
AWS
re:Invent

DOP313

Using performance analysis to build efficient applications at Amazon

Jas Chhabra

Principal Engineer

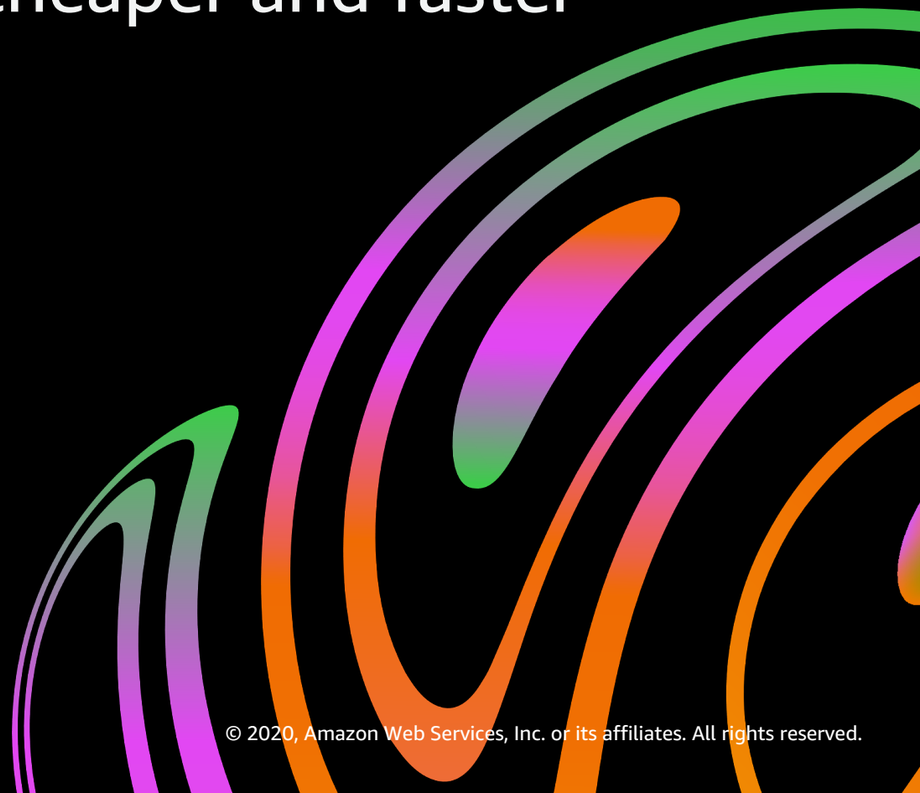
AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

- Why profile?
- Why profile in prod?
- Real-world stories on profiling to make apps cheaper and faster
- Takeaways



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why profile?



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon engineering priorities

- Security
- Durability
- Availability
- Speed
- Efficiency/Cost

Profiling focus

- Security
- Durability
- Availability
- Speed
- Efficiency/Cost

Profiling focus

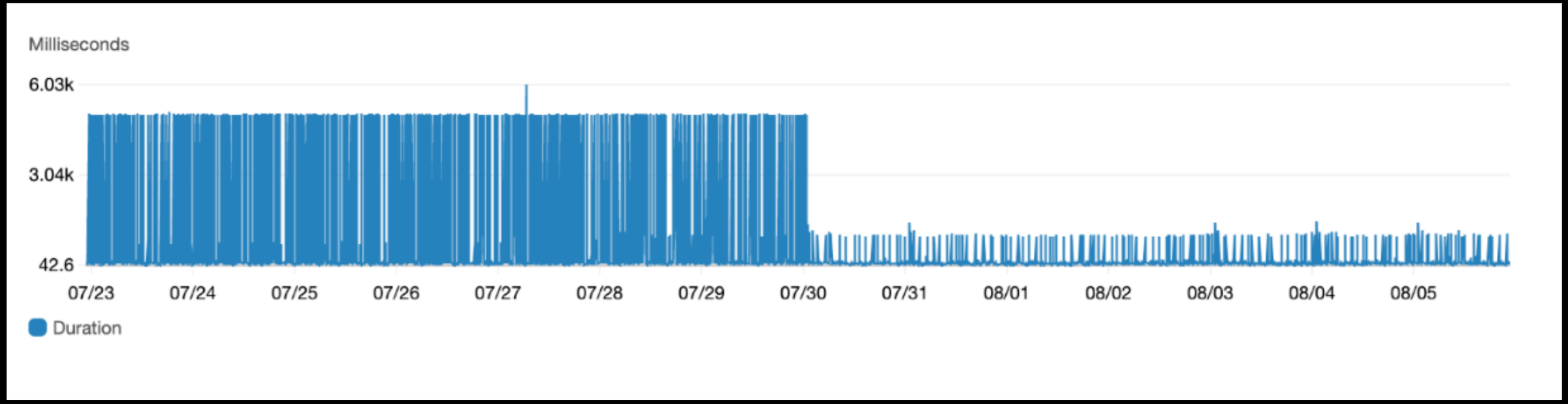
- Security
- Durability
- Availability
- Speed
- Efficiency/Cost

