GRAAL @TWITTER

EXPERIENCES IN A PRODUCTION ENVIRONMENT

@CHRISTHALINGER

#TWITTERVMTEAM

@CHRISTHALINGER

TWITTER

- Huge distributed system
- Comprised of many services
  - tweet service
  - user service
  - timeline service
  - social graph service
  - ...

SERVICES

- Many JVMs per service
- 1,000s of machines running 1,000s of JVMs
- Multiple datacenters

OPEN-SOURCE

https://engineering.twitter.com/opensource

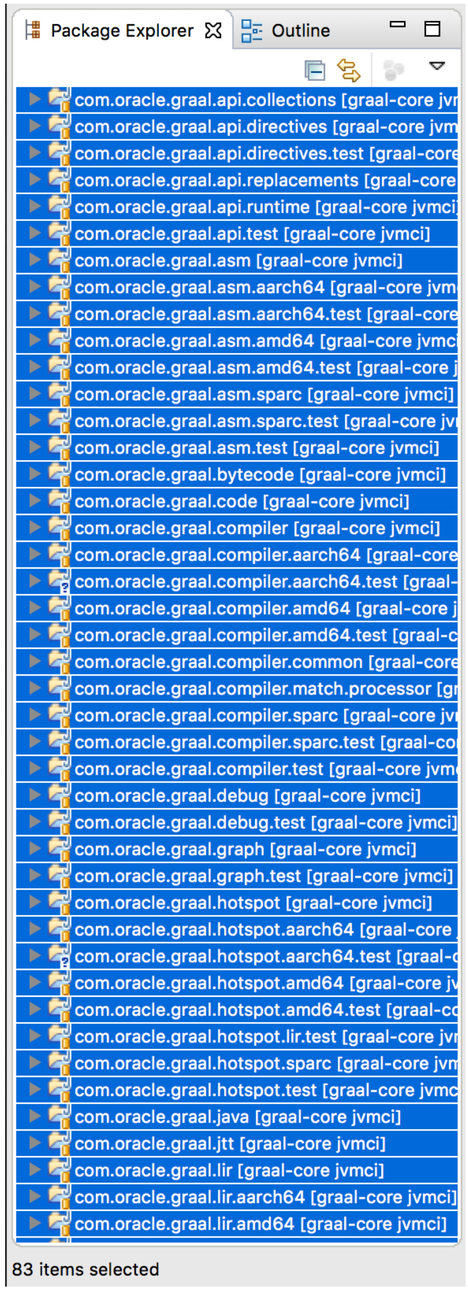
- Twitter is using many open-source projects
- ...and also gives back to the community:
  - https://engineering.twitter.com/opensource/projects
- Graal is a good fit
  - We are planning to contribute, too

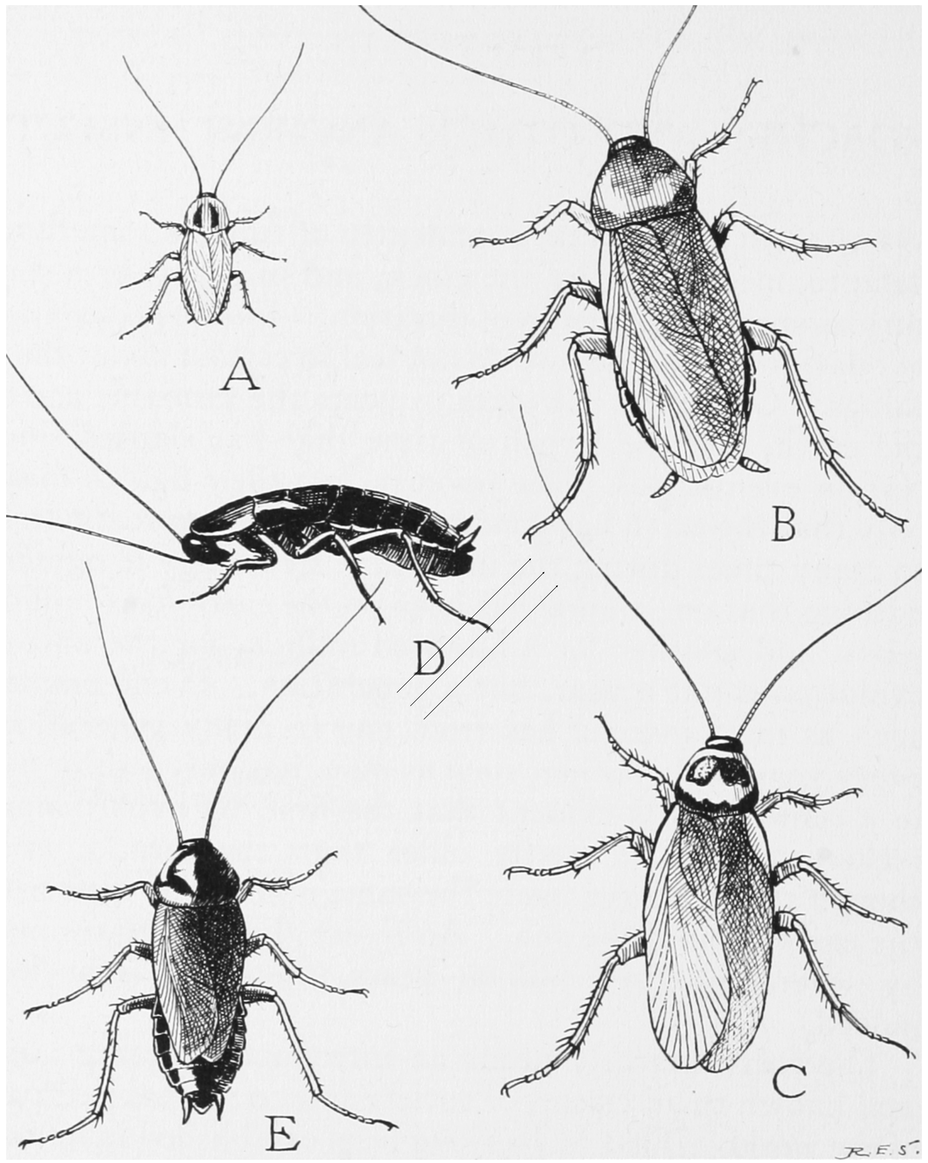
TWITTER'S JDK

- Based on OpenJDK 8u
- Will be open-sourced (soon!)
- Has:
  - JEP 243 backport (JVMCI)
  - Contrail (JFR replacement)
  - CMS improvements

WHY GRAAL?

- C2 is old and very complex
  - Learning curve way too steep
  - The last few years no new major optimizations
  - Reached its end-of-life already a while ago
- Graal's learning curve is much shorter
  - Advantages of Java
  - Modularized





https://en.wikipedia.org/wiki/Cockroach

BUG 1: #128

Graal should support same OSR compilations as C1 and C2

- https://github.com/graalvm/graal-core/issues/128
- `PrintCompilation` shows something like this:
  - `COMPILE SKIPPED: OSR with locks not supported`
- The exception looks like...

