elastic

Java Flight Recorder

The Secret Arrow in Your Quiver

Daniel Mitterdorfer
@dmitterd

Agenda

1. Java Flight Recorder: Why and What?
2. How do I use Java Flight Recorder?
3. Demos
4. Summary

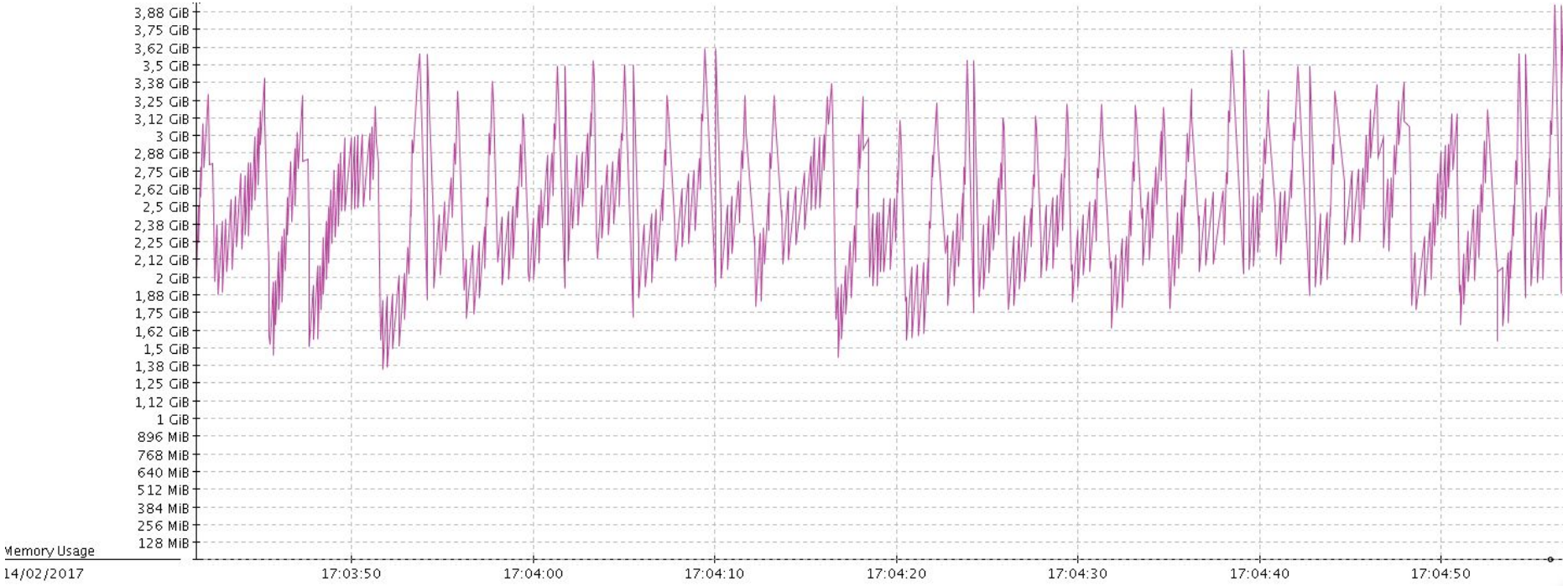
Why Java Flight Recorder?

Memory: Looks fine...

elastic

Why Java Flight Recorder?

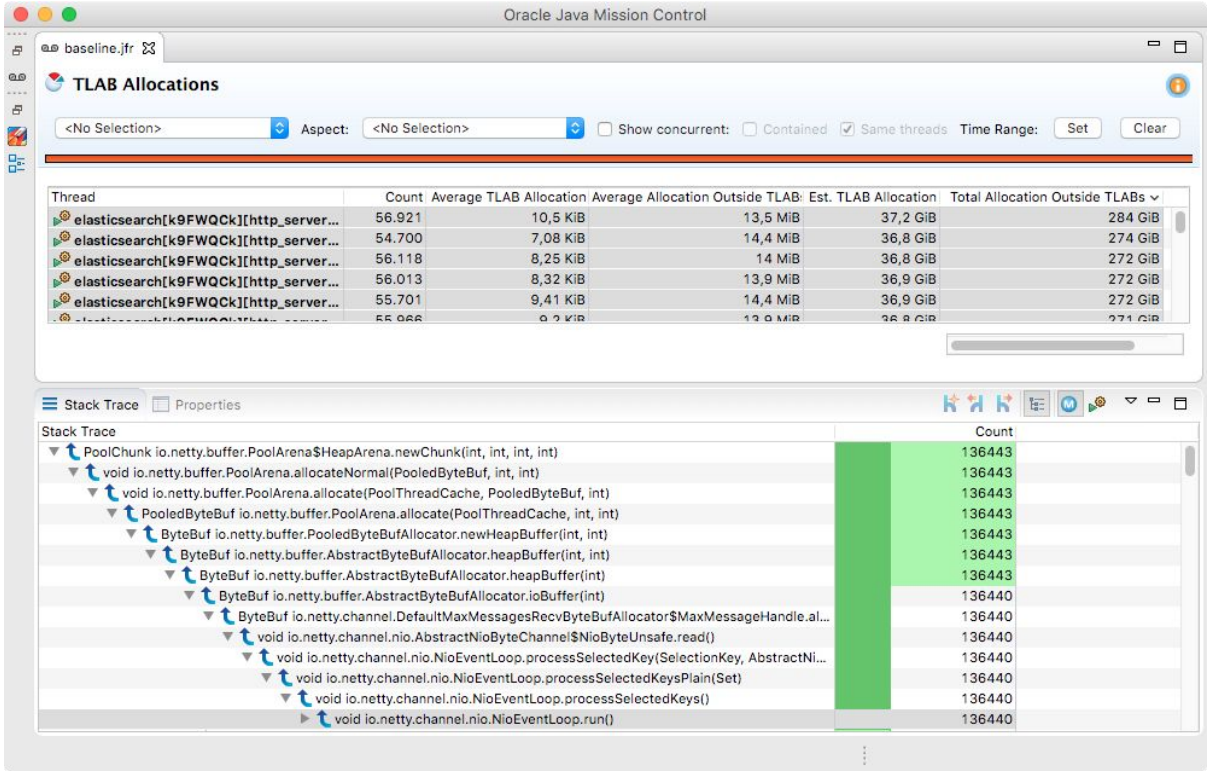
Memory: … but what's happening here?



elastic

Why Java Flight Recorder?

Memory: Why is it happening?



elastic

What is Java Flight Recorder?

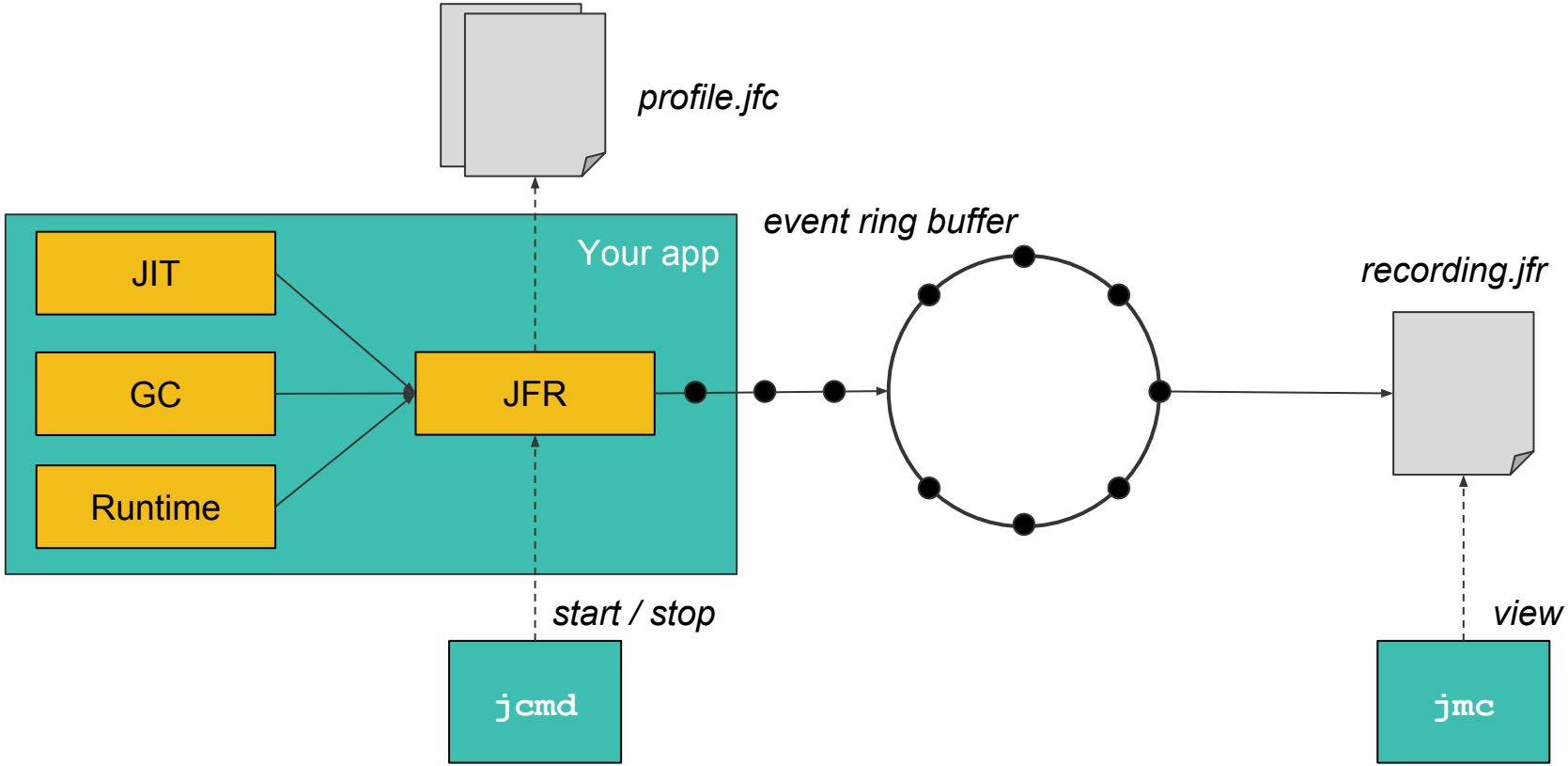
- Low-overhead sampling profiler
- Originally part of JRockit
- In **Oracle JDK** since Java 7u40