Cheaper and faster

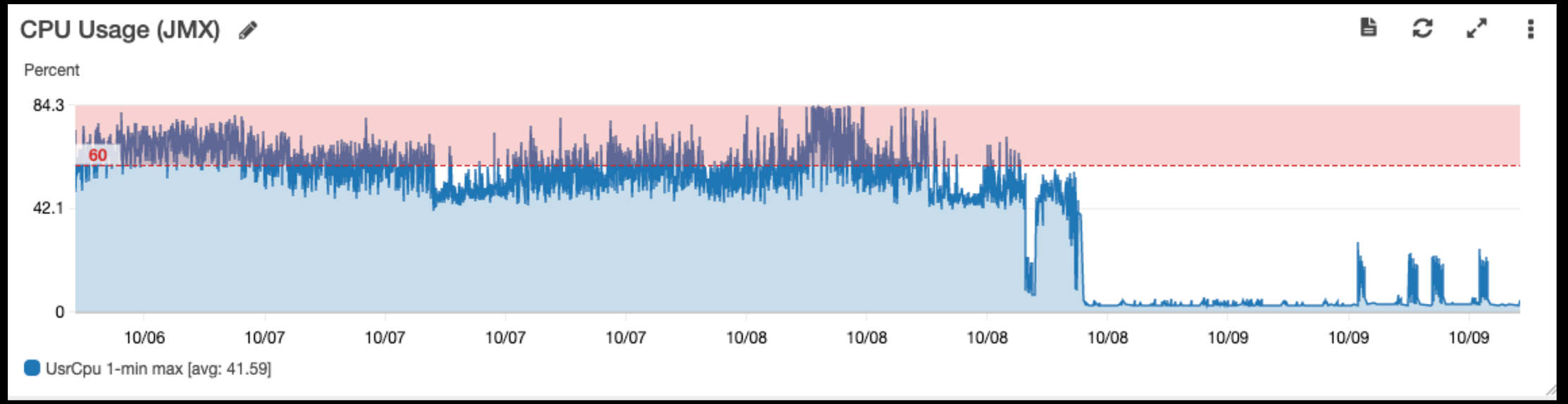
Increase customer happiness

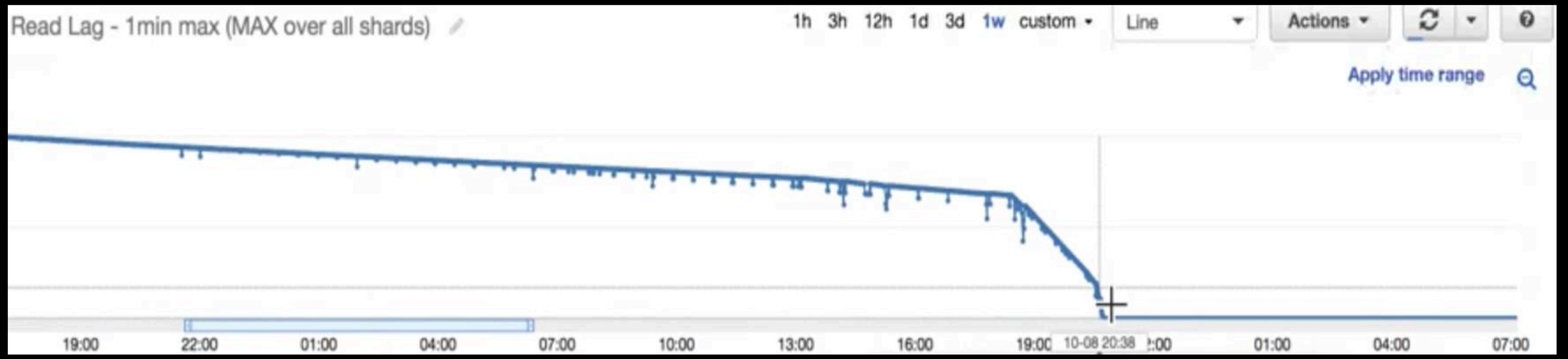
The best way to lose 3000 milliseconds fast.

Coming up: How an AWS service used Amazon CodeGuru Profiler to reduce P99 latency by 3000 milliseconds



What if I told that you are leaving money on the table?





Engineer: Lots of duct tape can fix anything

- Service under high load: Add more/bigger hardware
- Struggle to keep infrastructure costs down



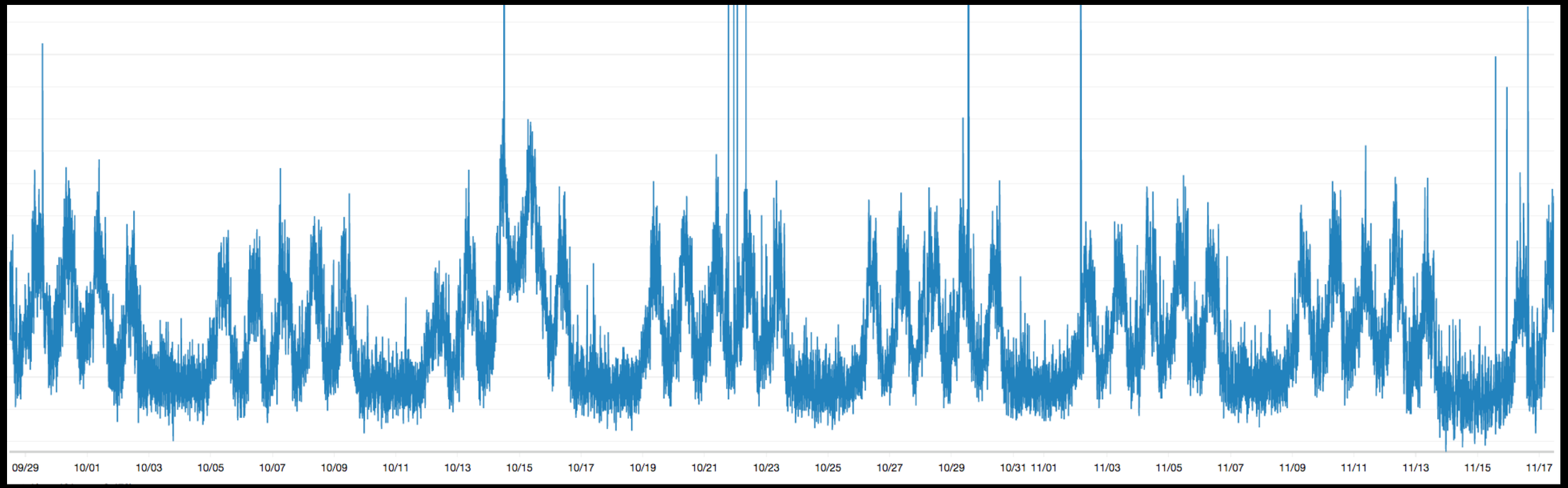
Why in prod?