BUG 1: #128

Graal should support same OSR compilations as C1 and C2

```
jdk.vm.ci.code.BailoutException: OSR with locks not supported
        at com.oracle.graal.hotspot.phases.OnStackReplacementPhase.getEntryMarker(OnStackReplacementPhase.java:139)
        at com.oracle.graal.hotspot.phases.OnStackReplacementPhase.run(OnStackReplacementPhase.java:69)
        at com.oracle.graal.phases.Phase.run(Phase.java:51)
        at com.oracle.graal.phases.BasePhase.apply(BasePhase.java:166)
        at com.oracle.graal.phases.BasePhase.apply(BasePhase.java:148)
        at com.oracle.graal.phases.PhaseSuite.run(PhaseSuite.java:154)
        at com.oracle.graal.phases.BasePhase.apply(BasePhase.java:166)
        at com.oracle.graal.phases.BasePhase.apply(BasePhase.java:148)
        at com.oracle.graal.compiler.GraalCompiler.emitFrontEnd(GraalCompiler.java:193)
        at com.oracle.graal.compiler.GraalCompiler.compile(GraalCompiler.java:175)
        at com.oracle.graal.compiler.GraalCompiler.compileGraph(GraalCompiler.java:162)
        at com.oracle.graal.hotspot.HotSpotGraalCompiler.compile(HotSpotGraalCompiler.java:179)
        at com.oracle.graal.hotspot.CompilationTask.runCompilation(CompilationTask.java:212)
        at com.oracle.graal.hotspot.HotSpotGraalCompiler.compileMethod(HotSpotGraalCompiler.java:128)
        at jdk.vm.ci.hotspot.HotSpotJVMCIRuntime.compileMethod(HotSpotJVMCIRuntime.java:409)
```

BUG 1: #128

Graal should support same OSR compilations as C1 and C2

- Tom and I discussed this for a bit...
- ...and we decided that it's not a big problem right now.
- Tiered compilation takes care of it
  - After a while the failing method will be flagged and compiled with C1

#128 Open

dougxc commented on Sep 15, 2016 Member

@christhalinger , you interested in having a go at this?

christhalinger commented on Sep 15, 2016 Member

Interested? Yes. Do I have time right now? No.

christhalinger commented on Mar 14, 2016 • edited

Member

While running some obfuscated code I see a lot of these:

```
250398 21839 %s!   4       ge::al @ 1586 (2018 bytes)   COMPILE SKIPPED: OSR with locks n
```

christhalinger commented on Mar 14, 2016 • edited

Member

I'm seeing similar bailouts with jorbis:

```
31342 22854 %s    4       com.jcraft.jorbis.Mdct::backward @ 103 (468 bytes)   COMPILE SI
31529 22856 %s    4       com.jcraft.jorbis.Mapping0::inverse @ 804 (867 bytes)   COMPILI
```

christhalinger commented on Mar 14, 2016 • edited

Member

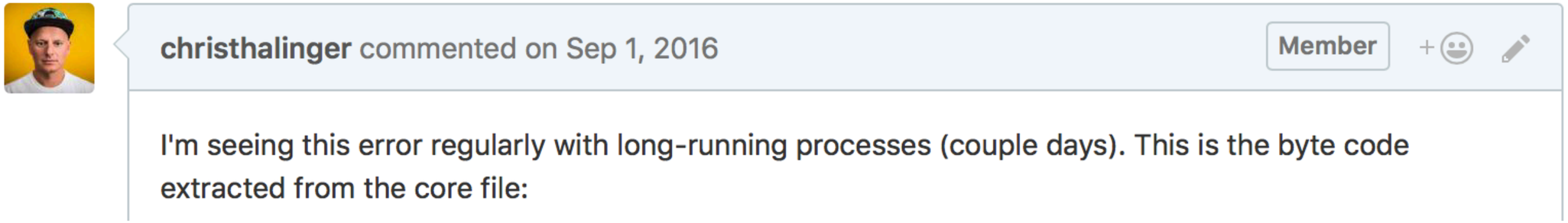
This is an interesting and possibly important one:

```
4045 13616 % !   4       java.util.concurrent.ConcurrentHashMap::transfer @ 376 (832 by
```

BUG 2: #204

fatal error: no original bytecode found in scala.collection.TraversableLike$class.builder$1(Lscala/collection/TraversableLike;Lscala/collection/generic/CanBuildFrom;)Lscala/collection/mutable/Bu
at bci 24

- https://github.com/graalvm/graal-core/issues/204



BUG 2: #204

fatal error: no original bytecode found in scala.collection.TraversableLike$class.builder$1(Lscala/collection/TraversableLike;Lscala/collection/generic/CanBuildFrom;)Lscala/collection/mutable/Bu
at bci 24

christhalinger commented on Sep 9, 2016 Member

Turns out that the https://github.com/mariusae/heapster agent is causing the problem. I could reproduce it locally with:

BUG 2: #204

fatal error: no original bytecode found in scala.collection.TraversableLike$class.builder$1(Lscala/collection/TraversableLike;Lscala/collection/generic/CanBuildFrom;)Lscala/collection/mutable/Bu
at bci 24

- https://github.com/mariusae/heapster

Heapster

Heapster provides an agent library to do heap profiling for JVM processes with output compatible with Google perftools.
The goal of Heapster is to be able to do meaningful (sampled) heap profiling in a production setting.

dougxc commented on Sep 12, 2016 • edited

Member

As I suspected, the problem is that we don't isolate snippet bytecode from instrumentation agents. In this case, it's the bytecode of `Double.valueOf` inlined into `com.oracle.graal.replacements.BoxingSnippets.doubleValueOf(double)`. Time to think of a solution.

tkrodriguez commented on Sep 12, 2016

Member

This isn't even code that we own so we don't have a good way around this. I think the assert in ReplacementsImpl.notifyNotInlined should be raised to a hard error since it was masking this issue. We might also need more checks in snippet handling in general so that this kind of misassociation of bci and method doesn't happen in other places.

dougxc commented on Sep 12, 2016

Member

I think it's time to bite the bullet and parse the bytecodes for snippets (and their callees) directly. I'm looking into the best way to do this.

#204 Closed

Showing **65 changed files** with **2,767 additions** and **446 deletions.**

BUG 3: #206

Netty 4.1.5 tests fail

- https://github.com/graalvm/graal-core/issues/206
- For the longest time we were seeing connection exceptions in the logs
- We didn't know where they were coming from
- Until I, independently, decided to run the Netty 4 test suite…

BUG 3: #206

Netty 4.1.5 tests fail

christhalinger commented on Sep 9, 2016 Member

Today we narrowed down where these odd connection exceptions we were seeing are coming from:

```
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] Netty ..................................... SUCCESS [  5.584 s]
[INFO] Netty/Common .............................. SUCCESS [01:15 min]
[INFO] Netty/Buffer .............................. FAILURE [ 48.142 s]
```

tkrodriguez commented on Sep 12, 2016 Member

[9/12/16, 11:07:31 AM] Tom Rodriguez: we seem to be writing the values with the wrong endianness.
[9/12/16, 11:07:39 AM] Tom Rodriguez: java.lang.AssertionError: expected:<54694> but was:<42709>
[9/12/16, 11:07:46 AM] Tom Rodriguez: $ tohex 42709
a6d5
$ tohex 54694
d5a6

tkrodriguez commented on Sep 12, 2016 Member

The problem is that the reverseBytes plugins for short and char do incorrect canonicalization that may eliminate that reverse completely.

