Performance at MongoDB

Understanding and Improving Performance at MongoDB

David Daly
Staff Performance Engineer
https://daviddaly.me/

Agenda

Basics of Performance Testing

Scaling Our Testing

Adding Features + Updates

Community Interactions

Basics of Performance Testing

Understand the performance of our software and when it changes

The focus for DSI was serving the more complex requirements of end-to-end system performance tests on real clusters, automating every step including provisioning of hardware, and generating consistent, repeatable results

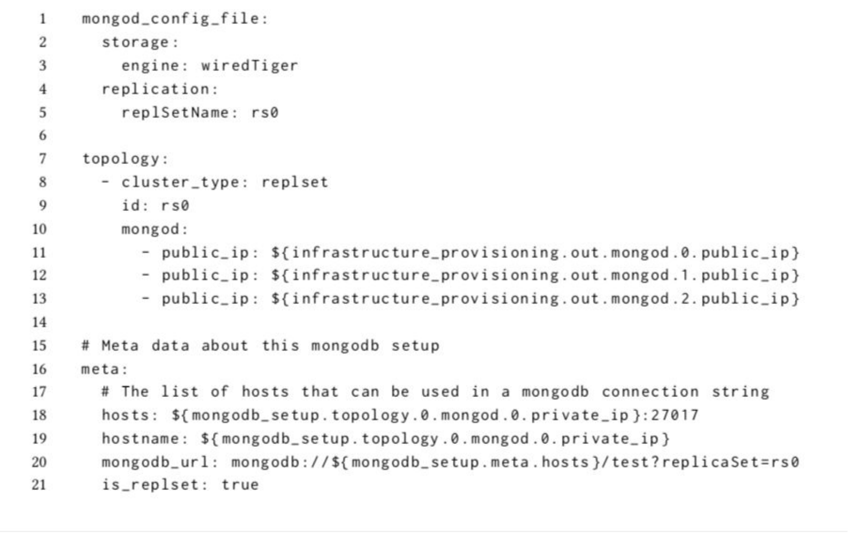
DSI Goals

- Full end-to-end automation
- Support both CI and manual testing
- Elastic, public cloud infrastructure
- Everything configurable
- All configuration via YAML
- Diagnosability
- Repeatability

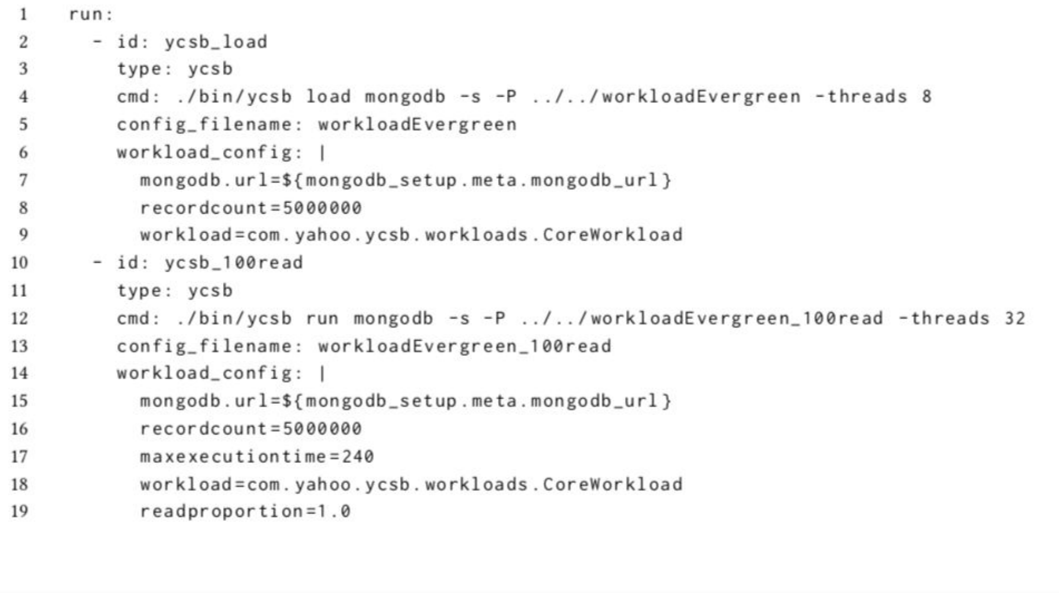
DSI Modules

- Bootstrap
- Infrastructure provisioning
- System setup
- Workload setup
- MongoDB setup
- Test Control
- Analysis
- Infrastructure teardown

Configuration Files



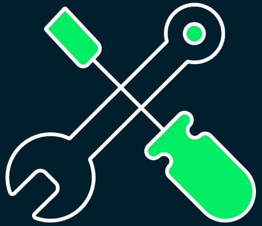
```
mongod_config_file:
  storage:
    engine: wiredTiger
  replication:
    replSetName: rs0

topology:
  - cluster_type: replset
    id: rs0
    mongod:
      - public_ip: ${infrastructure_provisioning.out.mongod.0.public_ip}
      - public_ip: ${infrastructure_provisioning.out.mongod.1.public_ip}
      - public_ip: ${infrastructure_provisioning.out.mongod.2.public_ip}

# Meta data about this mongodb setup
meta:
  # The list of hosts that can be used in a mongodb connection string
  hosts: ${mongodb_setup.topology.0.mongod.0.private_ip}:27017
  hostname: ${mongodb_setup.topology.0.mongod.0.private_ip}
  mongodb_url: mongodb://${mongodb_setup.meta.hosts}/test?replicaSet=rs0
  is_replset: true
```



```
run:
  - id: ycsb_load
    type: ycsb
    cmd: ./bin/ycsb load mongodb -s -P ../../workloadEvergreen -threads 8
    config_filename: workloadEvergreen
    workload_config: |
      mongodb.url=${mongodb_setup.meta.mongodb_url}
      recordcount=5000000
      workload=com.yahoo.ycsb.workloads.CoreWorkload
  - id: ycsb_100read
    type: ycsb
    cmd: ./bin/ycsb run mongodb -s -P ../../workloadEvergreen_100read -threads 32
    config_filename: workloadEvergreen_100read
    workload_config: |
      mongodb.url=${mongodb_setup.meta.mongodb_url}
      recordcount=5000000
      maxexecutiontime=240
      workload=com.yahoo.ycsb.workloads.CoreWorkload
      readproportion=1.0
```

Experiments

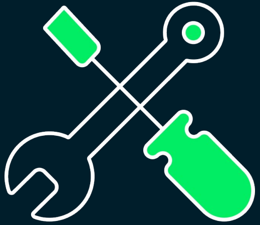
Controlling Noise



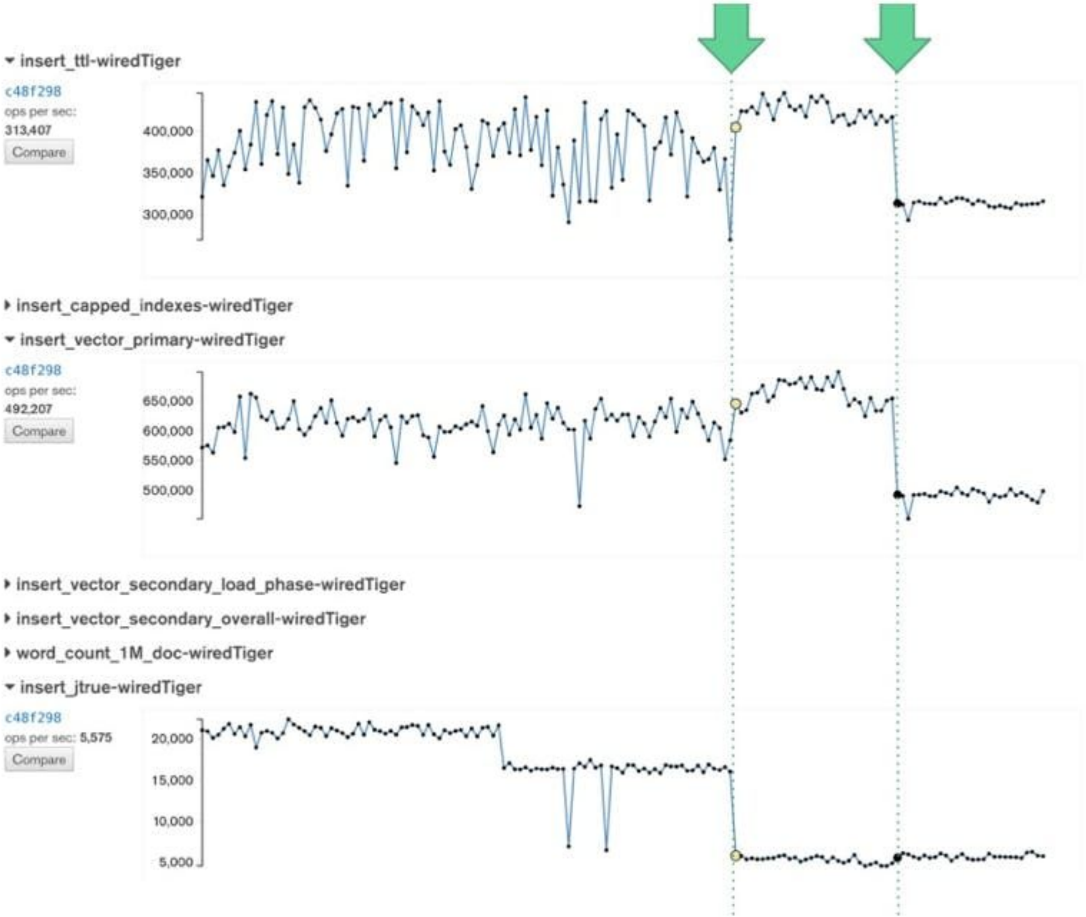
- Repeated Test Runs (5x)
- Canary Tests + Real Tests
- Most Assumptions Were Wrong

Results

Controlling Noise



- ✓ C3.8xlarge was lowest noise
- ✓ Replaced SSD with EBS PIOPS
- ✓ Disable Optimizations



▾ insert_ttl-wiredTiger

c48f298

ops per sec:

313,407

Compare

▸ insert_capped_indexes-wiredTiger

▾ insert_vector_primary-wiredTiger

c48f298

ops per sec:

492,207

Compare

▸ insert_vector_secondary_load_phase-wiredTiger

▸ insert_vector_secondary_overall-wiredTiger

▸ word_count_1M_doc-wiredTiger

▾ insert_jtrue-wiredTiger

c48f298

ops per sec: 5,575

Compare

○ SSD -> EBS

● CPU: No HT, single socket, scheduling

Decide if Perf Changed

Have human’s look at the graphs

We have a lot of graphs

Use a threshold

High false positive and false negative rate.
Identifies regressions on the wrong commit

Use Math

Use Change Point Detection algorithms and modern math.