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Prod vs. pre-prod

- Environment
- Traffic patterns
- Can’t go back in time



“But, if I run in prod, it will impact my service”

- Use a sampling profiler for near zero overhead on your service
- Example: CodeGuru Profiler

Important but not urgent

- Security
- Durability
- Availability
- **Delivering new products and features**
- Speed
- Efficiency/Cost

Limited time for this

What to pick?



Why not all three?



Build a culture: Make it easy

- 

Make it easy: Always on. Always available.

Build a culture: Create mechanisms



- **Make it easy:** Always on. Always available.
- **Mechanism:** Infrastructure cost is reviewed periodically.
- **Mechanism:** Share your wins. Celebrate your success.

How easy?

AWS Sr. PE David Yanacek finds inefficiencies during his coffee break



Result of culture change

- 1000s of apps use CodeGuru Profiler
- CodeGuru Profiler used to debug 100s of operational events
- Mentioned in 100s of performance wins
- 10s of millions of dollars saved

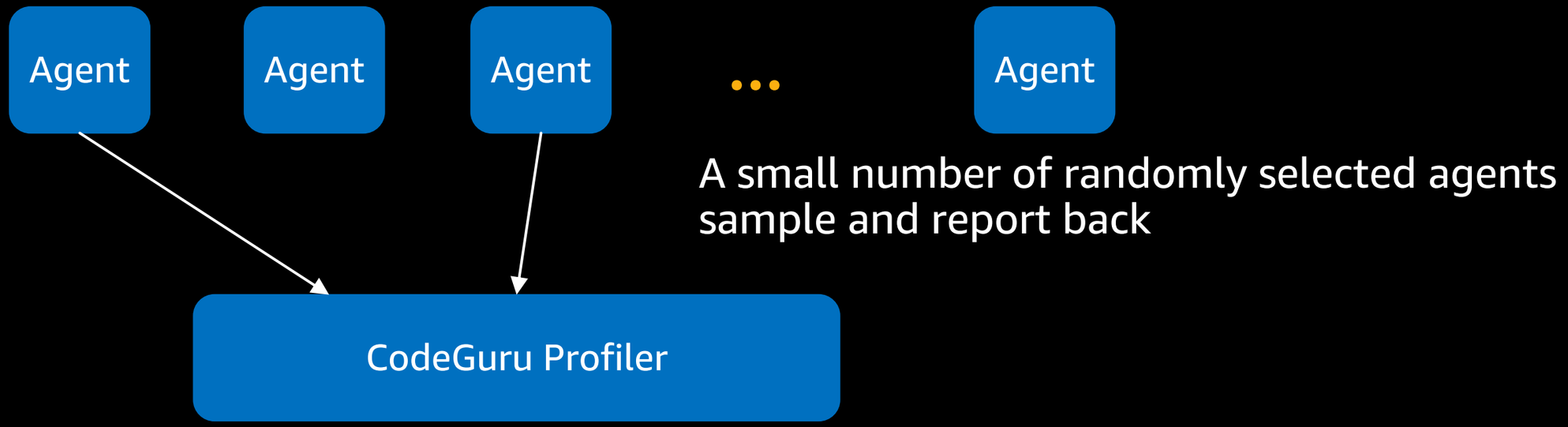
Profilers

- Event-based/tracing profilers
- Statistical/sampling profilers

How do sampling profilers work?

Example: Profiler used at Amazon (CodeGuru Profiler)

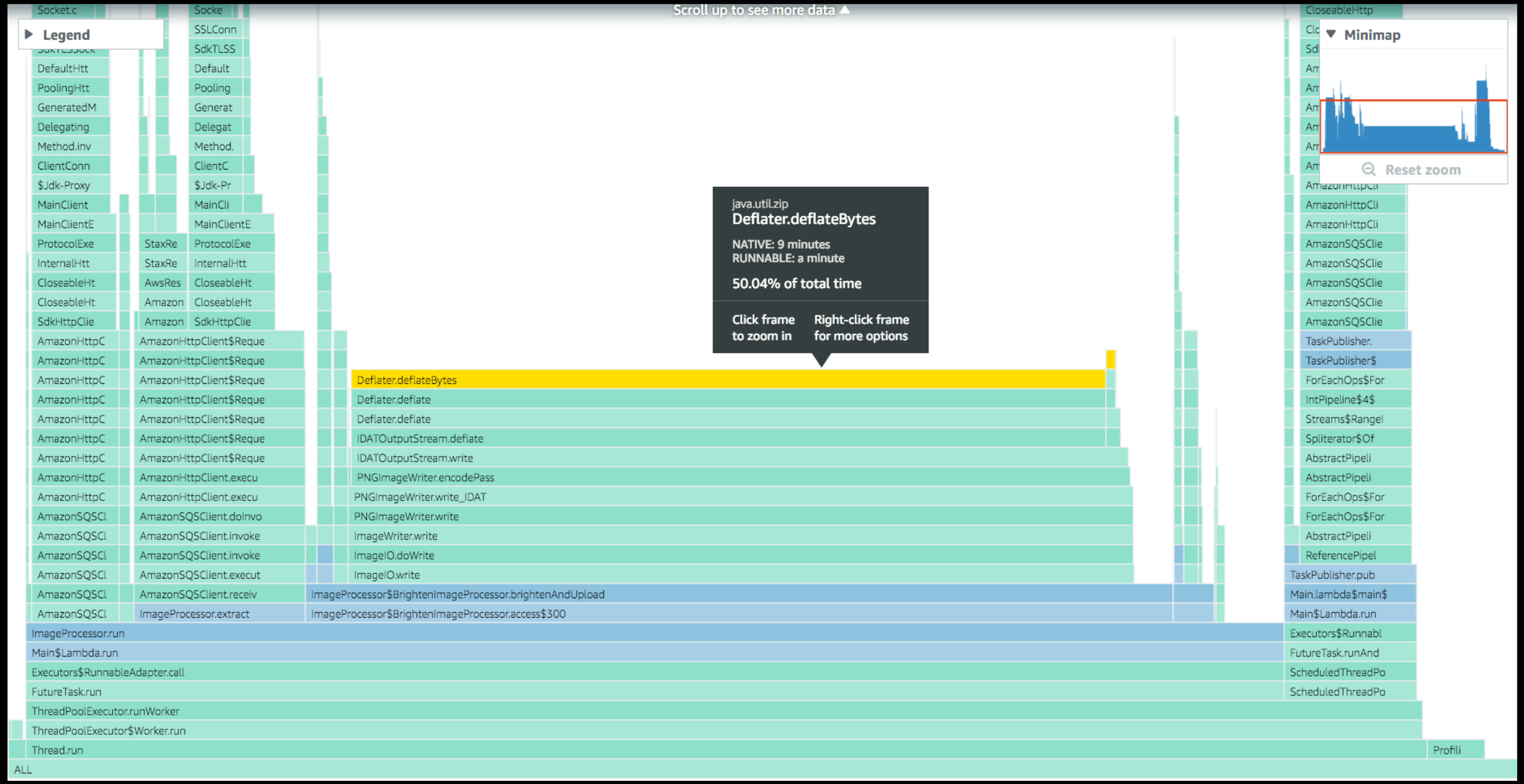
Agents sample infrequently. CPU load < 1%