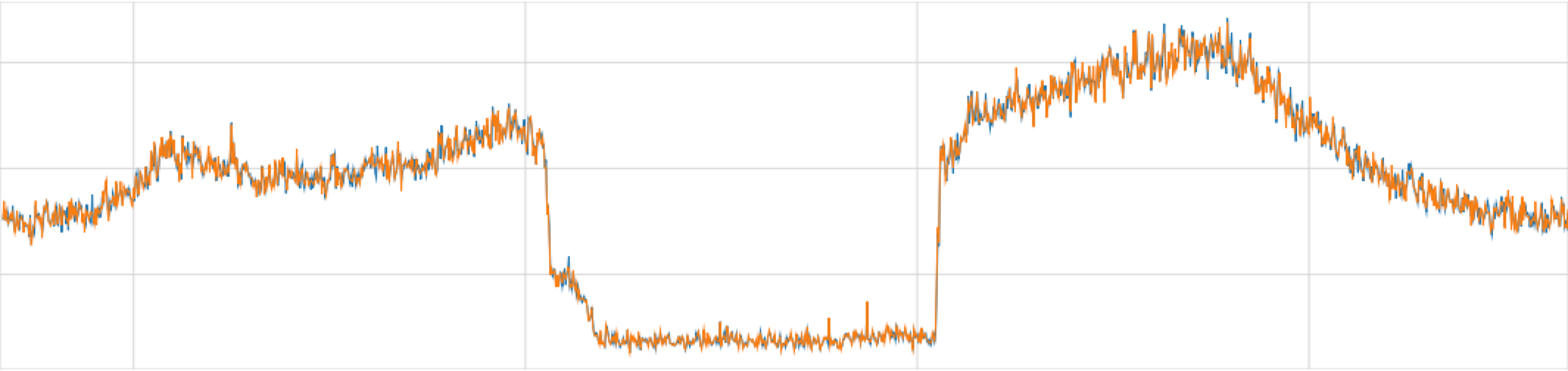
#206 Closed

Showing **5 changed files** with **68 additions** and **18 deletions.**

TWEET SERVICE

- Finagle-Thrift service
  - https://engineering.twitter.com/opensource/projects/finagle
- Testing properties:
  - Dedicated machines
  - All instances receive the exact same requests

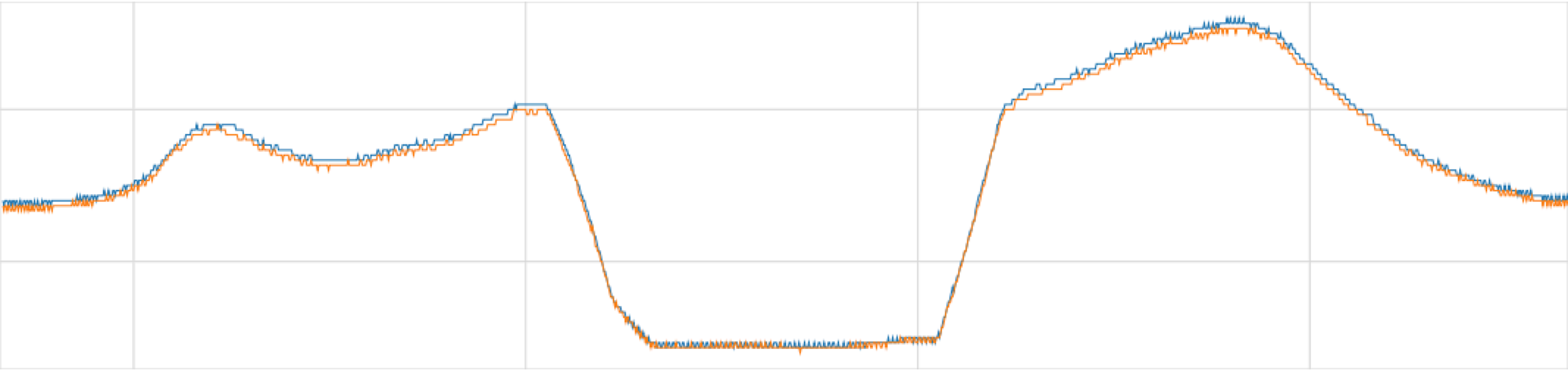
REQUESTS/SEC



C2 Graal

PS SCAVENGE CYCLES

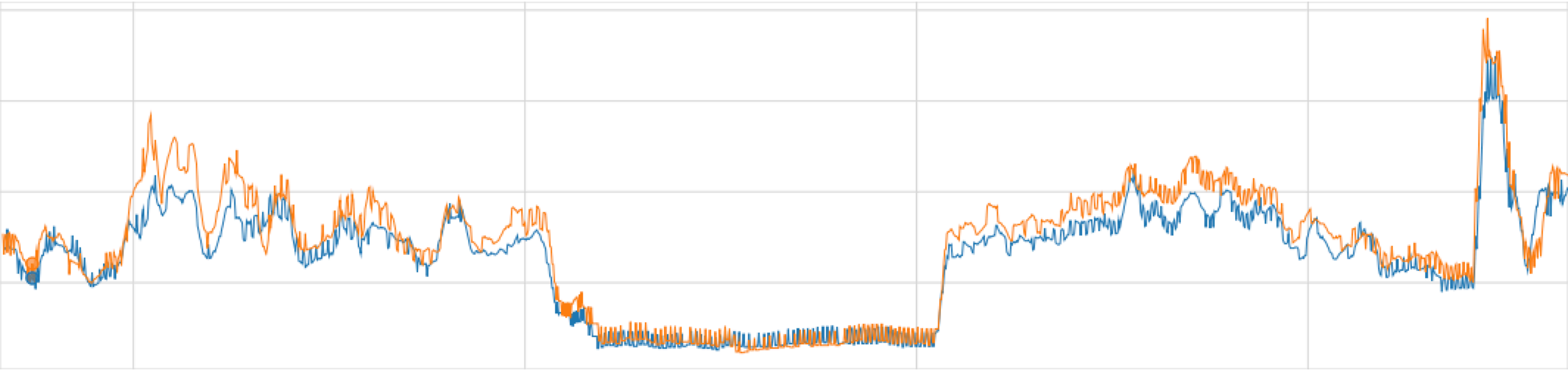
movingavg(60)



C2 Graal

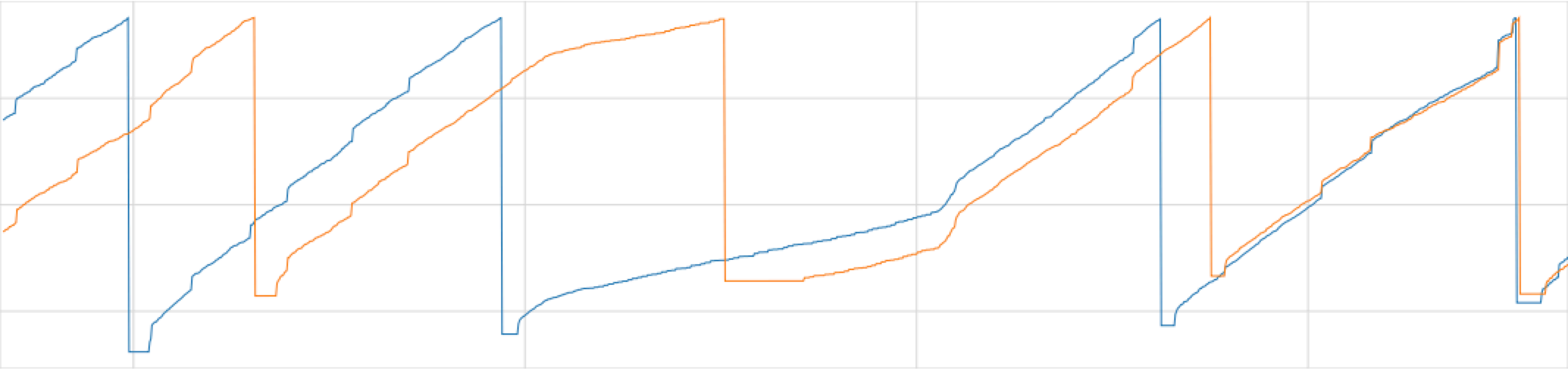
PS SCAVENGE TIME

movingavg(10)