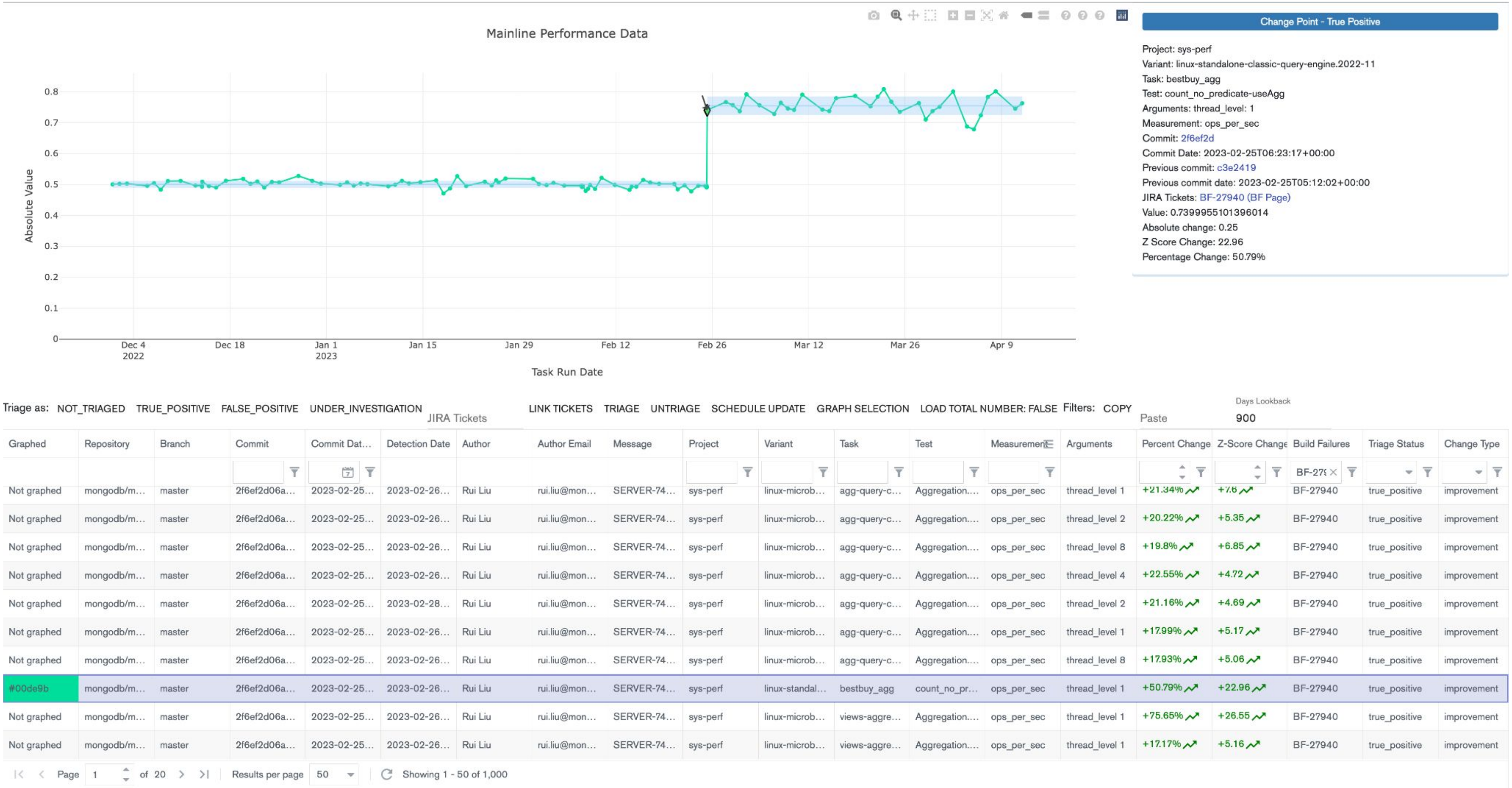
Problem

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.”



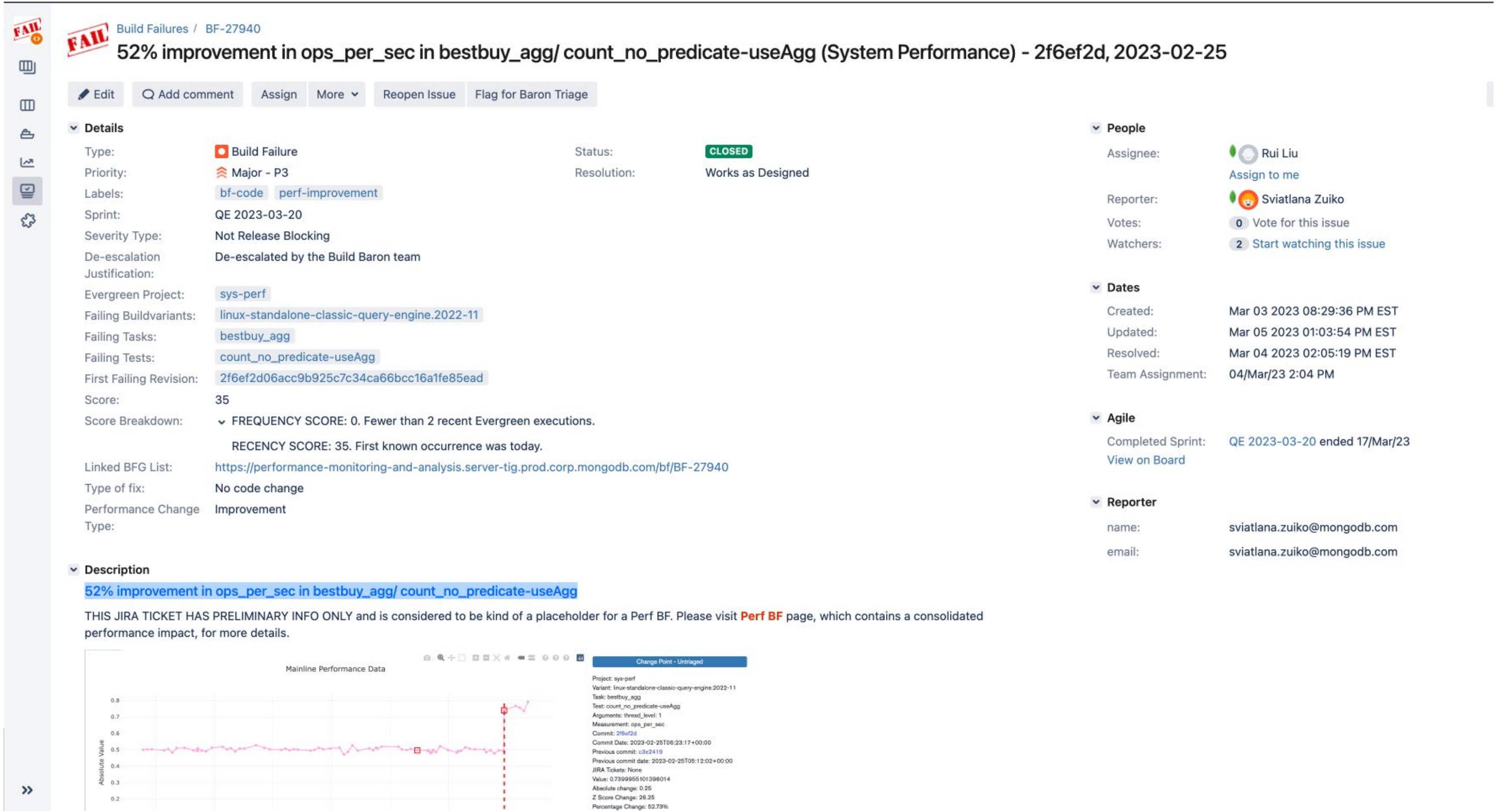
Change Point - True Positive

Project: sys-perf
Variant: linux-standalone-classic-query-engine.2022-11
Task: bestbuy_agg
Test: count_no_predicate-useAgg
Arguments: thread_level: 1
Measurement: ops_per_sec
Commit: 2f6ef2d
Commit Date: 2023-02-25T06:23:17+00:00
Previous commit: c3e2419
Previous commit date: 2023-02-25T05:12:02+00:00
JIRA Tickets: BF-27940 (BF Page)
Value: 0.7399955101396014
Absolute change: 0.25
Z Score Change: 22.96
Percentage Change: 50.79%

Triage as: NOT_TRIAGED TRUE_POSITIVE FALSE_POSITIVE UNDER_INVESTIGATION JIRA Tickets LINK TICKETS TRIAGE UNTRIAGE SCHEDULE UPDATE GRAPH SELECTION LOAD TOTAL NUMBER: FALSE Filters: COPY Paste Days Lookback 900