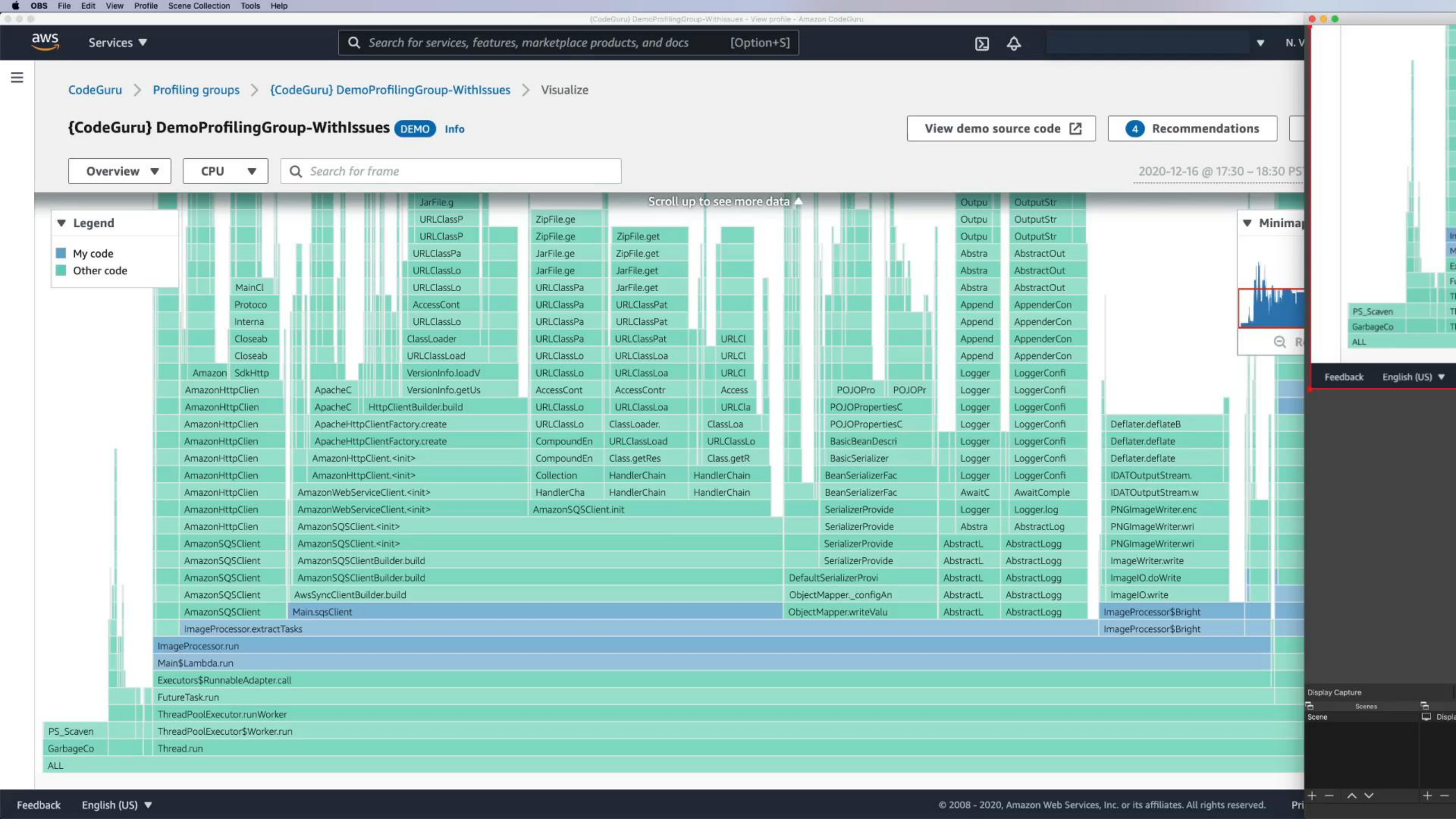


Why sampling profilers?