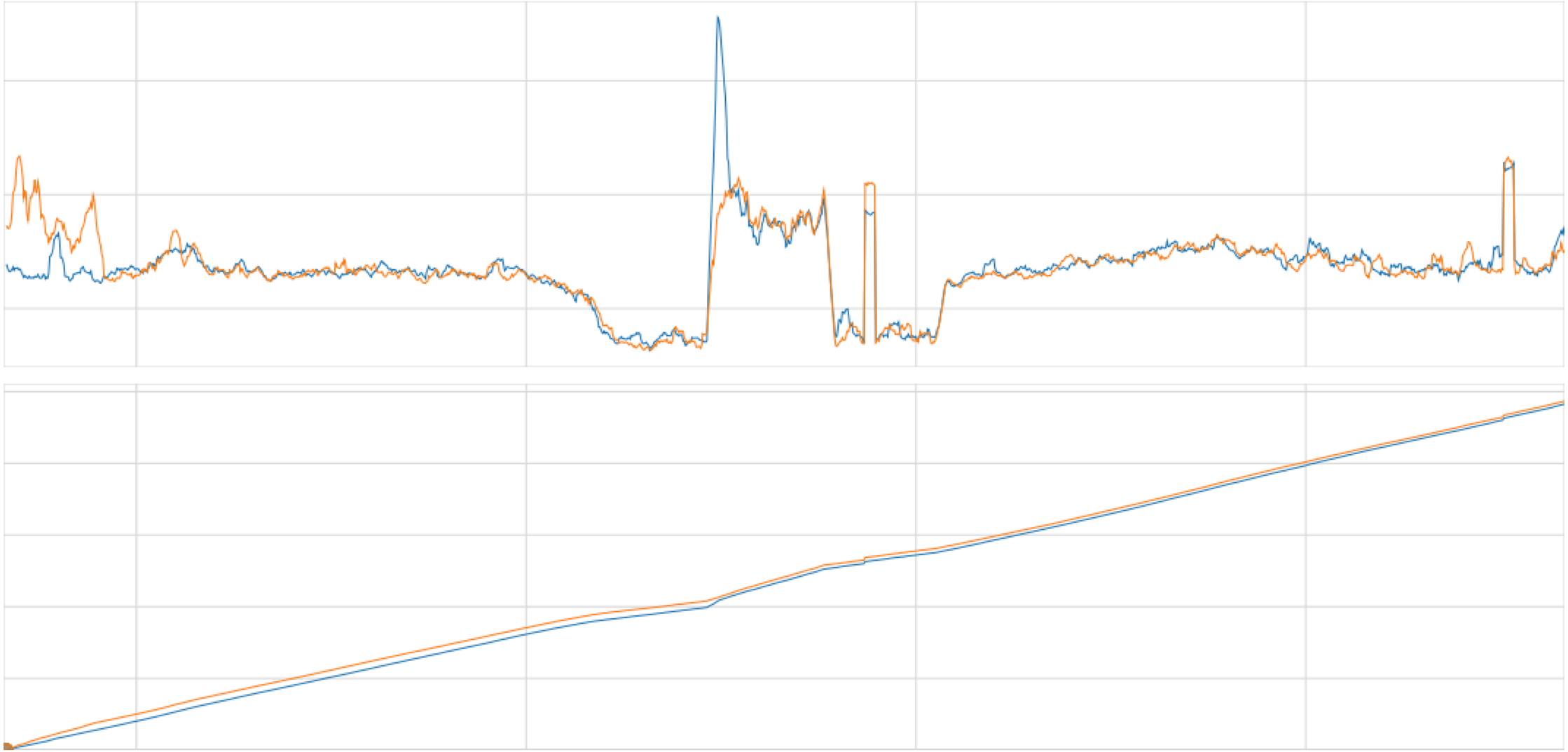


C2 Graal

OLD GEN USED

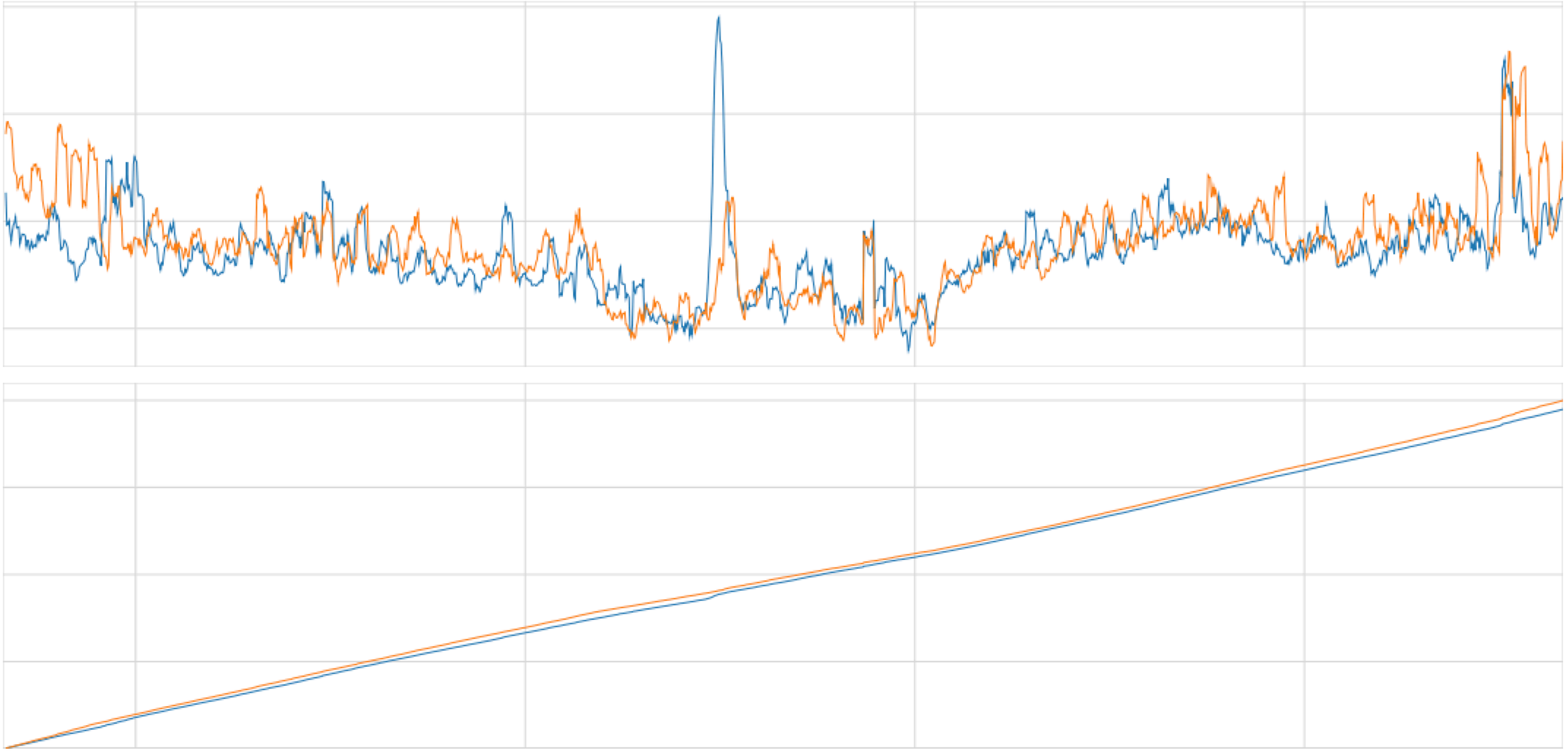


LATENCY P99



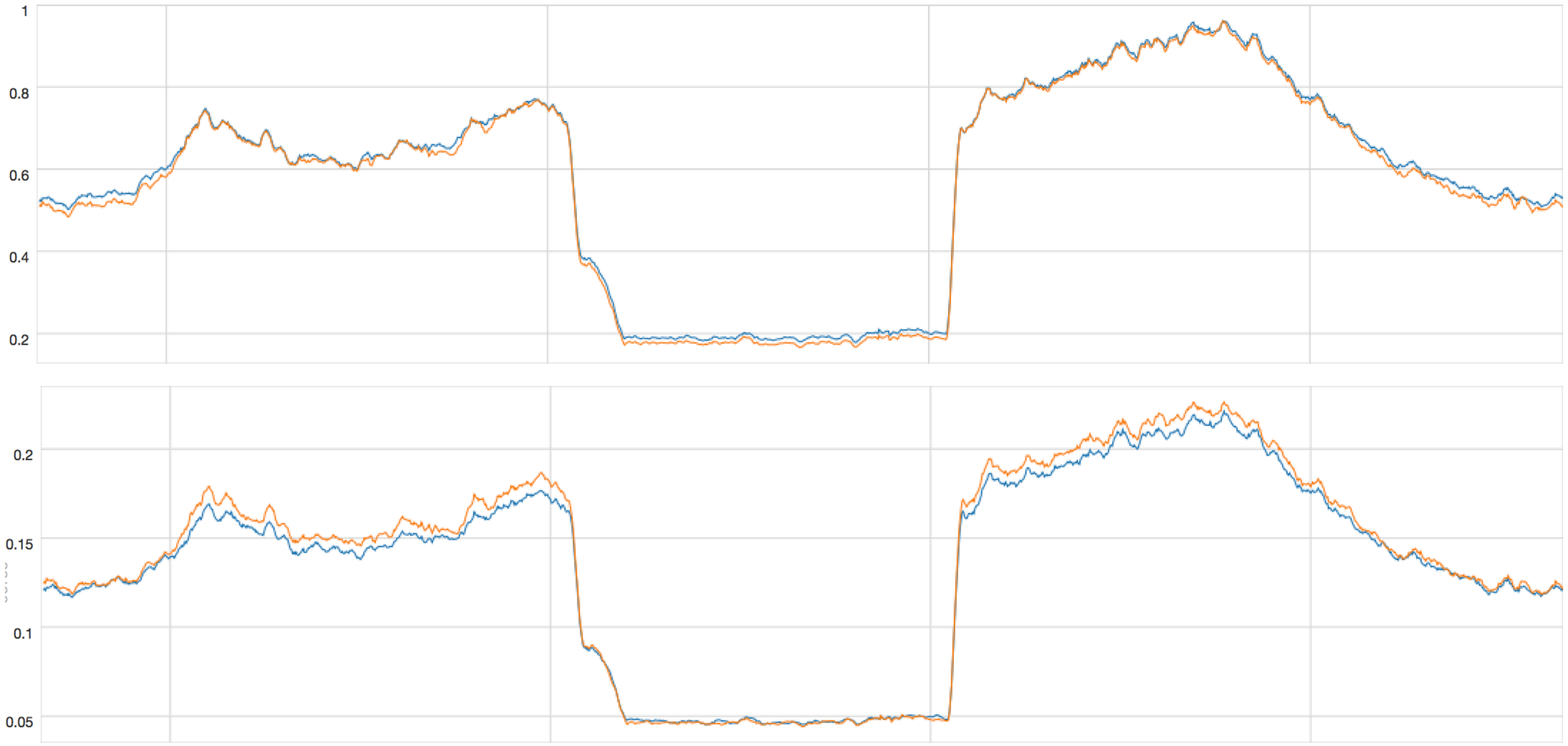
C2 Graal

LATENCY P9999



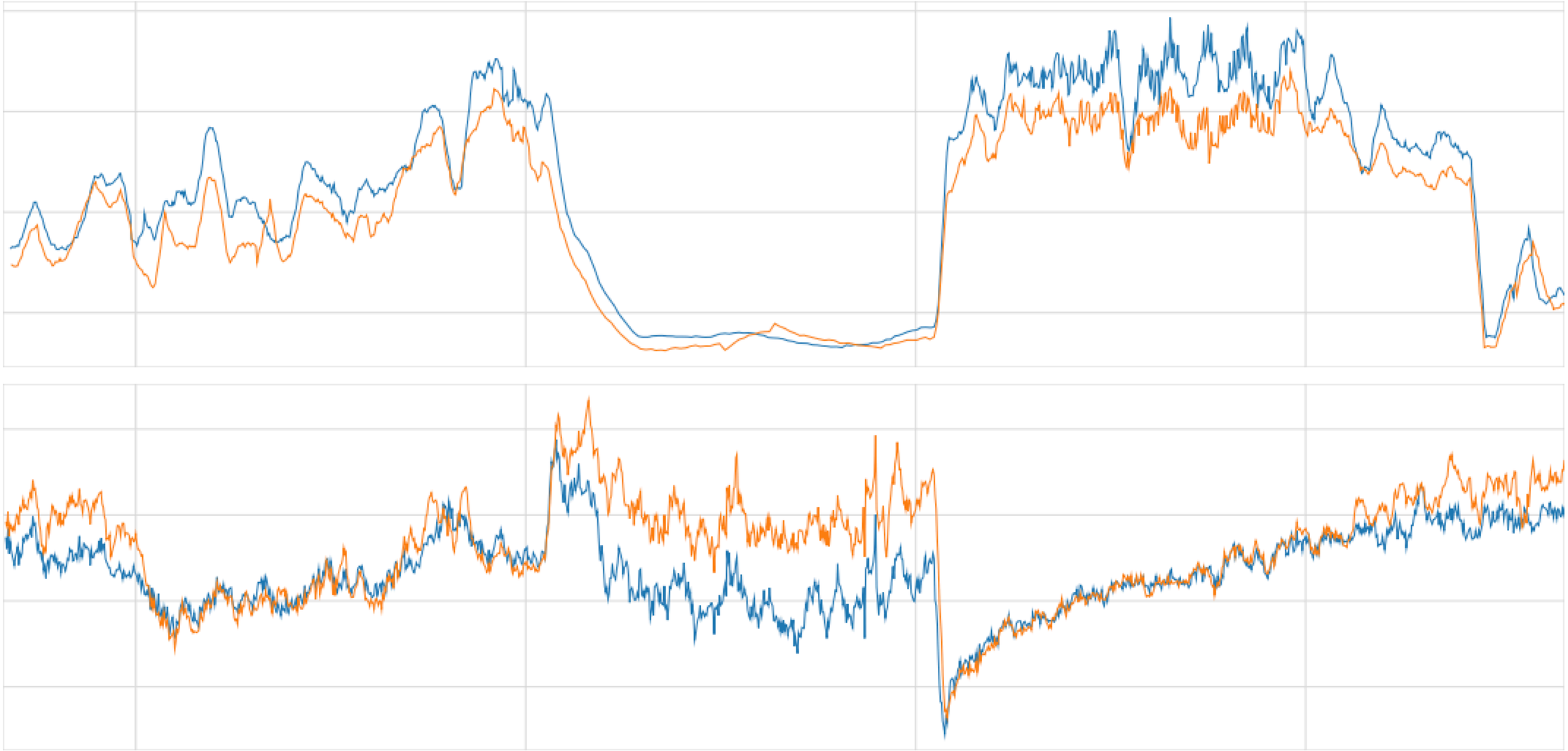
