PERFORMANCE TUNING TWITTER SERVICES WITH GRAAL AND MACHINE LEARNING

@CHRISTHALINGER

#TWITTERVMTEAM

@CHRISTHALINGER

Oracle Groundbreaker Ambassador

ABOUT ME...

CHRIS

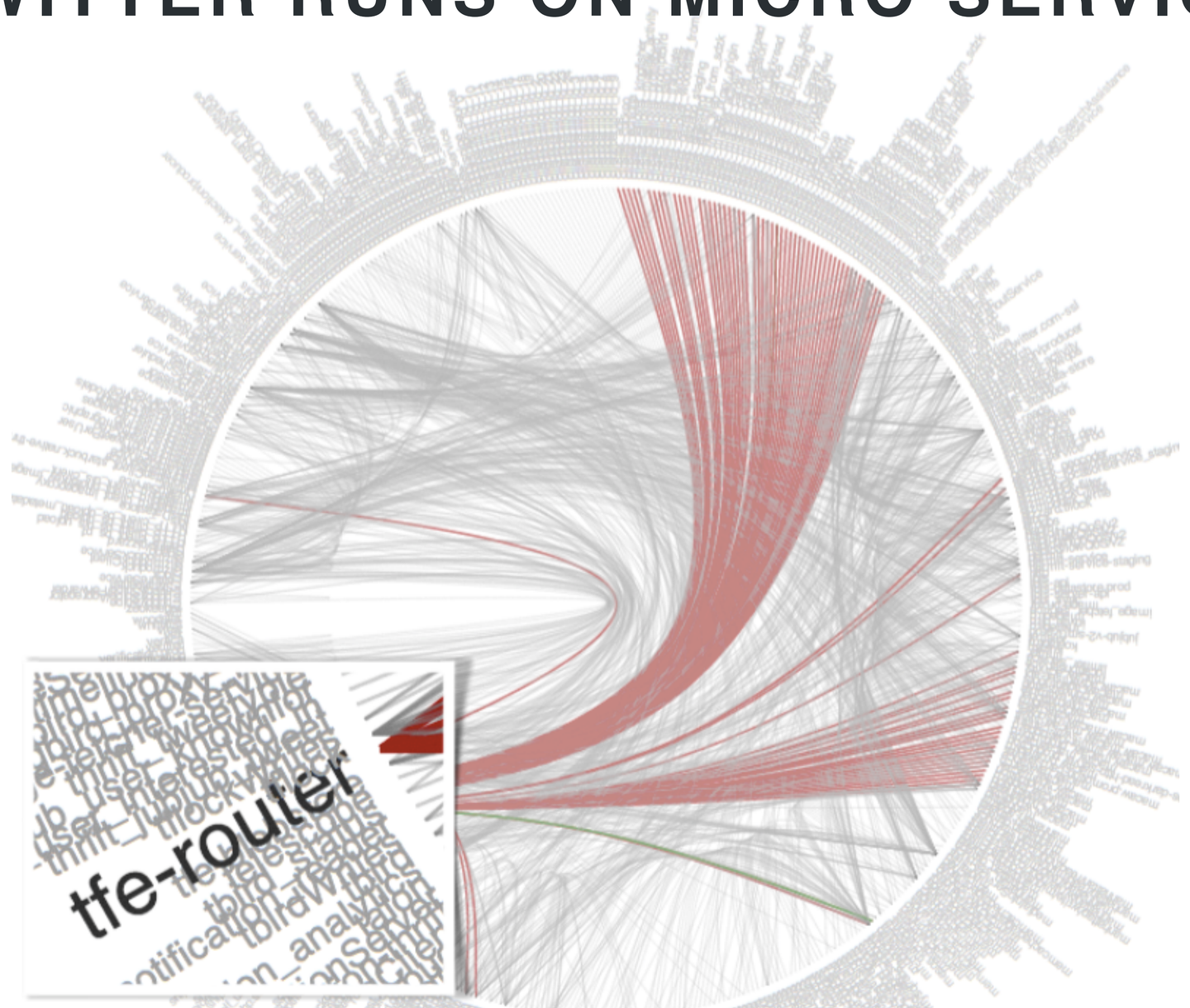
@christhalinger

currently @Twitter // former @Oracle, @SunMicrosystems // Oracle Groundbreaker Ambassador @groundbreakers // @LavaOneConf organizer // @HawaiiJUG leader

Haleiwa, HI

- Working on JVMs for over 14 years
- Sun Microsystems/Oracle HotSpot compiler team
  - JSR 292: Supporting Dynamically Typed Languages on the JavaTM Platform
  - JEP 243: Java-Level JVM Compiler Interface
  - JEP 295: Ahead-of-Time Compilation
- Twitter VM team

TWITTER RUNS ON MICRO SERVICES



- O(10^3) services
- O(10^5) service instances
- Heterogeneous hardware

Source: "How we built a metering and chargeback system to incentivize higher resource utilization of Twitter infrastructure", Micheal Arul, Vinu Charanya, *LinuxCon 2016*, Toronto, August 22-24, 2016.

PERFORMANCE OPTIMIZATION

- Hand-tuning doesn’t scale
  - few parameters handled manually
  - time-consuming, labor-intensive, error-prone
- Upgrades make optimality fleeting

Many services operate below optimality

PERFORMANCE TUNING

- Given a function $f(x_1, x_2, \ldots, x_n)$ defined over domain **X**

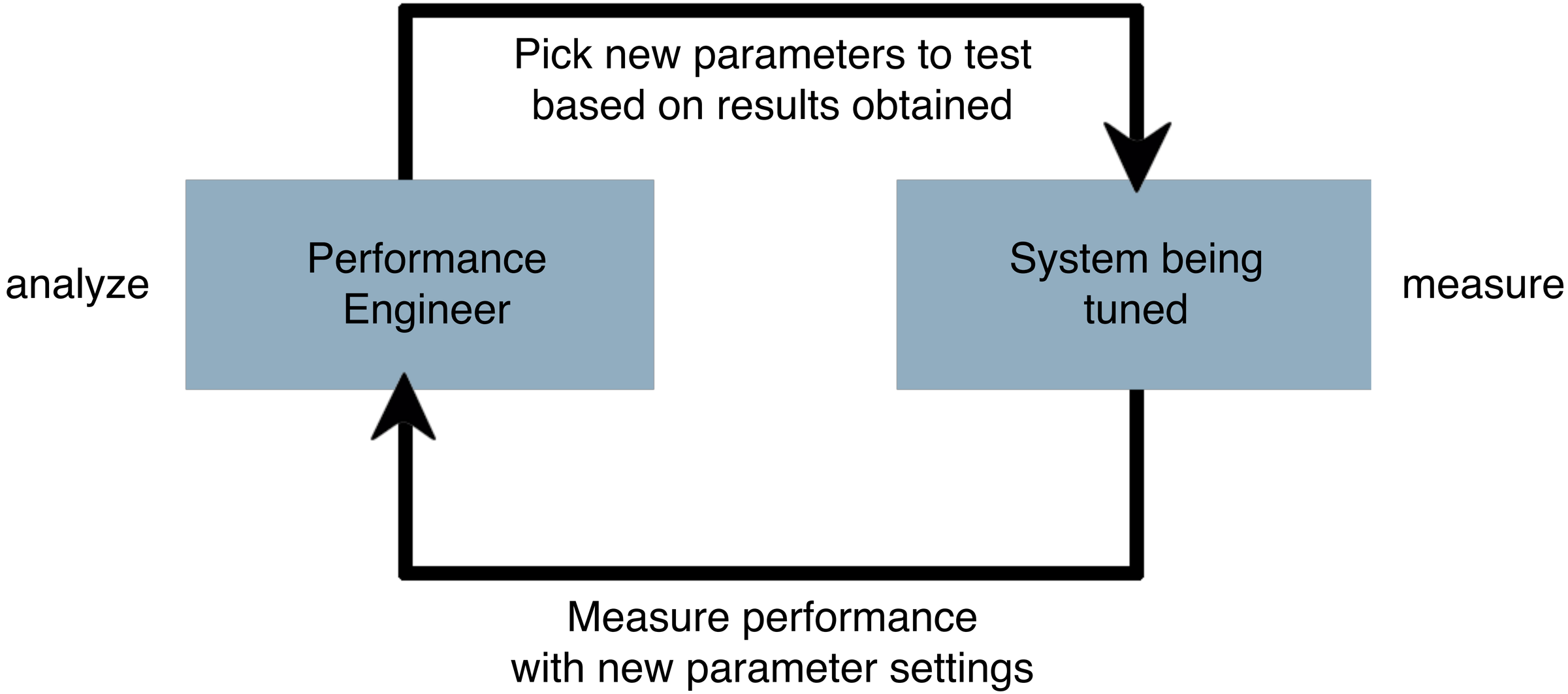
PERFORMANCE TUNING

- Given a function $f(x_1, x_2, \ldots, x_n)$ defined over domain **X**,
- find a configuration $\boldsymbol{A} = (a_1, a_2, \ldots, a_n)$ that maximizes **f**

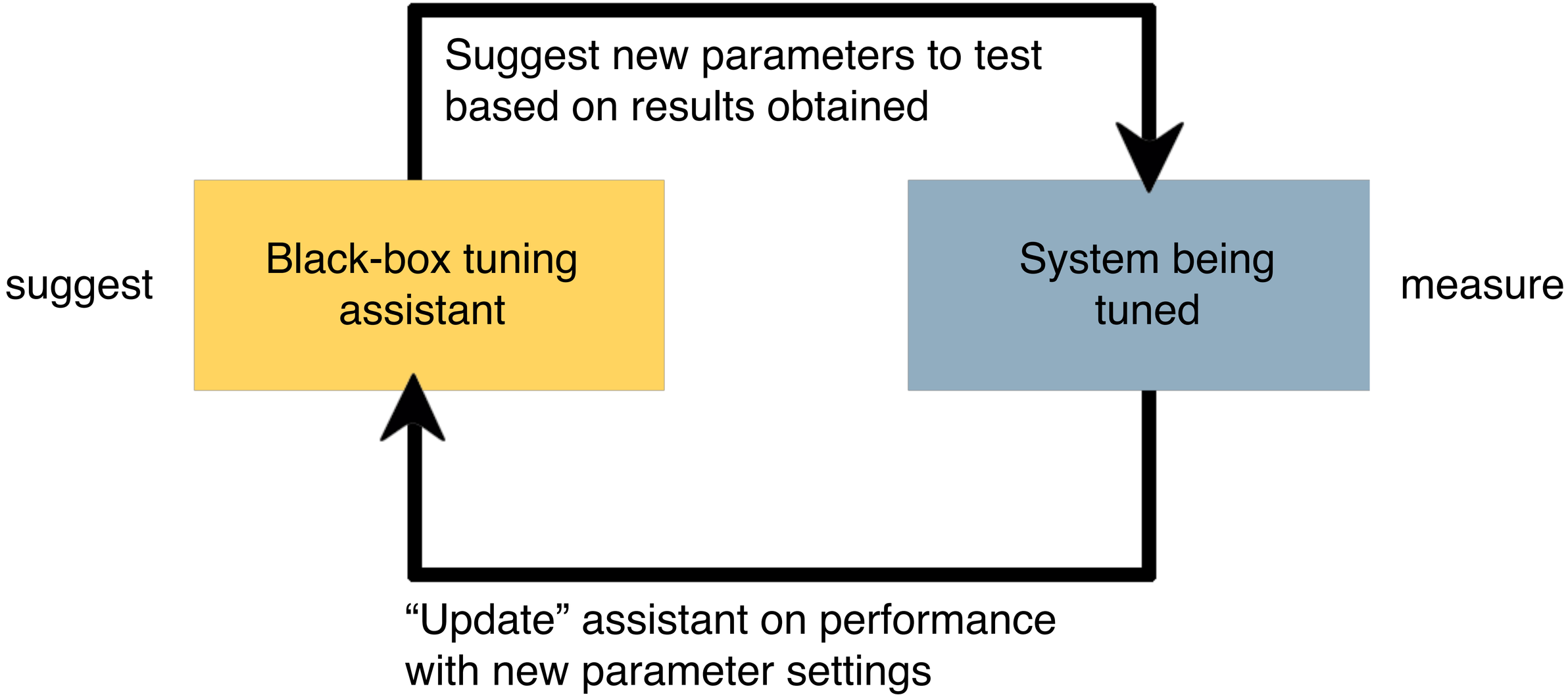
PERFORMANCE TUNING

- Given a function $f(x_1, x_2, \ldots, x_n)$ defined over domain **X**,
- find a configuration $\boldsymbol{A} = (a_1, a_2, \ldots, a_n)$ that maximizes **f**,
- subject to constraint predicate $G(x_1, x_2, \ldots, x_n)$

PERFORMANCE TUNING



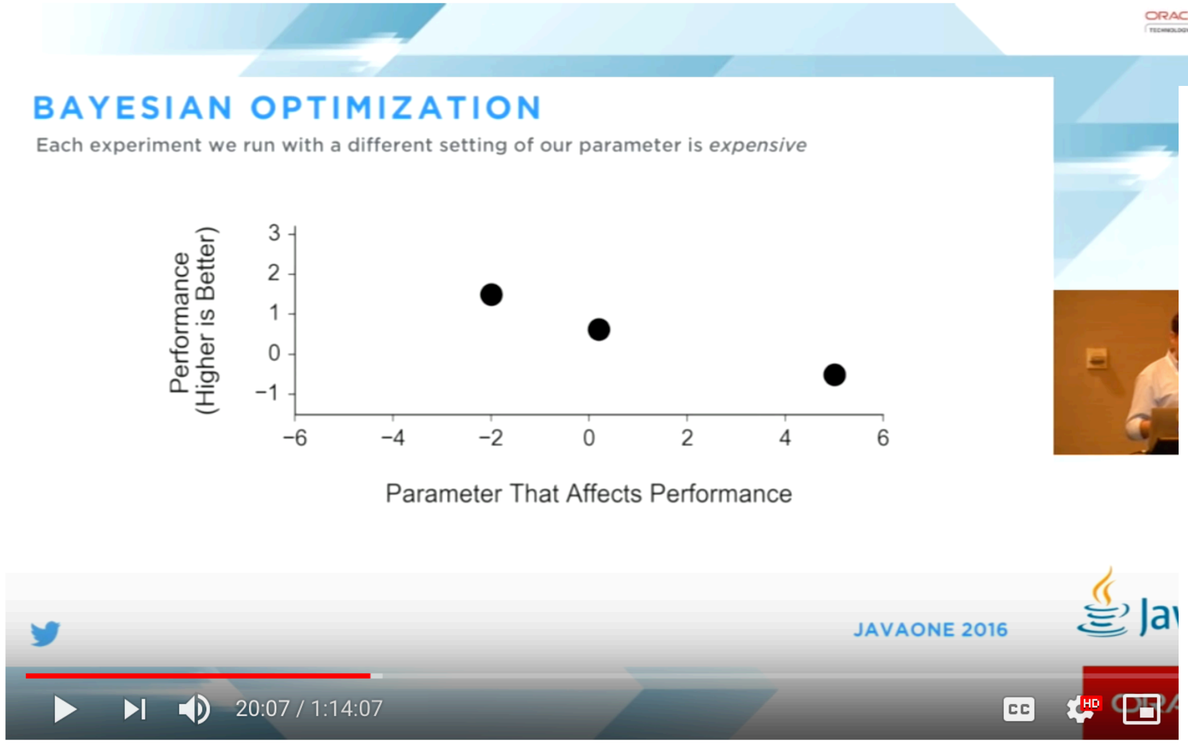
PERFORMANCE TUNING



OPTIMIZATION OF A BLACK BOX FUNCTION

- A technique from machine learning: **Bayesian Optimization**
- A method to learn (potentially noisy) objective functions
  - iteratively, efficiently
- Finds near-optima in few iterations
- Works well with non-linear, multi-modal, high dimensional functions

LISTEN TO THE EXPERTS



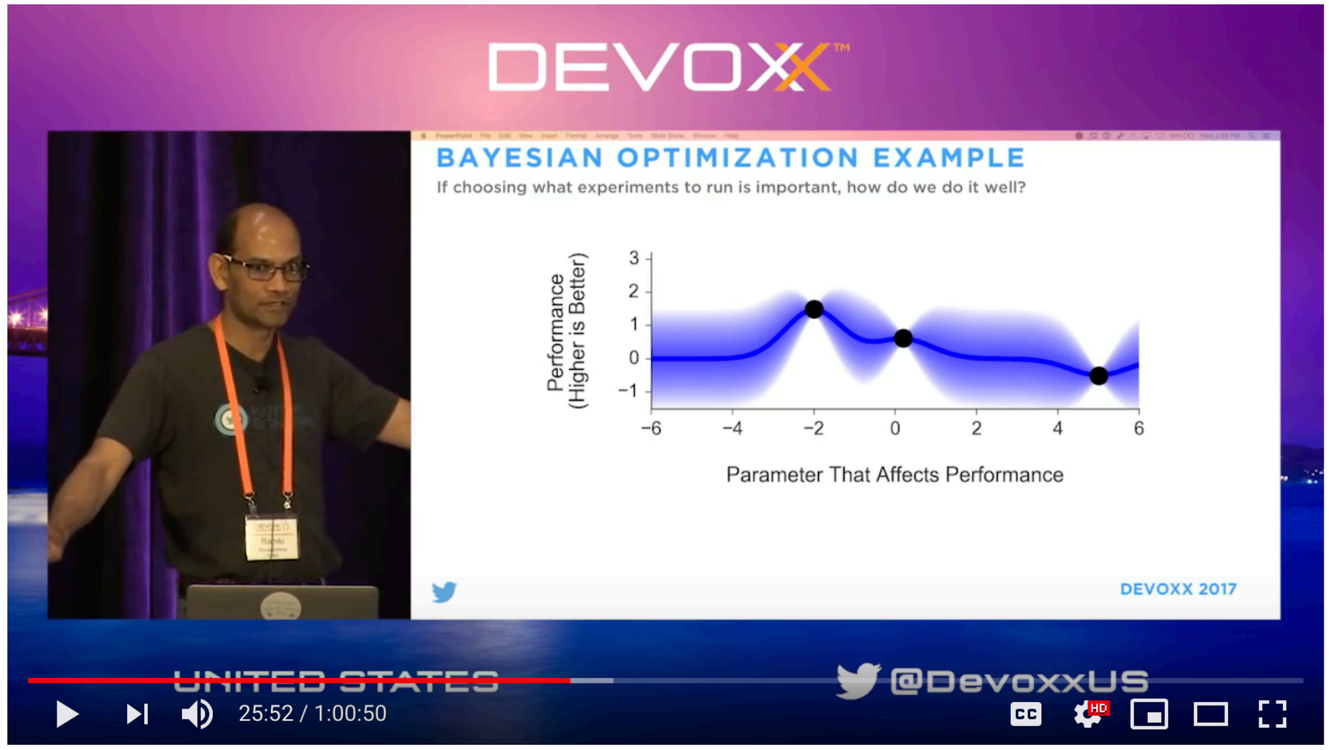
Automated Tuning of the JVM with Bayesian Optimization

1,997 views

21 0 SHARE S

Oracle Developers
Published on Sep 20, 2016

SUBSC



Continuous Optimization of MIcroservices using Machine Learning by Ramki Ramakrishna

