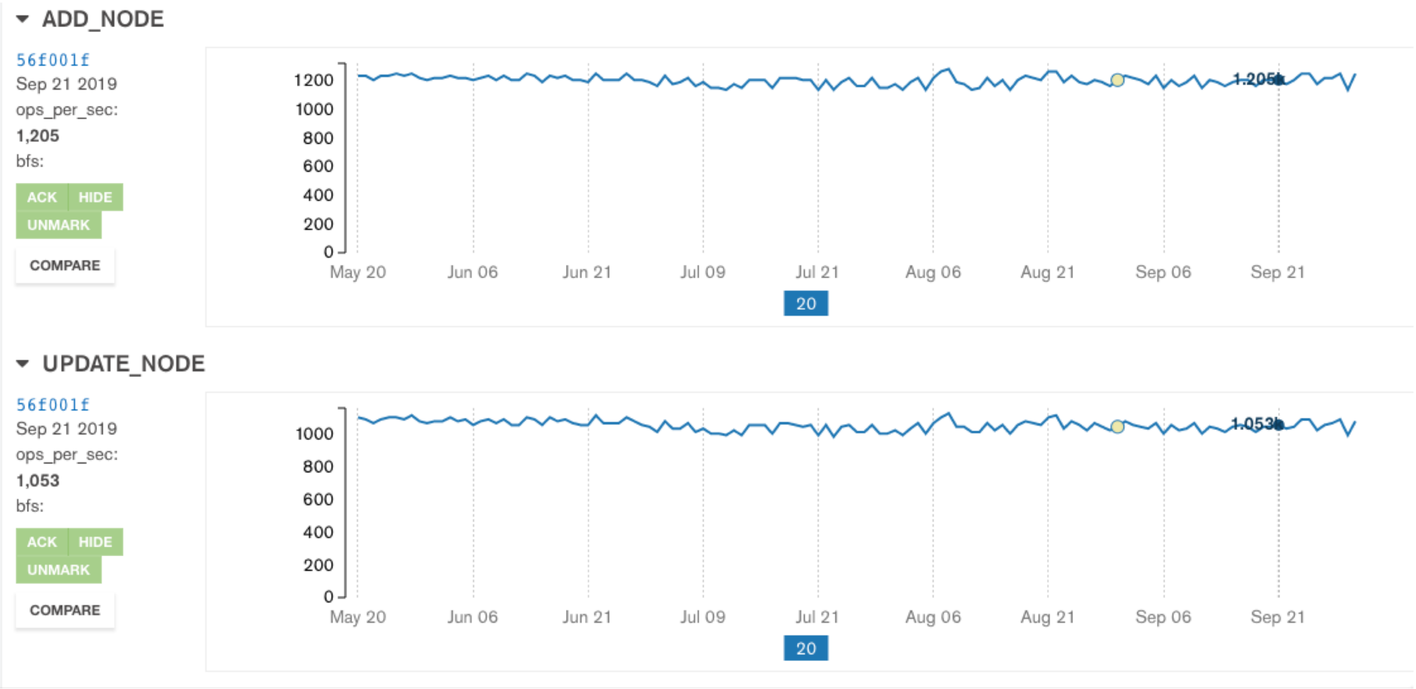
Report the results

Decide (and alert) if the performance changed

- Human looking at graphs – there are a lot of graphs

Visualize the result

Automate everything/Keep noise down

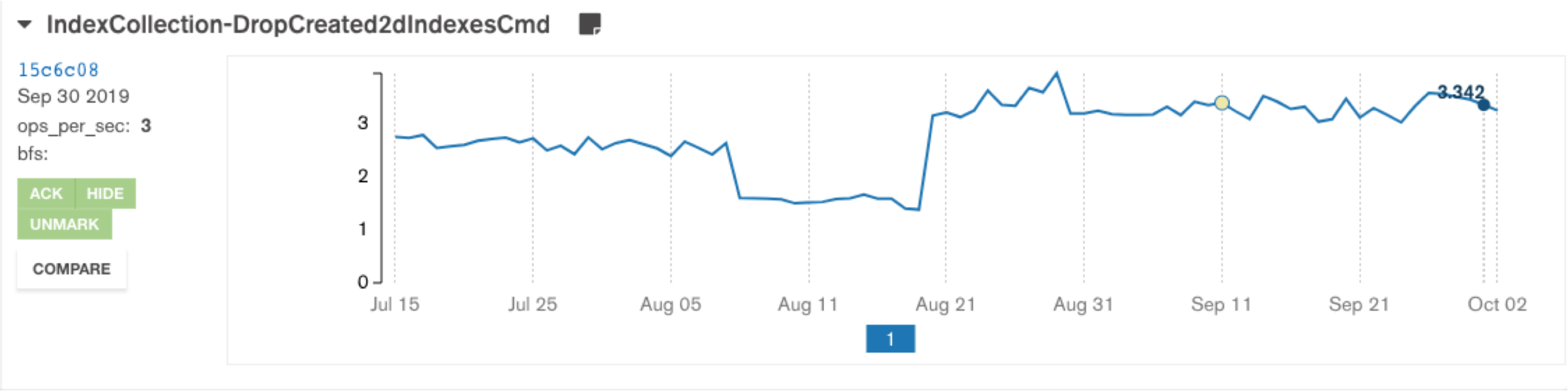


ADD_NODE

56f001f
Sep 21 2019
ops_per_sec:
1,205
bfs:

UPDATE_NODE