C2 Graal

USER/SYSTEM CPU TIME



C2 Graal

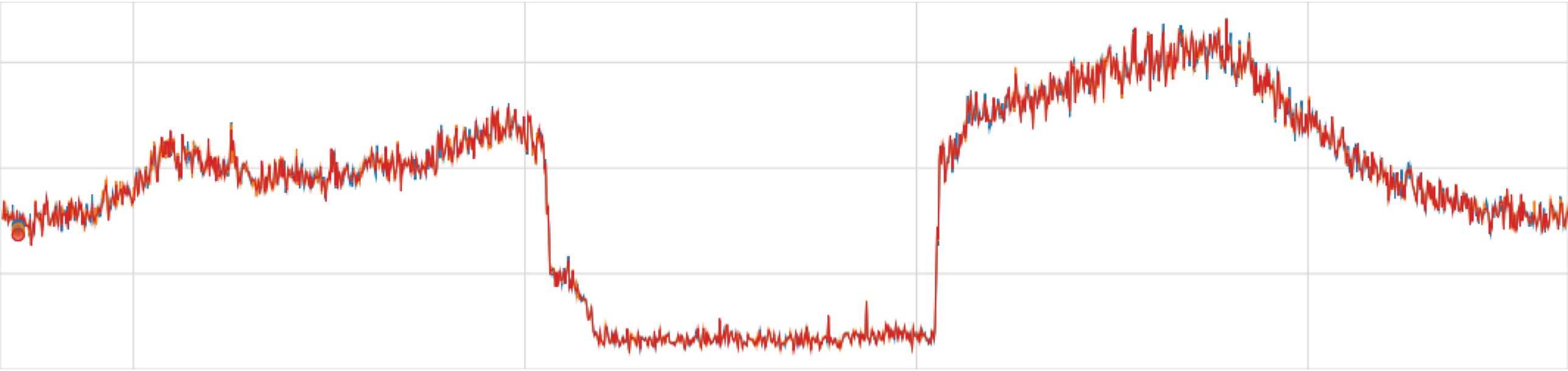
TWEETS/{GC,CPU} MSEC



C2 Graal

https://en.wikipedia.org/wiki/USS_Enterprise_(NCC-1701-A)

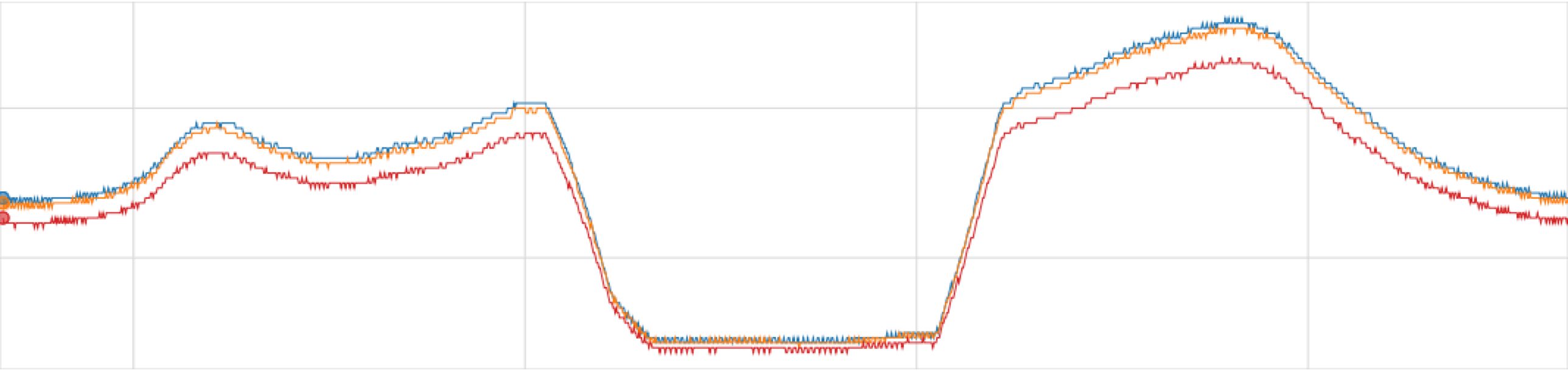