Graphed	Repository	Branch	Commit	Commit Dat…	Detection Date	Author	Author Email	Message	Project	Variant	Task	Test	Measurement	Arguments	Percent Change	Z-Score Change	Build Failures	Triage Status	Change Type
																	BF-27		
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-25…	2023-02-26…	Rui Liu	rui.liu@mon…	SERVER-74…	sys-perf	linux-microb…	agg-query-c…	Aggregation.…	ops_per_sec	thread_level 1	+21.34% ↗	+7.6 ↗	BF-27940	true_positive	improvement
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-25…	2023-02-26…	Rui Liu	rui.liu@mon…	SERVER-74…	sys-perf	linux-microb…	agg-query-c…	Aggregation.…	ops_per_sec	thread_level 2	+20.22% ↗	+5.35 ↗	BF-27940	true_positive	improvement
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-25…	2023-02-26…	Rui Liu	rui.liu@mon…	SERVER-74…	sys-perf	linux-microb…	agg-query-c…	Aggregation.…	ops_per_sec	thread_level 8	+19.8% ↗	+6.85 ↗	BF-27940	true_positive	improvement
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-25…	2023-02-26…	Rui Liu	rui.liu@mon…	SERVER-74…	sys-perf	linux-microb…	agg-query-c…	Aggregation.…	ops_per_sec	thread_level 4	+22.55% ↗	+4.72 ↗	BF-27940	true_positive	improvement
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-25…	2023-02-28…	Rui Liu	rui.liu@mon…	SERVER-74…	sys-perf	linux-microb…	agg-query-c…	Aggregation.…	ops_per_sec	thread_level 2	+21.16% ↗	+4.69 ↗	BF-27940	true_positive	improvement
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-25…	2023-02-26…	Rui Liu	rui.liu@mon…	SERVER-74…	sys-perf	linux-microb…	agg-query-c…	Aggregation.…	ops_per_sec	thread_level 1	+17.99% ↗	+5.17 ↗	BF-27940	true_positive	improvement
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-25…	2023-02-26…	Rui Liu	rui.liu@mon…	SERVER-74…	sys-perf	linux-microb…	agg-query-c…	Aggregation.…	ops_per_sec	thread_level 8	+17.93% ↗	+5.06 ↗	BF-27940	true_positive	improvement
#00de9b	mongodb/m…	master	2f6ef2d06a…	2023-02-25…	2023-02-26…	Rui Liu	rui.liu@mon…	SERVER-74…	sys-perf	linux-standal…	bestbuy_agg	count_no_pr…	ops_per_sec	thread_level 1	+50.79% ↗	+22.96 ↗	BF-27940	true_positive	improvement
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-25…	2023-02-26…	Rui Liu	rui.liu@mon…	SERVER-74…	sys-perf	linux-microb…	views-aggre…	Aggregation.…	ops_per_sec	thread_level 1	+75.65% ↗	+26.55 ↗	BF-27940	true_positive	improvement
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-25…	2023-02-26…	Rui Liu	rui.liu@mon…	SERVER-74…	sys-perf	linux-microb…	views-aggre…	Aggregation.…	ops_per_sec	thread_level 1	+17.17% ↗	+5.16 ↗	BF-27940	true_positive	improvement

Page 1 of 20 Results per page 50 Showing 1 - 50 of 1,000



Build Failures / BF-27940

52% improvement in ops_per_sec in bestbuy_agg/ count_no_predicate-useAgg (System Performance) - 2f6ef2d, 2023-02-25

Details

Type:	Build Failure
Priority:	Major - P3
Labels:	bf-code perf-improvement
Sprint:	QE 2023-03-20
Severity Type:	Not Release Blocking
De-escalation Justification:	De-escalated by the Build Baron team
Evergreen Project:	sys-perf
Failing Buildvariants:	linux-standalone-classic-query-engine.2022-11
Failing Tasks:	bestbuy_agg
Failing Tests:	count_no_predicate-useAgg
First Failing Revision:	2f6ef2d06acc9b925c7c34ca66bcc16a1fe85ead
Score:	35
Score Breakdown:	FREQUENCY SCORE: 0. Fewer than 2 recent Evergreen executions. RECENCY SCORE: 35. First known occurrence was today.
Linked BFG List:	https://performance-monitoring-and-analysis.server-tig.prod.corp.mongodb.com/bf/BF-27940
Type of fix:	No code change
Performance Change Type:	Improvement

Status:	CLOSED
Resolution:	Works as Designed

People

Assignee:	Rui Liu Assign to me
Reporter:	Sviatlana Zuiko
Votes:	0 Vote for this issue
Watchers:	2 Start watching this issue

Dates

Created:	Mar 03 2023 08:29:36 PM EST
Updated:	Mar 05 2023 01:03:54 PM EST
Resolved:	Mar 04 2023 02:05:19 PM EST
Team Assignment:	04/Mar/23 2:04 PM

Agile

Completed Sprint:	QE 2023-03-20 ended 17/Mar/23

View on Board

Reporter

name:	sviatlana.zuiko@mongodb.com
email:	sviatlana.zuiko@mongodb.com

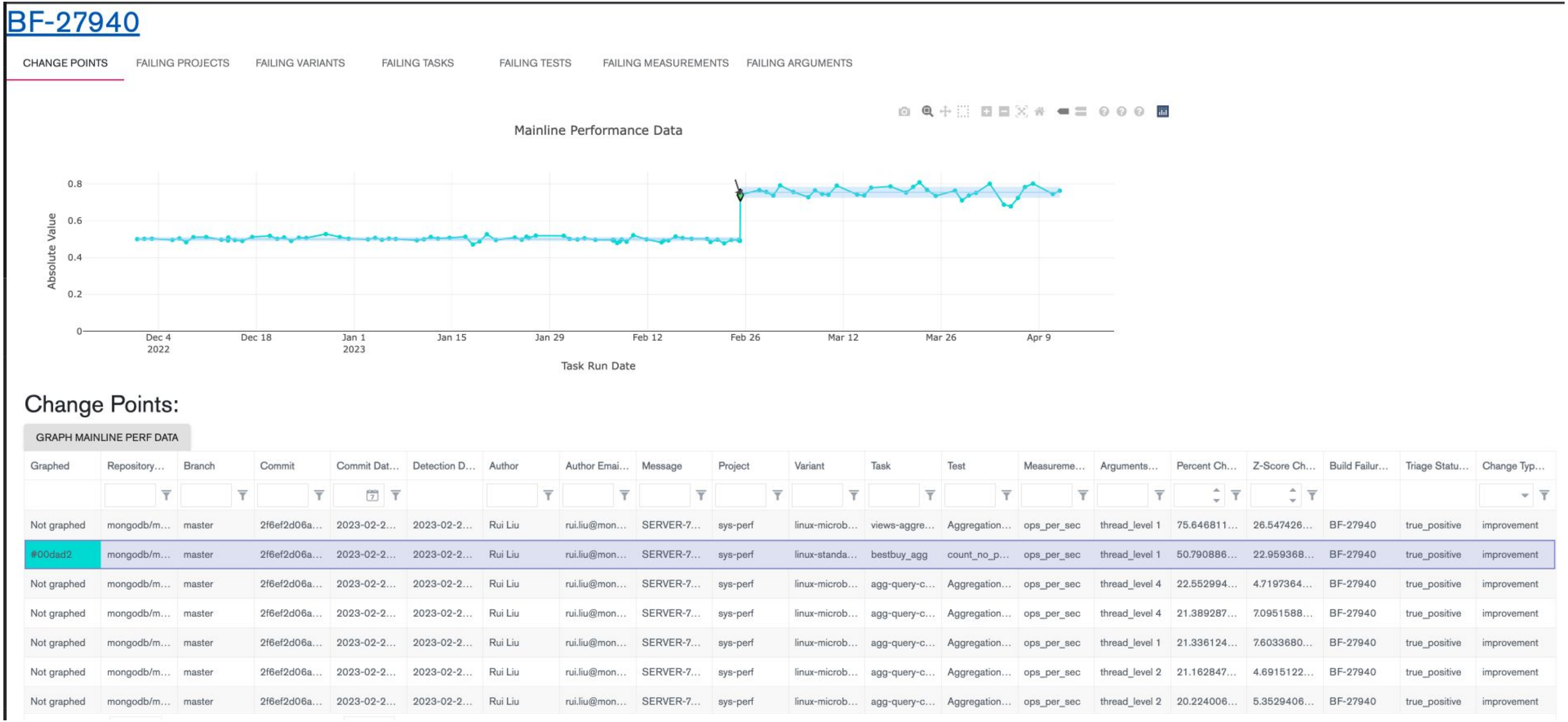
Description

52% improvement in ops_per_sec in bestbuy_agg/ count_no_predicate-useAgg

THIS JIRA TICKET HAS PRELIMINARY INFO ONLY and is considered to be kind of a placeholder for a Perf BF. Please visit **Perf BF** page, which contains a consolidated performance impact, for more details.

Change Point - Untriaged

Project: sys-perf
Variant: linux-standalone-classic-query-engine.2022-11
Task: bestbuy_agg
Test: count_no_predicate-useAgg
Arguments: thread_level: 1
Measurement: ops_per_sec
Commit: 2f6ef2d
Commit Date: 2023-02-25T06:23:17+00:00
Previous commit: c3e2419
Previous commit date: 2023-02-25T05:12:02+00:00
JIRA Tickets: None
Value: 0.7399955101398014
Absolute change: 0.25
Z Score Change: 26.25
Percentage Change: 52.73%



Change Points:

Graphed	Repository…	Branch	Commit	Commit Dat…	Detection D…	Author	Author Emai…	Message	Project	Variant	Task	Test	Measureme…	Arguments…	Percent Ch…	Z-Score Ch…	Build Failur…	Triage Statu…	Change Typ…
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-2…	2023-02-2…	Rui Liu	rui.liu@mon…	SERVER-7…	sys-perf	linux-microb…	views-aggre…	Aggregation…	ops_per_sec	thread_level 1	75.646811…	26.547426…	BF-27940	true_positive	improvement
#00dad2	mongodb/m…	master	2f6ef2d06a…	2023-02-2…	2023-02-2…	Rui Liu	rui.liu@mon…	SERVER-7…	sys-perf	linux-standa…	bestbuy_agg	count_no_p…	ops_per_sec	thread_level 1	50.790886…	22.959368…	BF-27940	true_positive	improvement
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-2…	2023-02-2…	Rui Liu	rui.liu@mon…	SERVER-7…	sys-perf	linux-microb…	agg-query-c…	Aggregation…	ops_per_sec	thread_level 4	22.552994…	4.7197364…	BF-27940	true_positive	improvement
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-2…	2023-02-2…	Rui Liu	rui.liu@mon…	SERVER-7…	sys-perf	linux-microb…	agg-query-c…	Aggregation…	ops_per_sec	thread_level 4	21.389287…	7.0951588…	BF-27940	true_positive	improvement
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-2…	2023-02-2…	Rui Liu	rui.liu@mon…	SERVER-7…	sys-perf	linux-microb…	agg-query-c…	Aggregation…	ops_per_sec	thread_level 1	21.336124…	7.6033680…	BF-27940	true_positive	improvement
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-2…	2023-02-2…	Rui Liu	rui.liu@mon…	SERVER-7…	sys-perf	linux-microb…	agg-query-c…	Aggregation…	ops_per_sec	thread_level 2	21.162847…	4.6915122…	BF-27940	true_positive	improvement
Not graphed	mongodb/m…	master	2f6ef2d06a…	2023-02-2…	2023-02-2…	Rui Liu	rui.liu@mon…	SERVER-7…	sys-perf	linux-microb…	agg-query-c…	Aggregation…	ops_per_sec	thread_level 2	20.224006…	5.3529406…	BF-27940	true_positive	improvement