56f001f
Sep 21 2019
ops_per_sec:
1,053
bfs:



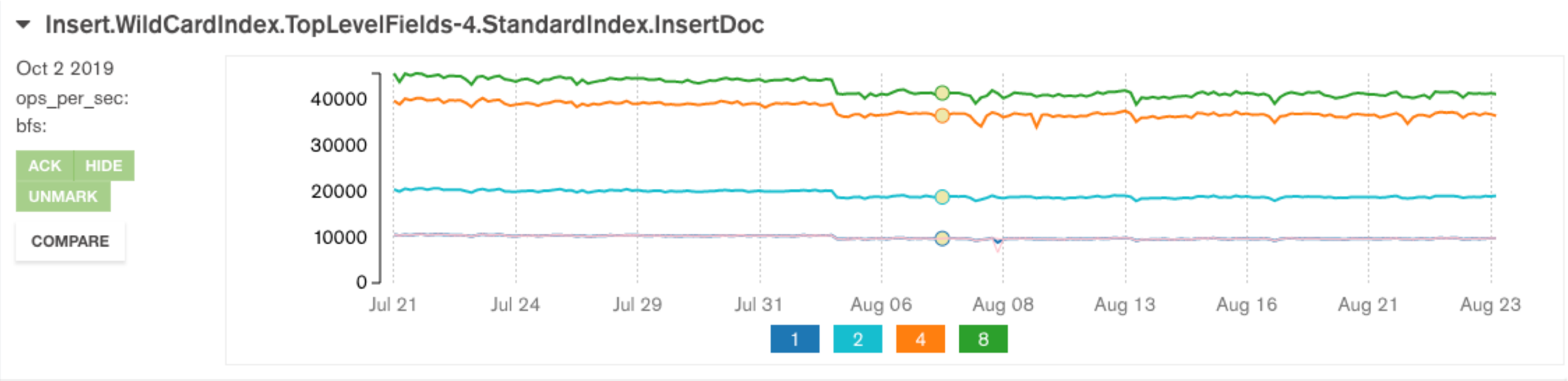
IndexCollection-DropCreated2dIndexesCmd

15c6c08

Sep 30 2019

ops_per_sec: **3**

bfs:



Insert.WildCardIndex.TopLevelFields-4.StandardIndex.InsertDoc

Oct 2 2019

ops_per_sec:

bfs:

Performance Testing in Continuous Integration (V1)