FLIGHT RECORDER
4GB FLASH/GPS
DO NOT OPEN


MISSION CONTROL CENTER
FCR-1 est. 1965
BETA= 16.1
ALT = 216.5
D/N: 0:17:40
GMT=32/18:29
LAT = 4.7
ALT = 196.7
LON= -52.8
SHK
ZKEP
DJS
ULD
USK
AQE
T27W
T27SW
SUN
SAA
-150
-120
-90
-60
-30
0
30
60
90
120
150

elastic

Java Flight Recorder from 10.000 feet



elastic

How do I use JFR?

Enabling Flight Recorder

Attach to a running application

- Start

```
jcmd $PID JFR.start
```

- Dump

```
jcmd $PID JFR.dump recording=1 filename=/tmp/rec.jfr
```

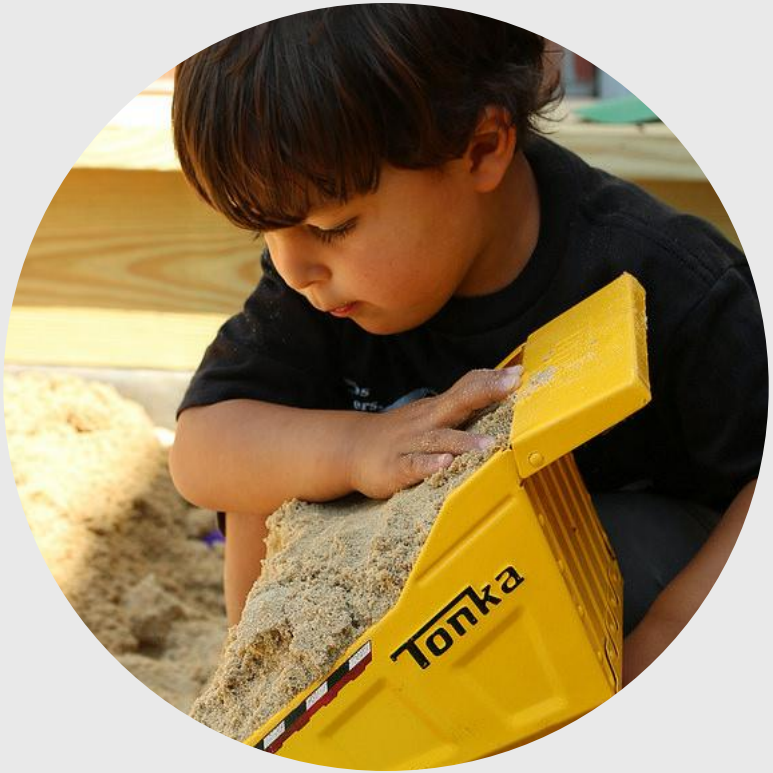
- Stop:

```
jcmd $PID JFR.stop recording=1
```

Enabling Flight Recorder

JVM flags (JDK 8 / 9)

```
-XX:+UnlockCommercialFeatures
-XX:+FlightRecorder
```



OR*

* May change in the future: See https://twitter.com/hirt/status/905500198297894912

Avoiding Safepoint Bias

JVM flags (JDK 8 / 9)

```
-XX:+UnlockCommercialFeatures
-XX:+FlightRecorder
-XX:+UnlockDiagnosticVMOptions
-XX:+DebugNonSafepoints
```

Safepoints?

Safepoints: Purpose

When the JVM needs to take control...

1. Halt all application threads (“safepoint”)
2. Do housekeeping tasks:
   - Garbage Collection
   - Take Thread Dumps
   - Deoptimize code
   - ….
3. Let application threads continue

Safepoint Polling*

- JVM “arms” a page in memory (on Linux: `mprotect`)
- Application threads poll and raise SEGV if page is armed

* specific to JIT-compiled code in OpenJDK: http://hg.openjdk.java.net/jdk9/jdk9/hotspot/file/tip/src/share/vm/runtime/safepoint.cpp

Safepoint Polling: Problems

Some operations defer safepoint polls

- Counted loops with `int` counters (considered “short”-running)
- Some I/O operations: e.g. writing mmapped buffers, `System.arrayCopy()`

JVMTI vs AsyncGetCallTrace

JVM Tool Interface (JVMTI)

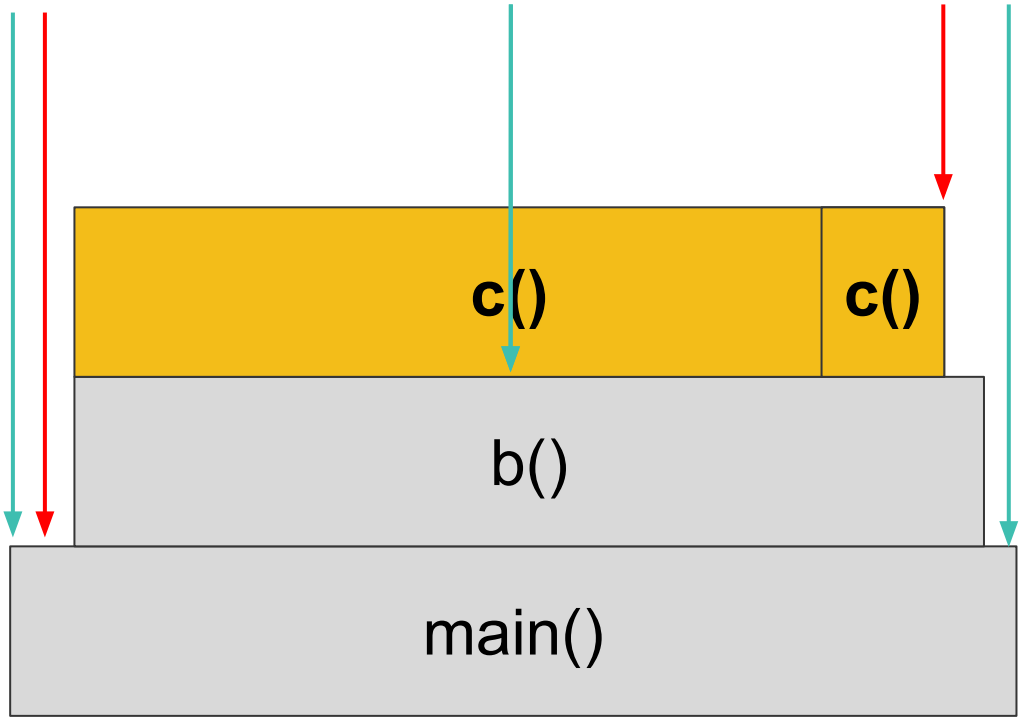
- “Official” interface for debuggers and profilers
- Captures stacks at safepoints
- Used by almost every profiler

JVMTI vs AsyncGetCallTrace

AsyncGetCallTrace

- Unofficial API
- Captures stacks at any time
- Used by Java Flight Recorder, Honest Profiler, Oracle Developer Studio