+ Always on. Always available.
+ Avoids side effects
+ Low impact
- Potential statistical inaccuracies. Avoided by:
  - Random sampling
  - Aggregate samples to ensure statistical significance and low error

Profiler visualizations




OBS File Edit View Profile Scene Collection Tools Help
{CodeGuru} DemoProfilingGroup-WithIssues - View profile - Amazon CodeGuru
aws Services Search for services, features, marketplace products, and docs [Option+S]
CodeGuru > Profiling groups > {CodeGuru} DemoProfilingGroup-WithIssues > Visualize
{CodeGuru} DemoProfilingGroup-WithIssues DEMO Info
View demo source code
4 Recommendations
Overview CPU Search for frame
2020-12-16 @ 17:30 – 18:30 PS
Scroll up to see more data
Legend
My code
Other code
Minimap
Feedback English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Display Capture
Scenes
Scene

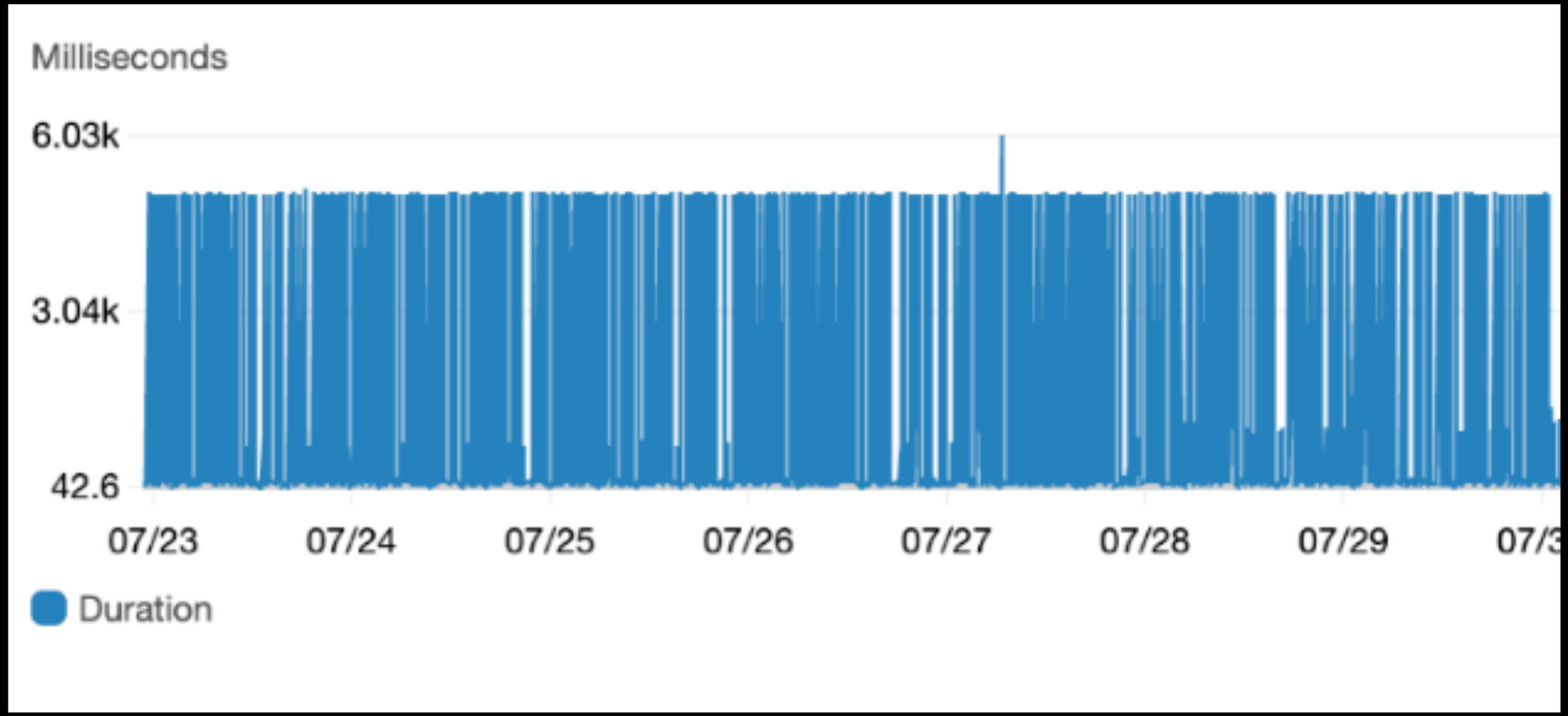
Real-world stories



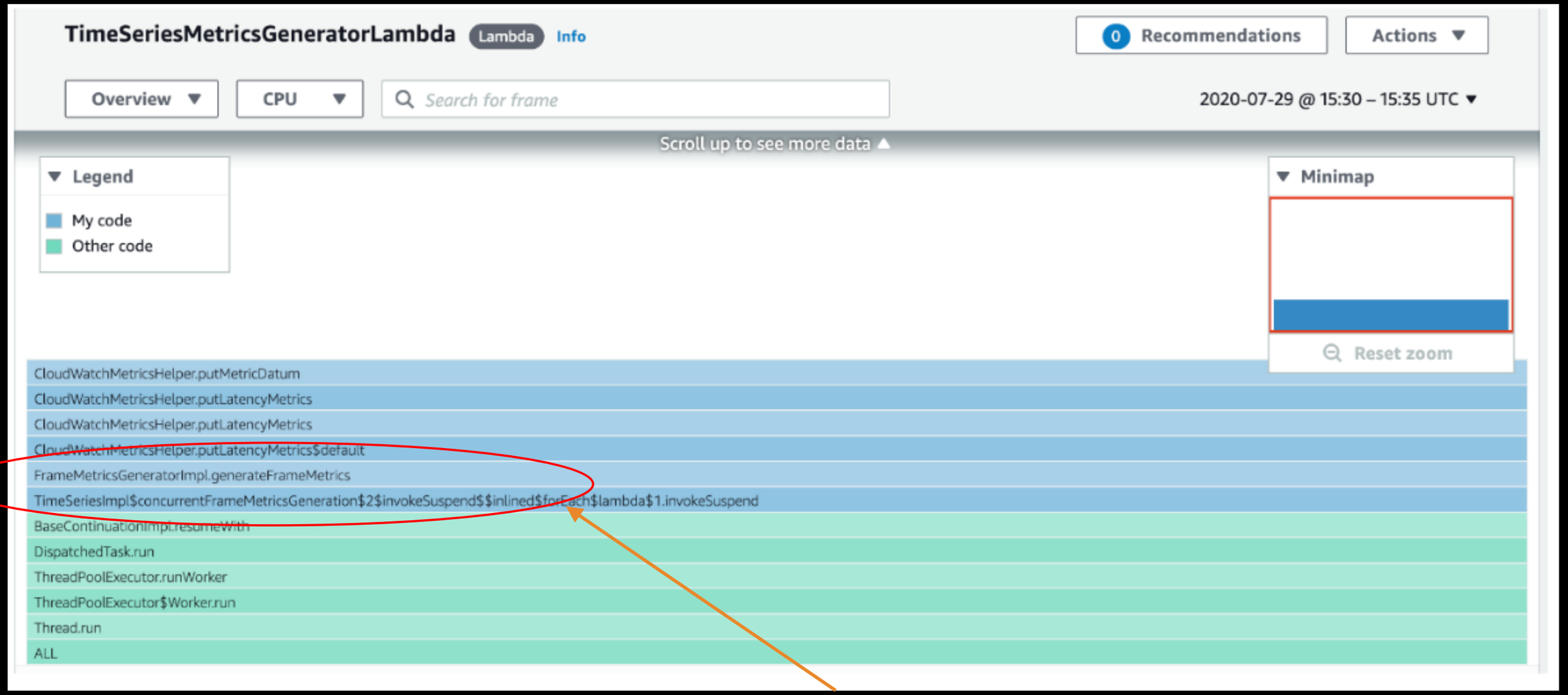
AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Issue: P99 latency spikes