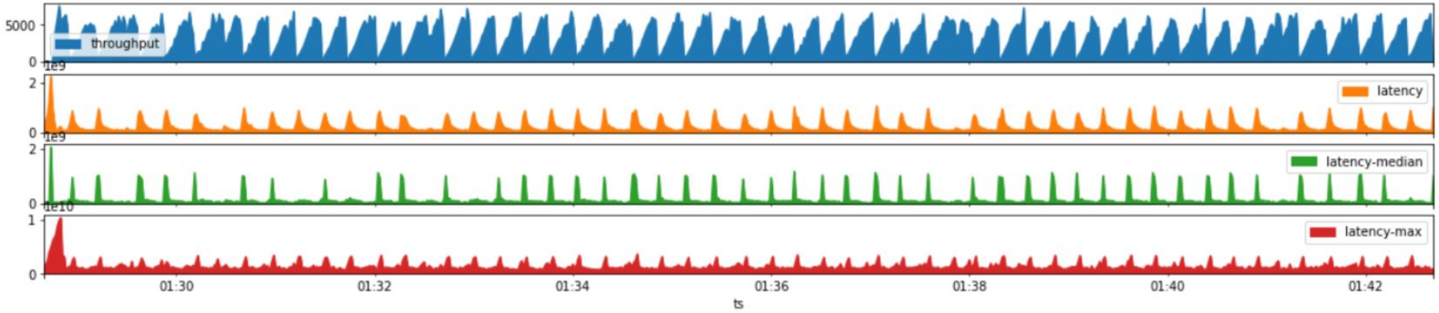
Scaling Our Testing

New Tools

- ✓ Load Generator – Measure Everything
- ✓ More Metrics
- ✓ Measure System Metrics

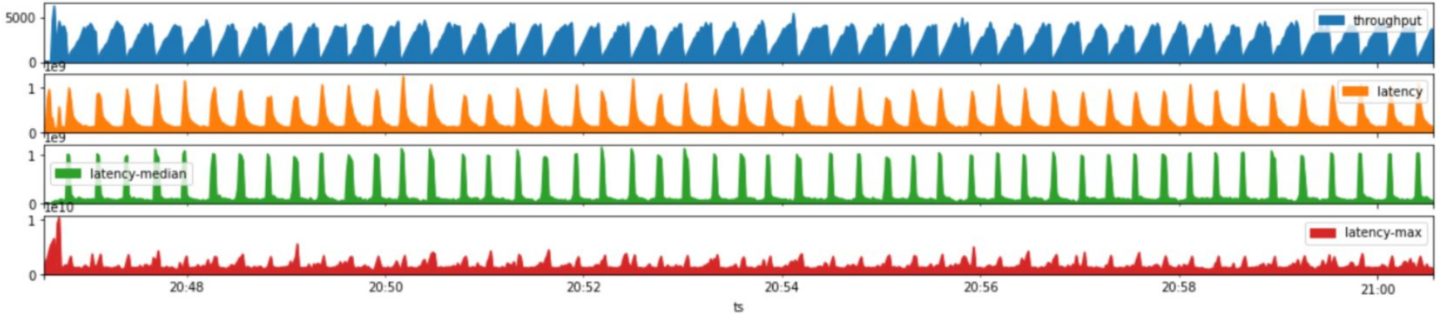
[53]:

```
plot(li512)
```

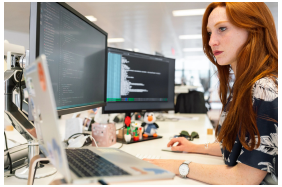


[54]:

```
plot(su512)
```

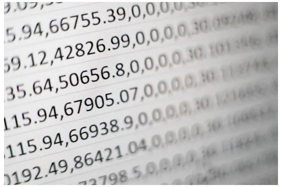


Problem is getting harder



X

X



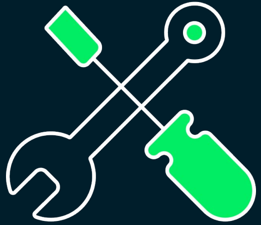
=

It Keeps Growing 5-10x a year!

	Variants	Tasks	Results
2018	10	80	
2019	17	170	2,195
2020	24	342	17,240
2021	30	395	157,834
2022	39	655	599,130
recent	54	1,054	1,051,150

Add Process

Throw People At the Problem



- ✓ Dedicated Build Barons
- ✓ Monthly Review of Performance
- ✓ Try to Automate More

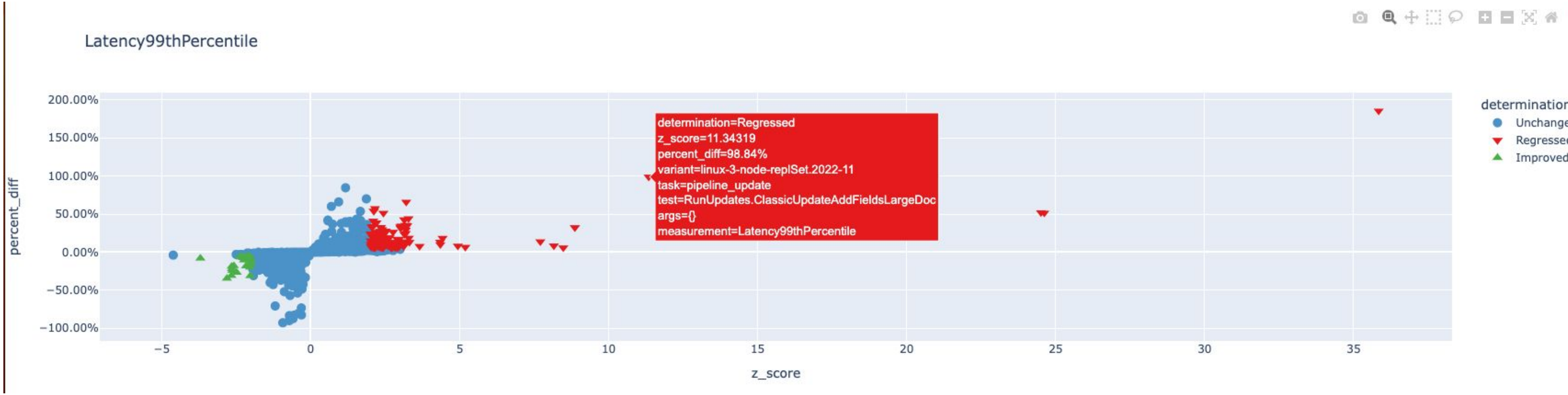
Adding Features + Updates

New Dashboards

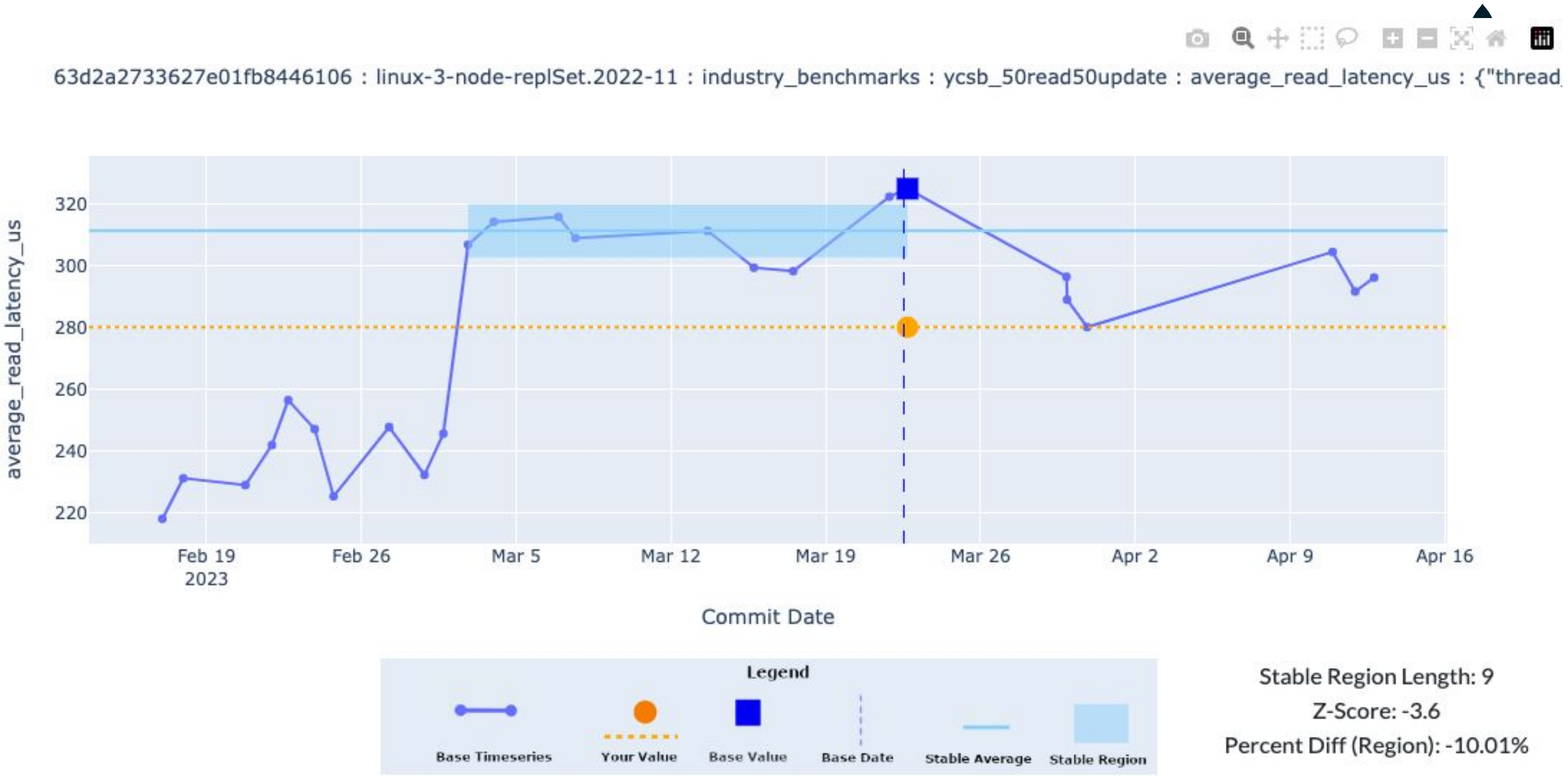
Compare experiment to mainline

Compare a branch/or feature on to disabled (SBE build)