REQUESTS/SEC



C2 Graal Enterprise Graal

PS SCAVENGE CYCLES

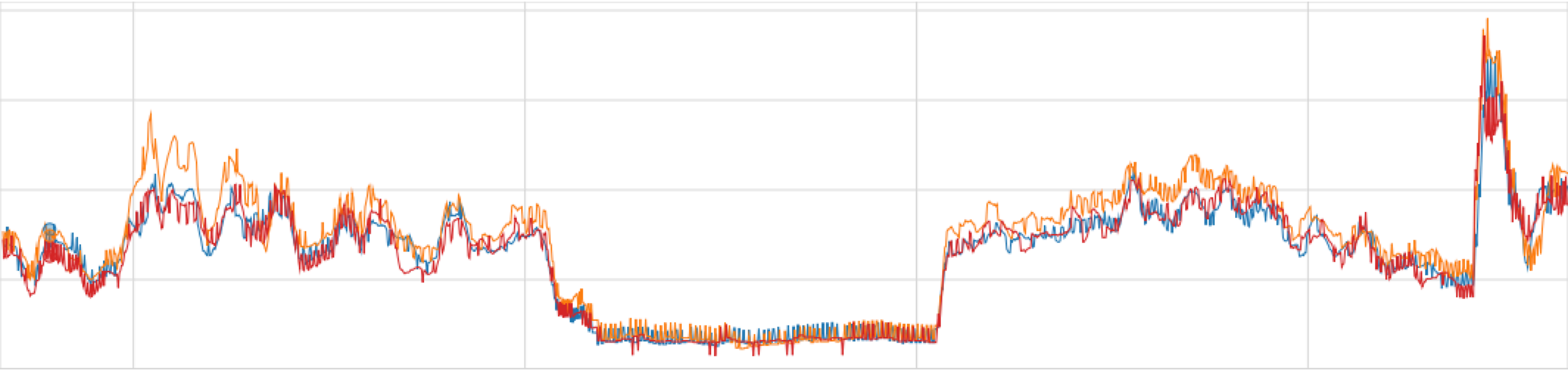
movingavg(60)



C2 Graal Enterprise Graal

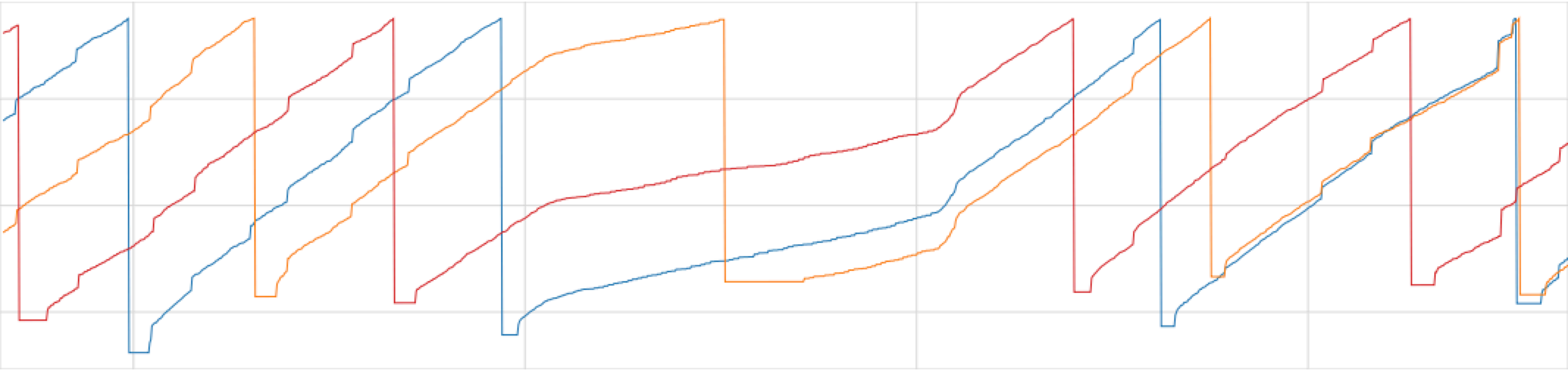
PS SCAVENGE TIME

movingavg(10)