1,358 views

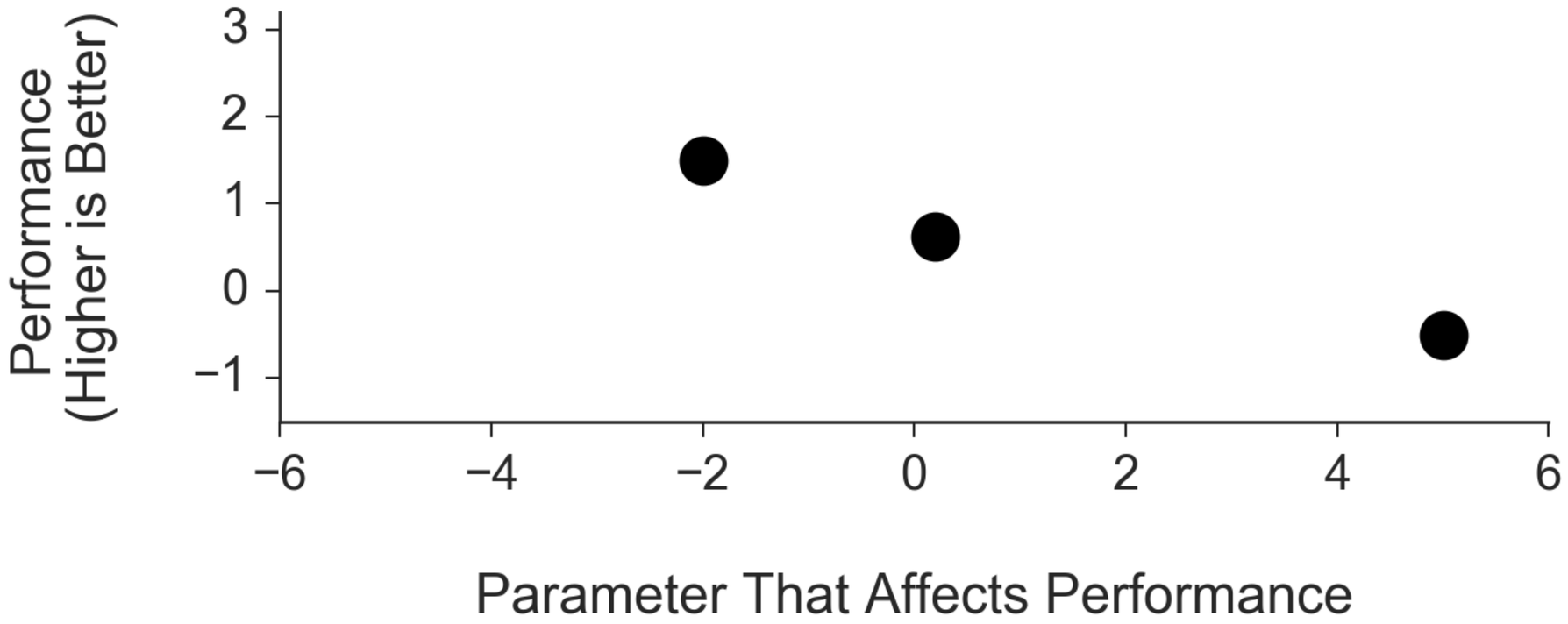
19 1 SHARE SAVE

Devoxx
Published on Apr 19, 2017

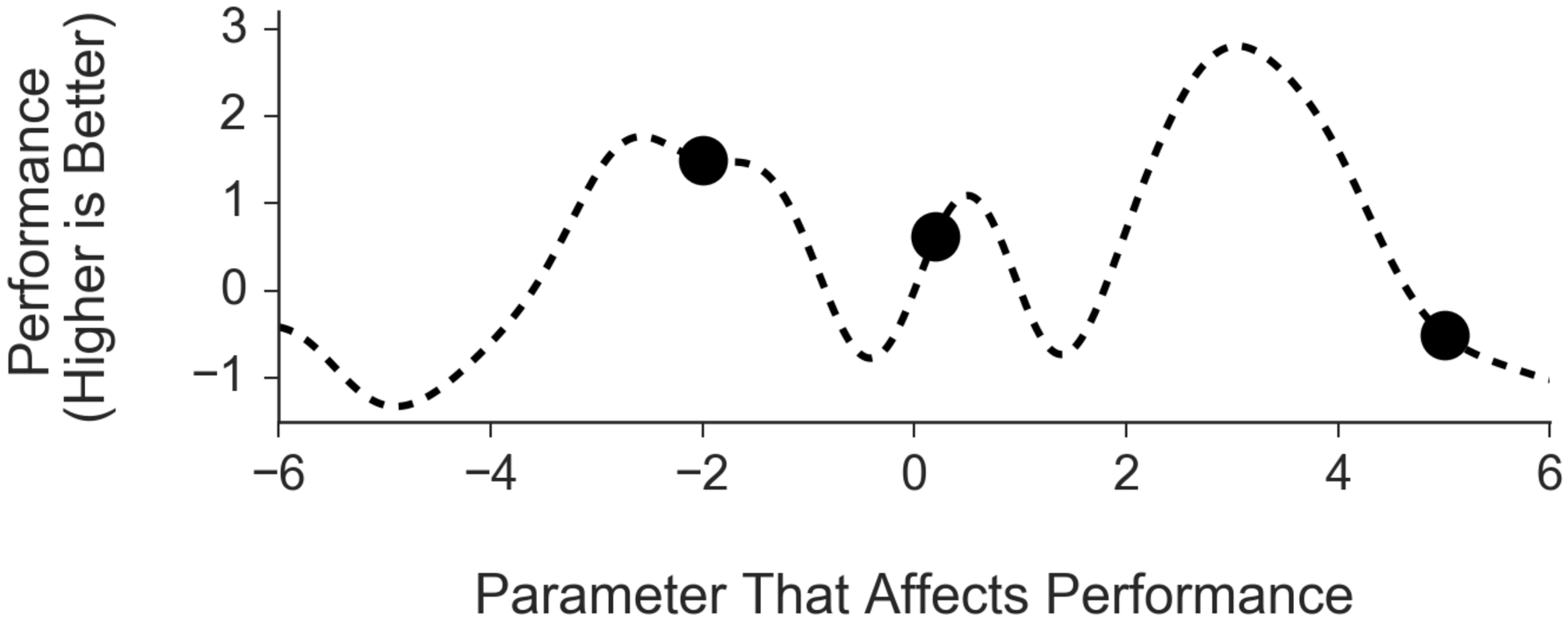
SUBSCRIBE 67K

@CHRISTHALINGER I #TWITTERVMTEAM

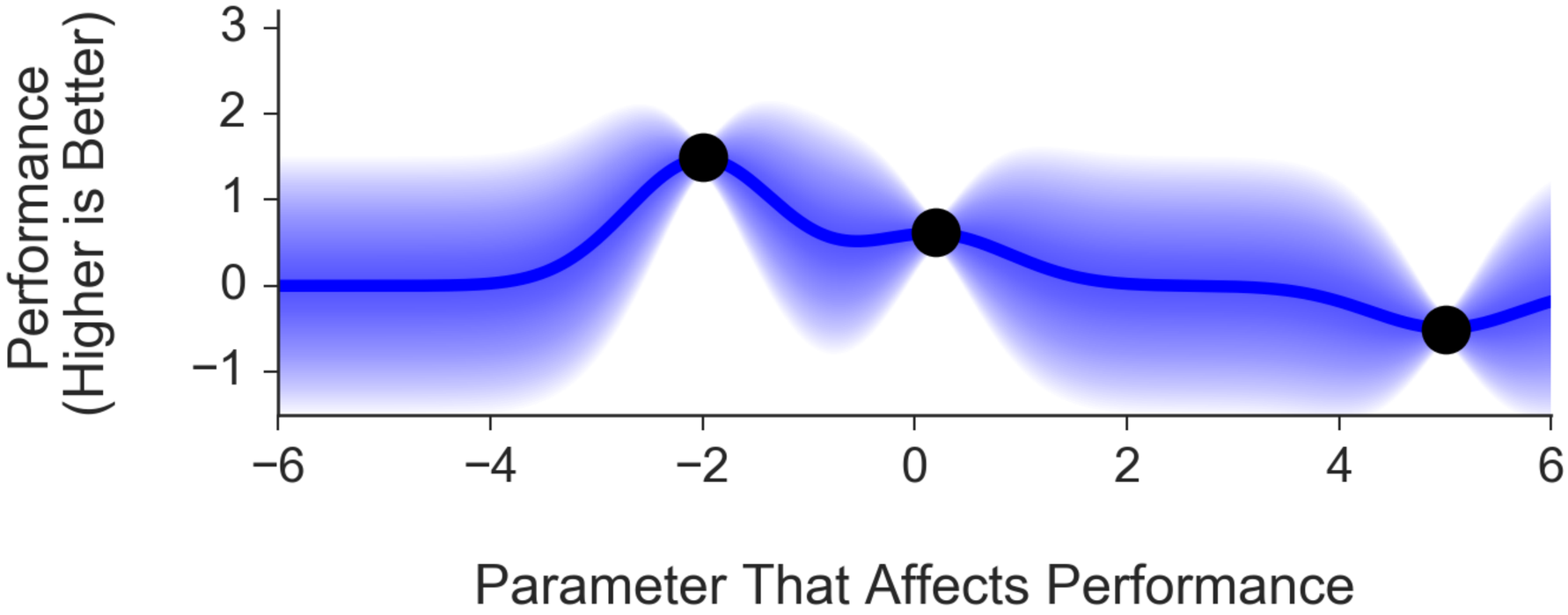
BAYESIAN OPTIMIZATION EXAMPLE



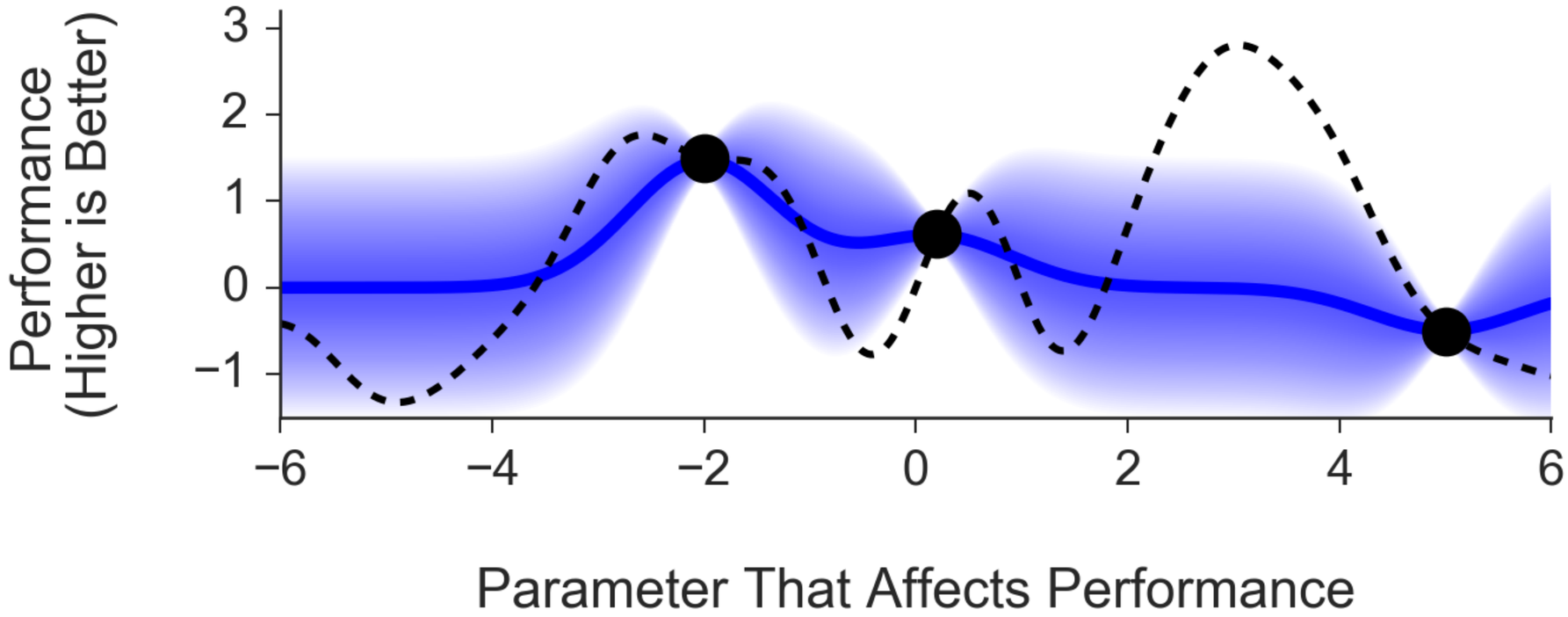
BAYESIAN OPTIMIZATION EXAMPLE



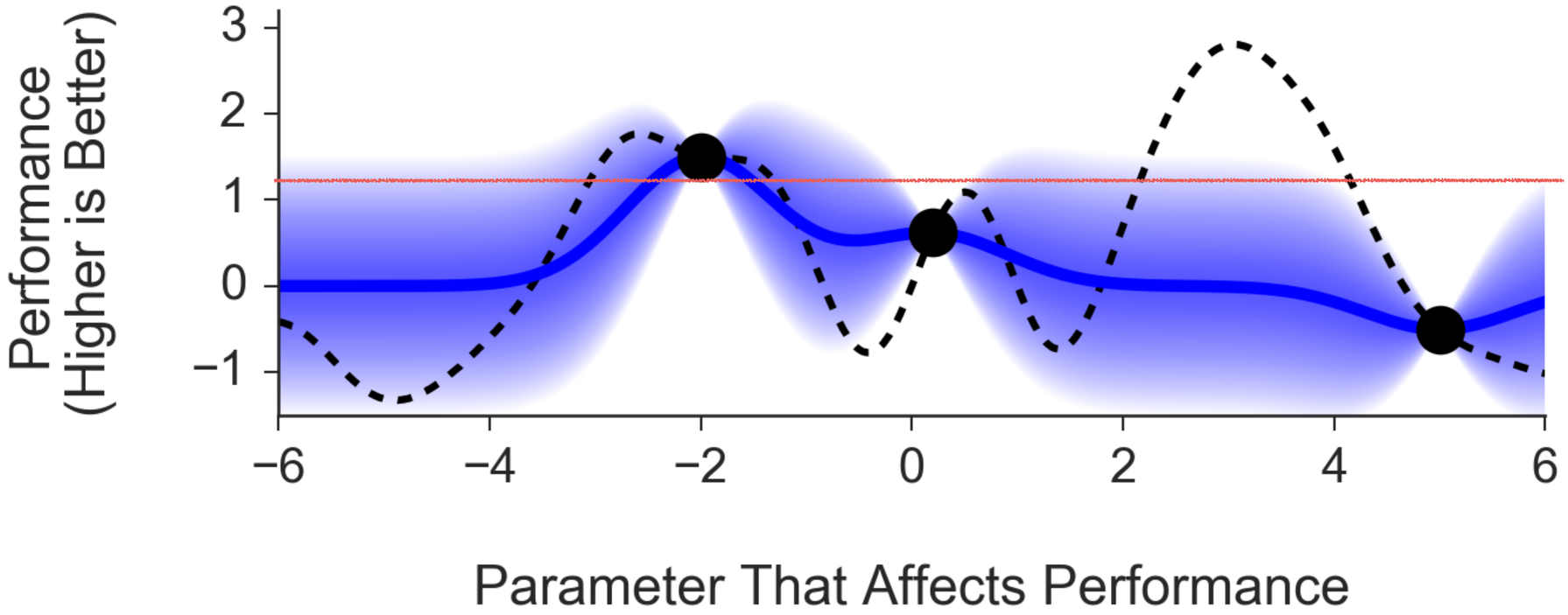
BAYESIAN OPTIMIZATION EXAMPLE

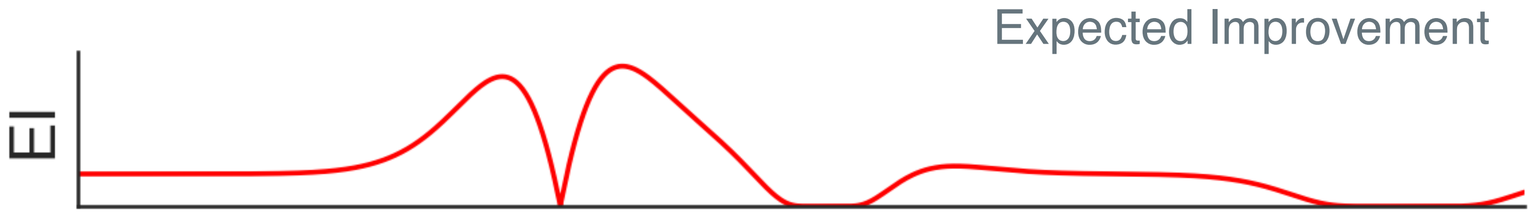


BAYESIAN OPTIMIZATION EXAMPLE

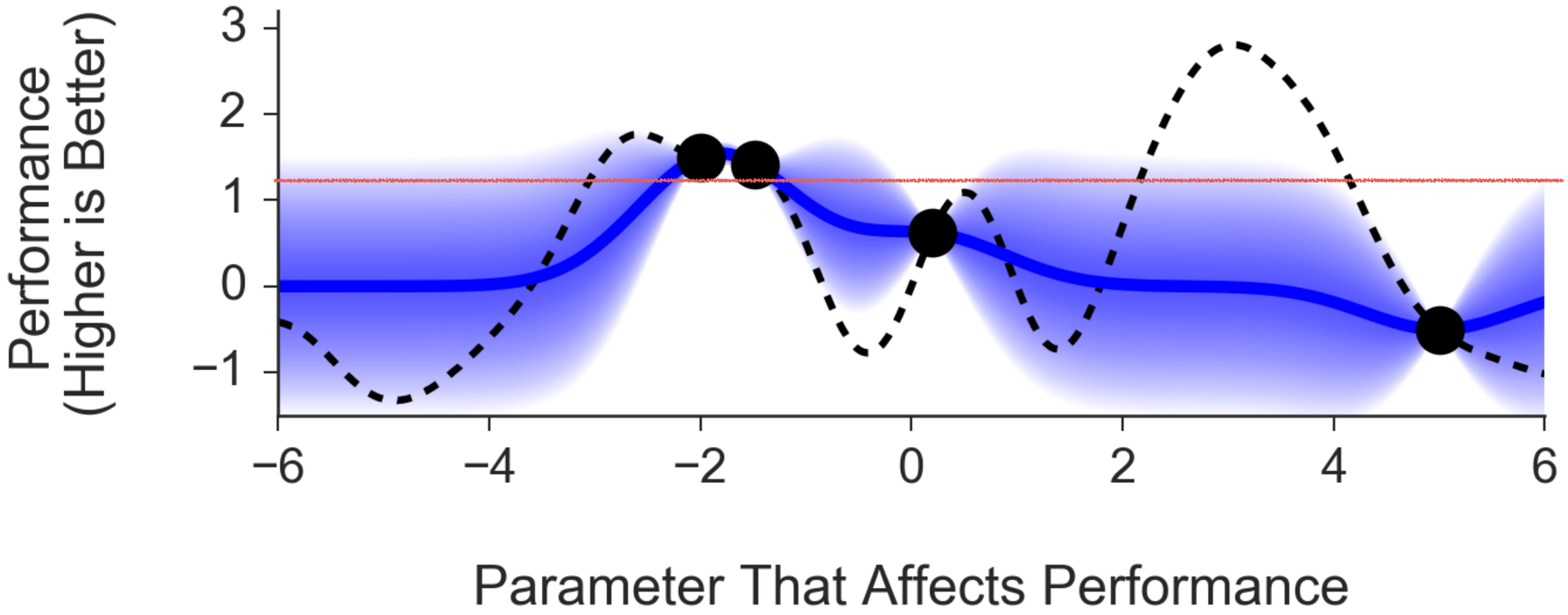


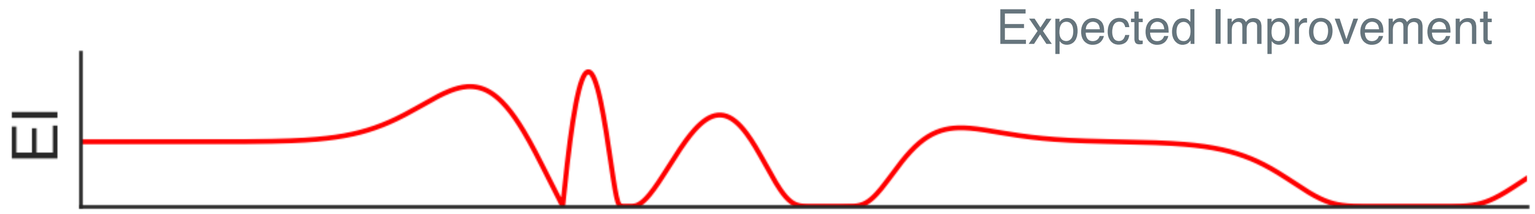
BAYESIAN OPTIMIZATION EXAMPLE



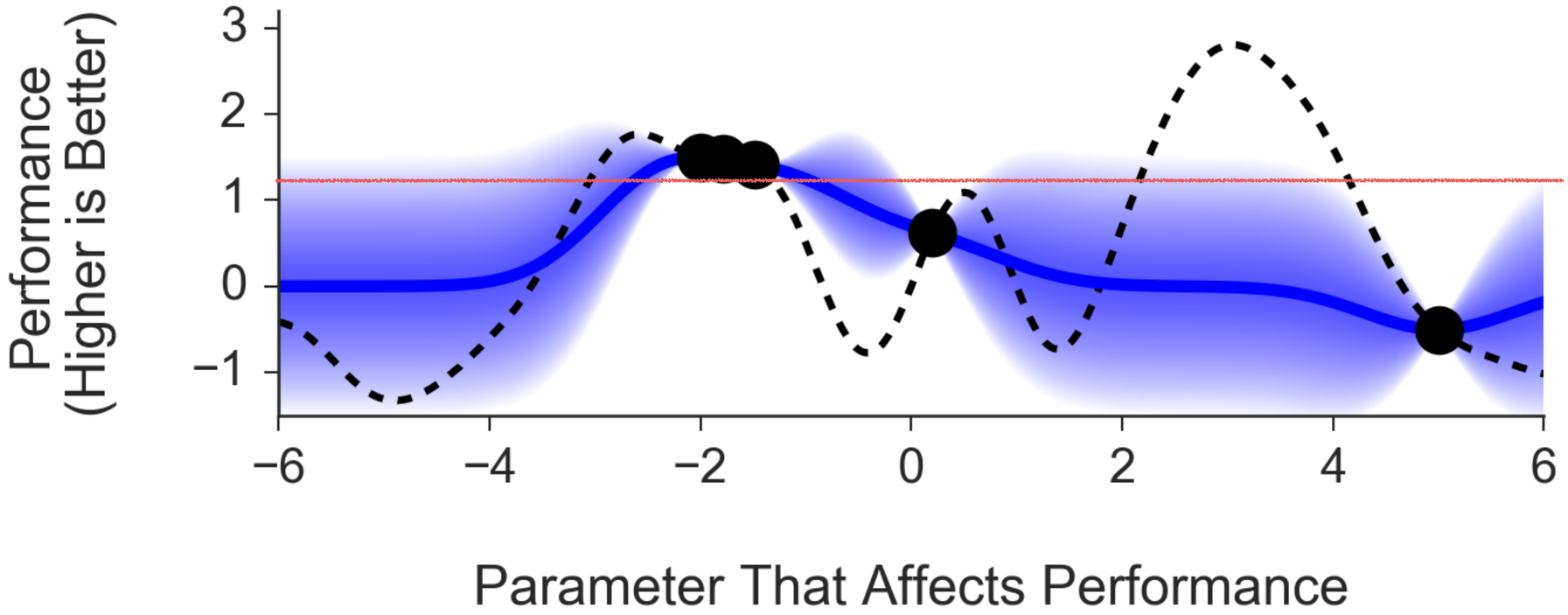


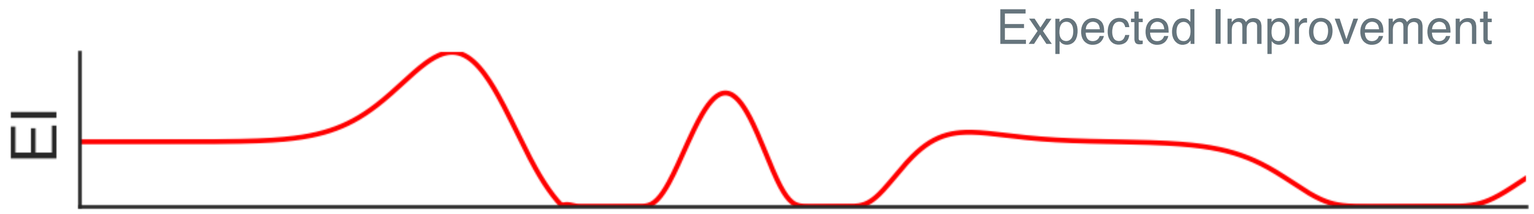
BAYESIAN OPTIMIZATION EXAMPLE



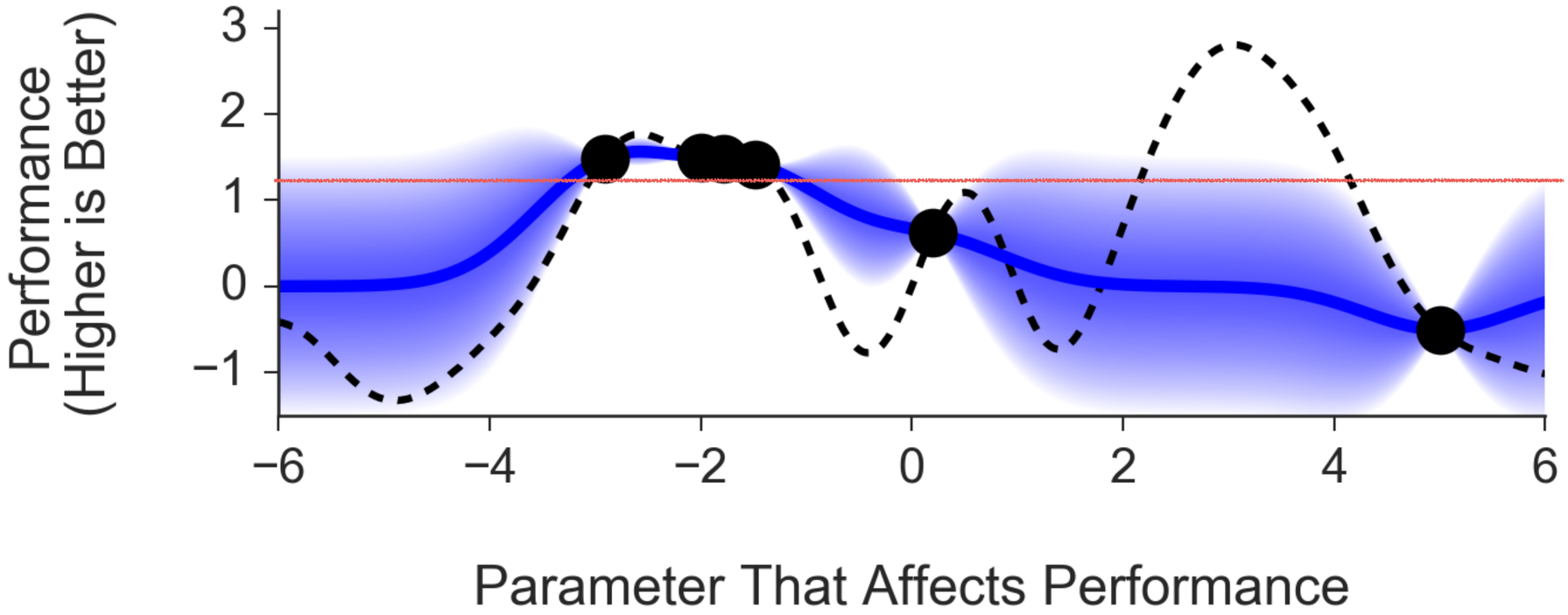


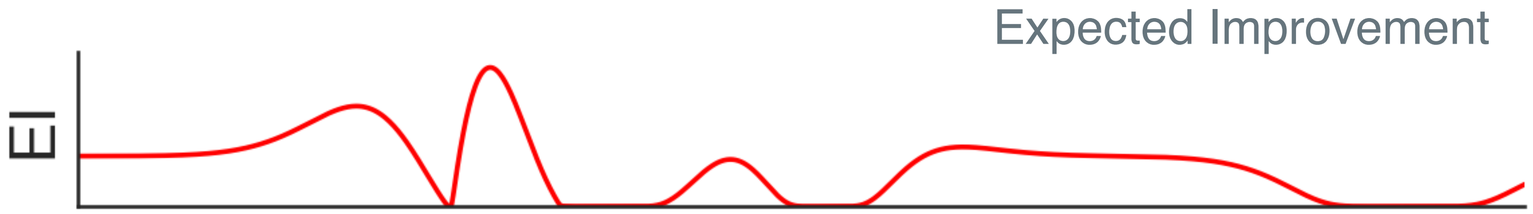
BAYESIAN OPTIMIZATION EXAMPLE



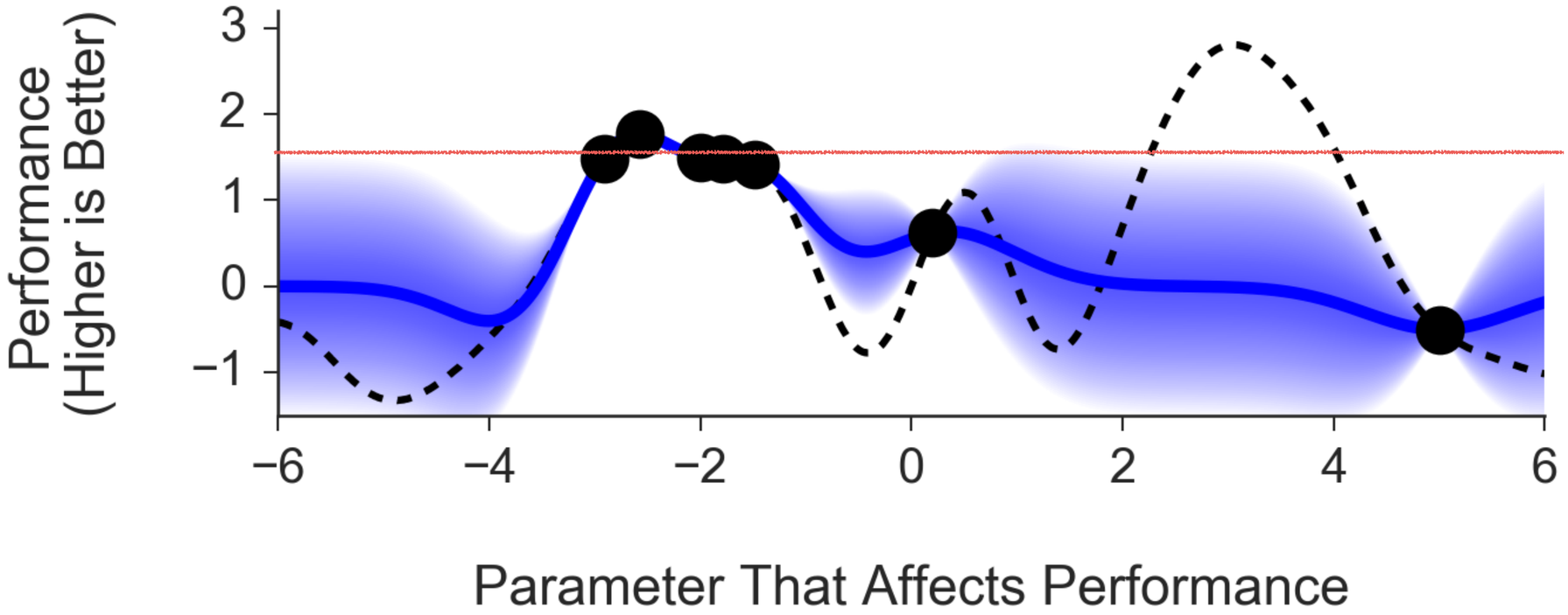


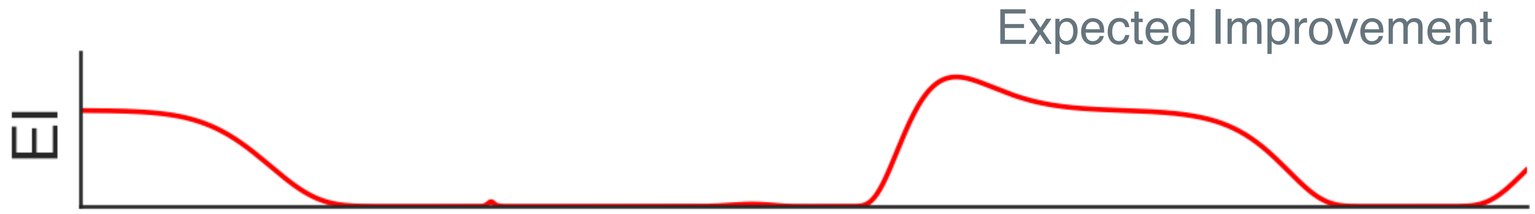
BAYESIAN OPTIMIZATION EXAMPLE



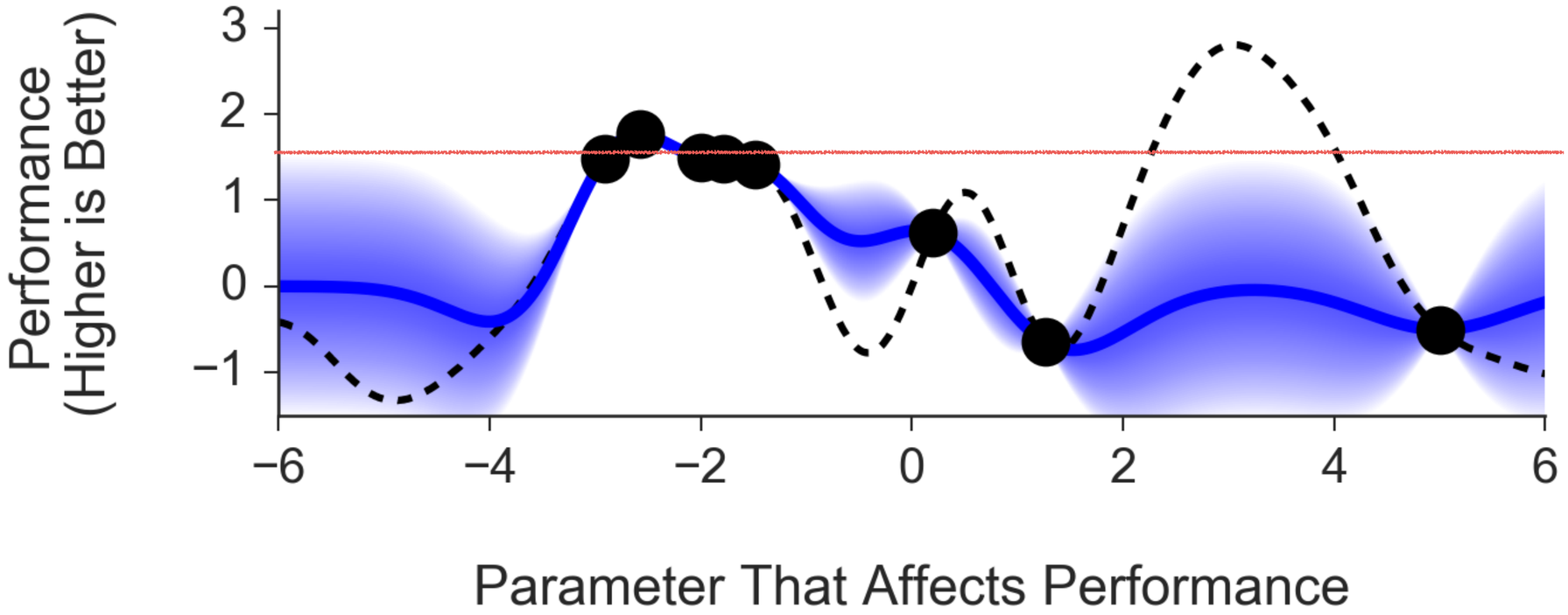


BAYESIAN OPTIMIZATION EXAMPLE

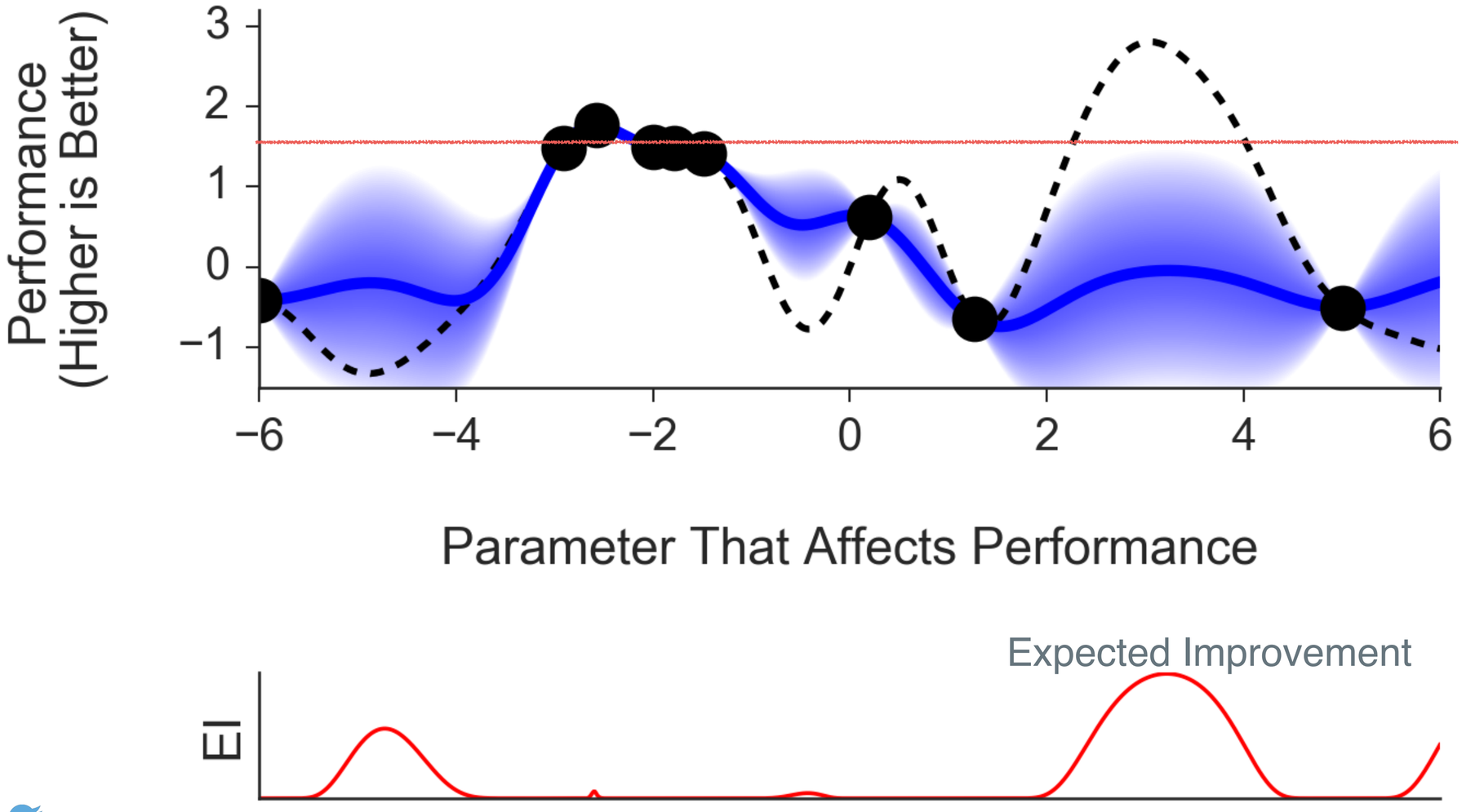




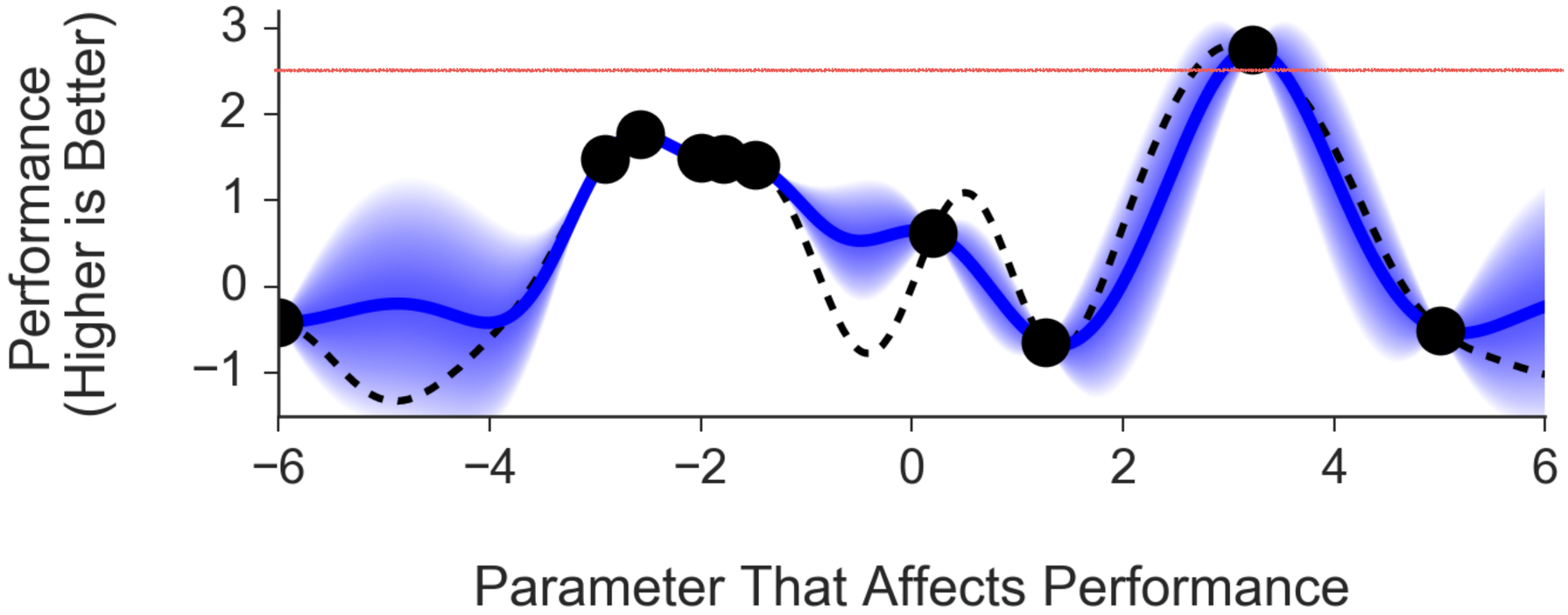
BAYESIAN OPTIMIZATION EXAMPLE



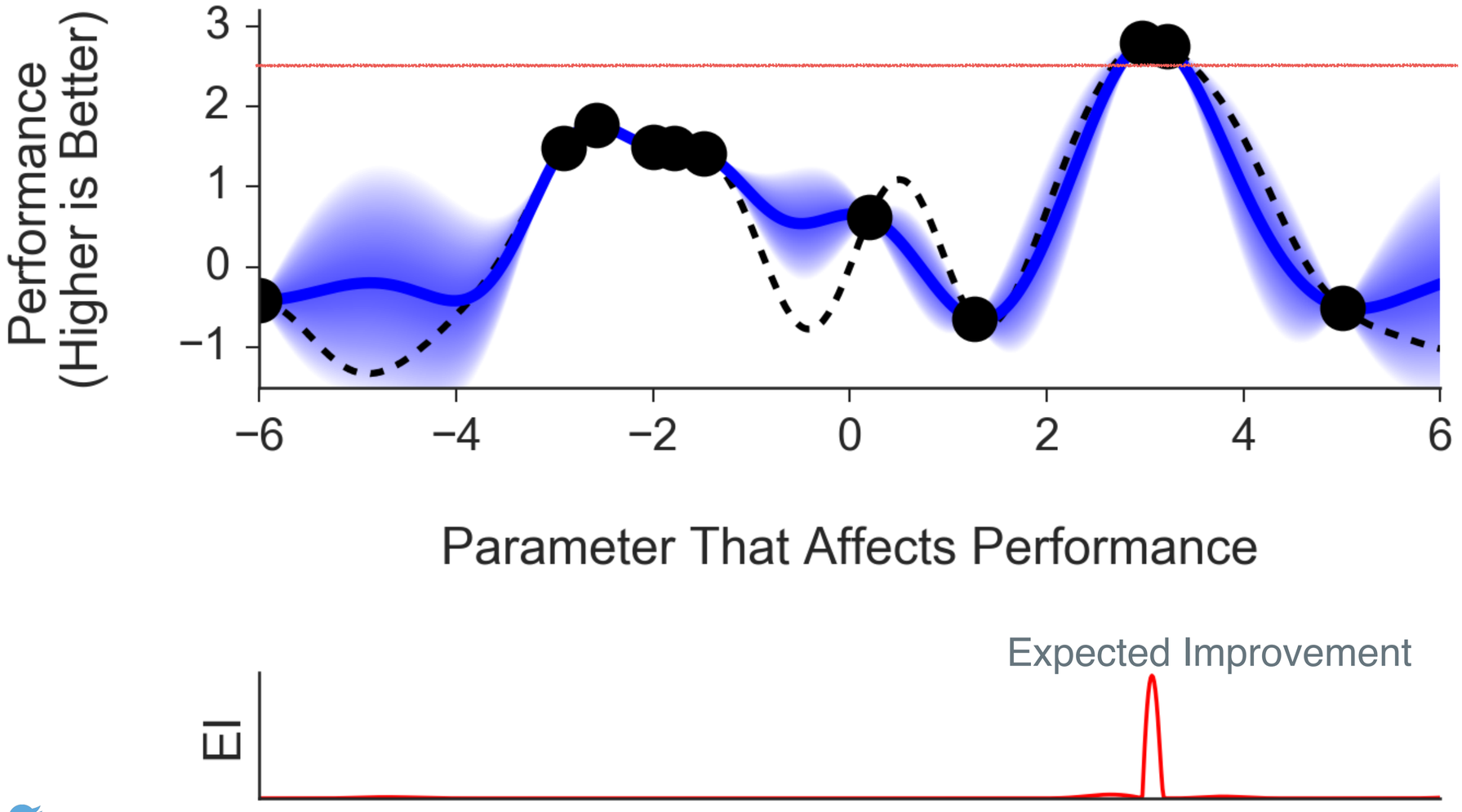
BAYESIAN OPTIMIZATION EXAMPLE



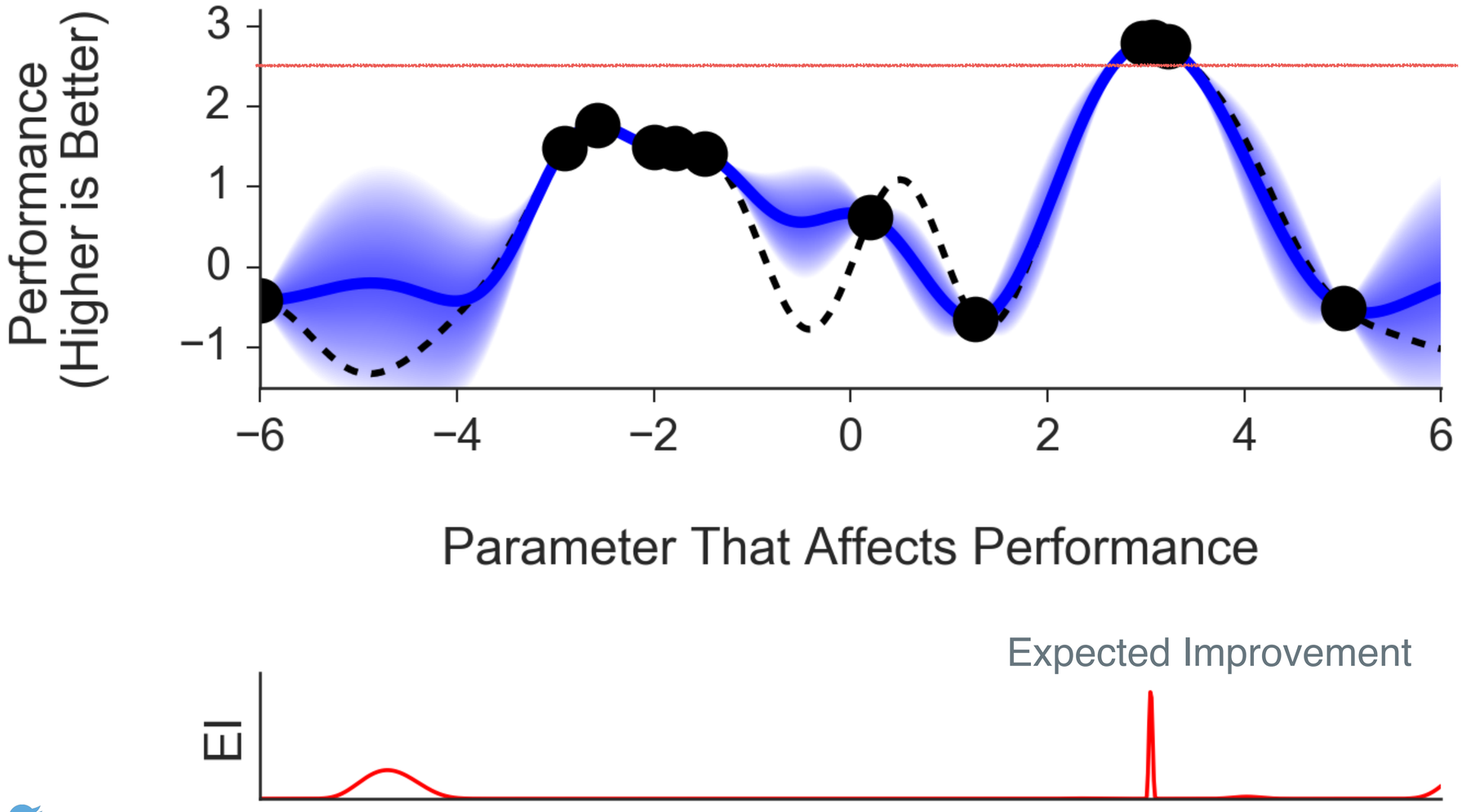
BAYESIAN OPTIMIZATION EXAMPLE



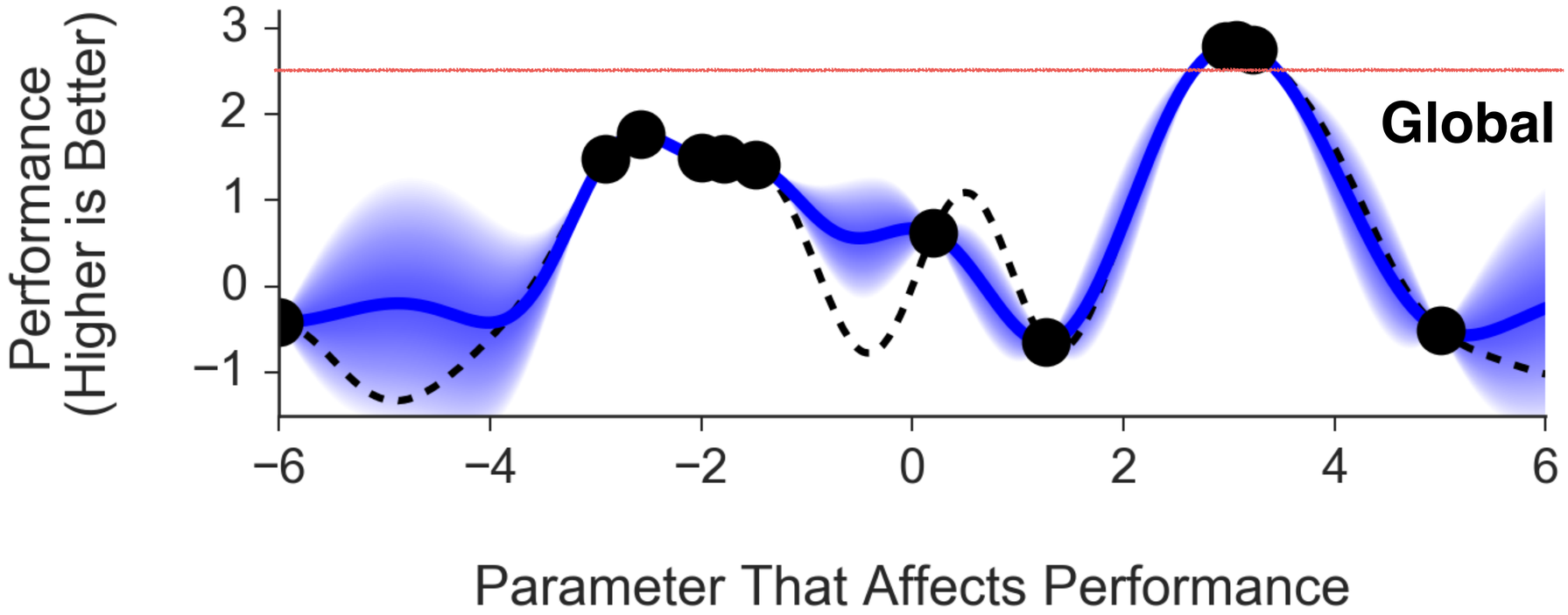
BAYESIAN OPTIMIZATION EXAMPLE



BAYESIAN OPTIMIZATION EXAMPLE



BAYESIAN OPTIMIZATION EXAMPLE