All enabled from existing, but requiring more work



determination=Regressed
z_score=11.34319
percent_diff=98.84%
variant=linux-3-node-replSet.2022-11
task=pipeline_update
test=RunUpdates.ClassicUpdateAddFieldsLargeDoc
args={}
measurement=Latency99thPercentile

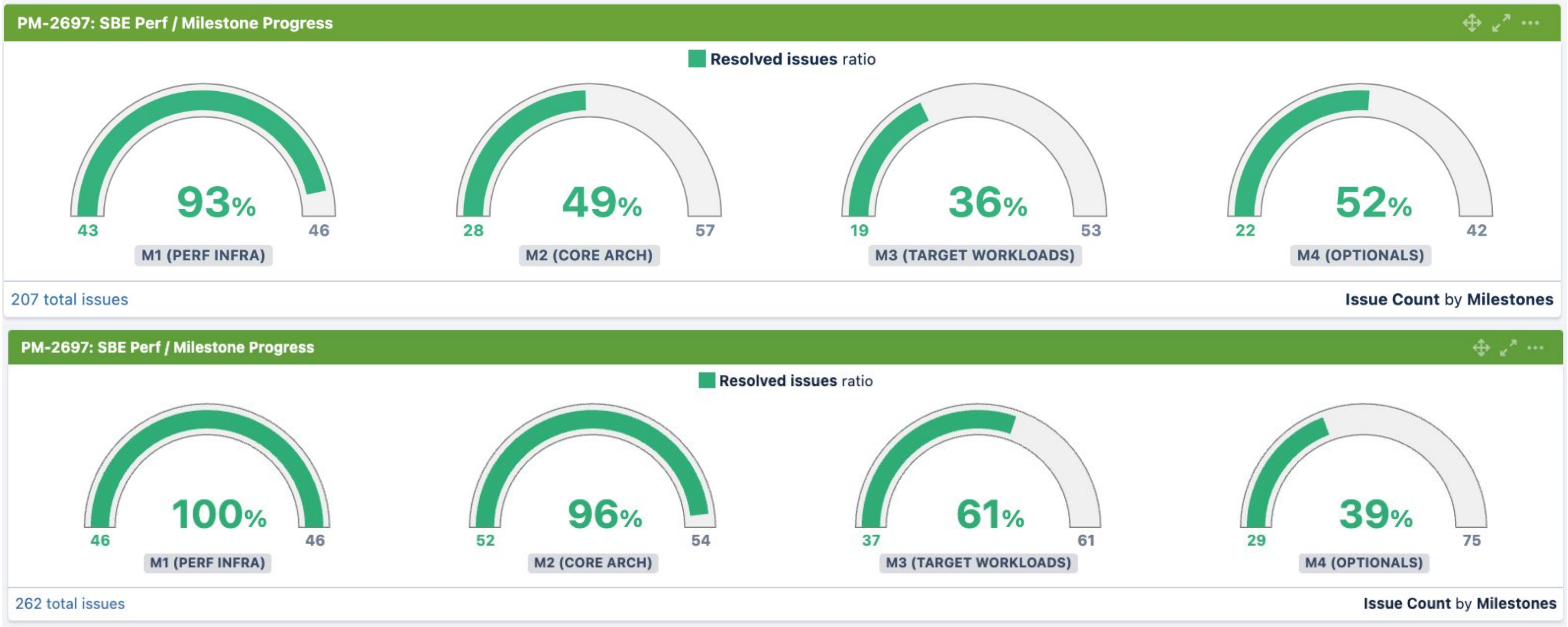


63d2a2733627e01fb8446106 : linux-3-node-replSet.2022-11 : industry_benchmarks : ycsb_50read50update : average_read_latency_us : {"thread

Stable Region Length: 9

Z-Score: -3.6

Percent Diff (Region): -10.01%



PM-2697: SBE Perf / Milestone Progress

🟩 **Resolved issues** ratio

207 total issues

Issue Count by **Milestones**

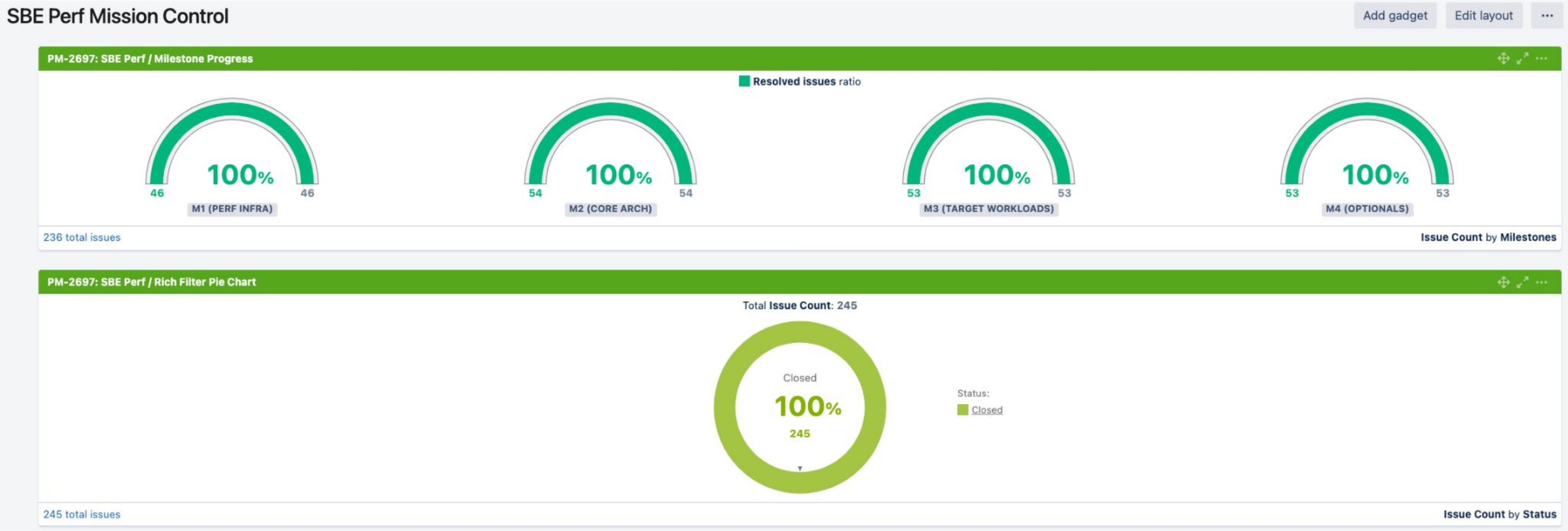
PM-2697: SBE Perf / Milestone Progress

🟩 **Resolved issues** ratio

262 total issues

Issue Count by **Milestones**

Feature Dashboard



Updating Test Systems

Pinned Systems

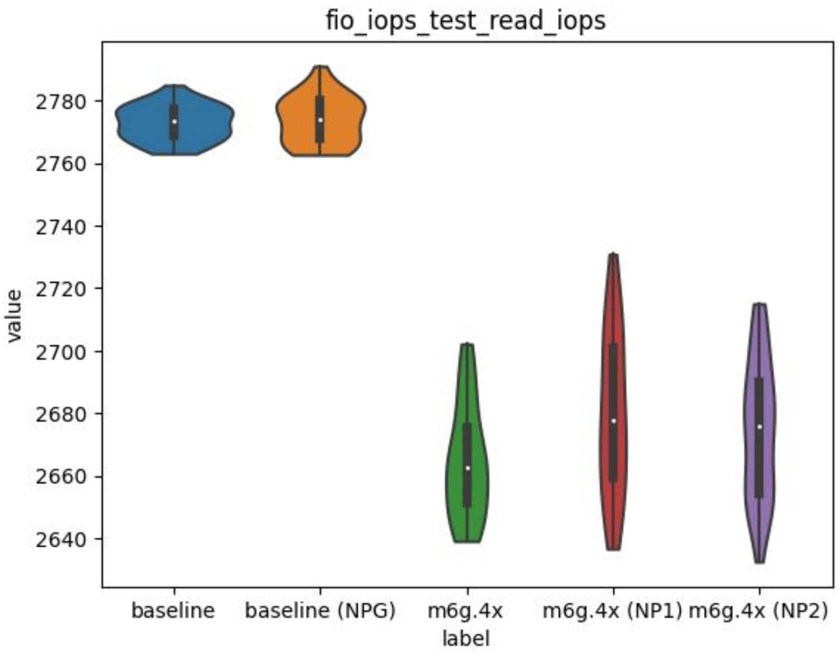
Systems became old!