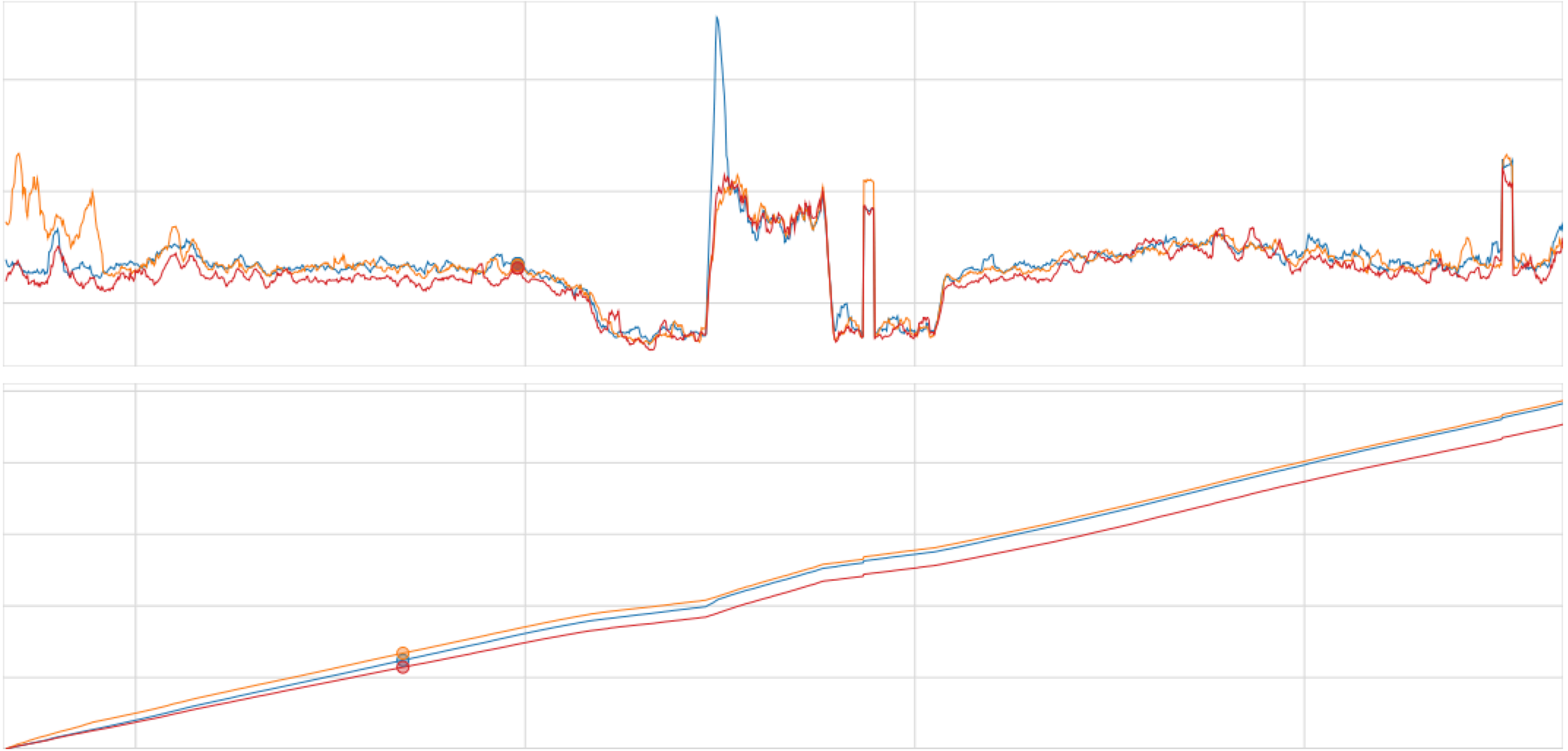
C2 Graal Enterprise Graal

OLD GEN USED



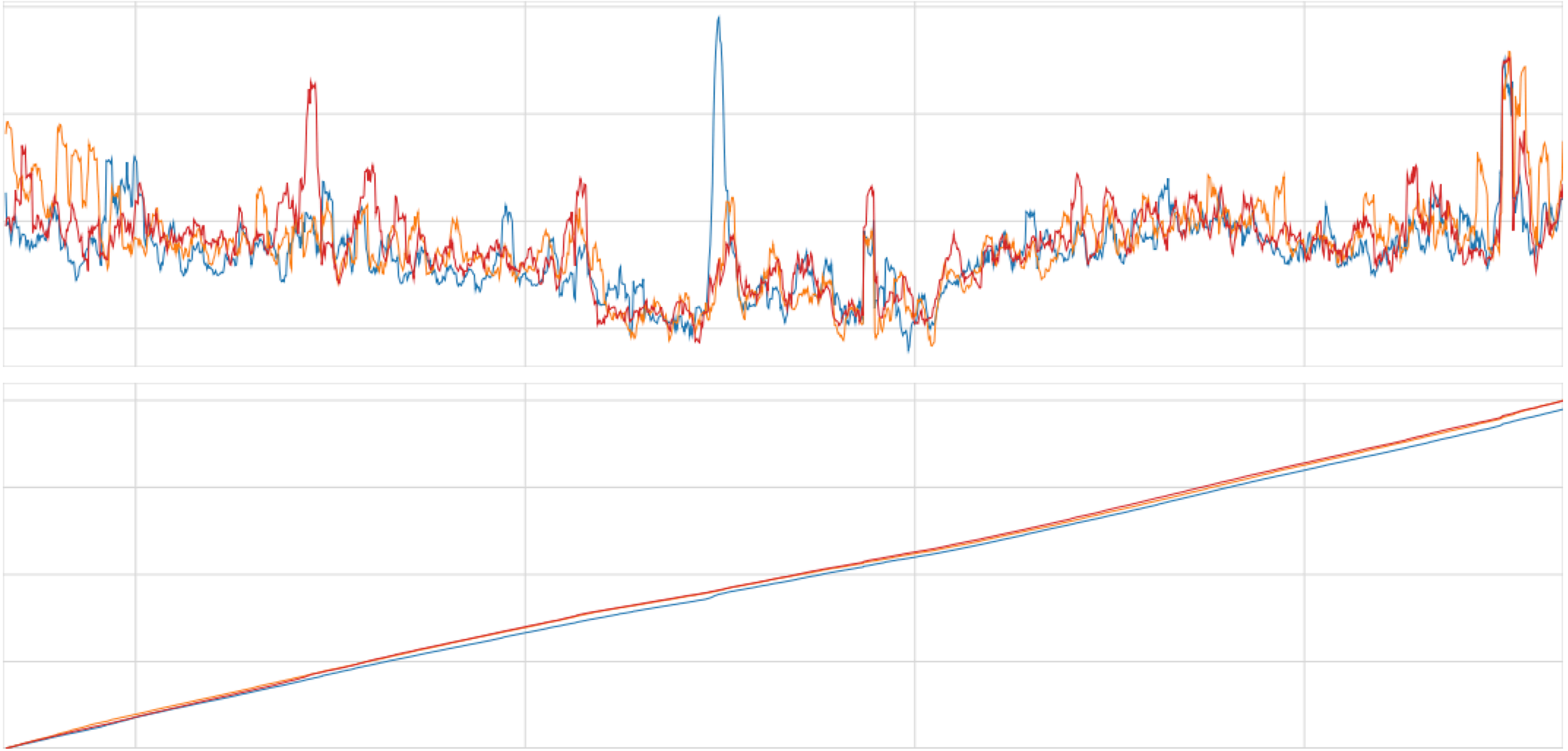
C2 Graal Enterprise Graal

LATENCY P99



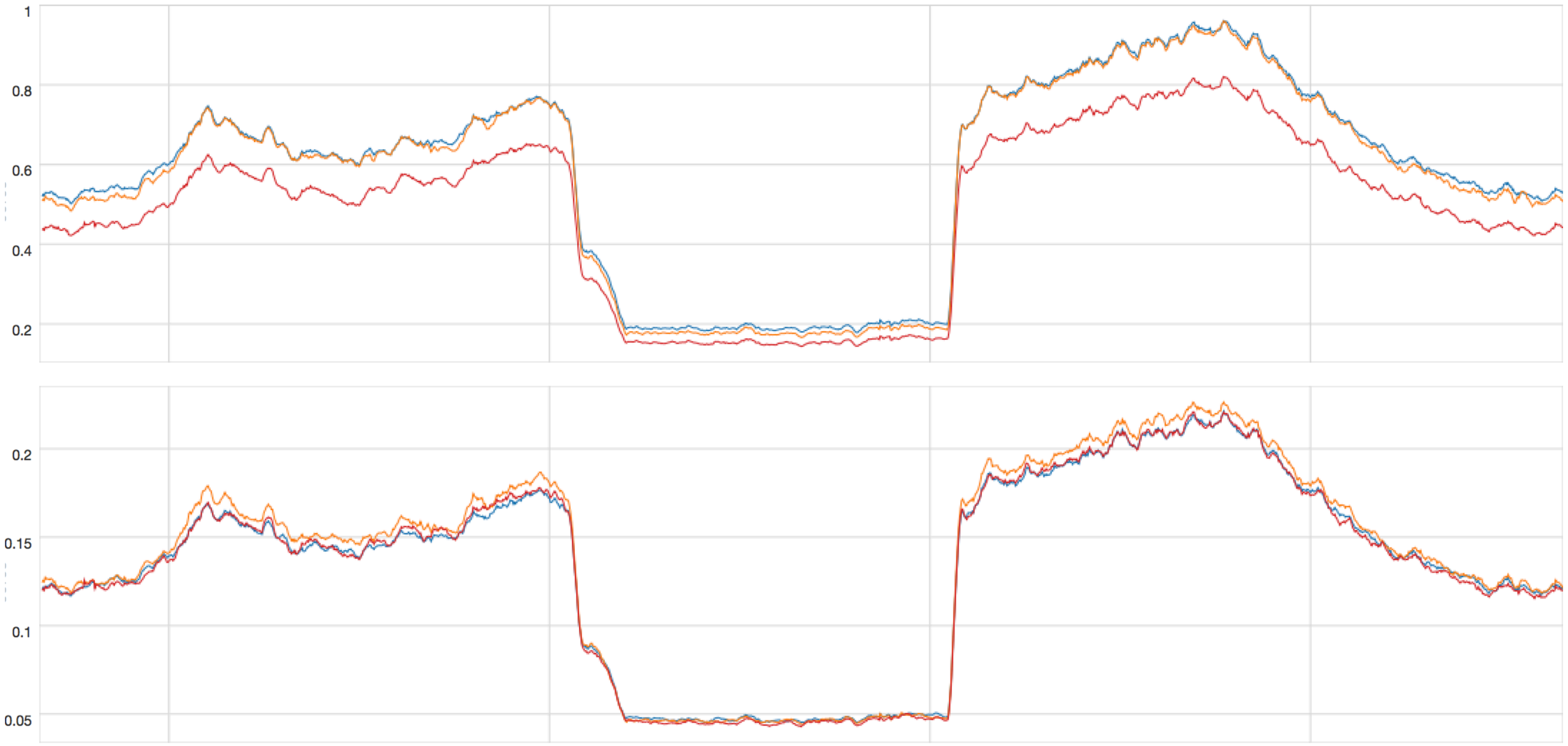