JVMTI: Safepoint Bias

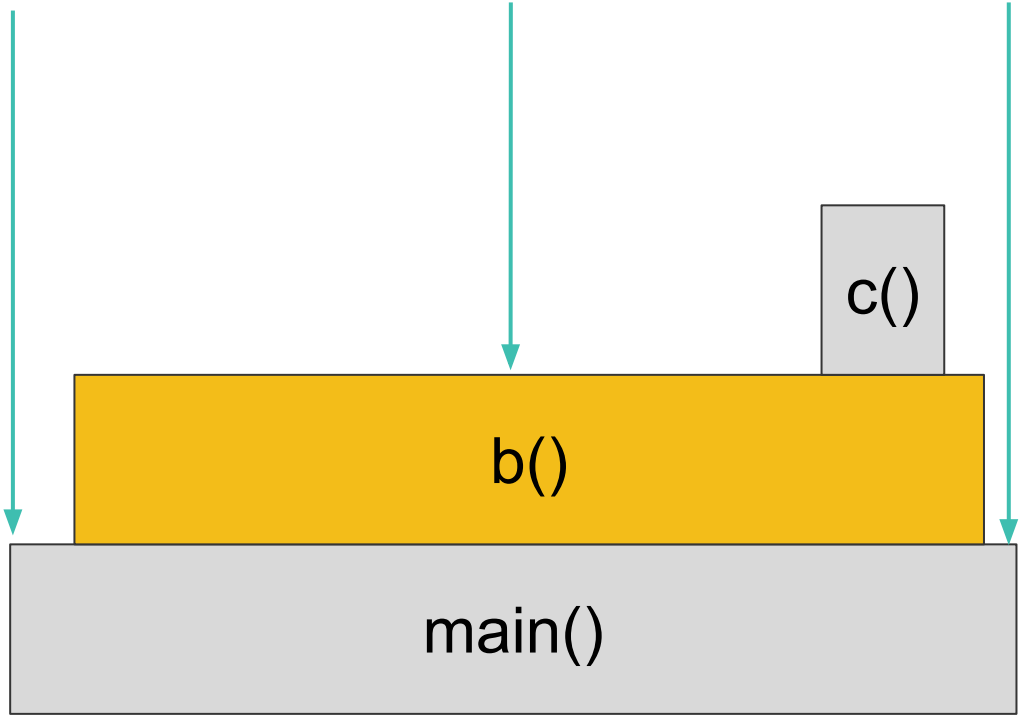


`JvmtiEnv::GetAllStackTraces()`

Safepoint Poll

elastic

AsyncGetCallTrace: No Safepoint Bias



AsyncGetCallTrace()

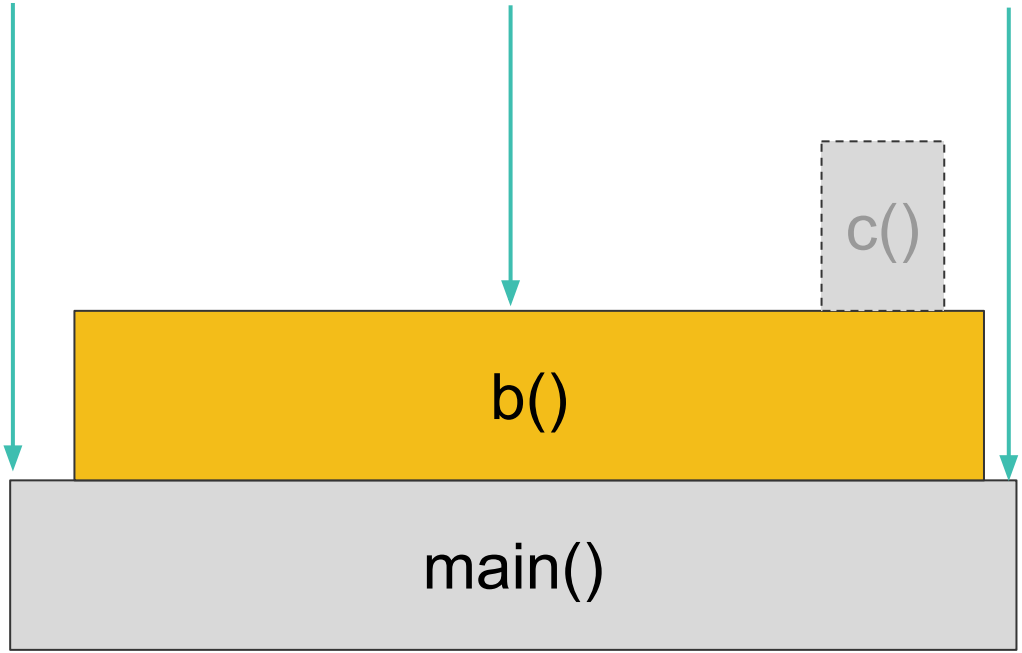
elastic

AsyncGetCallTrace is not a silver bullet

- Sampling profiler → sampling bias
- Sampling can fail:
  - GC active
  - Code is about to be deoptimized
  - Top stack frame is not a Java frame
  - ...

Sampling Bias

`c()` is “invisible”

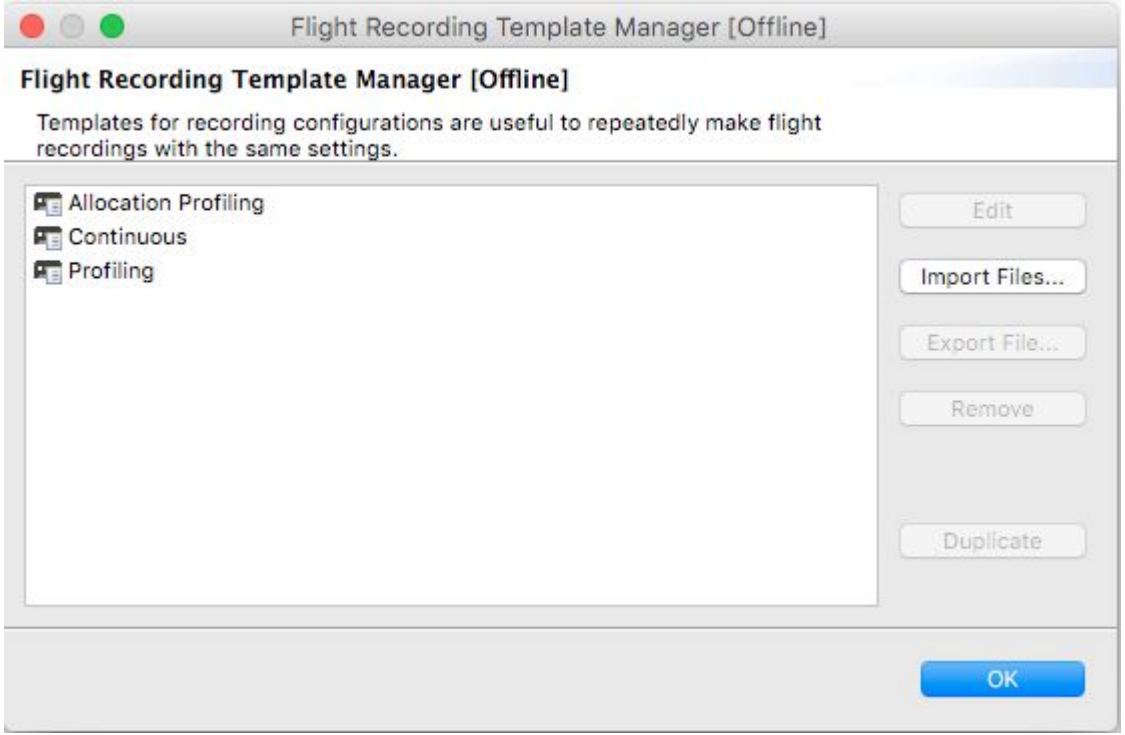


`AsyncGetCallTrace()`

elastic

Customized Recording Settings

Editing



elastic

Customized Recording Settings

Usage

- Stored as `*.jfc` in
  - `$JAVA_HOME/jre/lib/jfr/` (Java 8)
  - `$JAVA_HOME/lib/jfr/` (Java 9)
- To use `memory.jfc`:

```
-XX:StartFlightRecording=settings=memory
```

Starting a Profiling Recording (Fixed Time)

JVM flags (JDK 8 / 9)

```
-XX:+UnlockCommercialFeatures
-XX:+FlightRecorder
-XX:+UnlockDiagnosticVMOptions
-XX:+DebugNonSafepoints
-XX:StartFlightRecording=delay=240s,duration=60s,
settings=profile,filename=rec.jfr
```

Starting a Continuous Recording

JVM flags (JDK 8)

```
-XX:+UnlockCommercialFeatures
-XX:+FlightRecorder
-XX:+UnlockDiagnosticVMOptions
-XX:+DebugNonSafepoints
-XX:StartFlightRecording=defaultrecording=true
```

Not needed in JDK 9

Low Overhead < 2%

Source: https://www.oracle.com/technetwork/java/javaseproducts/mission-control/java-mission-control-wp-2008279.pdf

Dump a recording at application exit

JVM flags (JDK 8)

```
-XX:+UnlockCommercialFeatures
-XX:+FlightRecorder
-XX:+UnlockDiagnosticVMOptions
-XX:+DebugNonSafepoints
-XX:StartFlightRecording=defaultrecording=true
-XX:FlightRecorderOptions=disk=true,maxage=0s,maxsize=0,
dumponexit=true,dumponexitpath=/rec.jfr
```