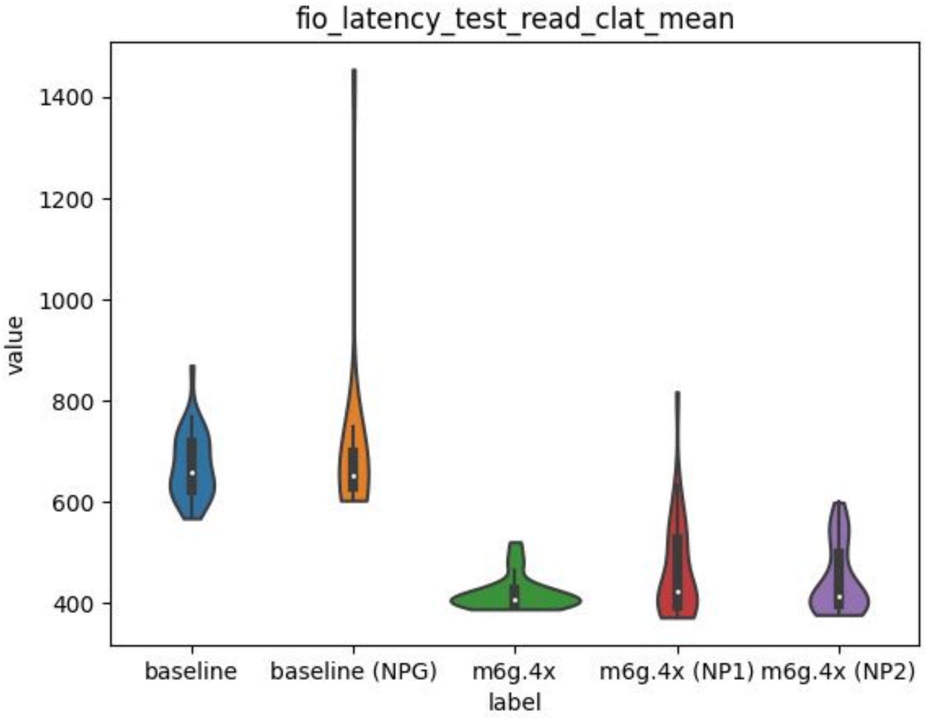
Must Control Noise

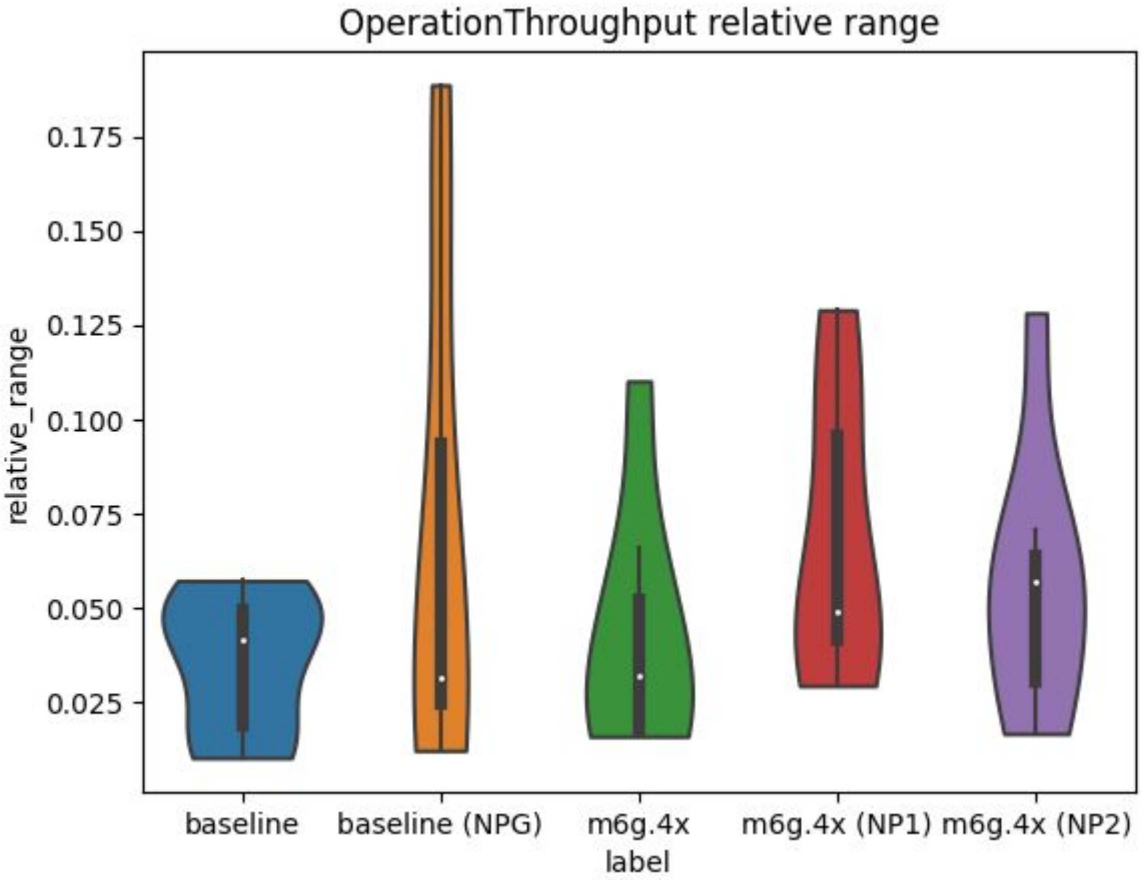
Ran several extensive experiments

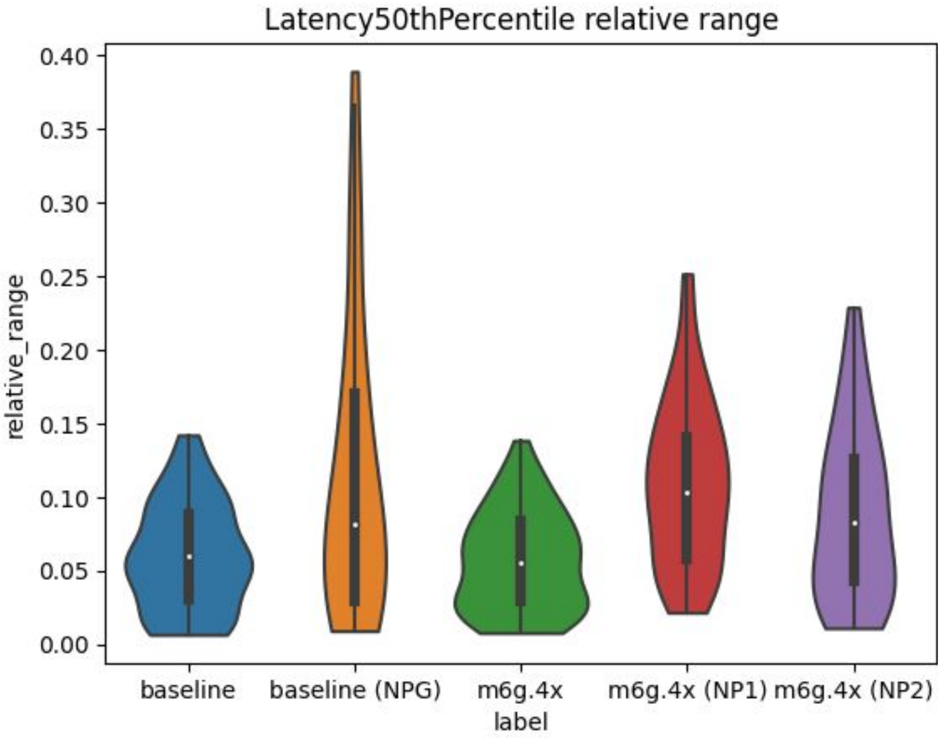
How to Upgrade

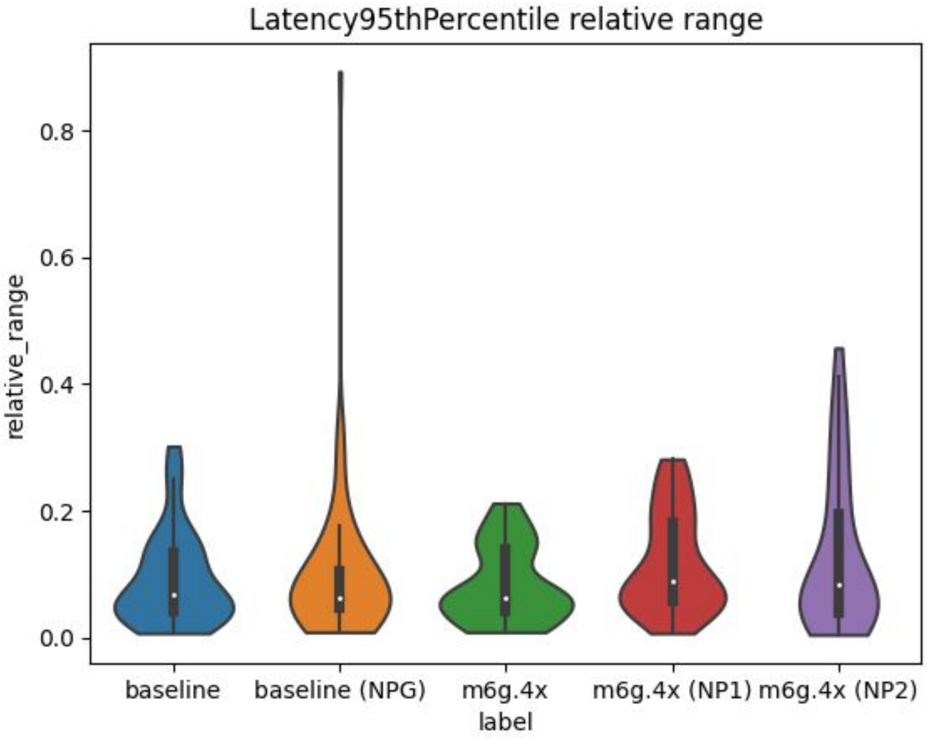
Added versioning to our system. Add new configurations – phase out old ones


fio_iops_test_read_iops
value
2780
2760
2740
2720
2700
2680
2660
2640
baseline
baseline (NPG)
m6g.4x
m6g.4x (NP1)
m6g.4x (NP2)
label


fio_latency_test_read_clat_mean
value
1400
1200
1000
800
600
400
baseline
baseline (NPG)
m6g.4x
m6g.4x (NP1)
m6g.4x (NP2)
label


OperationThroughput relative range
relative_range
0.175
0.150
0.125
0.100
0.075
0.050
0.025
baseline
baseline (NPG)
m6g.4x
m6g.4x (NP1)
m6g.4x (NP2)
label


Latency50thPercentile relative range
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
relative_range
baseline
baseline (NPG)
m6g.4x
m6g.4x (NP1)
m6g.4x (NP2)
label