Global optimum discovered

WE LIKE BAYESIAN OPTIMIZATION…

- …because it is
  - Non-parametric
  - Robust
  - Extensible
  - Battle-tested on many types of real-world, high-impact problems

Experiments

WHAT IS AUTOTUNE?

- BOaaS
  - Bayesian Optimization-as-a-Service (Whetlab)
- Autotune
  - Driver for the experiments for explorations suggested by Bayesian Optimization

WHAT IS GRAAL?

- Java Virtual Machine Just-in-Time (JIT) compiler
- Actively developed by Oracle Labs
  - http://openjdk.java.net/projects/graal/
  - https://github.com/oracle/graal
- Uses JVMCI (JEP 243)
- Written in Java

LEARN MORE ABOUT GRAAL

Graal: How to use the new JVM JIT compiler in real life by Chris Thalinger

2,060 views

43 0 SHARE SAVE ...

Devoxx
Published on Mar 27, 2018

SUBSCRIBE 67K

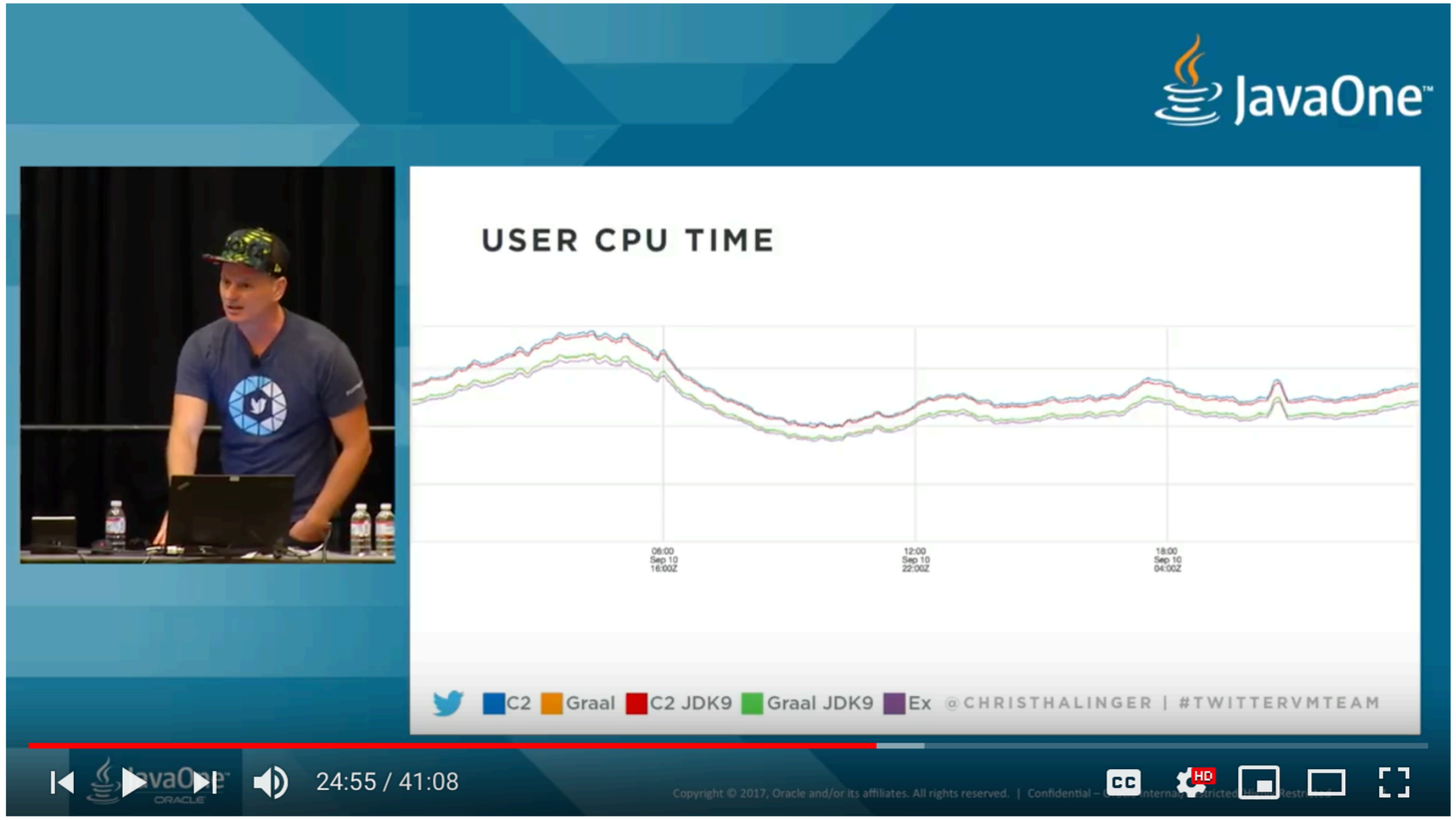
WHICH PARAMETERS?

```
graal.TrivialInliningSize = 10                                          [Integer]
          Graphs with less than this number of nodes are trivial and therefore
          always inlined.


graal.MaximumInliningSize = 300                                         [Integer]
          Inlining is explored up to this number of nodes in the graph for each
          call site.


graal.SmallCompiledLowLevelGraphSize = 300                              [Integer]
          If the previous low-level graph size of the method exceeds the
          threshold, it is not inlined.
```

PREVIOUS WORK