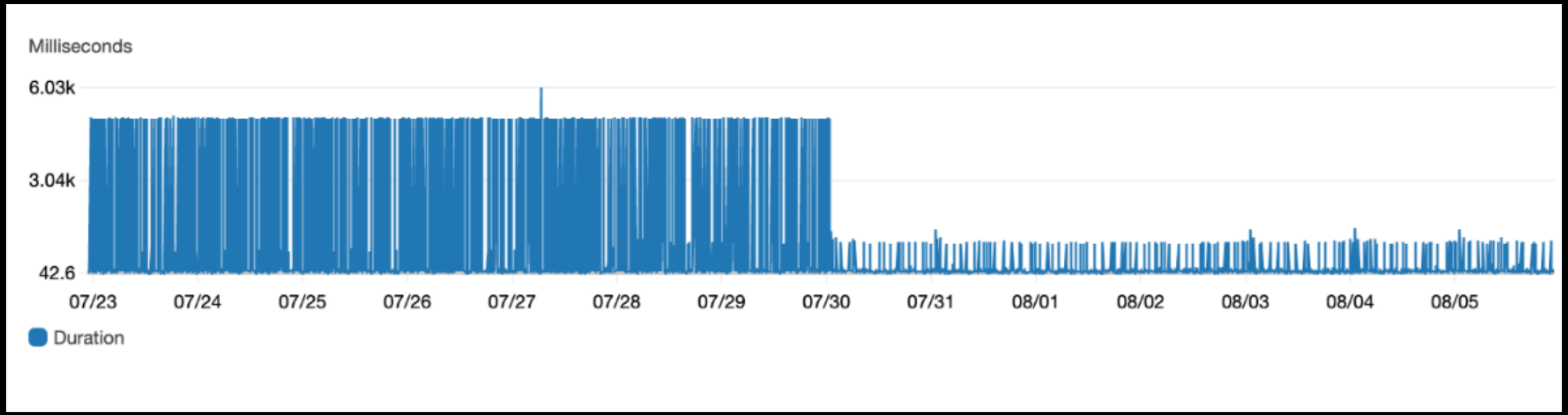
Insight



Thread publishing cloud-watch metrics in a synchronized block.
Rest of the threads were waiting.

Result

Team changed the `TimeSeriesMetricsGeneratorLambda` Lambda code, to publish CloudWatch metrics at the end of the function.



Issue

One of the Amazon retail services was experiencing latency delays and growing infrastructure costs.

Insight #1: Excessive garbage collection



Insight #2: Response metadata cache

AWS HTTP response metadata cache causing overhead

Why did this issue match?

Your profile spent **2.86%** of the RUNNABLE and BLOCKED time on frames related to this issue, but we would expect less than **1%** of your profile to be spent on these frames under most circumstances.

The top matched frames were `ResponseMetadataCache.add` .

Description

The AWS SDK v1.x HTTP client by default saves response metadata for debug purposes, which can cause performance overhead (both CPU usage and lock contention), especially in high throughput applications.