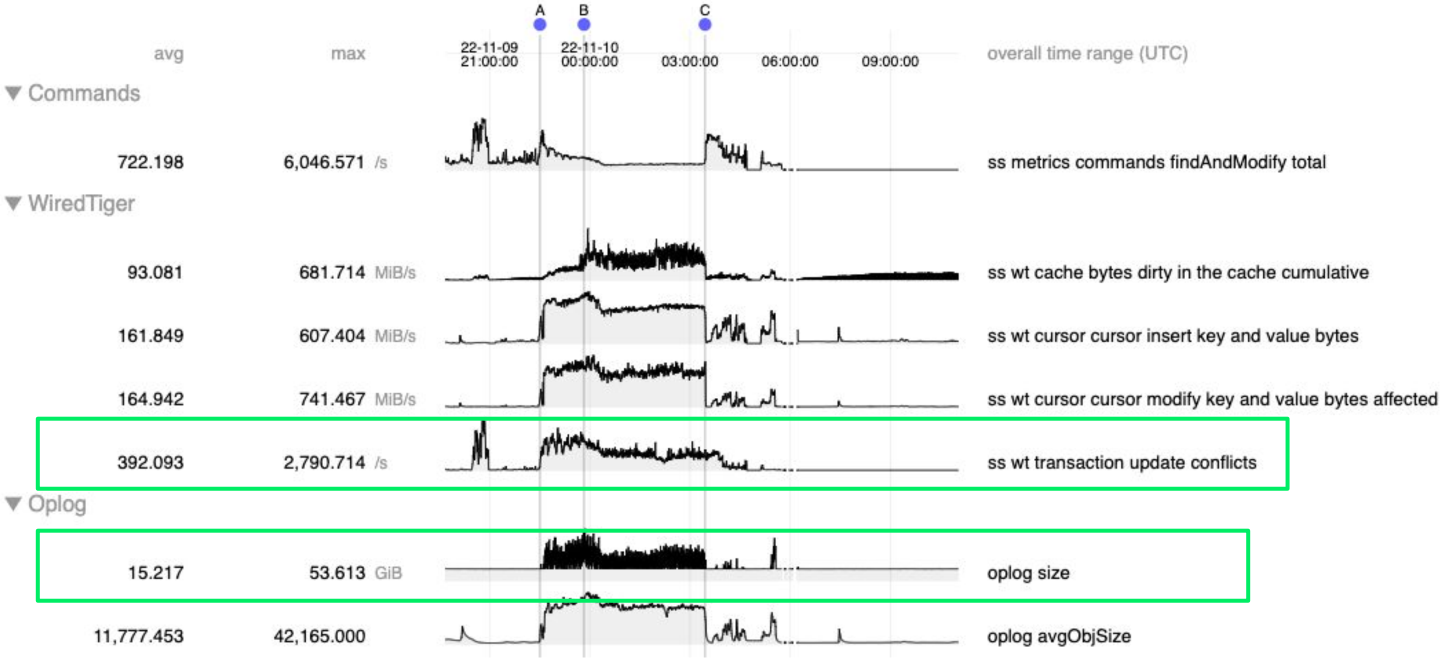

Latency95thPercentile relative range
0.8
0.6
0.4
0.2
0.0
relative_range
baseline
baseline (NPG)
m6g.4x
m6g.4x (NP1)
m6g.4x (NP2)
label

Diagnosis

Visualization of telemetry

Profiling support

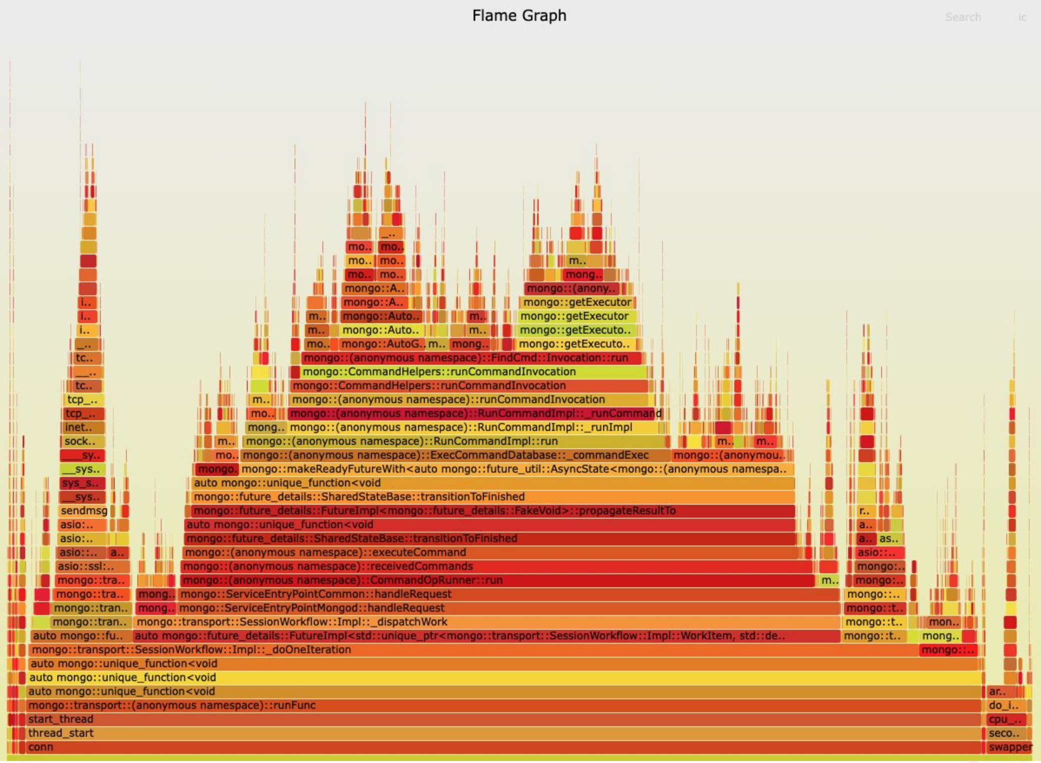
FTDC(t2)



avg	max	overall time range (UTC)
▼ Commands		
722.198	6,046.571 /s	ss metrics commands findAndModify total
▼ WiredTiger		
93.081	681.714 MiB/s	ss wt cache bytes dirty in the cache cumulative
161.849	607.404 MiB/s	ss wt cursor cursor insert key and value bytes
164.942	741.467 MiB/s	ss wt cursor cursor modify key and value bytes affected
392.093	2,790.714 /s	ss wt transaction update conflicts
▼ Oplog		
15.217	53.613 GiB	oplog size
11,777.453	42,165.000	oplog avgObjSize

Profiling with Flamegraphs

- Can automatically rerun task to
  - Profile
  - Create Flame Graph

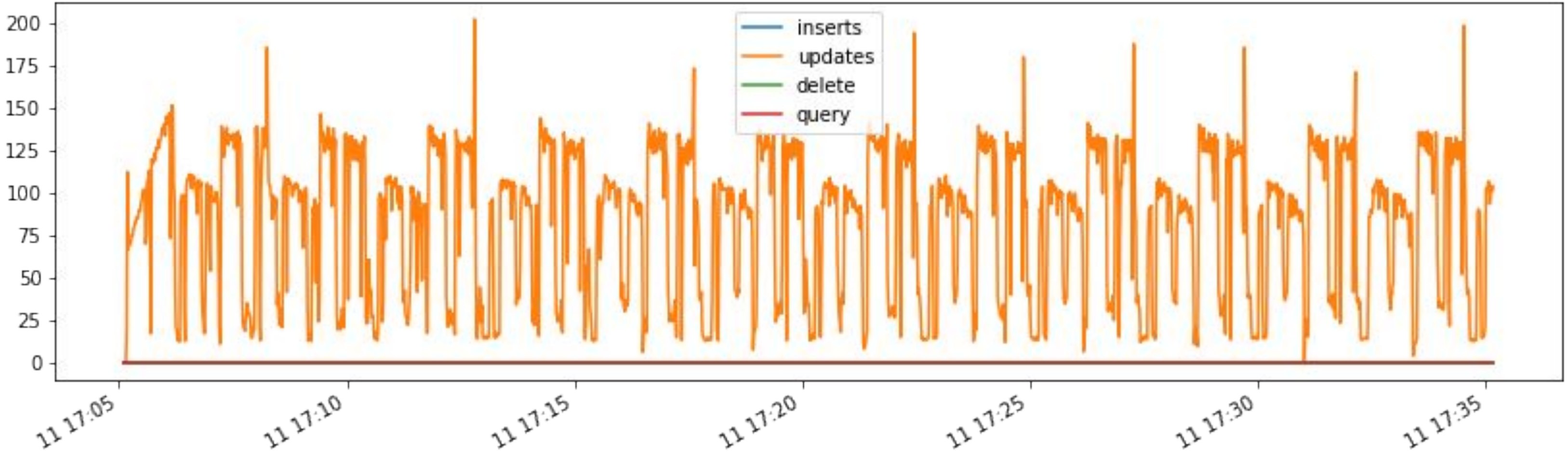


[Brendan Gregg FlameGraphs]

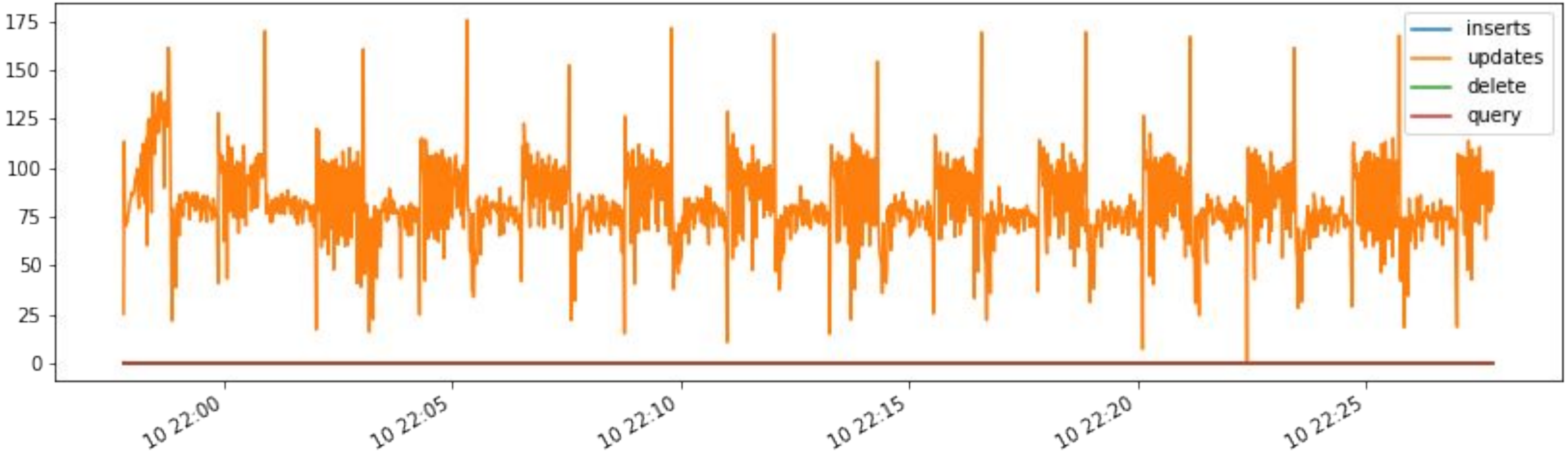
Stability >> Raw Performance

How do you measure stability?