Twitter's Quest for a Wholly Graal Runtime

2,665 views

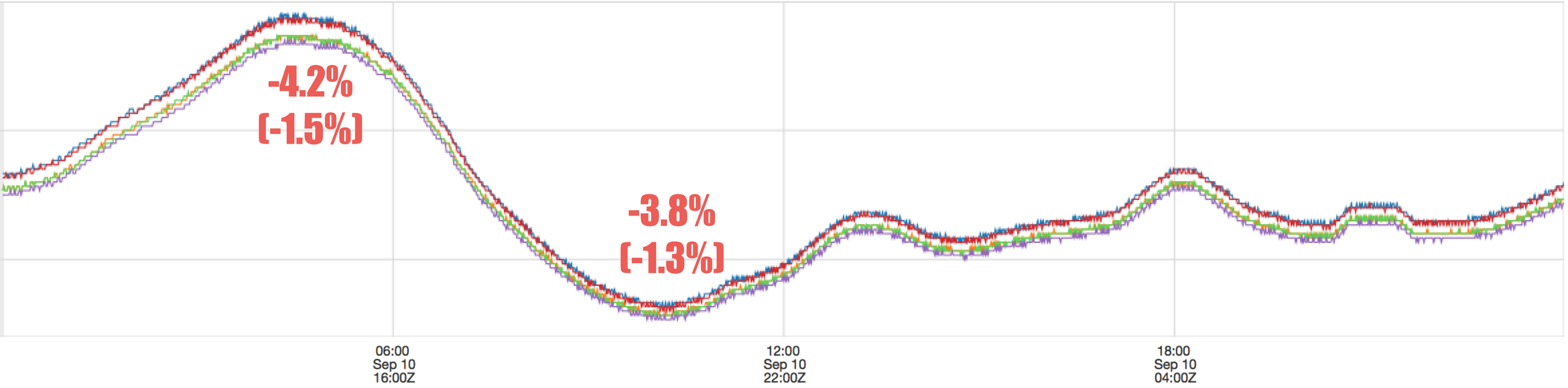
41 4 SHARE SAVE

Java
Published on Oct 2, 2017

SUBSCRIBE 79K

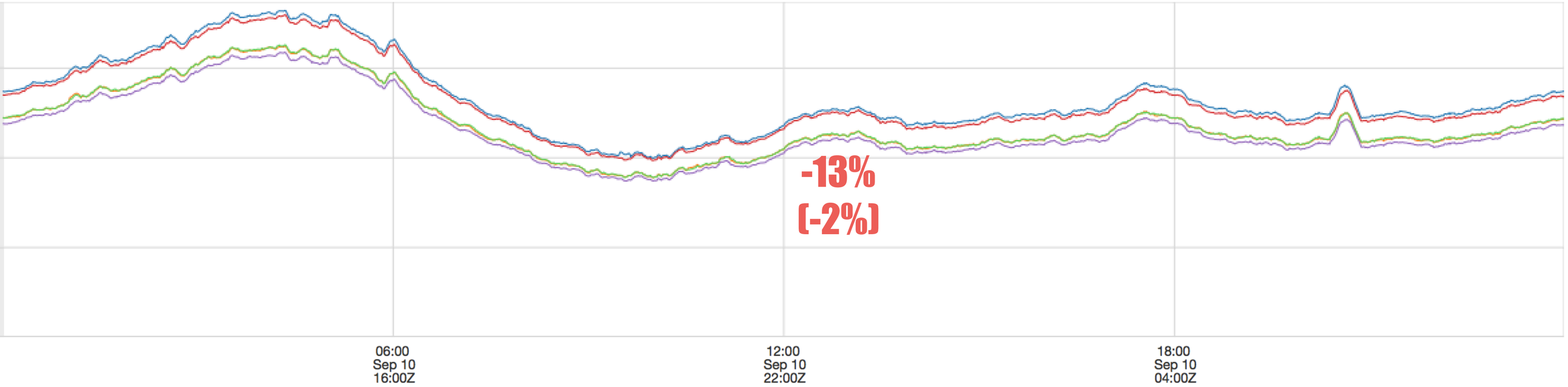
PS SCAVENGE CYCLES

movingavg(60)



C2 Graal C2 JDK9 raal JDK9 E

USER CPU TIME



C2 Graal C2 JDK9 raal JDK9 E

WHAT RANGES?

```
{
  "name" : "Dgraal.TrivialInliningSize",
  "whetlabName" : "GraalTrivialInliningSize",
  "parameterType": "integer",
  "min" : "10",
  "max" : "25"
},

{
  "name" : "Dgraal.MaximumInliningSize",
  "whetlabName" : "GraalMaximumInliningSize",
  "parameterType": "integer",
  "min" : "200",
  "max" : "500"
}
```

```
{
  "name" : "Dgraal.SmallCompiledLowLevelGraphSize",
  "whetlabName" : "GraalSmallCompiledLowLevelGraphSize",
  "parameterType": "integer",
  "min" : "200",
  "max" : "650"
},
```

TEST SETUP

- 1 dedicated machine per instance/configuration
  - All instances receive the exact same requests
- Run with `jvmci-0.30` and `graal-vm-0.22`
  - Default tiered C1/Graal setup

Experiment 1: Tweet Service

TWEET SERVICE

- Finagle-Thrift service
  - https://twitter.github.io/finagle/

Finagle is an extensible RPC system for the JVM, used to construct high-concurrency servers. Finagle implements uniform client and server APIs for several protocols, and is designed for high performance and concurrency. Most of Finagle's code is protocol agnostic, simplifying the implementation of new protocols.