Setup a system under test

Run a workload

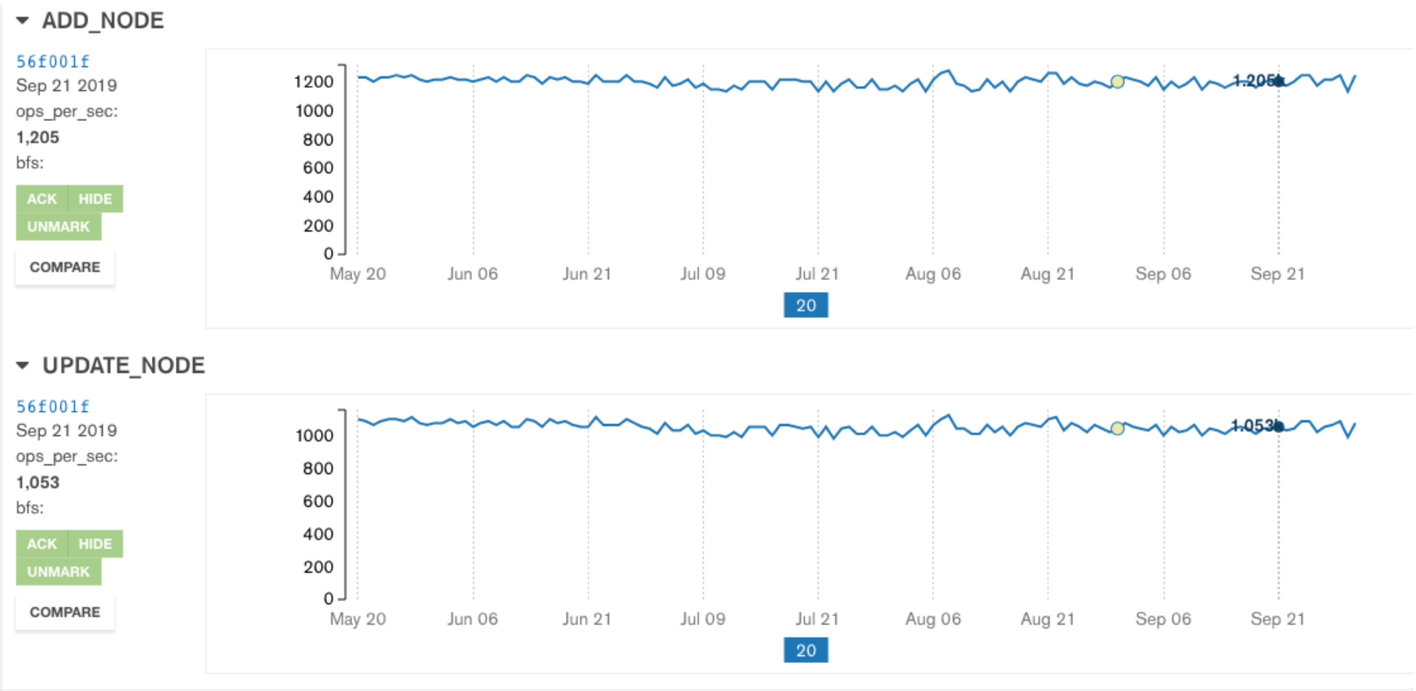
Report the results

Decide (and alert) if the performance changed

- Alert if performance drops more than 10% from baseline

Visualize the result

Automate everything/Keep noise down

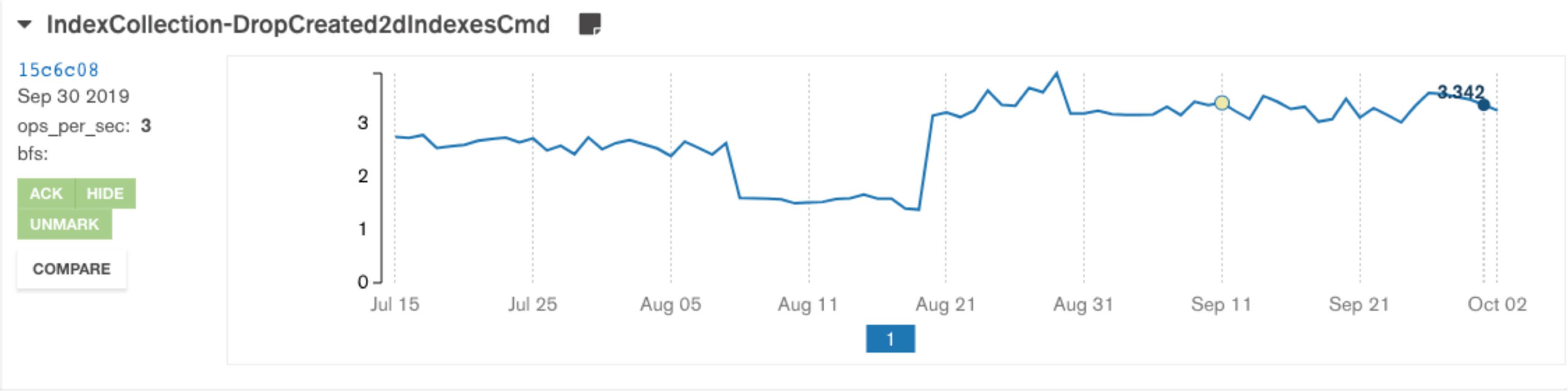


ADD_NODE

56f001f
Sep 21 2019
ops_per_sec:
1,205
bfs:

UPDATE_NODE

56f001f
Sep 21 2019
ops_per_sec:
1,053
bfs:



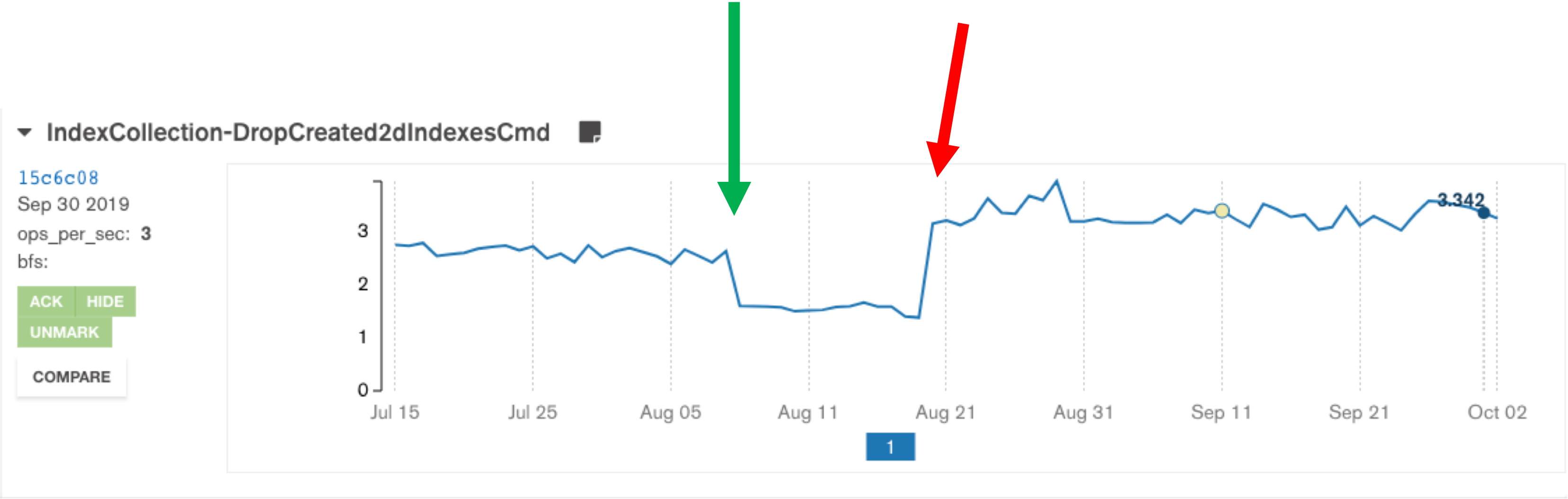
IndexCollection-DropCreated2dIndexesCmd

15c6c08

Sep 30 2019

ops_per_sec: 3

bfs:



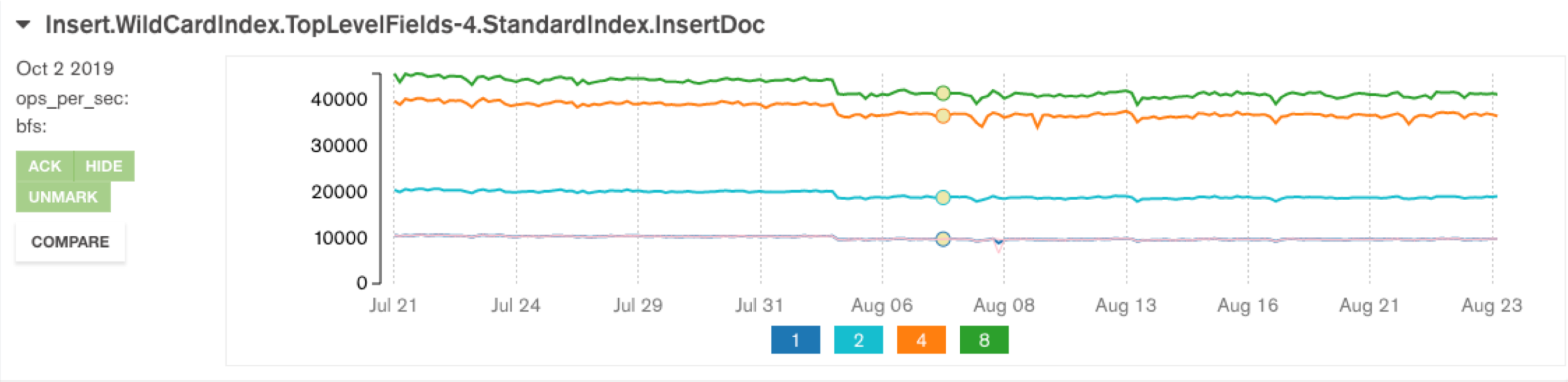
IndexCollection-DropCreated2dIndexesCmd

15c6c08

Sep 30 2019

ops_per_sec: 3

bfs:

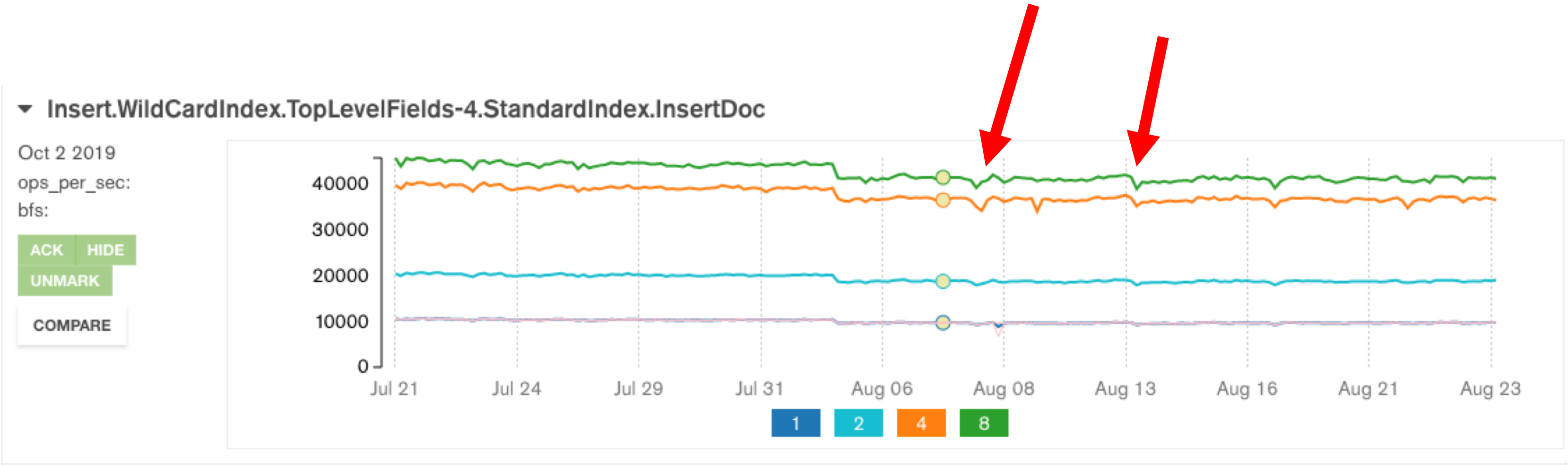


Insert.WildCardIndex.TopLevelFields-4.StandardIndex.InsertDoc

Oct 2 2019

ops_per_sec:

bfs:

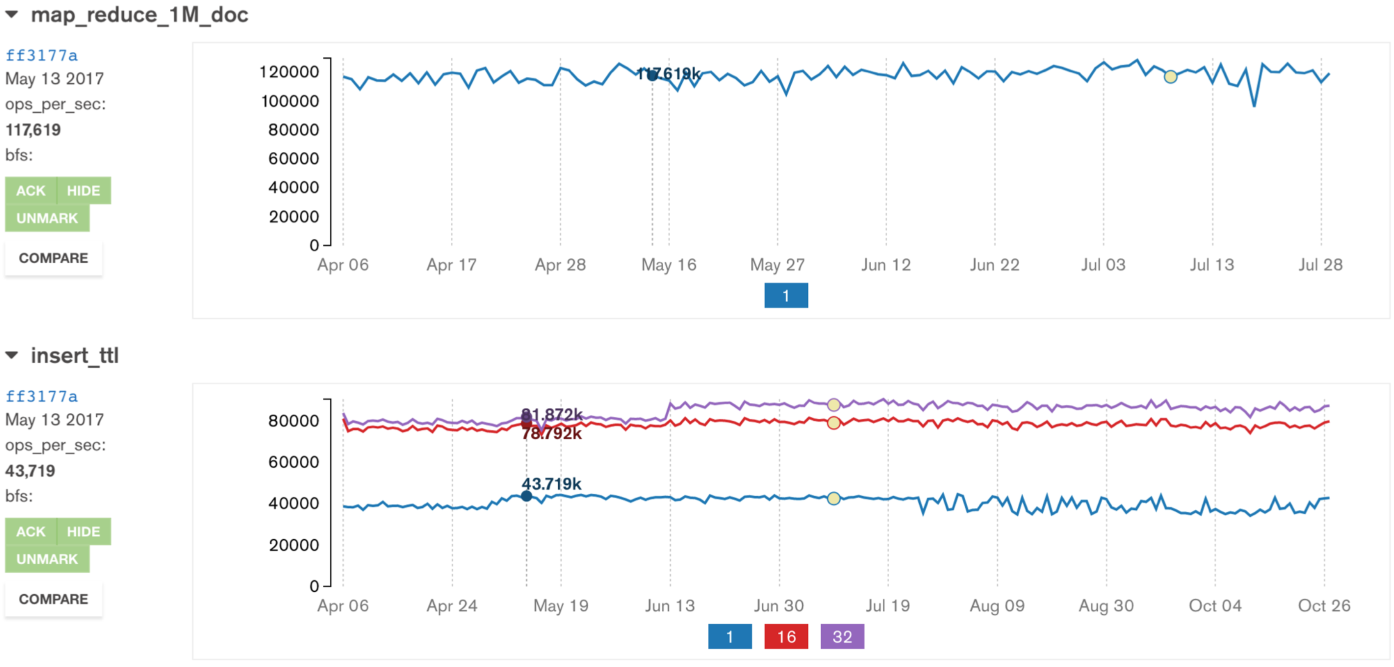


Insert.WildCardIndex.TopLevelFields-4.StandardIndex.InsertDoc

Oct 2 2019

ops_per_sec:

bfs:

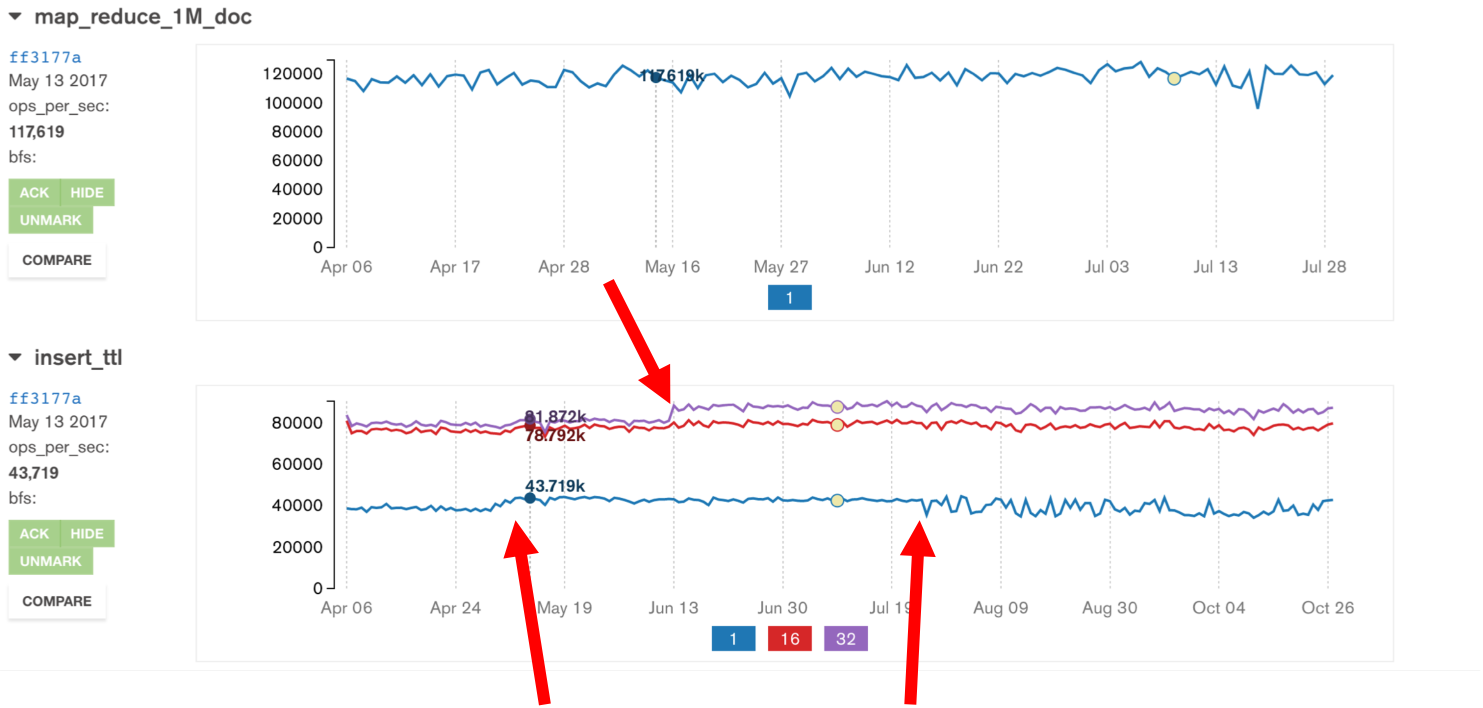


map_reduce_1M_doc

ff3177a
May 13 2017
ops_per_sec:
117,619
bfs:

insert_ttl

ff3177a
May 13 2017
ops_per_sec:
43,719
bfs:



map_reduce_1M_doc

ff3177a
May 13 2017
ops_per_sec:
117,619
bfs:

insert_ttl

ff3177a
May 13 2017
ops_per_sec:
43,719
bfs:

Thresholds Are Awful

But better than version 0!

Problems

- False positives – some tests are noisier than others
- False negatives – miss any change less than the threshold
- Identifying regressions at the wrong time
  - E.g., 8% drop doesn’t cross threshold, but a week later 8% drop + 3% noise cross the threshold

Problem

Problem Statement

Detect which commits change the performance of the software (as measured by our performance tests) in the presence of the noise from the testing infrastructure.

Change Point Detection

"Change point analysis is the process of detecting distributional changes within time-ordered observations."