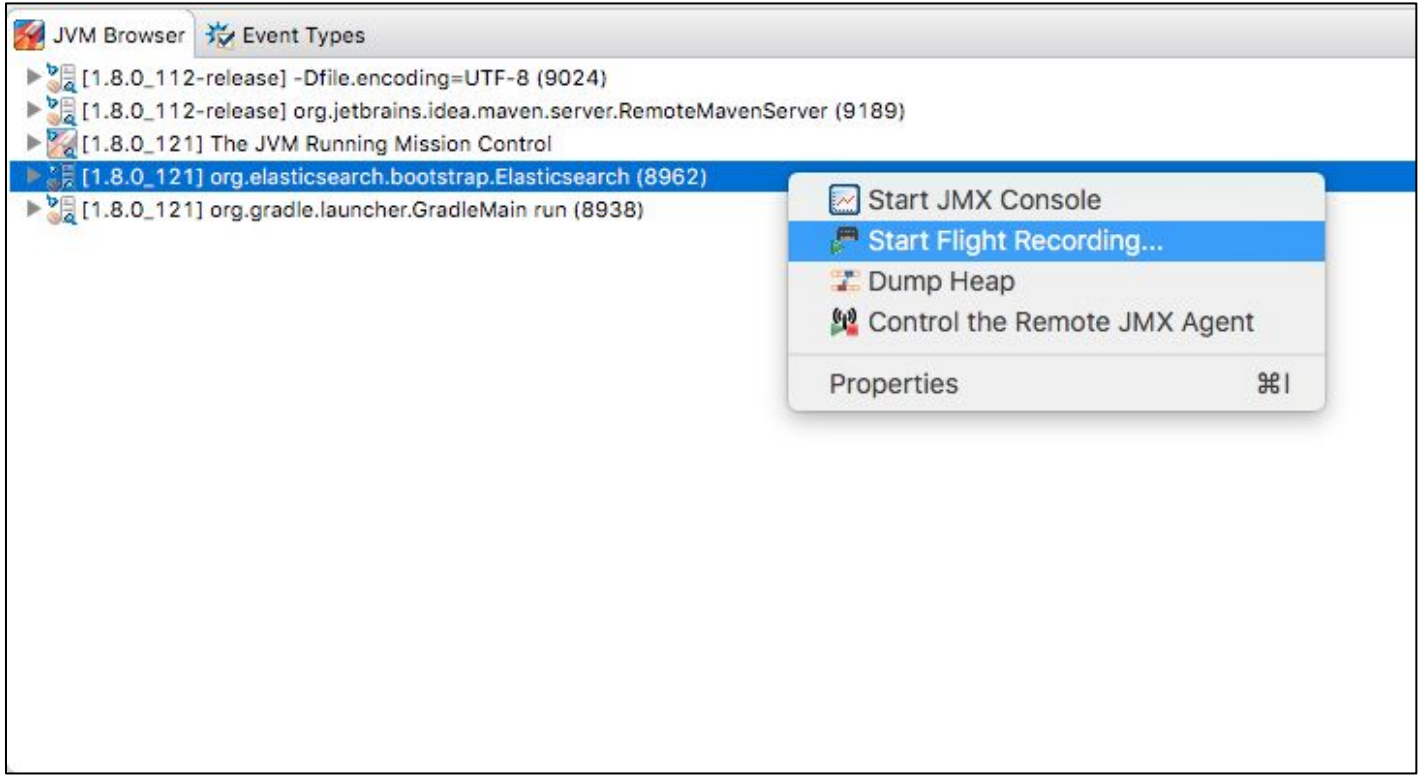
Dump a recording at application exit

JVM flags (JDK 9)

```
-XX:+UnlockCommercialFeatures
-XX:+FlightRecorder
-XX:+UnlockDiagnosticVMOptions
-XX:+DebugNonSafepoints
-XX:StartFlightRecording=disk=true,maxage=0s,maxsize=0,
    dumponexit=true,filename=/rec.jfr
```