TWEET SERVICE: OBJECTIVE

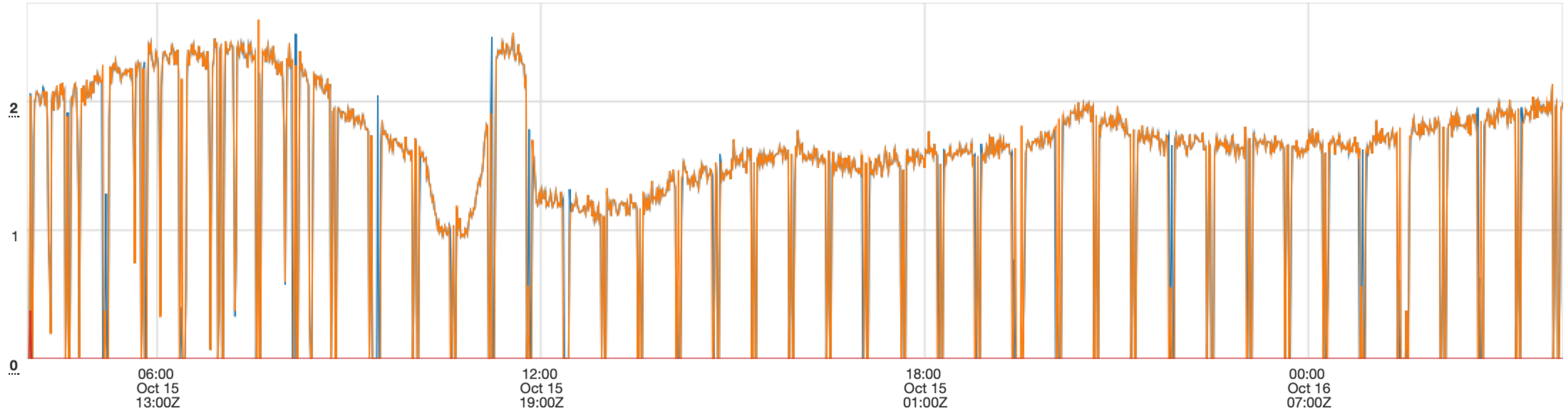
```
  "objectiveQuery" : "(1 / movingavg(10, avg(rate(ts(SUM, mesos.container, sd.cthalinger.devel.tweetypie-staging-basic-graal-jdk8.
{{instanceId}}, cpus_user_time_secs)) / 60)))",
```

TWEET SERVICE: CONSTRAINTS

```
  "constraintQueries" : [
    {
      "query" : "(100 * rate(sum(ts(SUM, mesos.container, sd.cthalinger.devel.tweetypie-staging-basic-graal-jdk8.{{instanceId}},
cpus_nr_throttled)))) / rate(sum(ts(SUM, mesos.container, sd.cthalinger.devel.tweetypie-staging-basic-graal-jdk8.{{instanceId}},
cpus_nr_periods)))",
      "isMax" : "true",
      "threshold" : "10.0",
      "period" : "1.minutes"
    }
  ],
```

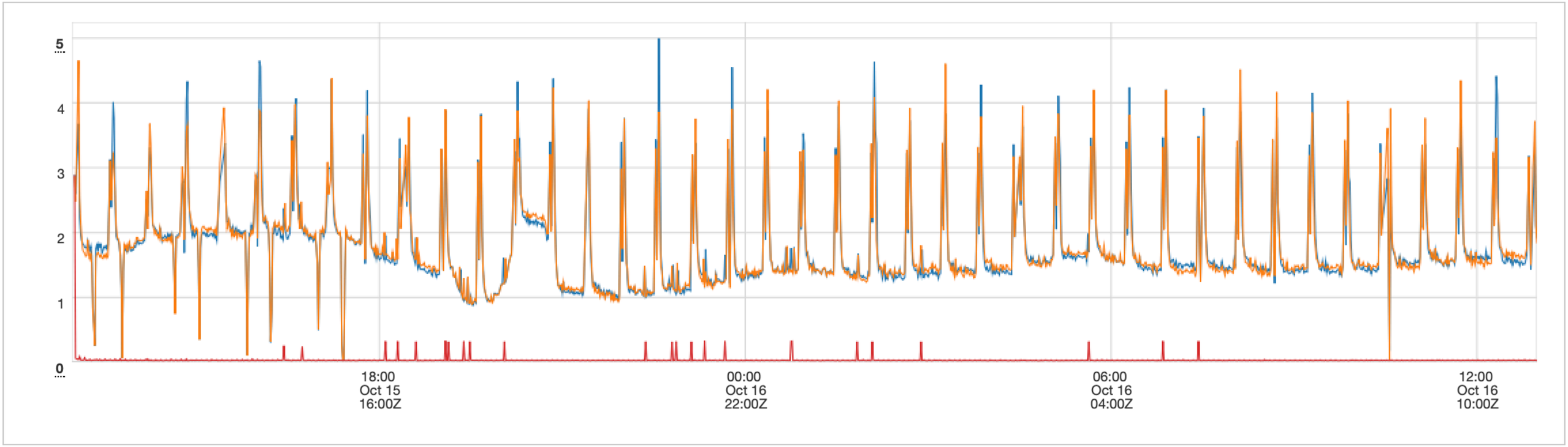
TWEET SERVICE: REQUESTS/SEC

24 hours



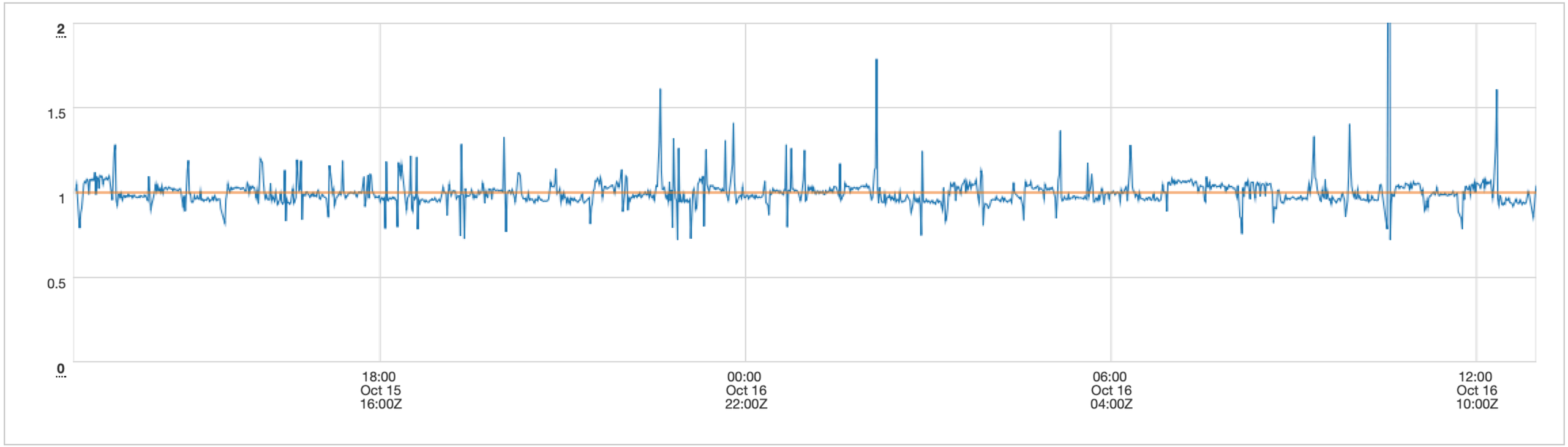
experiment ontrol

TWEET SERVICE: USER CPU TIME



experiment ontrol

TWEET SERVICE: USER CPU TIME - RATIO



experiment ontrol

TWEET SERVICE: AUTOTUNE RESULTS

44 results, 3 parameters. View description

Table | Charts

Best | Completed | Constraint Violated | In Progress

Add Result | Select all | Deselect all

Status	Result ID	Objective ↑	AutotuneOutcome	GraalTrivialInliningSize	GraalMaximumInliningSize	GraalSmallCompil
●	315930	1.0838		21	486	590
●	315961	1.0813		15	476	576
●	315986	1.0638		21	284	597
●	315976	1.0636		22	265	456
●	315985	1.0584		17	359	485
●	315968	1.0566		24	232	379

TWEET SERVICE: AUTOTUNE RESULTS

44 results, 3 parameters. View description

Table | Charts

Best | Completed | Constraint Violated | In Progress

Add Result | Select all | Deselect all

Status	Result ID	Objective ↑	AutotuneOutcome	GraalTrivialInliningSize	GraalMaximumInliningSize	GraalSmallCompil
●	315930	1.0838		21	486	590
●	315961	1.0813		15	476	576
●	315986	1.0638		21	284	597
●	315976	1.0636		22	265	456
●	315985	1.0584		17	359	485
●	315968	1.0566		24	232	379

(10-25) (200-500) (200-650)

TWEET SERVICE: AUTOTUNE RESULTS

44 results, 3 parameters. View description

Table Charts

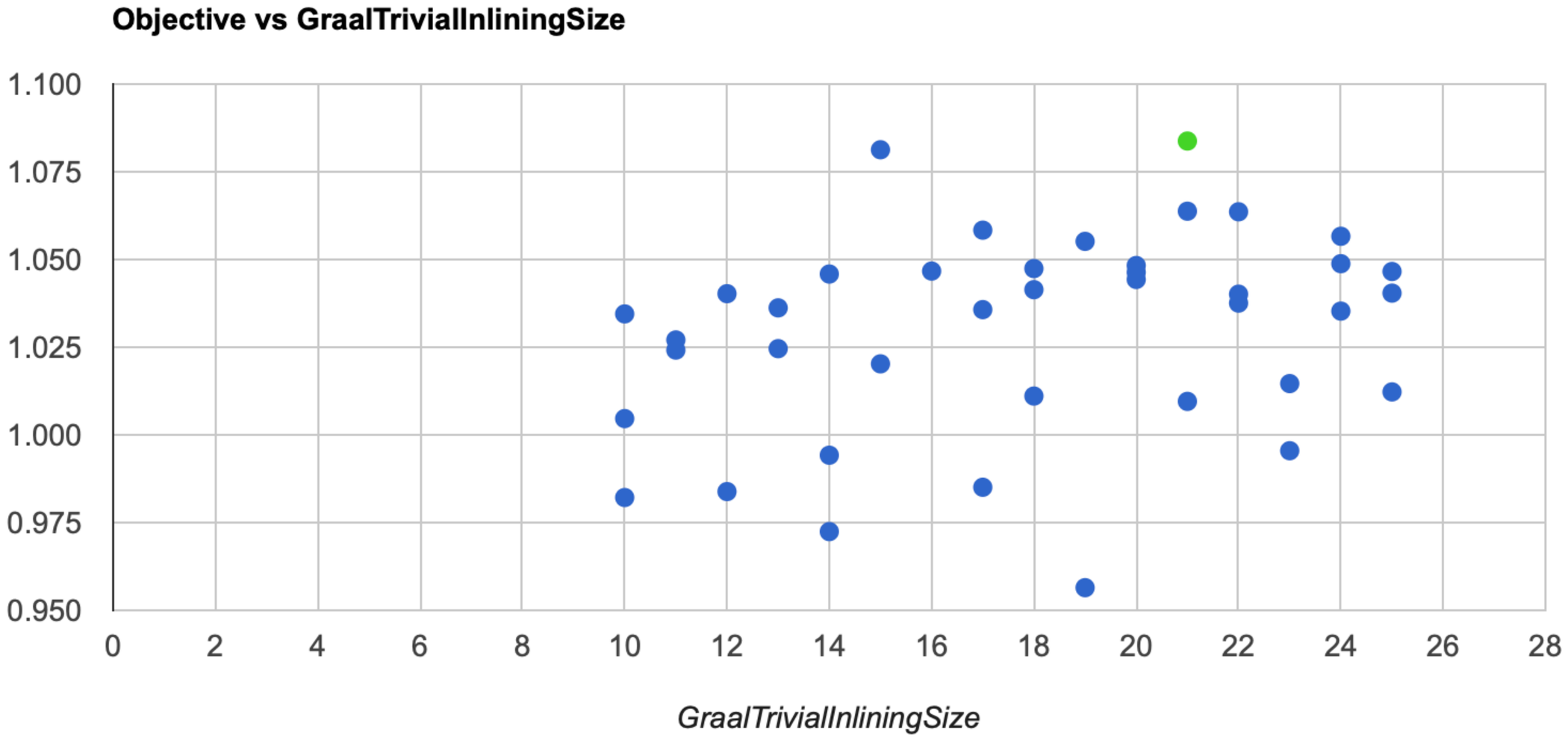
Best Completed Constraint Violated In Progress

Add Result | Select all | Deselect all

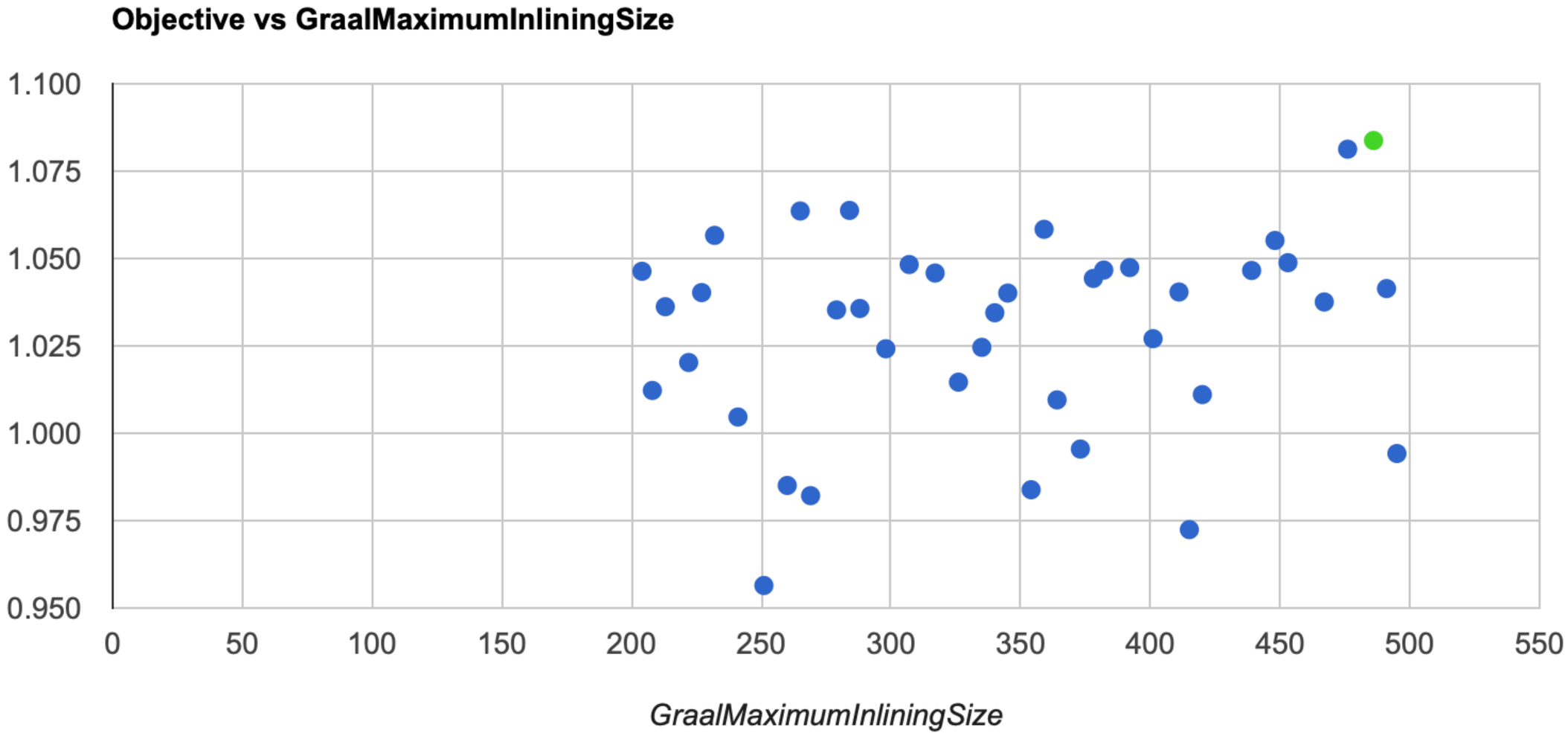
Status	Result ID	Objective ↓	AutotuneOutcome	GraalTrivialInliningSize	GraalMaximumInliningSize	GraalSmallCompiledLowLevel
●	315982			16	302	287
●	315987			13	434	372
●	315971			12	458	266
●	315947			16	429	337
●	315962	0.95645		19	251	238
●	315977	0.97244		14	415	231

(10-25) (200-500) (200-650)