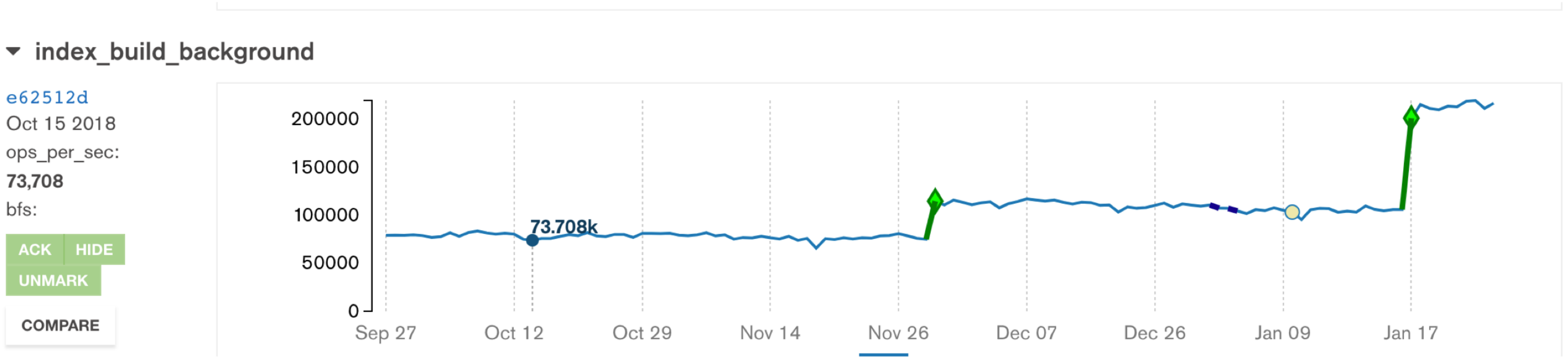
Support For Change Point Detection

Calculate the change points

Visualize change points on trend graphs

Change point dashboard for triage

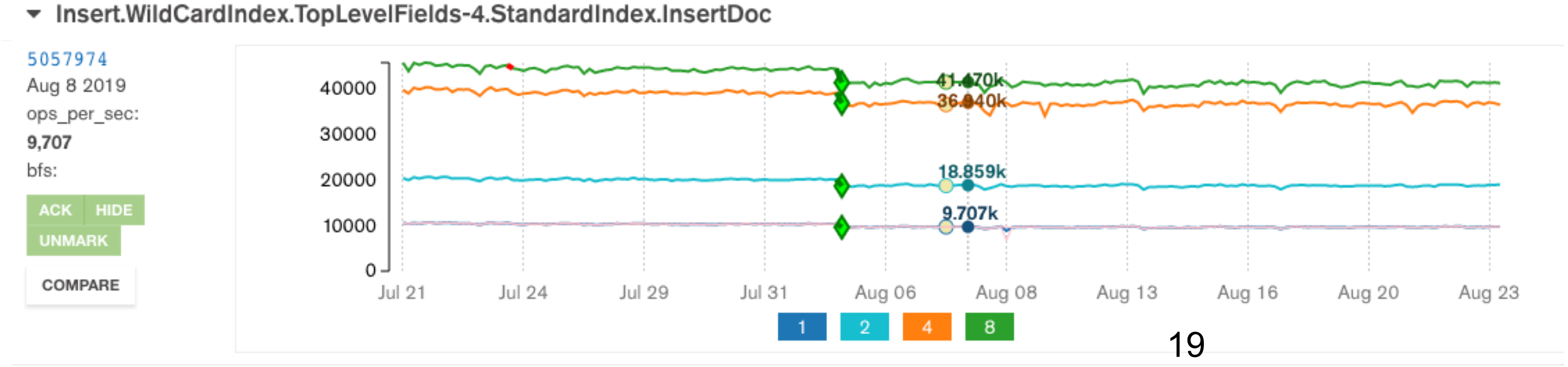
- Verify and isolate
- Create JIRA tickets



index_build_background

e62512d
Oct 15 2018
ops_per_sec:
73,708
bfs:

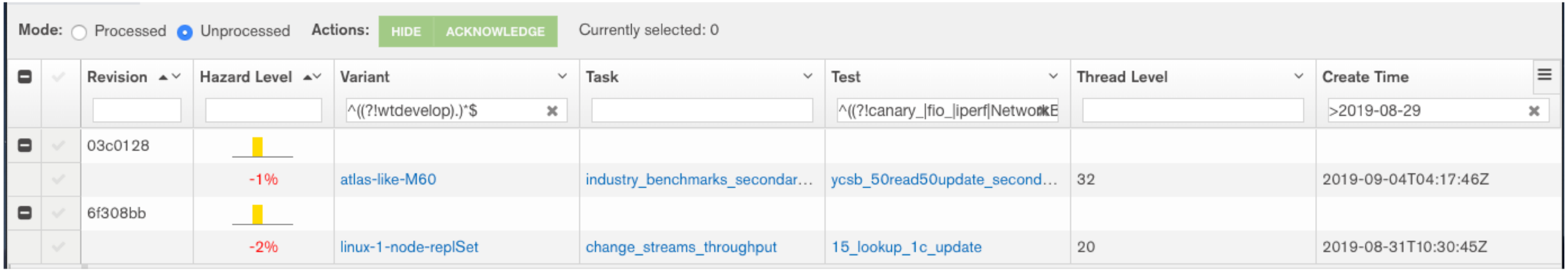
ACK HIDE
UNMARK
COMPARE



Insert.WildCardIndex.TopLevelFields-4.StandardIndex.InsertDoc

5057974
Aug 8 2019
ops_per_sec:
9,707
bfs:

ACK HIDE
UNMARK
COMPARE



Mode: ○ Processed ● Unprocessed Actions: HIDE ACKNOWLEDGE Currently selected: 0

⊟	✓	Revision	Hazard Level	Variant	Task	Test	Thread Level	Create Time
				^((?!wtdevelop).)*$ ✖		^((?!canary_|fio_|iperf|NetworkE		>2019-08-29 ✖
⊟	✓	03c0128						
	✓		-1%	atlas-like-M60	industry_benchmarks_secondar…	ycsb_50read50update_second…	32	2019-09-04T04:17:46Z
⊟	✓	6f308bb						
	✓		-2%	linux-1-node-replSet	change_streams_throughput	15_lookup_1c_update	20	2019-08-31T10:30:45Z

Impact: Does it Work?

Yes – Game Changing for us (but could still be better)

Qualitatively

- A human can process all the results
- Finding changes with smaller magnitude
- Finding changes faster → Regressions fixed sooner
- Recognizing improvements

Quantitatively

- E-divisive didn’t miss any real changes caught by the threshold system
- From 1% of notifications being useful to 67%

Work with (Help) Us

We have real world problems and would love to work with the community

- Noise Reduction work
- DBTest.io: “Automated System Performance Testing at MongoDB”
- LTB Talk: “How to Waste Time and Money Testing the Performance of a Software Product.”

Our code is open source: signal-processing-algorithms, infrastructure code

Our regression environment is open, and the platform is open source

Our performance data is not open source, but we’re working to share it with academics mongoDB®

Thank you

David Daly | Lead Engineer -- Performance | @daviddaly44 | https://daviddaly.me/