Enabling Flight Recorder

From Mission Control

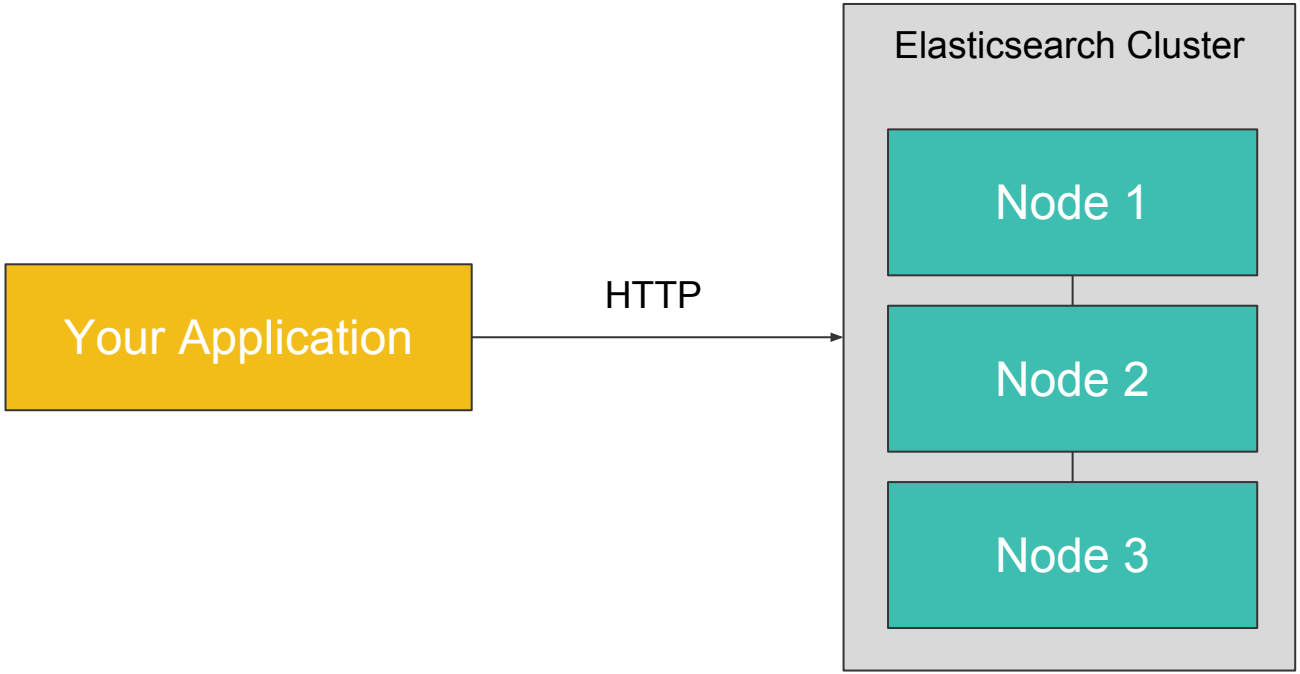


elastic

Enough talking: Demos

Example Application: Elasticsearch

Open Source Distributed Search Engine



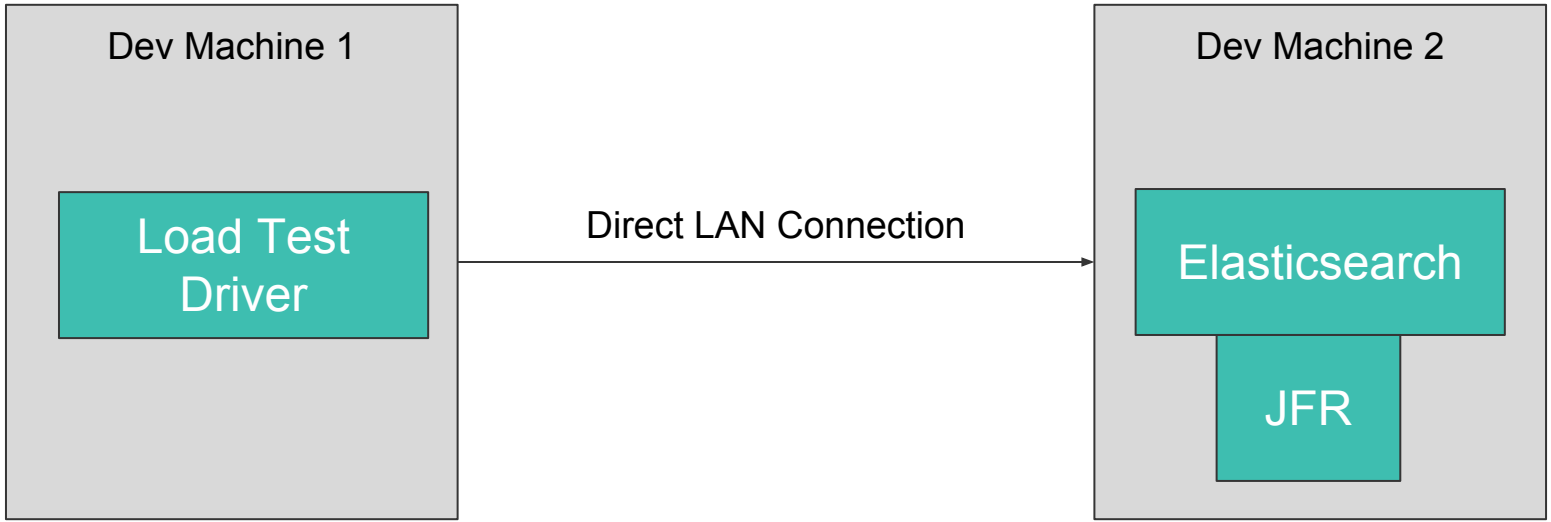
elastic

Example Application: Elasticsearch

Key Operations and Metrics

- **Adding Documents to the Index:** Throughput
- **Querying:** Latency

My Usual Setup

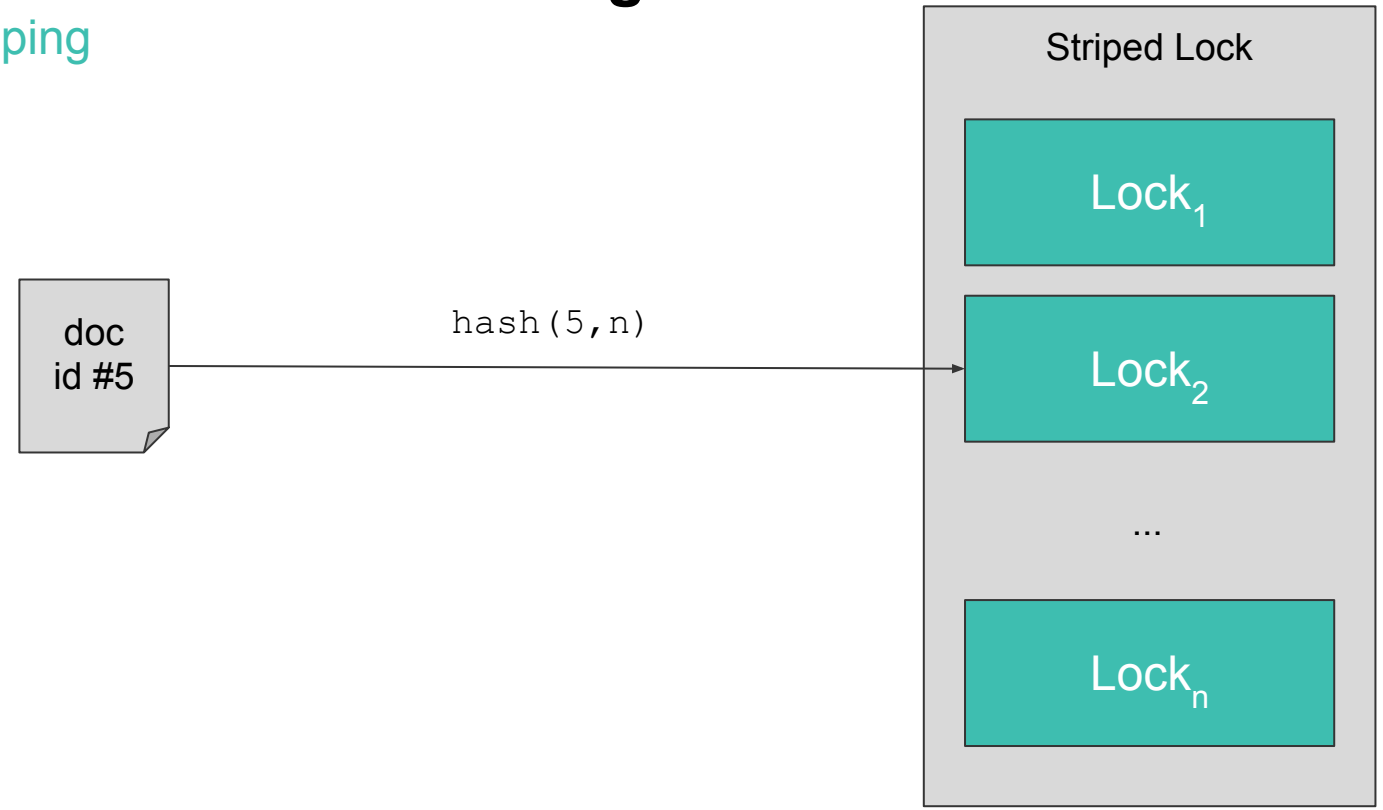


Demo: Contention Profiling

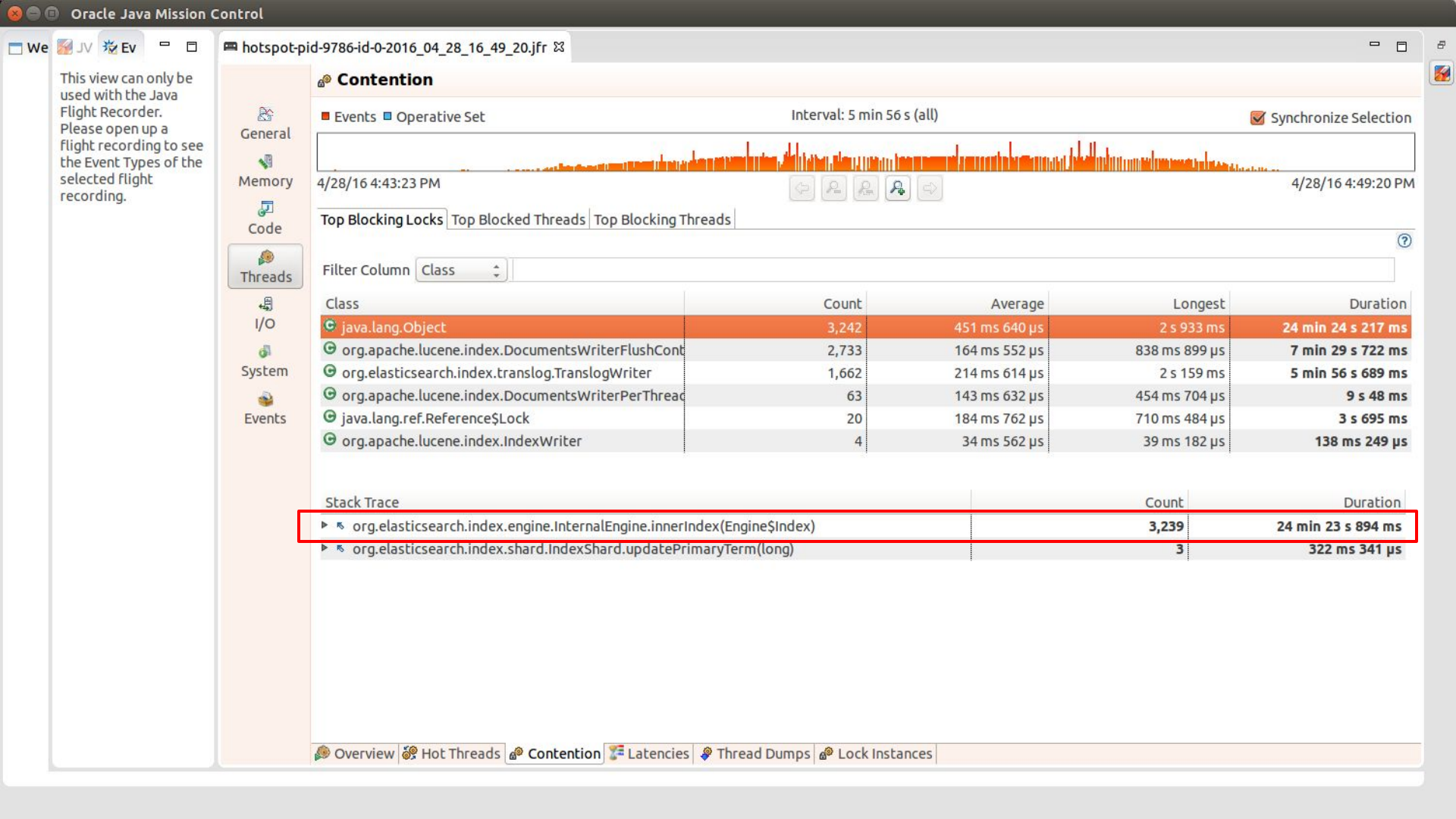
elastic/elasticsearch#18053

Demo: Contention Profiling

Lock Striping

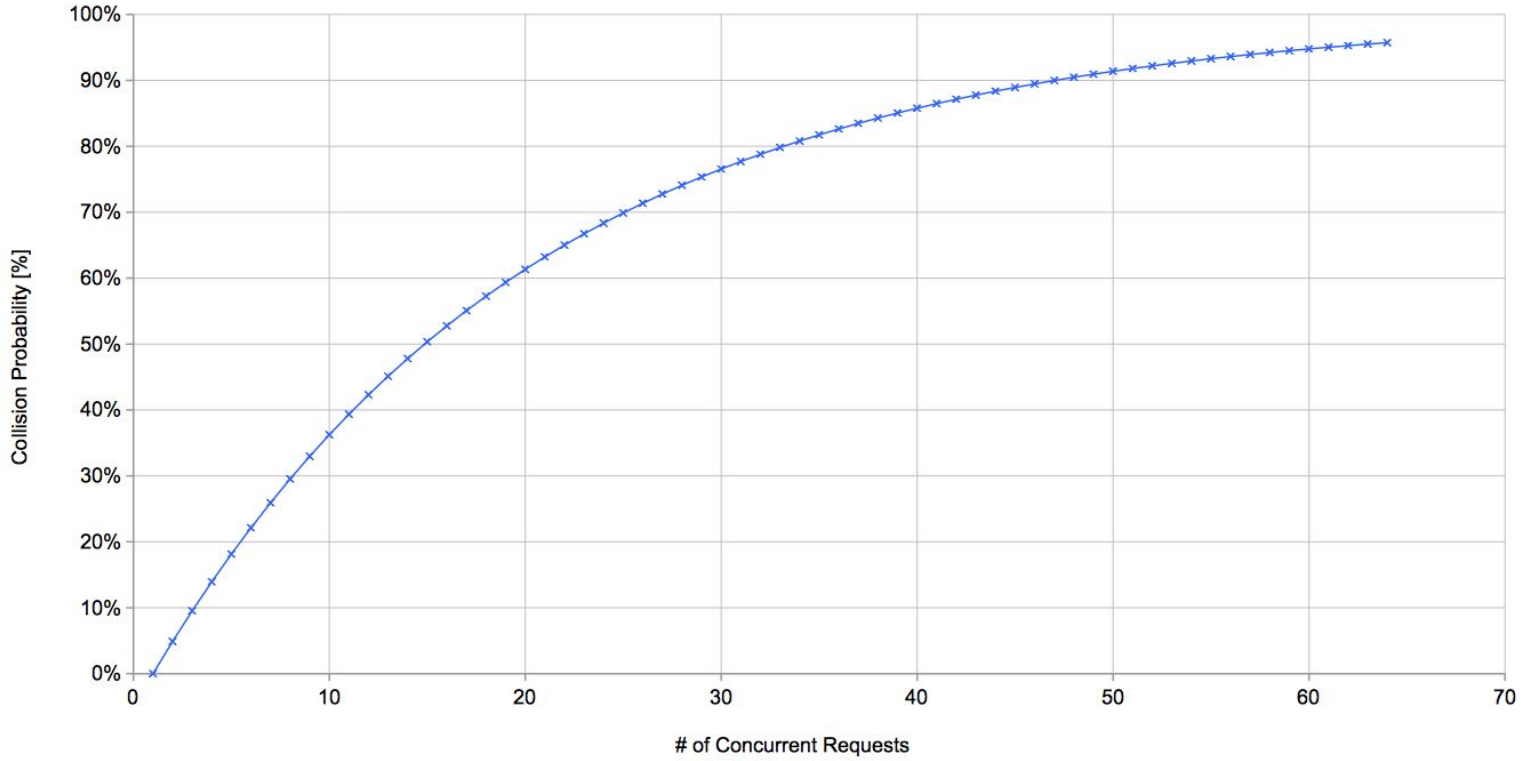


elastic


Oracle Java Mission Control
We
JV
Ev
This view can only be used with the Java Flight Recorder. Please open up a flight recording to see the Event Types of the selected flight recording.
hotspot-pid-9786-id-0-2016_04_28_16_49_20.jfr
Contention
General
Memory
Code
Threads
I/O
System
Events
Events
Operative Set
Interval: 5 min 56 s (all)