▼ **Suggested resolution steps**

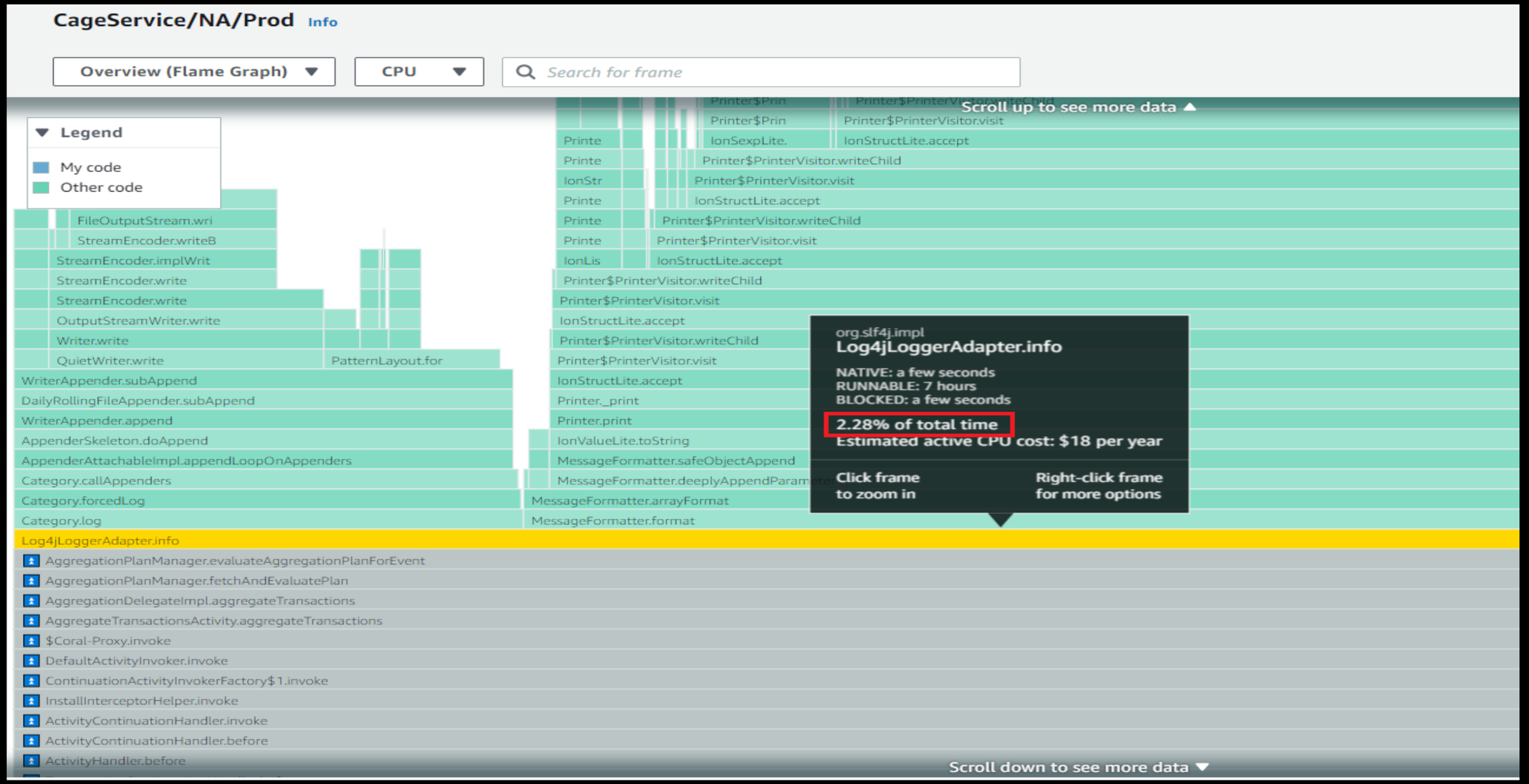
If you don't explicitly use this cache for something, you can disable it. From the official docs:

> Response metadata is typically used for troubleshooting issues with AWS support staff when services aren't acting as expected. While this feature is useful for debugging it adds overhead and disabling it may be desired in high throughput applications.

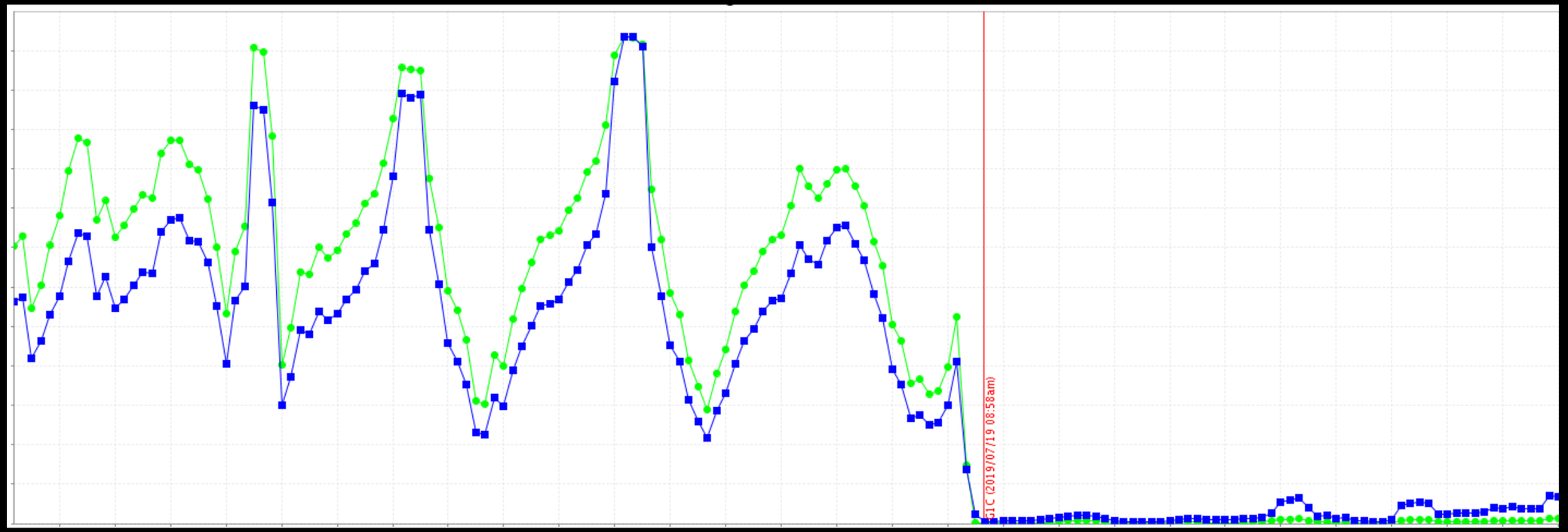
```
ClientConfiguration config = new ClientConfiguration()
    ... // any other setup you need for your application
    .withCacheResponseMetadata(false); // improve client performance

DynamoDB ddb = AmazonDynamoDBClient.builder()
    .withClientConfiguration(config)
    .build();
```