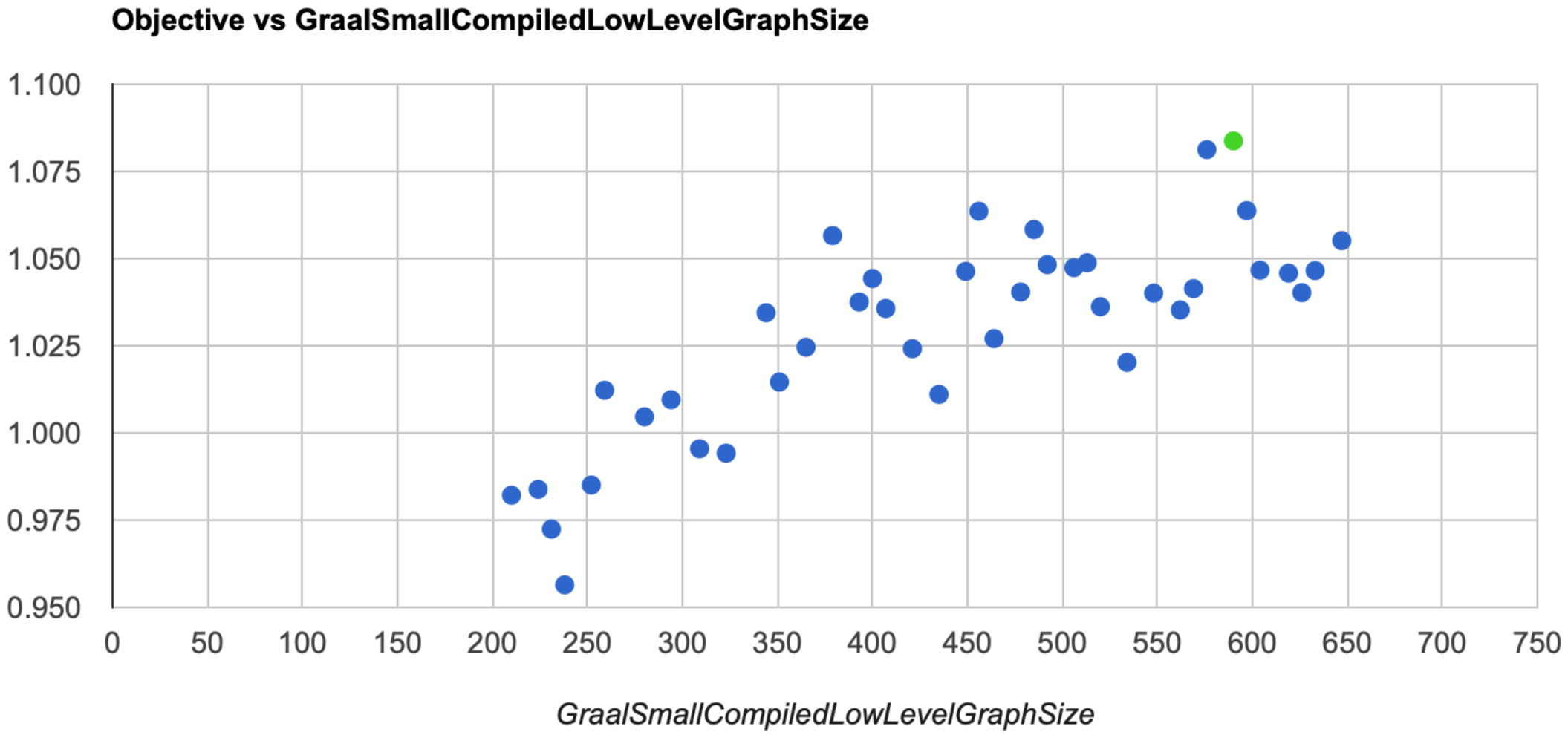
TWEET SERVICE: AUTOTUNE - CHARTS



TWEET SERVICE: AUTOTUNE - CHARTS

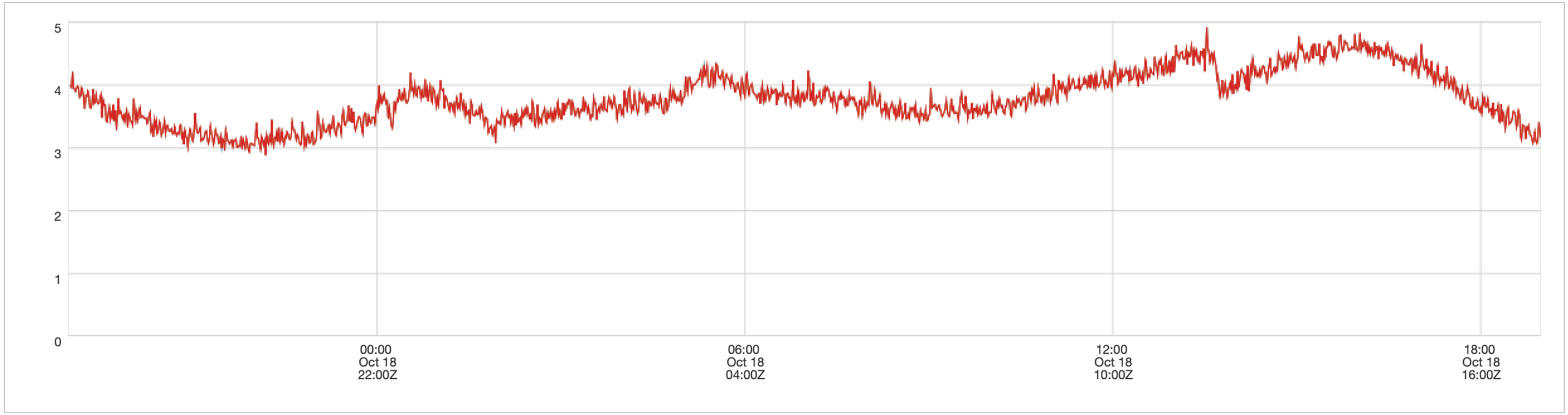


TWEET SERVICE: AUTOTUNE - CHARTS



TWEET SERVICE: REQUESTS/SEC

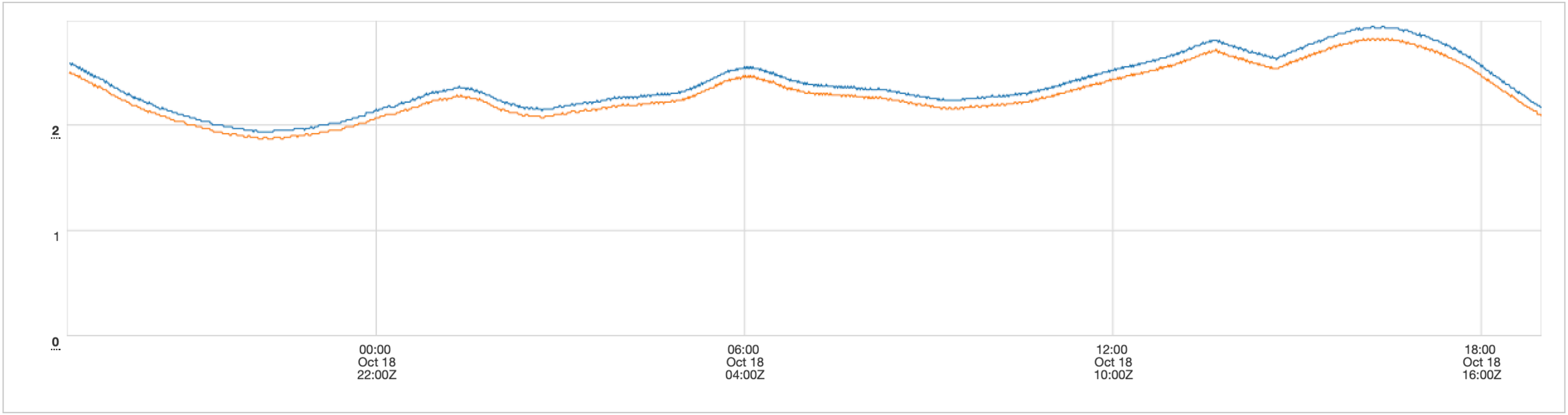
24 hours



C2 Graal Autotune

TWEET SERVICE: PS SCAVENGE CYCLES

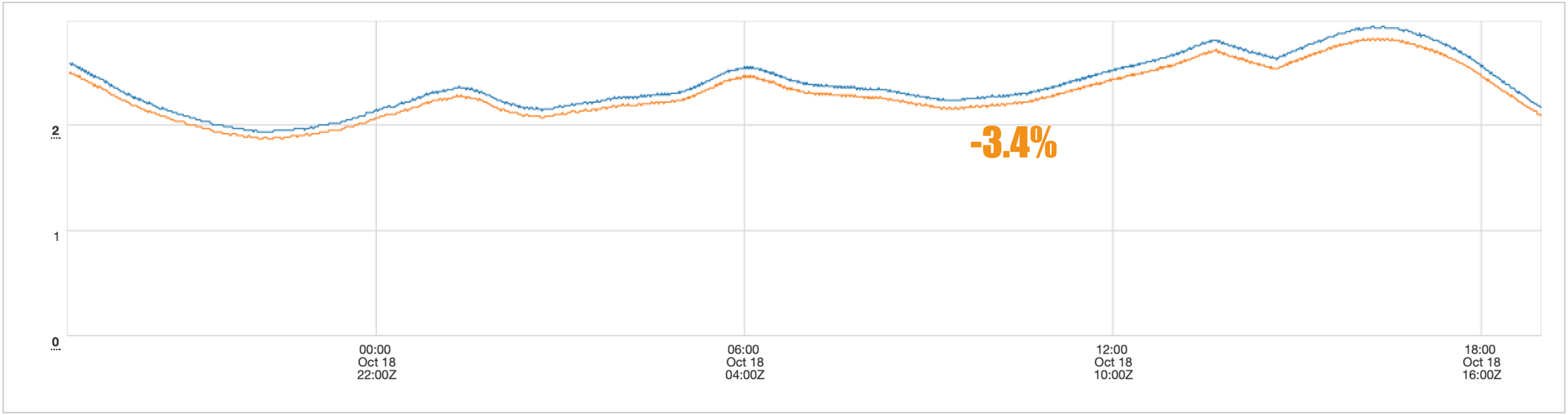
movingavg(60)



C2 Graal

TWEET SERVICE: PS SCAVENGE CYCLES

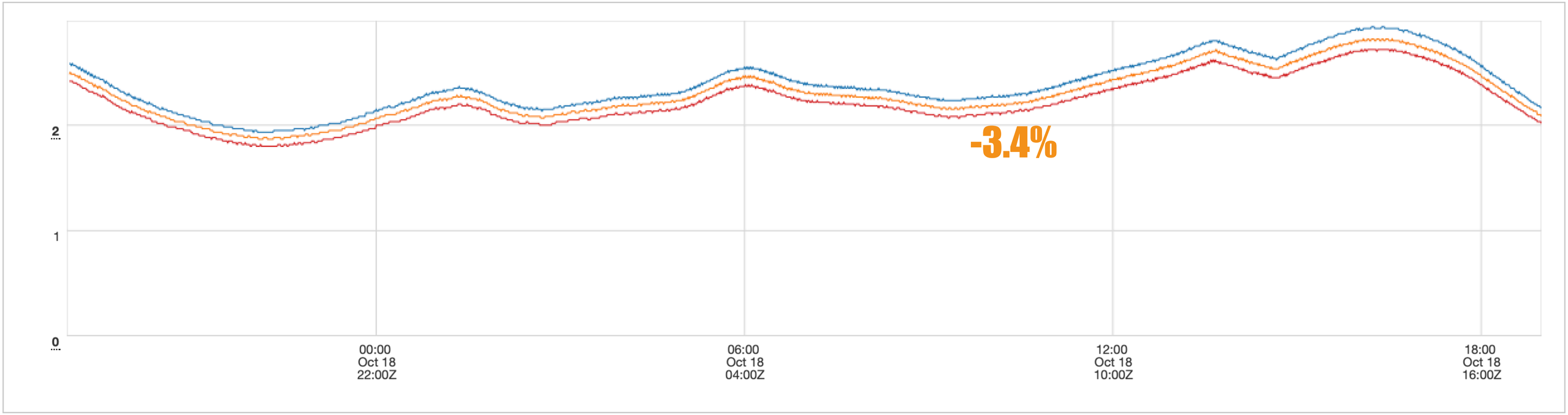
movingavg(60)



C2 Graal

TWEET SERVICE: PS SCAVENGE CYCLES

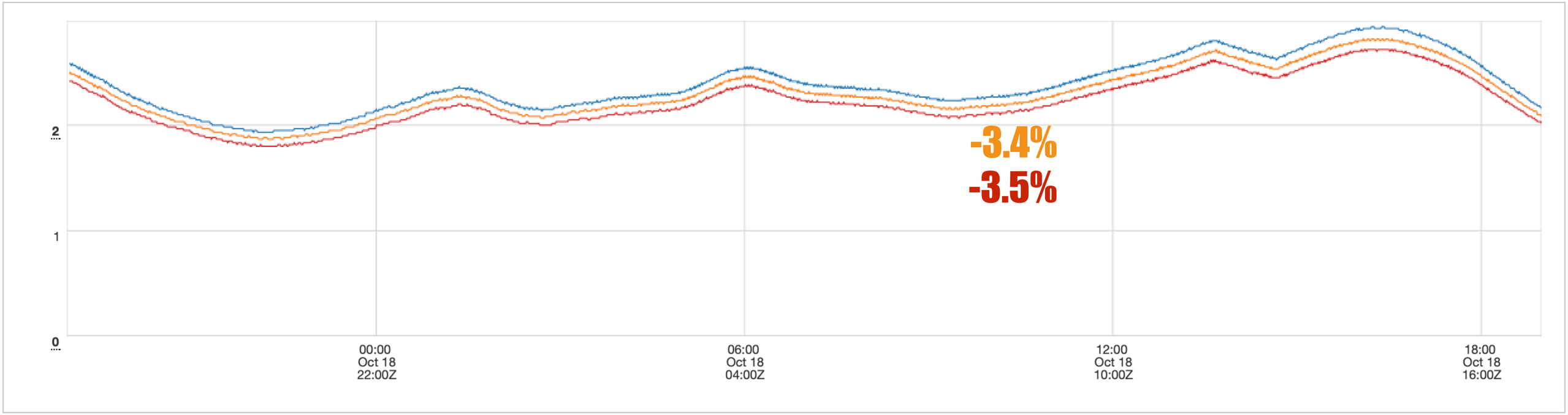
movingavg(60)



C2 Graal Autotune

TWEET SERVICE: PS SCAVENGE CYCLES

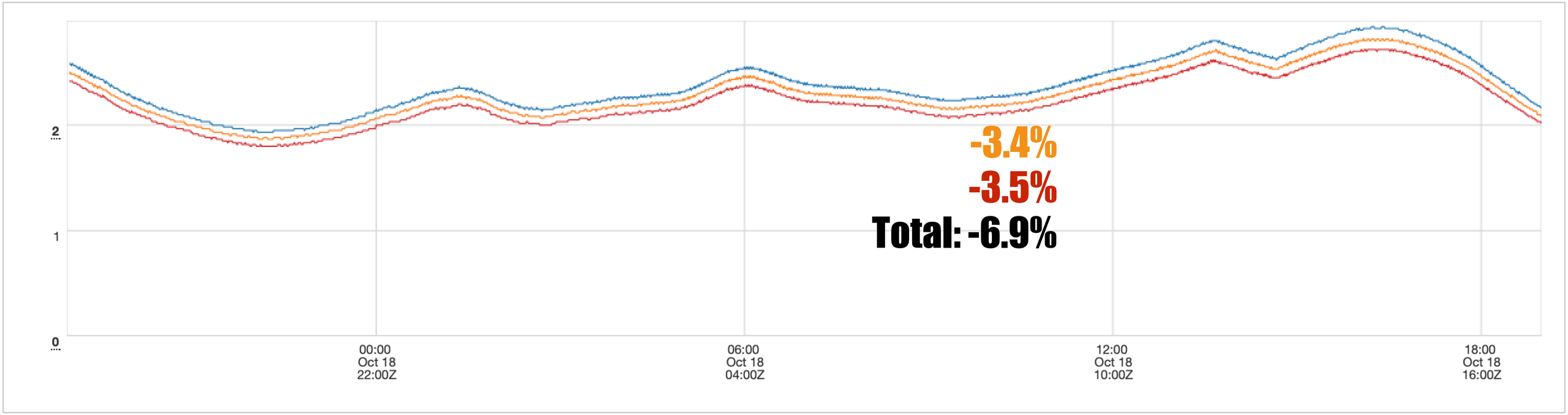
movingavg(60)



C2 Graal Autotune

TWEET SERVICE: PS SCAVENGE CYCLES

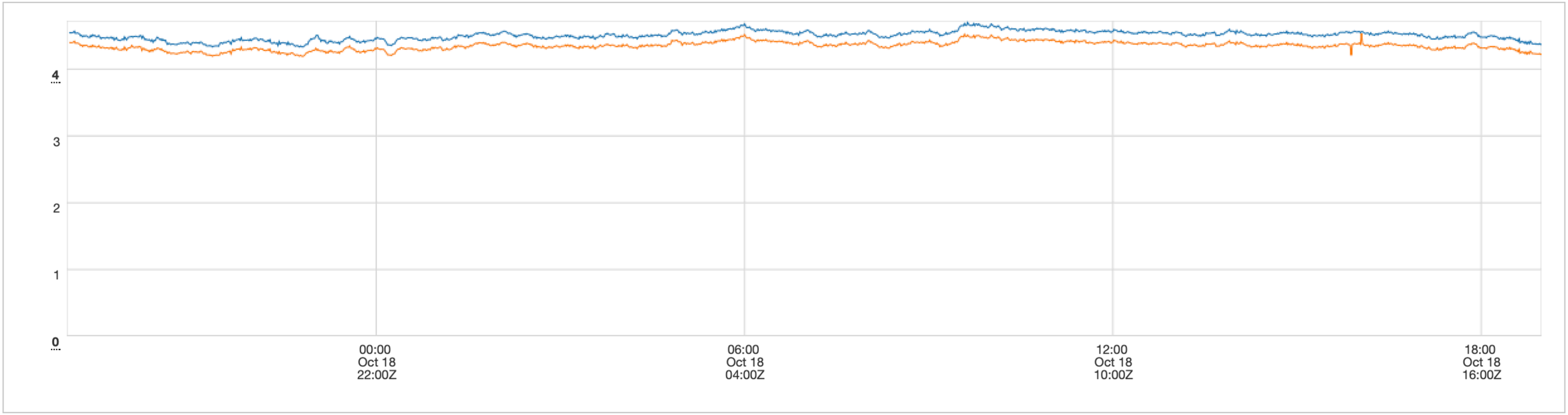
movingavg(60)



C2 Graal Autotune

TWEET SERVICE: ALLOCATED BYTES/TWEET

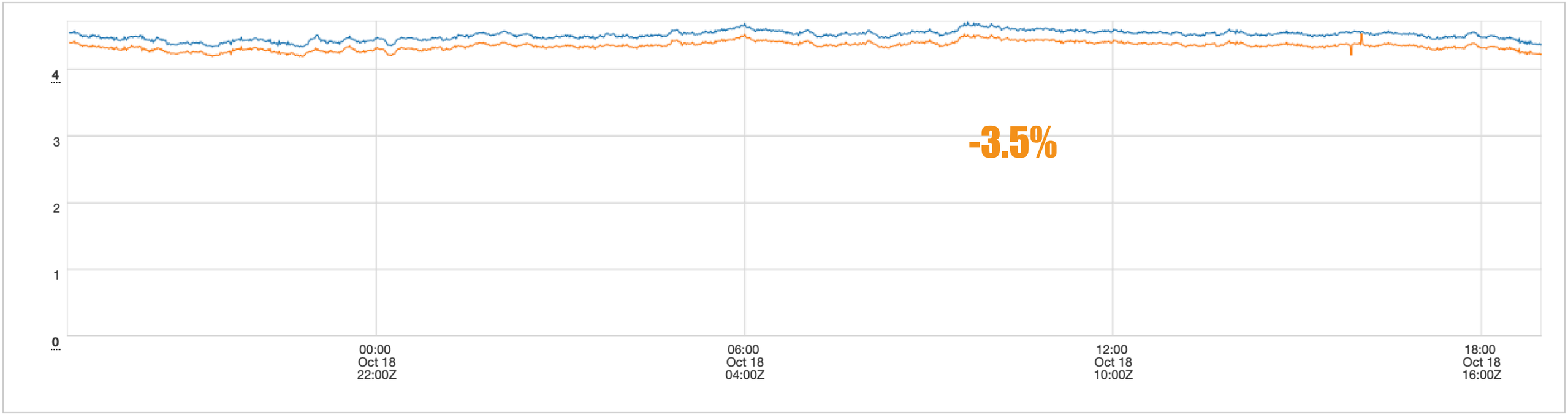
movingavg(10)



C2 Graal

TWEET SERVICE: ALLOCATED BYTES/TWEET

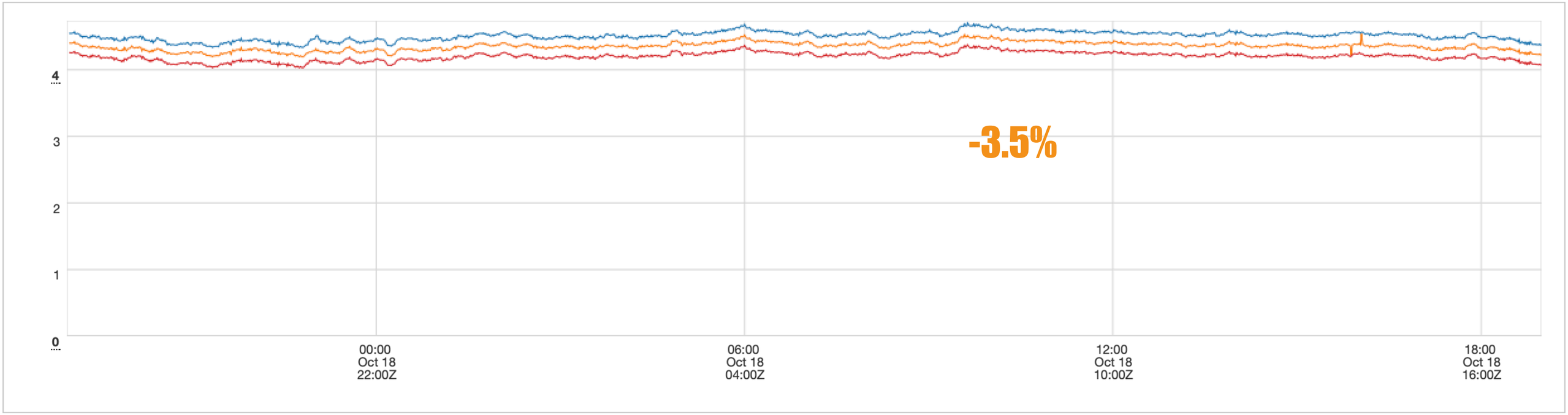
movingavg(10)



C2 Graal

TWEET SERVICE: ALLOCATED BYTES/TWEET

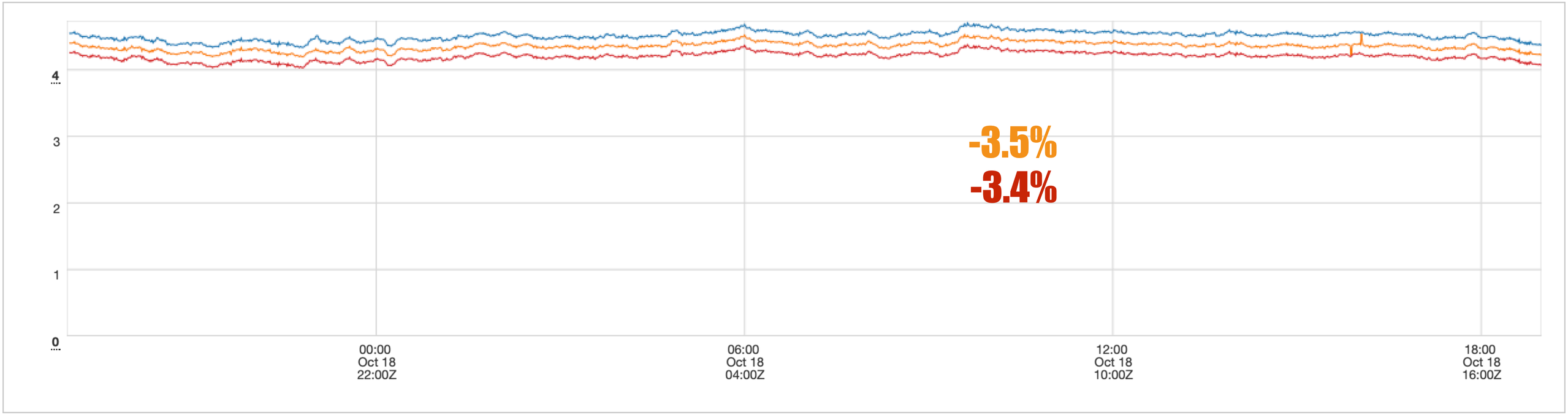
movingavg(10)