Synchronize Selection
4/28/16 4:43:23 PM
4/28/16 4:49:20 PM
Top Blocking Locks
Top Blocked Threads
Top Blocking Threads
Filter Column
Class
Class
Count
Average
Longest
Duration
java.lang.Object
3,242
451 ms 640 µs
2 s 933 ms
24 min 24 s 217 ms
org.apache.lucene.index.DocumentsWriterFlushCont
2,733
164 ms 552 µs
838 ms 899 µs
7 min 29 s 722 ms
org.elasticsearch.index.translog.TranslogWriter
1,662
214 ms 614 µs
2 s 159 ms
5 min 56 s 689 ms
org.apache.lucene.index.DocumentsWriterPerThread
63
143 ms 632 µs
454 ms 704 µs
9 s 48 ms
java.lang.ref.Reference$Lock
20
184 ms 762 µs
710 ms 484 µs
3 s 695 ms
org.apache.lucene.index.IndexWriter
4
34 ms 562 µs
39 ms 182 µs
138 ms 249 µs
Stack Trace
Count
Duration
org.elasticsearch.index.engine.InternalEngine.innerIndex(Engine$Index)
3,239
24 min 23 s 894 ms
org.elasticsearch.index.shard.IndexShard.updatePrimaryTerm(long)
3
322 ms 341 µs
Overview
Hot Threads
Contention
Latencies
Thread Dumps
Lock Instances