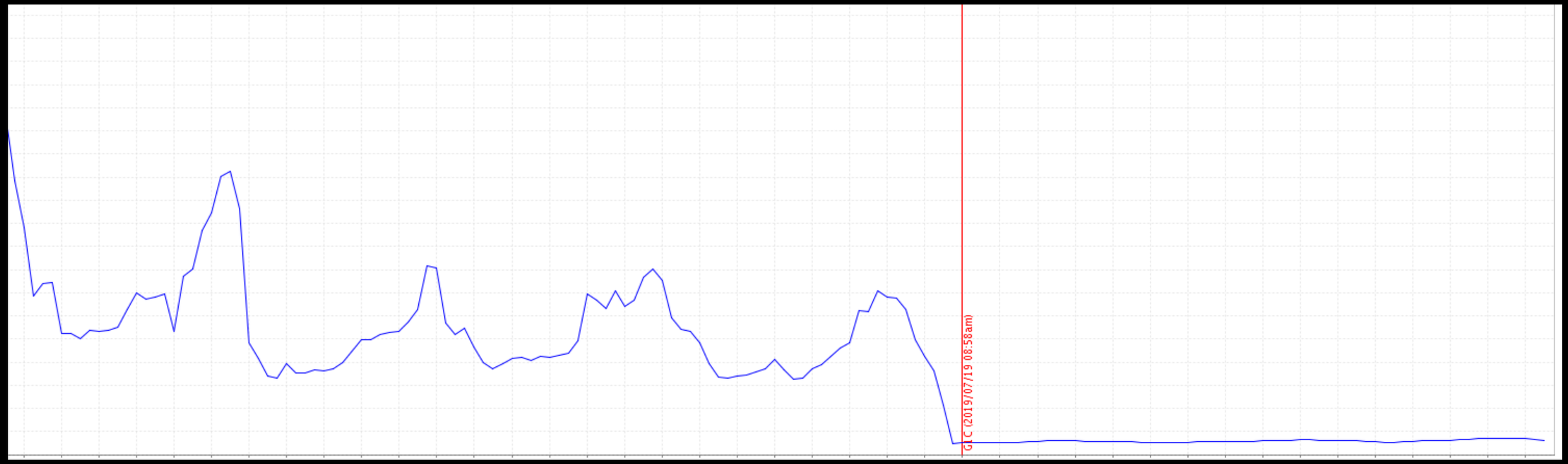
Insight #3: Excessive logging



Result: Garbage collector



Result: Latency



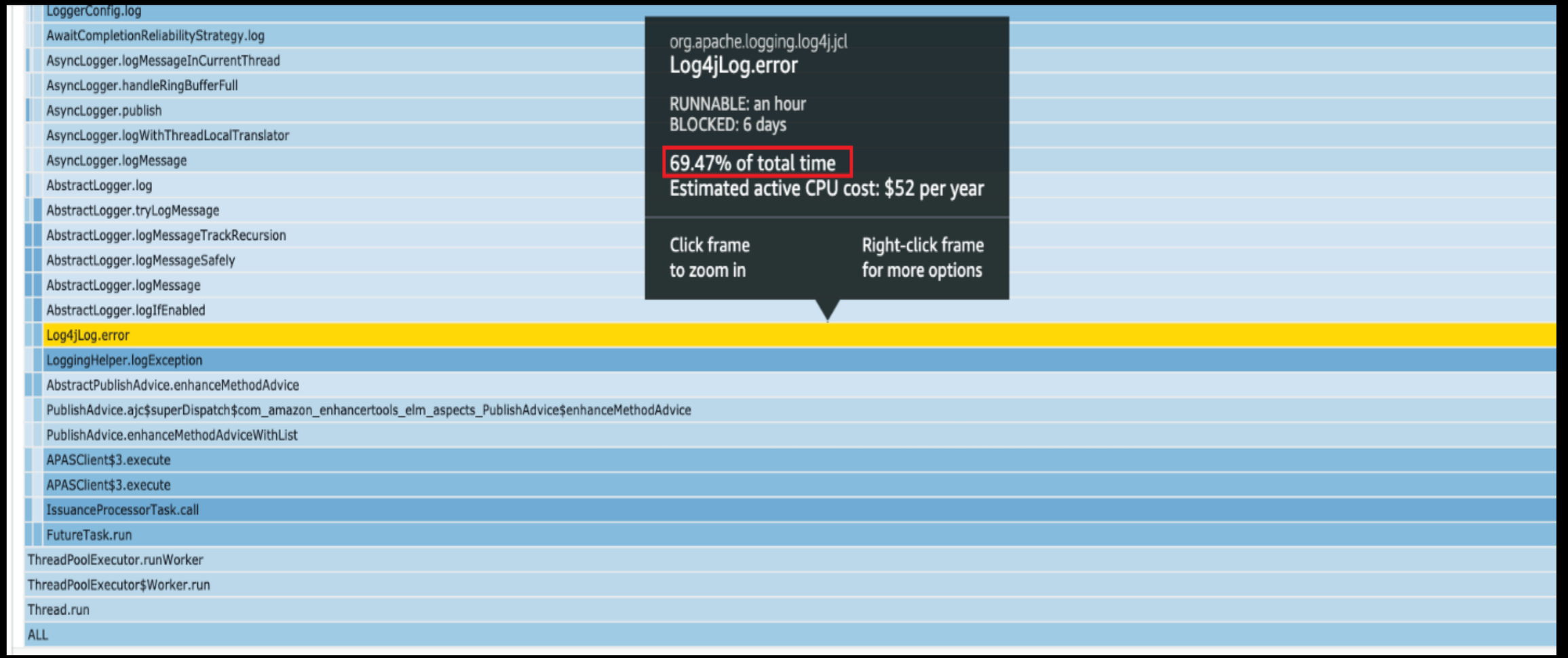
Result

- 25% lower infrastructure cost
- P99 latency reduced from 2500 msec to 250 msec
- Cheaper and faster

Issue

- High latency on checkout
- Immediate mitigation was to add hardware

Insight: Excessive exception logging



Result

- Reduced latency and scaled-back fleet
- Cheaper and faster

Takeaways



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why profile?

- Save infrastructure cost
- Reduce latency
- Debug and catch operation issues

Why profile in prod?

- Real workloads and environment
- Won’t impact your service

Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.