Not well



200
175
150
125
100
75
50
25
0
inserts
updates
delete
query
11 17:05
11 17:10
11 17:15
11 17:20
11 17:25
11 17:30
11 17:35

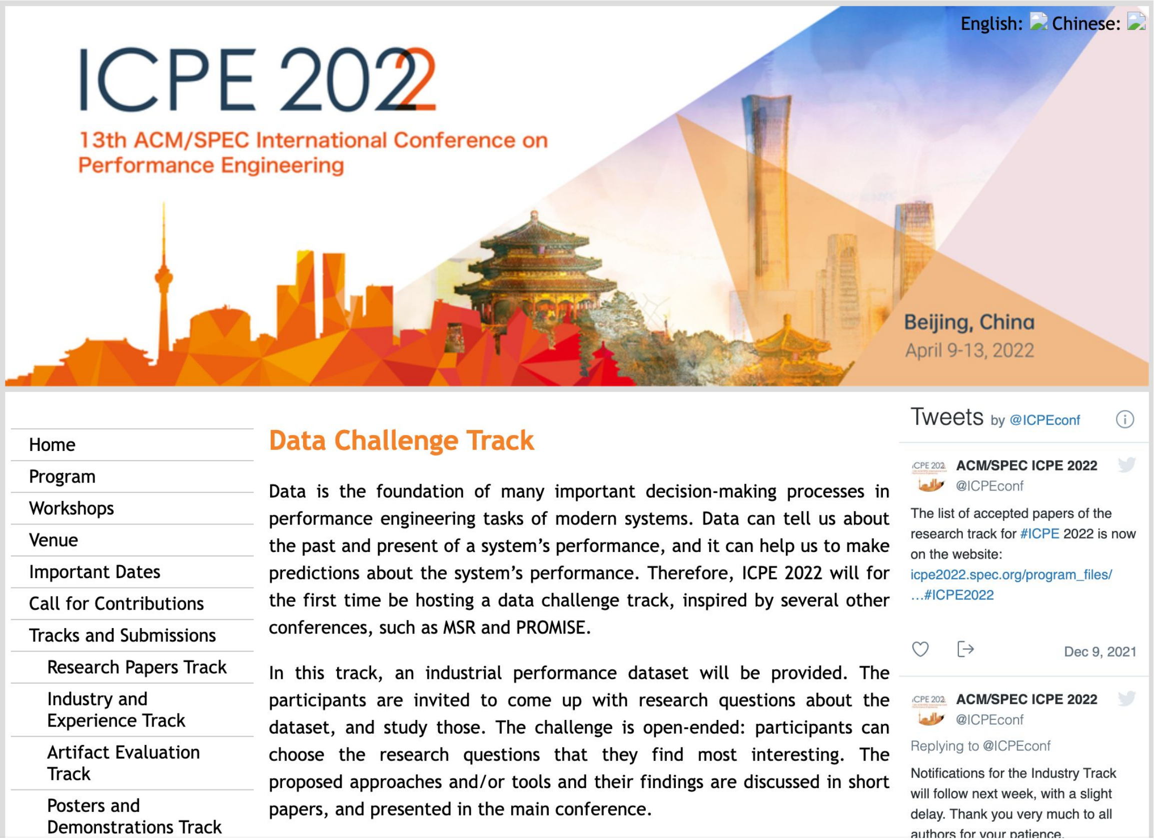


175
150
125
100
75
50
25
0
inserts
updates
delete
query
10 22:00
10 22:05
10 22:10
10 22:15
10 22:20
10 22:25

Community Interactions

Data Challenge

- CFP
- Notes
- Data
- 4 Accepted Papers



Data Challenge Track

Data is the foundation of many important decision-making processes in performance engineering tasks of modern systems. Data can tell us about the past and present of a system's performance, and it can help us to make predictions about the system's performance. Therefore, ICPE 2022 will for the first time be hosting a data challenge track, inspired by several other conferences, such as MSR and PROMISE.

In this track, an industrial performance dataset will be provided. The participants are invited to come up with research questions about the dataset, and study those. The challenge is open-ended: participants can choose the research questions that they find most interesting. The proposed approaches and/or tools and their findings are discussed in short papers, and presented in the main conference.

ACM/SPEC ICPE 2022
@ICPEconf

The list of accepted papers of the research track for #ICPE 2022 is now on the website: icpe2022.spec.org/program_files/ …#ICPE2022

Dec 9, 2021

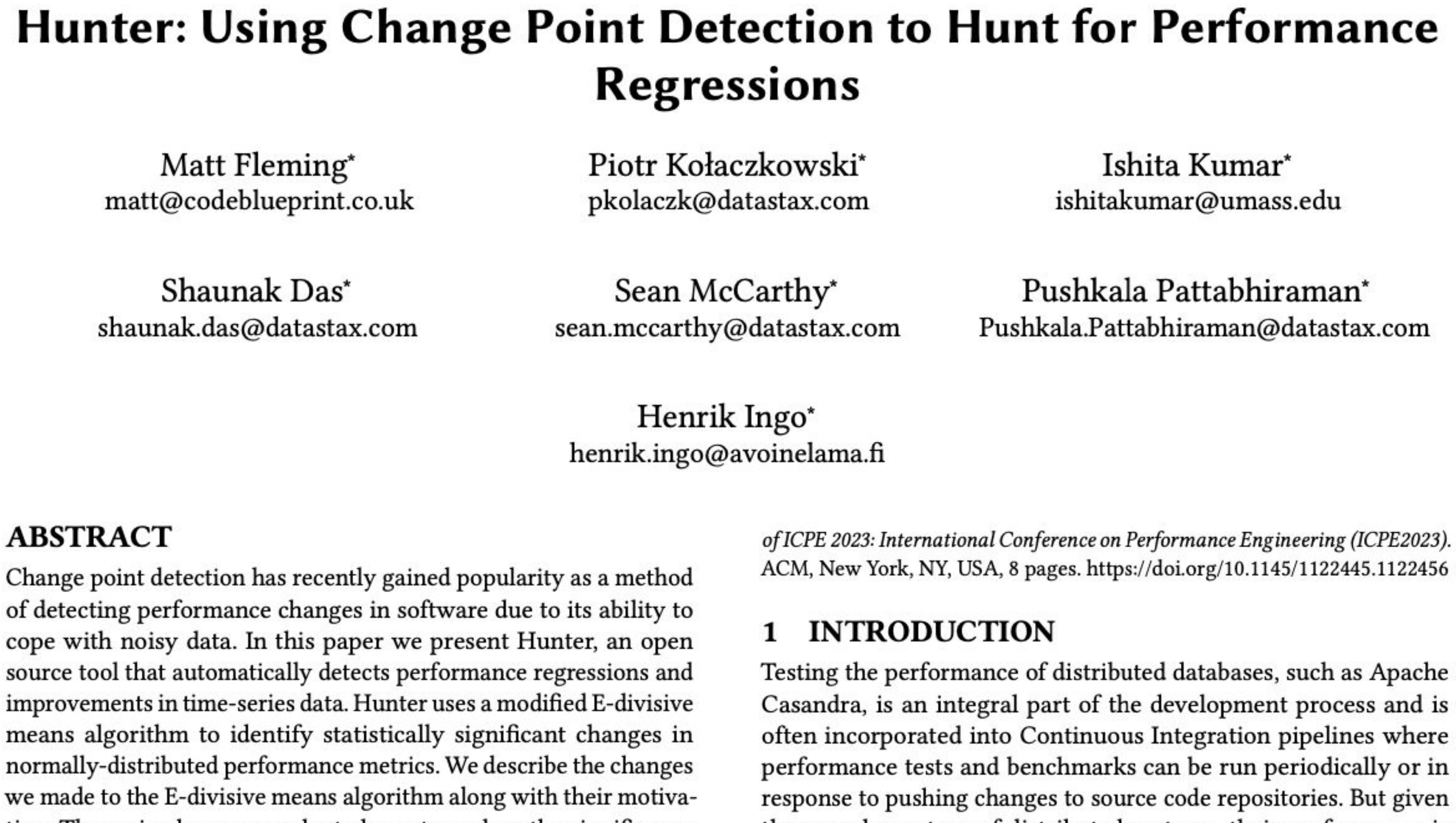
ACM/SPEC ICPE 2022
@ICPEconf

Replying to @ICPEconf

Notifications for the Industry Track will follow next week, with a slight delay. Thank you very much to all authors for your patience.

SPEC-RG

Hunter Paper Slide



Hunter: Using Change Point Detection to Hunt for Performance Regressions

Matt Fleming*
matt@codeblueprint.co.uk

Piotr Kołaczkowski*
pkolaczk@datastax.com

Ishita Kumar*
ishitakumar@umass.edu

Shaunak Das*
shaunak.das@datastax.com

Sean McCarthy*
sean.mccarthy@datastax.com

Pushkala Pattabhiraman*
Pushkala.Pattabhiraman@datastax.com

Henrik Ingo*
henrik.ingo@avoinelama.fi

ABSTRACT

Change point detection has recently gained popularity as a method of detecting performance changes in software due to its ability to cope with noisy data. In this paper we present Hunter, an open source tool that automatically detects performance regressions and improvements in time-series data. Hunter uses a modified E-divisive means algorithm to identify statistically significant changes in normally-distributed performance metrics. We describe the changes we made to the E-divisive means algorithm along with their motiva-

of ICPE 2023: International Conference on Performance Engineering (ICPE2023). ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Testing the performance of distributed databases, such as Apache Casandra, is an integral part of the development process and is often incorporated into Continuous Integration pipelines where performance tests and benchmarks can be run periodically or in response to pushing changes to source code repositories. But given

Work With 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"
- ICPE 2020: "The Use of Change Point Detection to Identify Software Performance Regressions in a Continuous Integration System" (video)
- ICPE 2021: Creating a Virtuous Cycle in Performance Testing at MongoDB

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

Final Thoughts

For academics

Consider the practical implications
Humans have to use these things and they have to scale
There are always things we need to work on. We don't need to find the optimal solution for anything, just a good enough solution for the most important things.

Practitioners

Automate everything. Consider the humans. Leverage what's out there, and share what you learn

Thank You