Demo: Contention Profiling

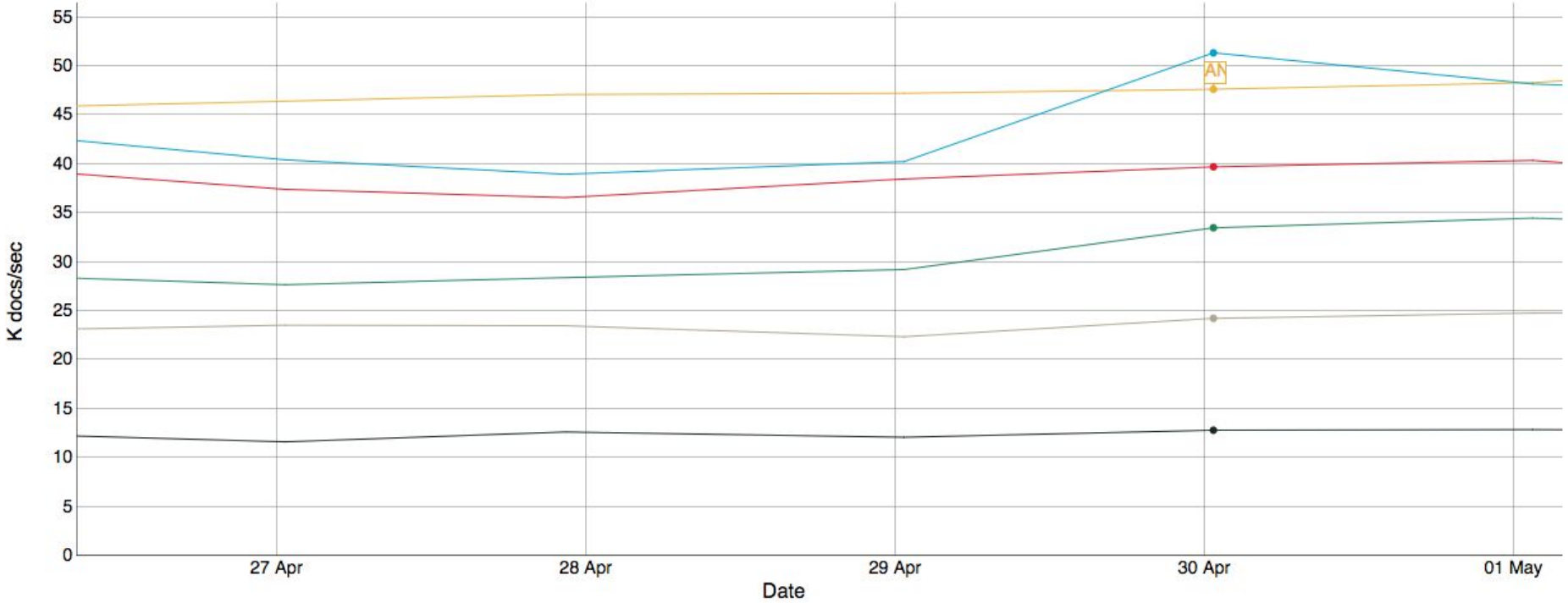
Collision Probability



elastic

Demo: Contention Profiling

Throughput Comparison



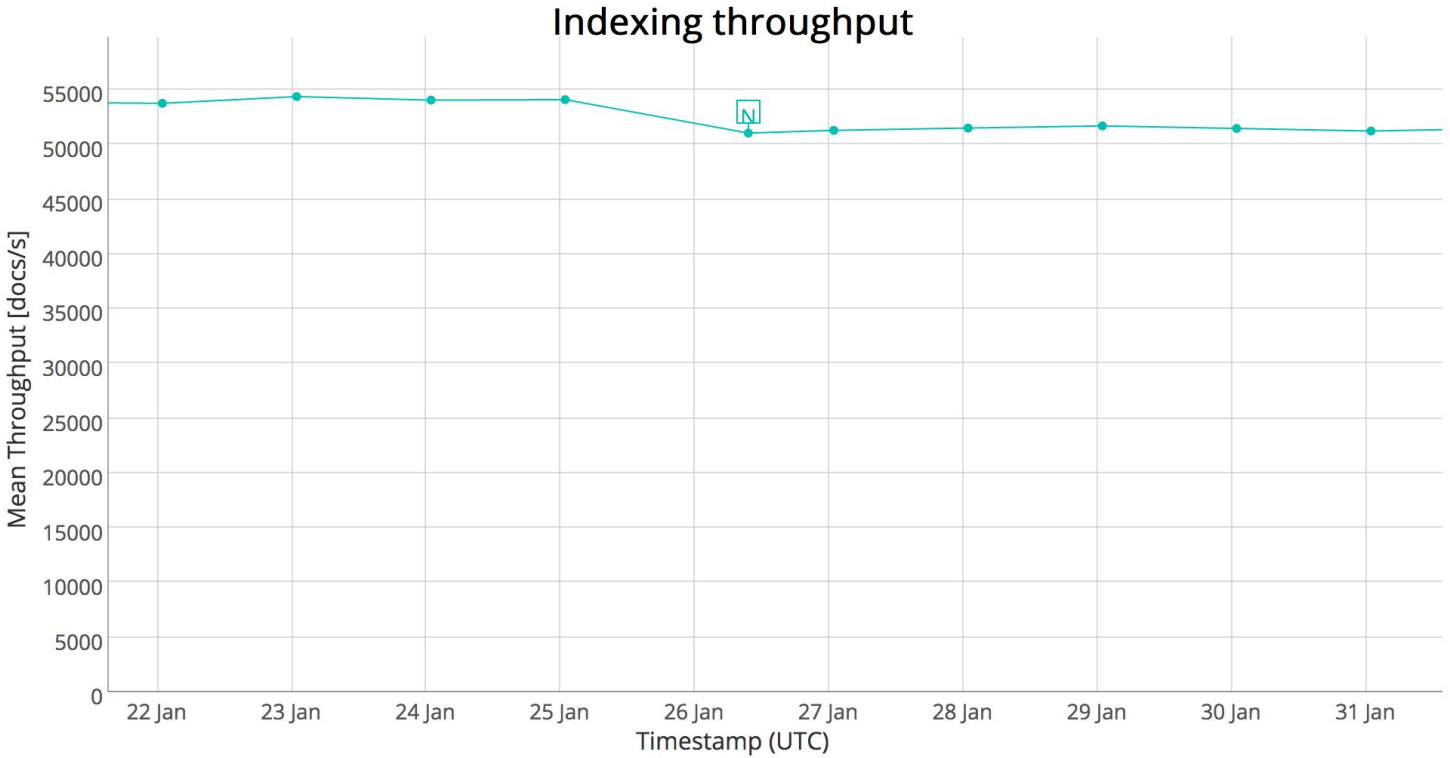
Source: https://elasticsearch-benchmarks.elastic.co/classic/

Demo: Allocation Profiling

elastic/elasticsearch#23185

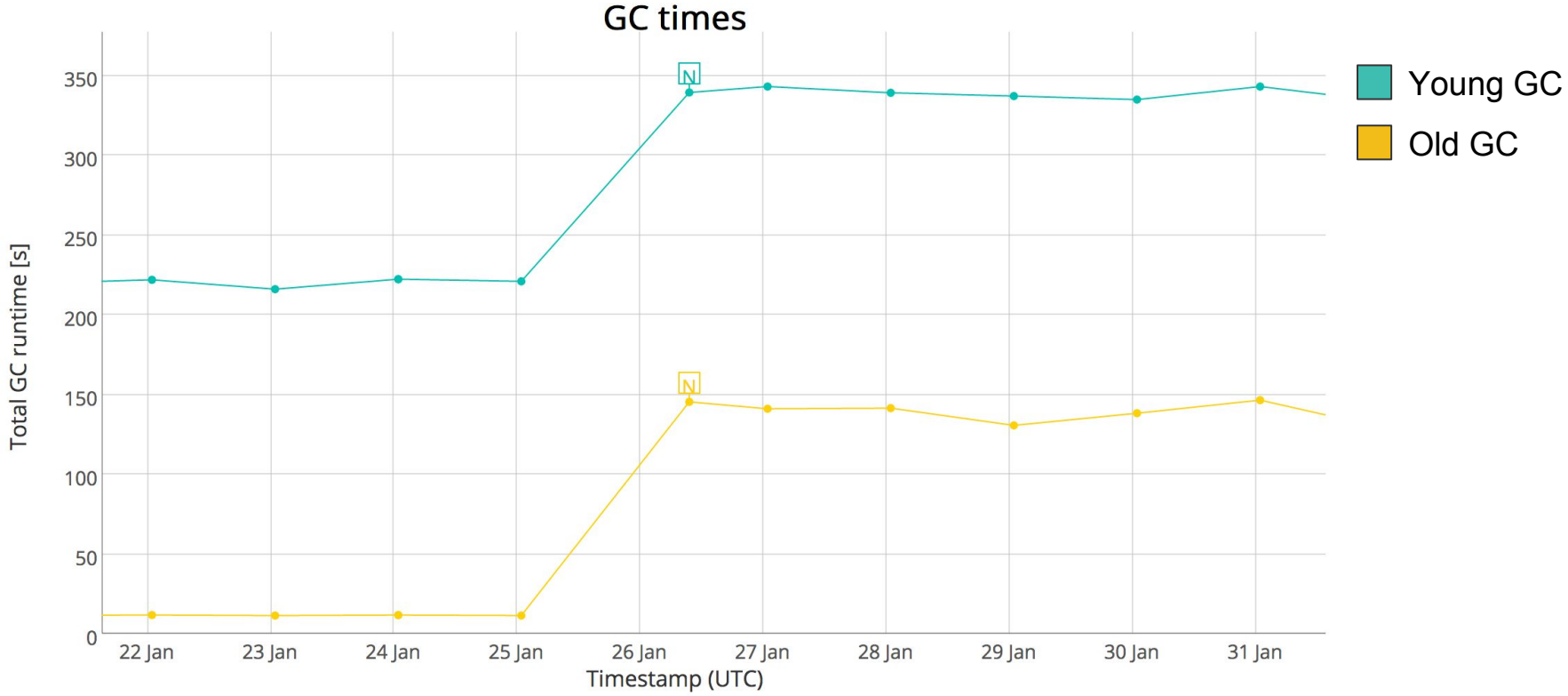
Demo: Allocation Profiling

Background:elastic/elasticsearch#23185



Source: https://elasticsearch-benchmarks.elastic.co/index.html#tracks/nyc-taxis/nightly/1y

Demo: Allocation Profiling

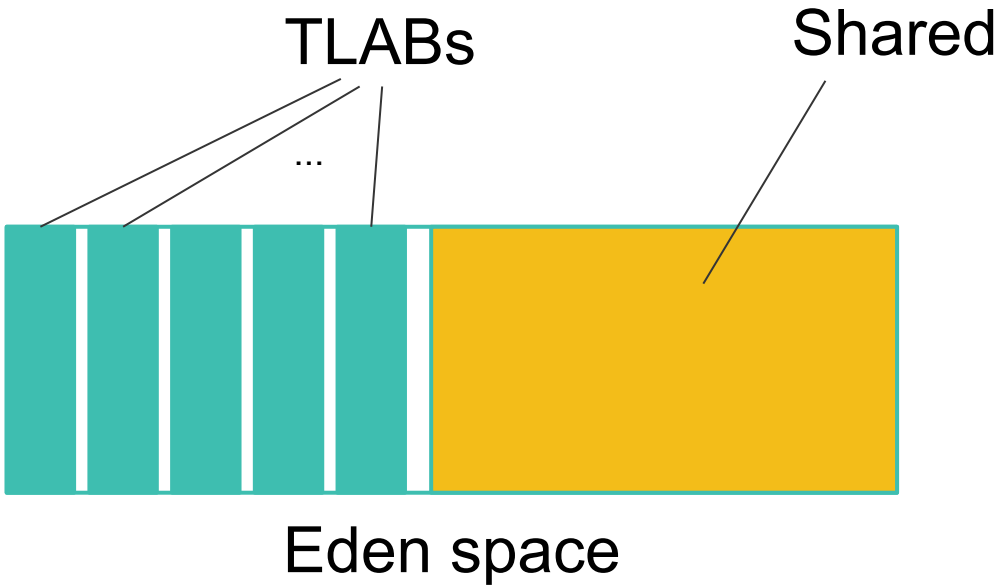


Source: https://elasticsearch-benchmarks.elastic.co/index.html#tracks/nyc-taxis/nightly/1y