C2 Graal Autotune

TWEET SERVICE: ALLOCATED BYTES/TWEET

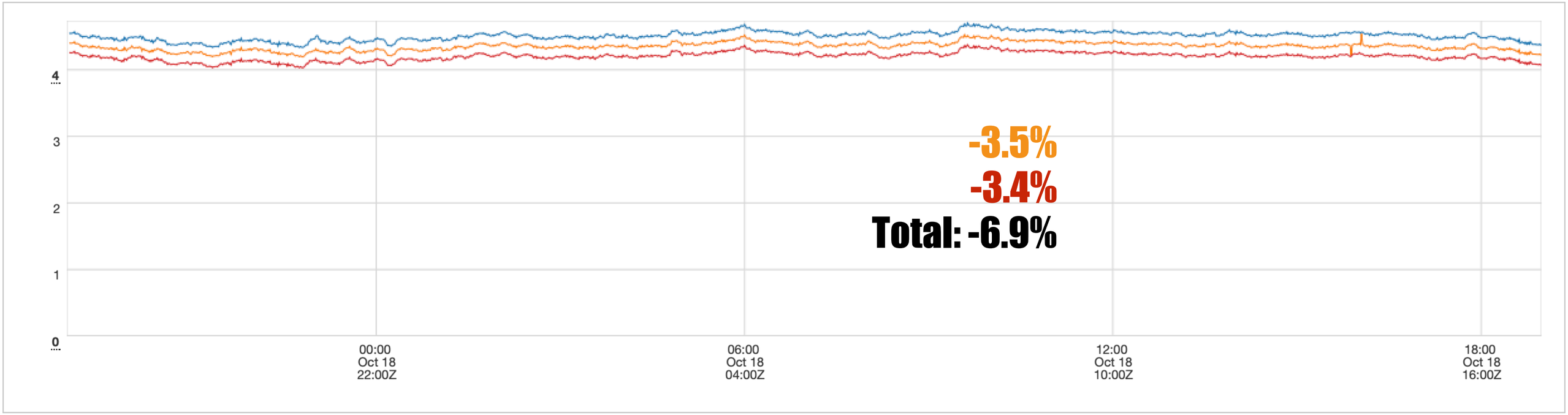
movingavg(10)



C2 Graal Autotune

TWEET SERVICE: ALLOCATED BYTES/TWEET

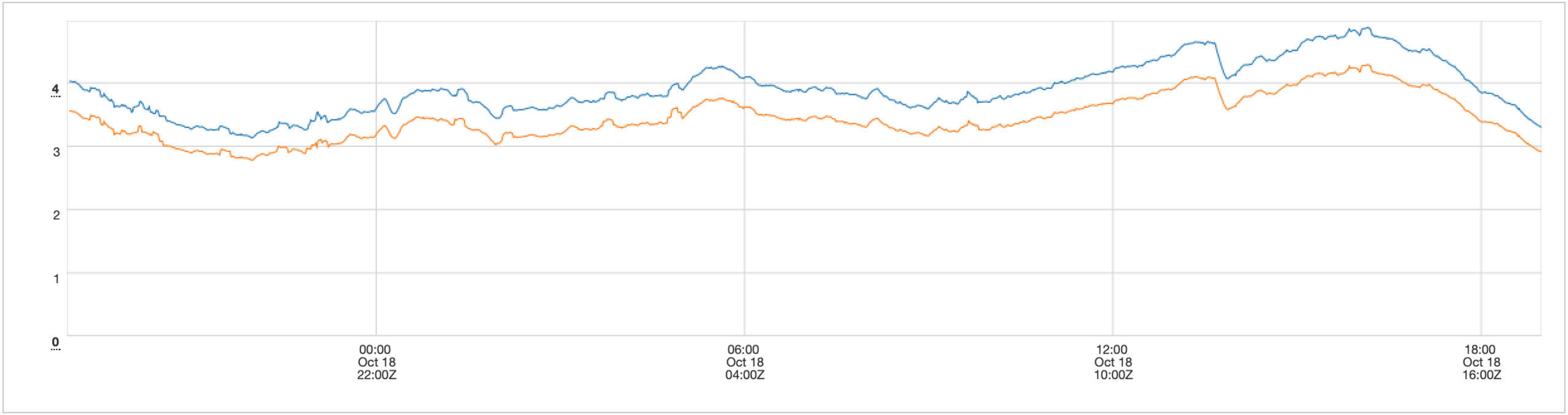
movingavg(10)



C2 Graal Autotune

TWEET SERVICE: USER CPU TIME

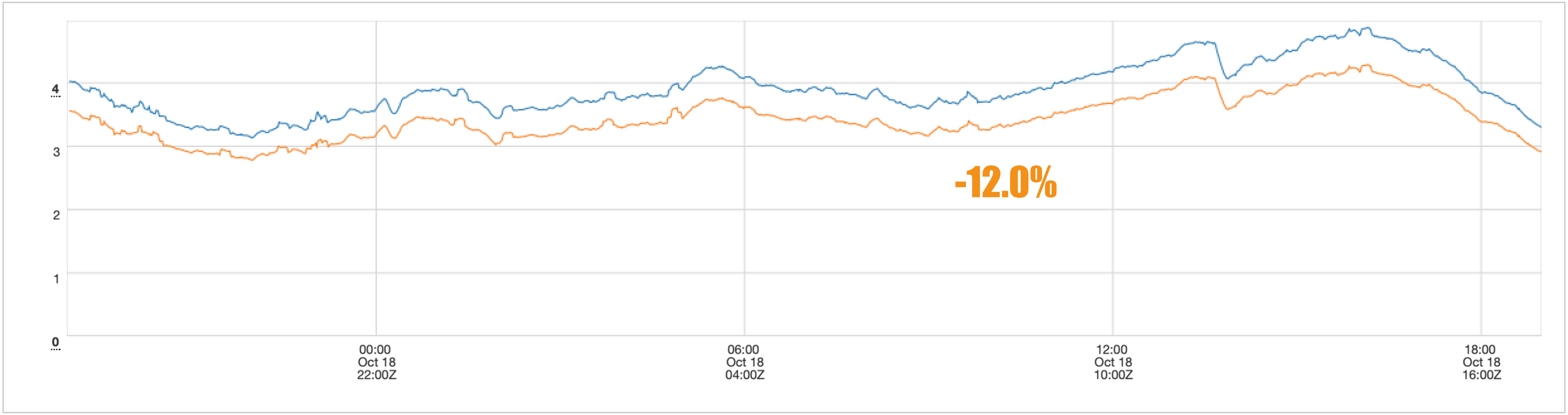
movingavg(10)



C2 Graal

TWEET SERVICE: USER CPU TIME

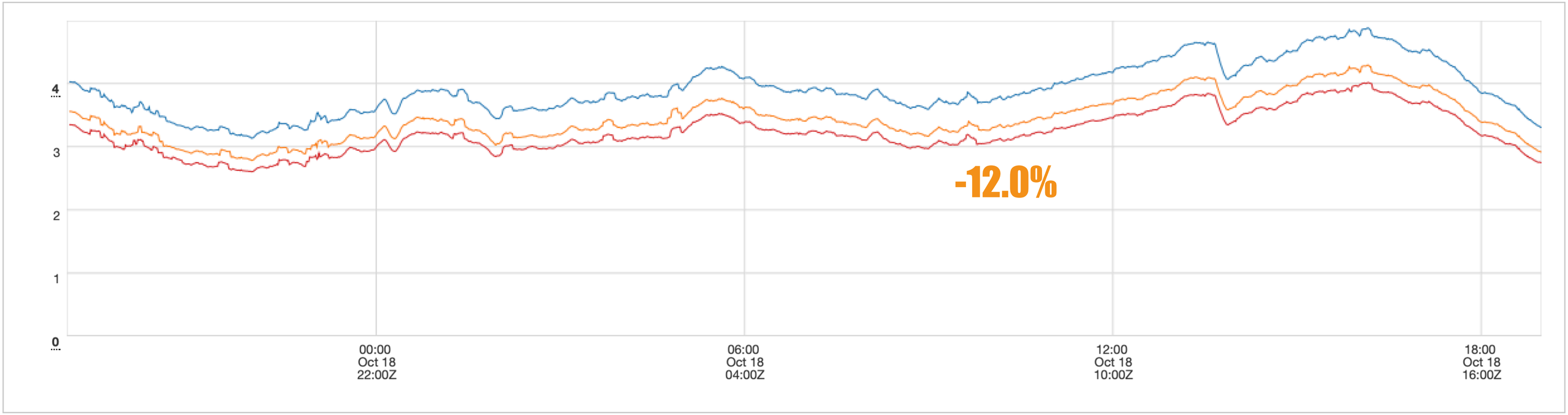
movingavg(10)



C2 Graal

TWEET SERVICE: USER CPU TIME

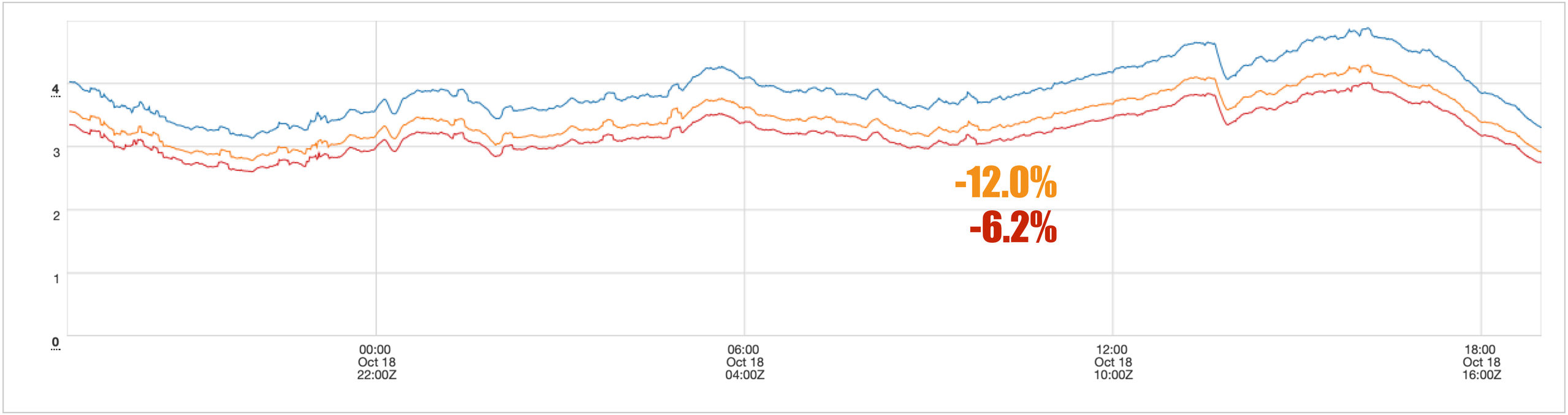
movingavg(10)



C2 Graal Autotune

TWEET SERVICE: USER CPU TIME

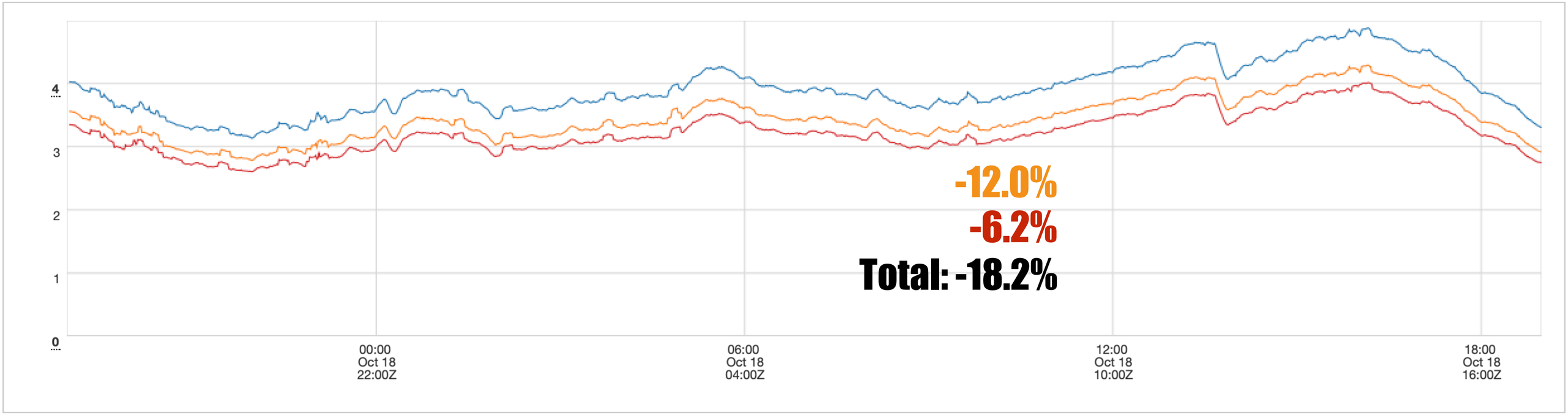
movingavg(10)



C2 Graal Autotune

TWEET SERVICE: USER CPU TIME

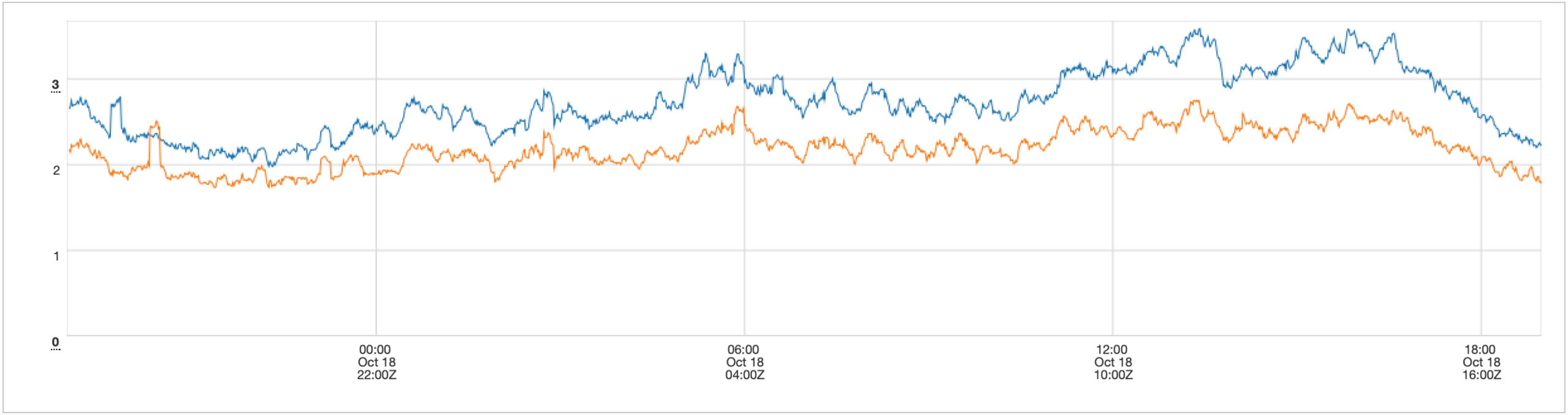
movingavg(10)



C2 Graal Autotune

TWEET SERVICE: LATENCY P99

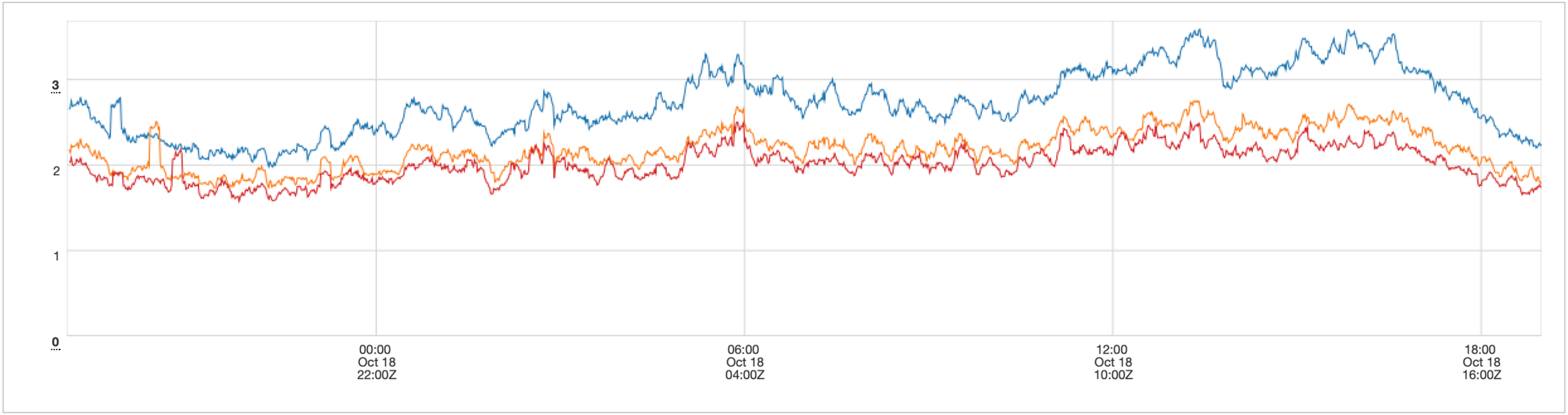
movingavg(10)



C2 Graal

TWEET SERVICE: LATENCY P99

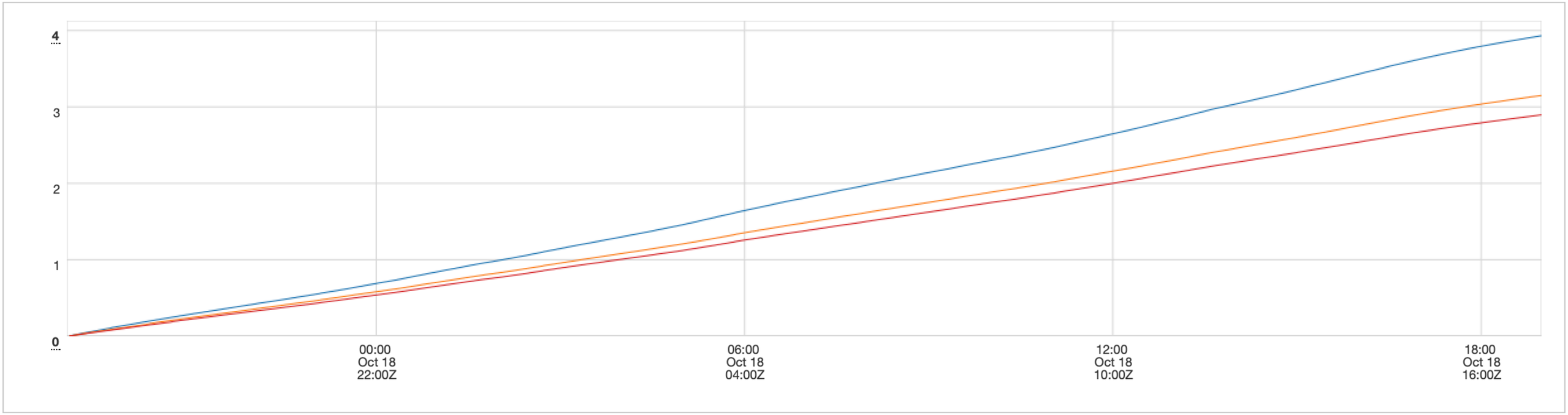
movingavg(10)



C2 Graal Autotune

TWEET SERVICE: LATENCY P99

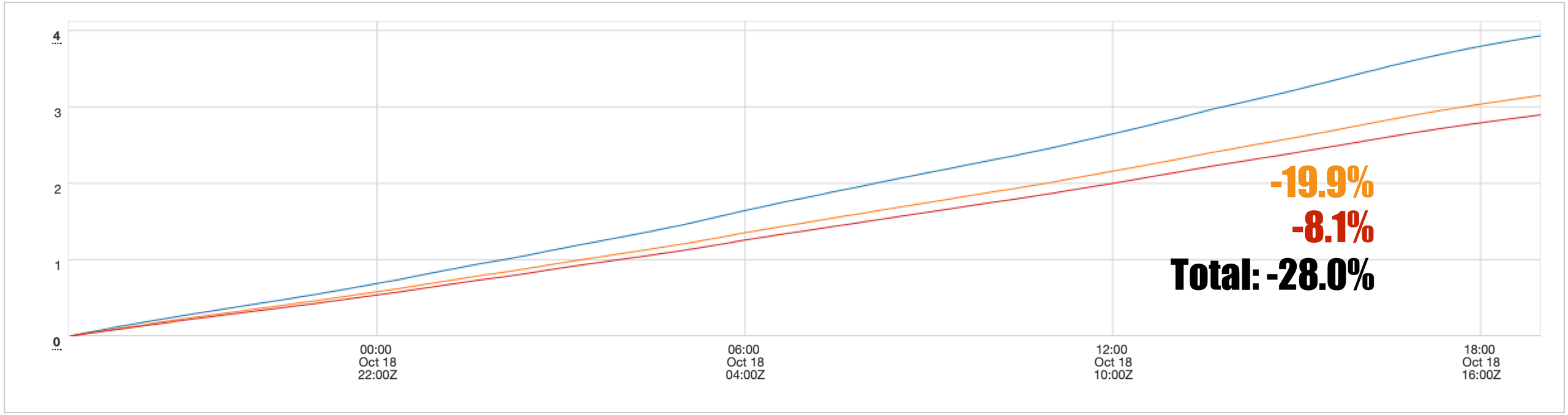
integrate()



C2 Graal Autotune

TWEET SERVICE: LATENCY P99

integrate()



C2 Graal Autotune

Experiment 2: Social Graph

SOCIAL GRAPH

- Finagle-Thrift service

Mission

Provide a centralized and useful abstraction for managing many-to-many relationships at Twitter.