C2 Graal Enterprise Graal

LATENCY P9999



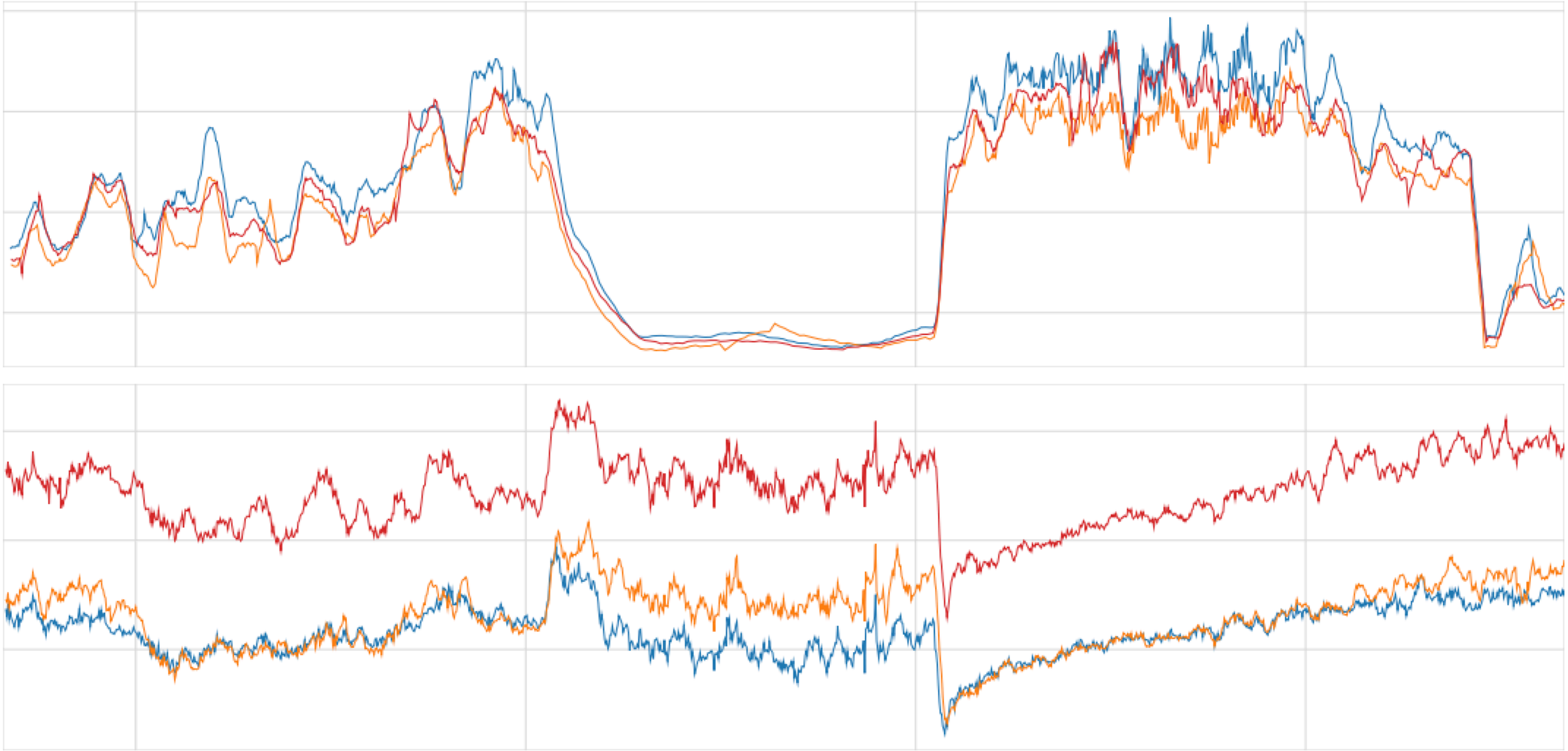
C2 Graal Enterprise Graal

USER/SYSTEM CPU TIME



C2 Graal Enterprise Graal

TWEETS/{GC,CPU} MSEC



C2 Graal Enterprise Graal

#TWITTERVMTEAM