elastic

Demo: Allocation Profiling

Thread Local Allocation Buffers (TLABs) in the JVM

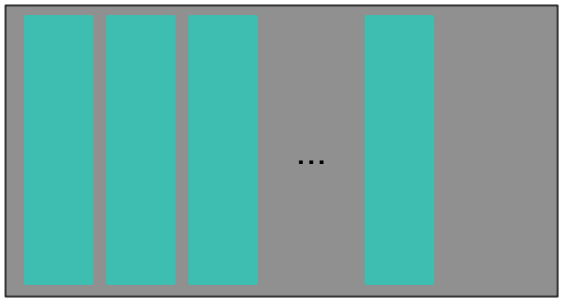


elastic

Demo: Allocation Profiling

Buffers in Netty

Many Buffers - One per read



One HTTP Request

Demo: Allocation Profiling

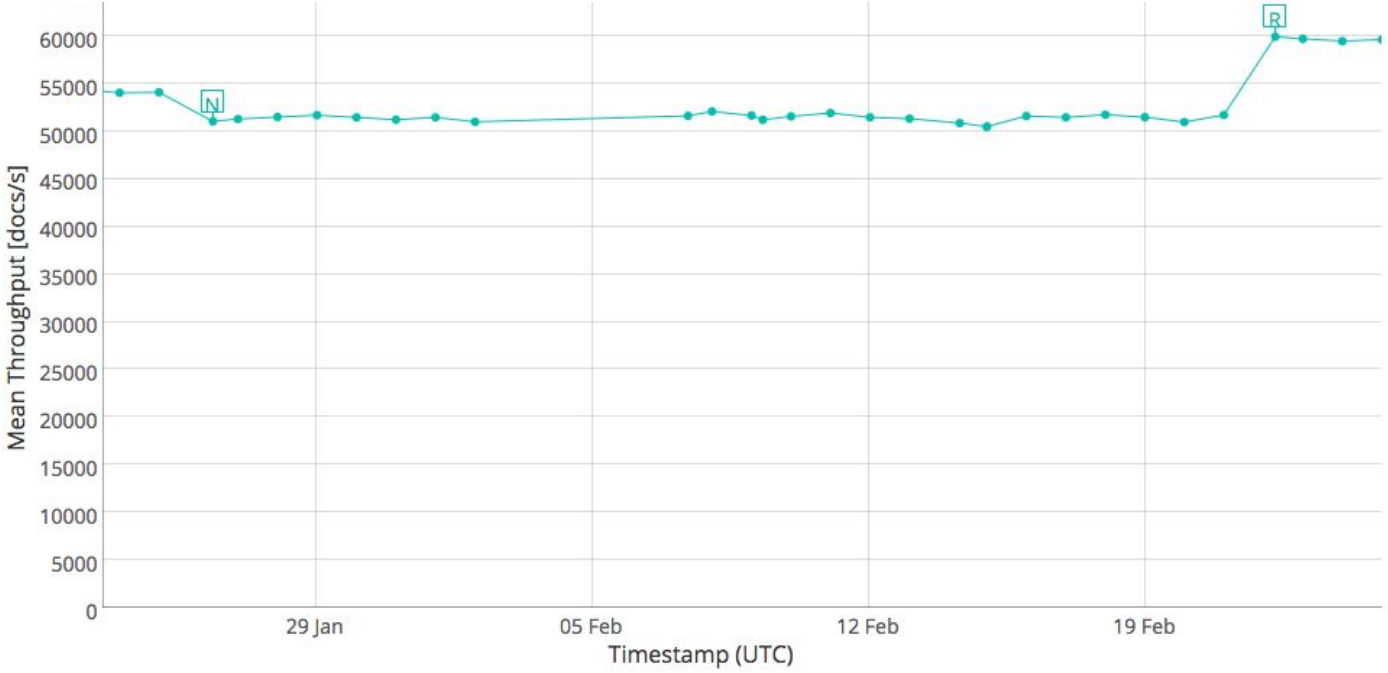
Problem: MTU

- **MTU Loopback:** 65536 bytes
- **MTU Ethernet:** 1500 bytes

→ Lower internal buffer sizes

Demo: Allocation Profiling

Throughput Comparison

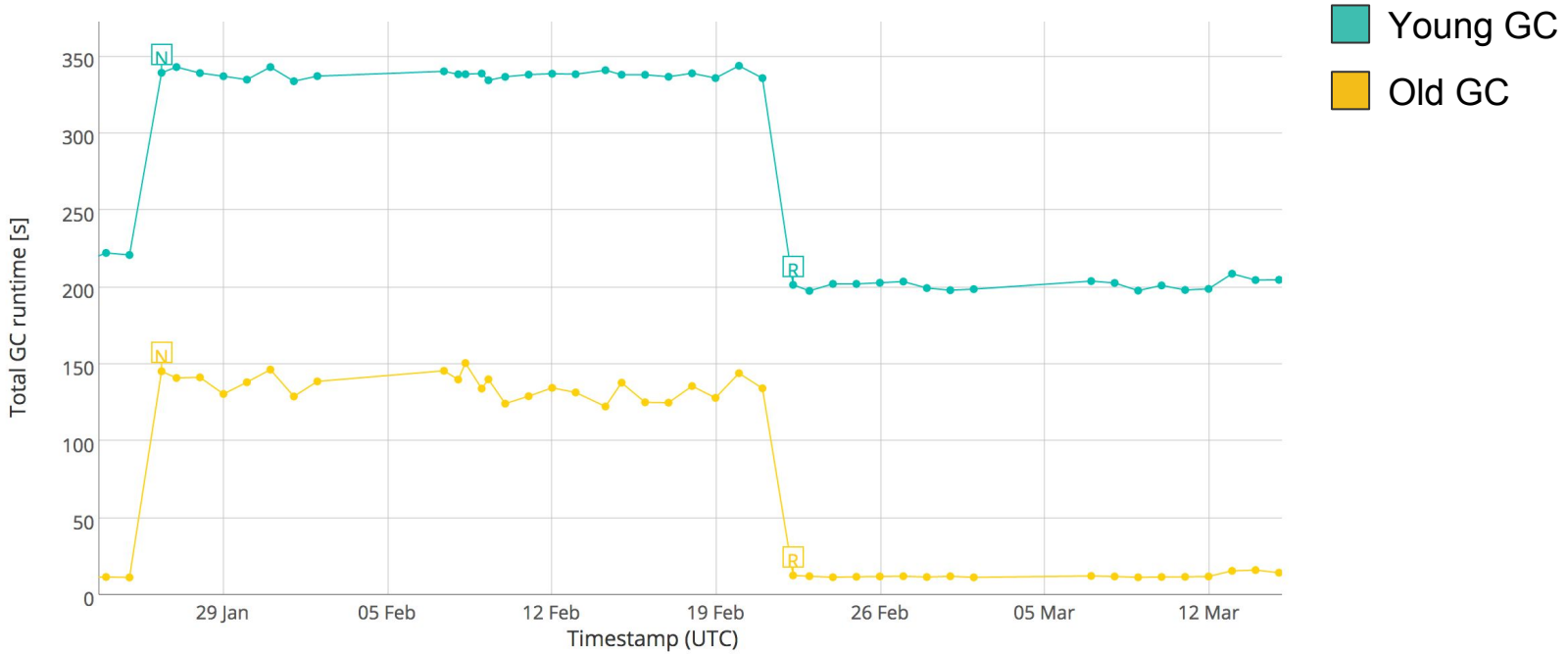


Source: https://elasticsearch-benchmarks.elastic.co/index.html#tracks/nyc-taxis/nightly/1y

elastic

Demo: Allocation Profiling

GC Times Comparison



Source: https://elasticsearch-benchmarks.elastic.co/index.html#tracks/nyc-taxis/nightly/1y

elastic

Summary & Outlook

JFR: A versatile tool

- Commercial extension of Oracle JDK
- Low overhead, customizable profiles
- Profiling and Continuous Mode
- Analyze various performance issues: CPU, Memory, Locking, ...

JFR and JDK 9

- JDK 9 brings a new version of Mission Control
- Flight recording also produced on crashes (e.g. OOME)
- Adds supported APIs
  - Create your own events
  - Parse flight recordings
- More events available

Source: https://www.jug.ch/events/slides/170215_SwissJUG2017_zurich_sneakpeek.pdf

References and Further Reading

- AsyncGetCallTrace: http://psy-lob-saw.blogspot.de/2016/06/the-pros-and-cons-of-agct.html
- Safepoints and Safepoint Bias
  - http://psy-lob-saw.blogspot.de/2015/12/safepoints.html
  - http://psy-lob-saw.blogspot.de/2016/02/why-most-sampling-java-profilers-are.html
- Flight Recorder Docs: https://docs.oracle.com/javacomponents/jmc-5-5/jfr-runtime-guide/
- Generally recommended: http://hirt.se/

Slides

https://daniel.mitterdorfer.name/talks/

Image Credits

- So klein und schon ein flight recorder by Uwe Hauck (used with written permission of the author)
- Nasa Social February 2017 by Nasa Johnson (license: CC BY-NC-ND 2.0)
- Tyler Greene Dodges Eric Young by Paul Martinez (license: CC BY-NC-ND 2.0)
- IMG_2857 by John M (license: CC BY-NC-ND 2.0)
- Greed by las - initially (license: CC BY-NC-ND 2.0)
- IMG_9861 by 7263255 (license: CC BY-SA 2.0)
- 2014 Bahrain GP - FP3 & Qualifying by CaterhamF1 (license: CC BY-NC-ND 2.0)