SOCIAL GRAPH: OBJECTIVE

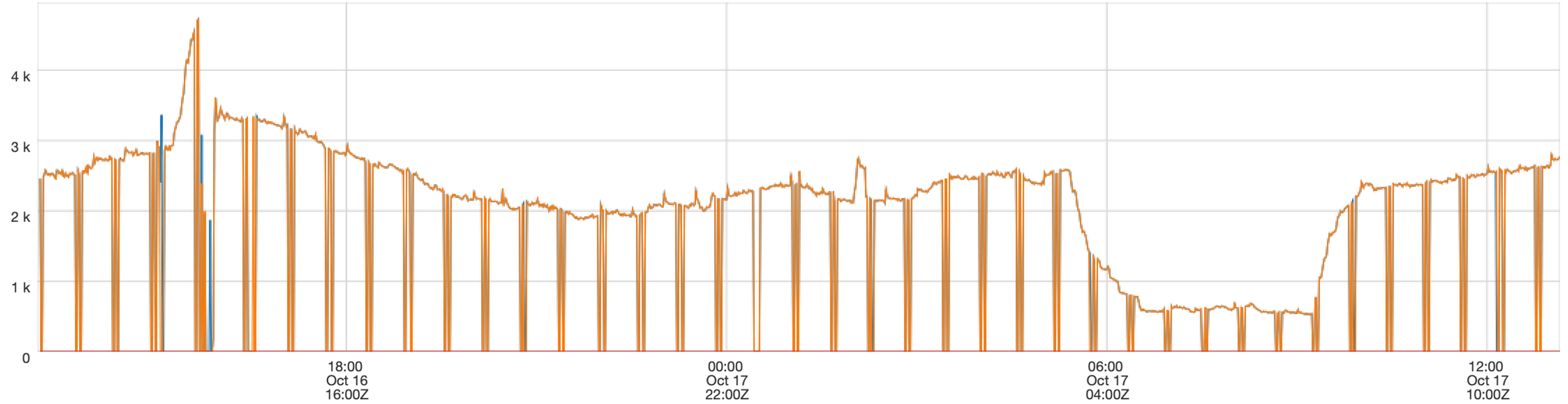
```
  "objectiveQuery" : "(1 / movingavg(10, avg(rate(ts(SUM, mesos.container, sd.cthalinger.devel.socialgraph-basic-graal-cms-jdk8.{{instanceId}}, cpu
s_user_time_secs)) / 60)))",
```

SOCIAL GRAPH: CONSTRAINTS

```
  "constraintQueries" : [
    {
      "query" : "(100 * rate(sum(ts(SUM, mesos.container, sd.cthalinger.devel.sd.cthalinger.devel.socialgraph-basic-graal-cms-jdk8.{{instanceId}},
cpus_nr_throttled)))) / rate(sum(ts(SUM, mesos.container, sd.cthalinger.devel.sd.cthalinger.devel.socialgraph-basic-graal-cms-jdk8.{{instanceId}},
cpus_nr_periods)))",
      "isMax" : "true",
      "threshold" : "10.0",
      "period" : "1"
    }
```

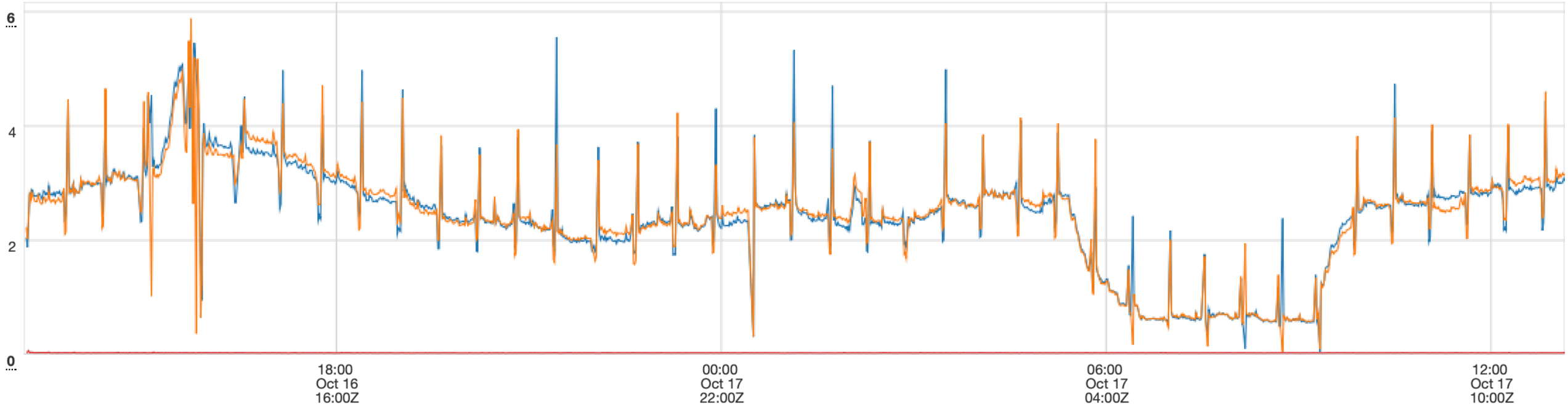
SOCIAL GRAPH: REQUESTS/SEC

24 hours



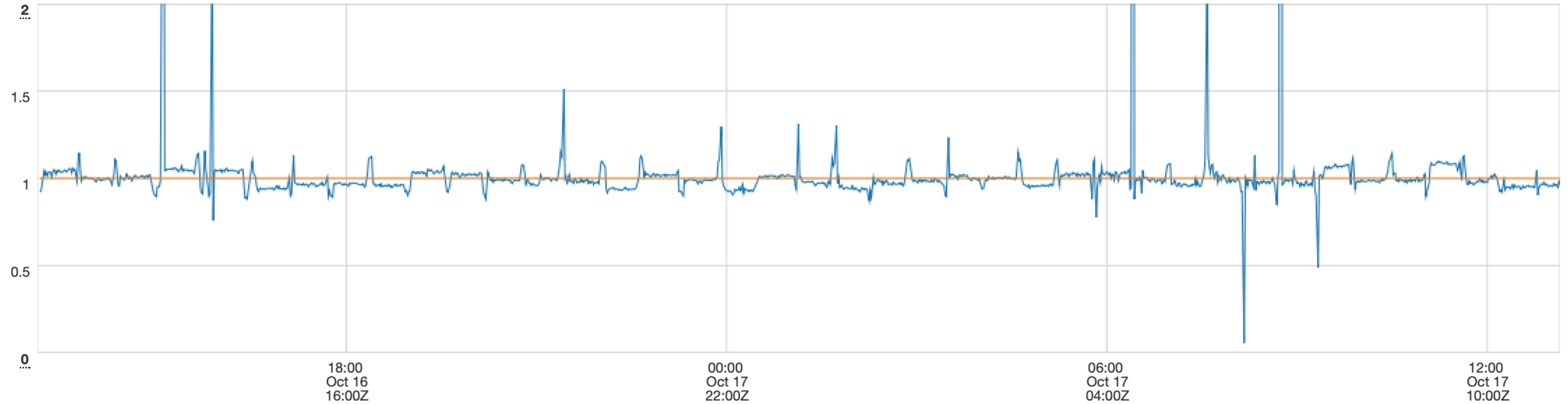
experiment ontrol

SOCIAL GRAPH: USER CPU TIME



experiment ontrol

SOCIAL GRAPH: USER CPU TIME - RATIO



experiment ontrol

SOCIAL GRAPH: AUTOTUNE RESULTS

44 results, 3 parameters. View description

Table | Charts

Best | Completed | Constraint Violated | In Progress

Add Result | Select all | Deselect all

Status	Result ID	Objective ↑	AutotuneOutcome	GraalTrivialInliningSize	GraalMaximumInliningSize	GraalSmallCompiledLowLevelG
●	316007	1.0759		23	398	646
●	316026	1.0757		12	299	469
●	316023	1.0719		22	337	526
●	316010	1.0684		13	285	589
●	316004	1.0642		19	323	533
●	315988	1.0468		17	483	321

(10-25) (200-500) (200-650)

SOCIAL GRAPH: AUTOTUNE RESULTS

44 results, 3 parameters. View description

Table | Charts

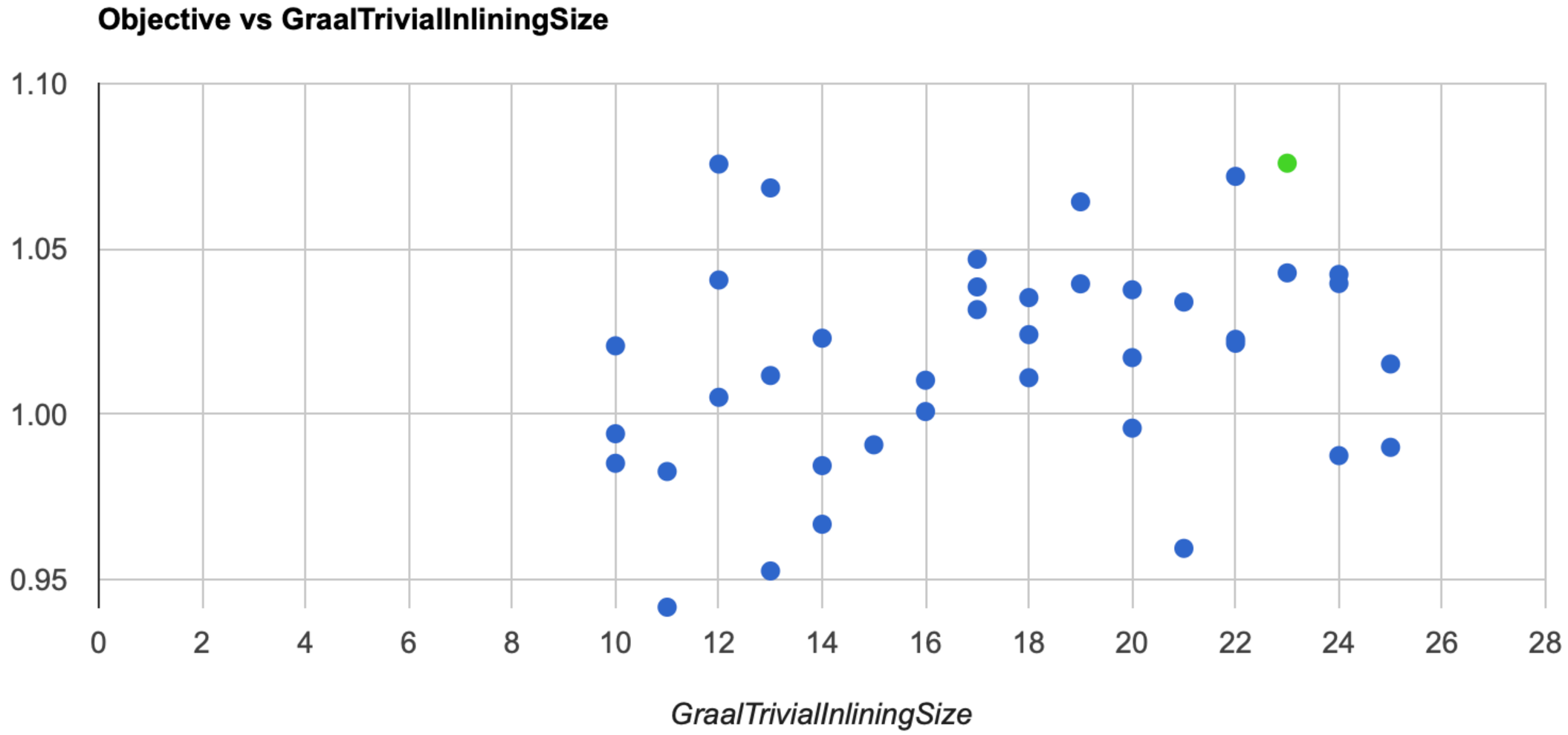
Best | Completed | Constraint Violated | In Progress

Add Result | Select all | Deselect all

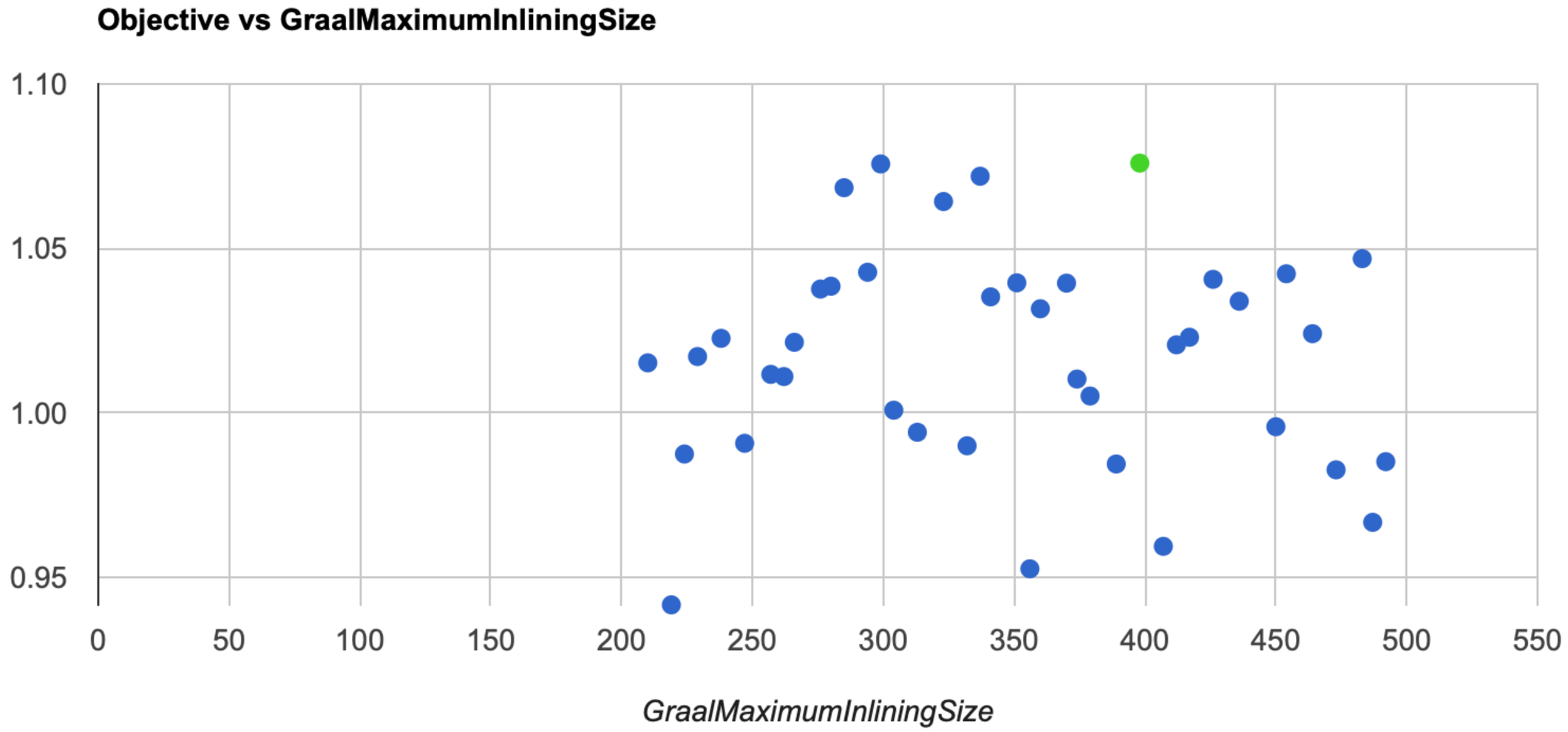
Status	Result ID	Objective ↓	AutotuneOutcome	GraalTrivialInliningSize	GraalMaximumInliningSize	GraalSmallCompiledLowLevelG
●	315993			15	445	378
●	315998			16	200	631
●	316031			25	431	554
●	316028			21	205	441
●	315994	0.94146		11	219	265
●	316029	0.95243		13	356	216

(10-25) (200-500) (200-650)

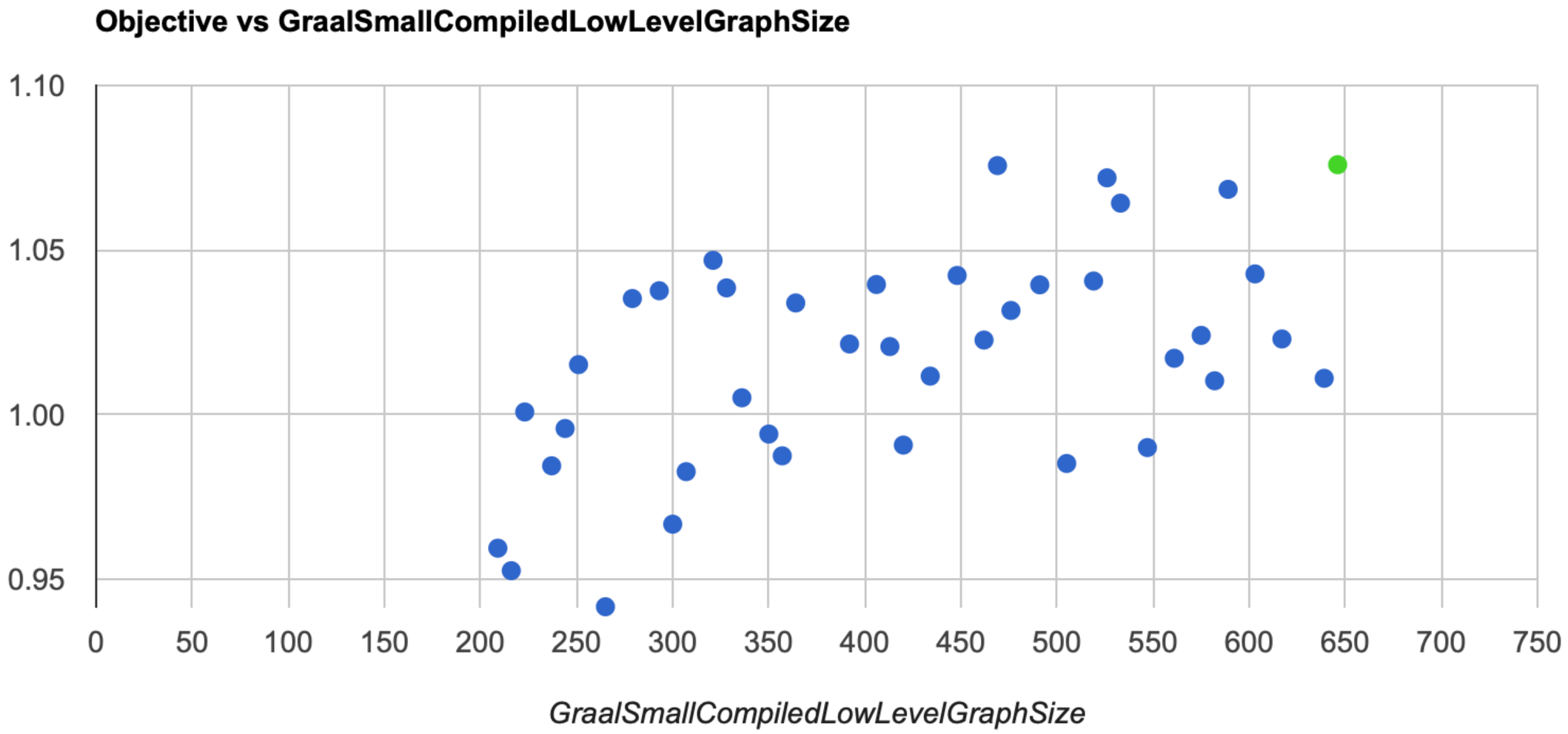
SOCIAL GRAPH: AUTOTUNE - CHARTS



SOCIAL GRAPH: AUTOTUNE - CHARTS

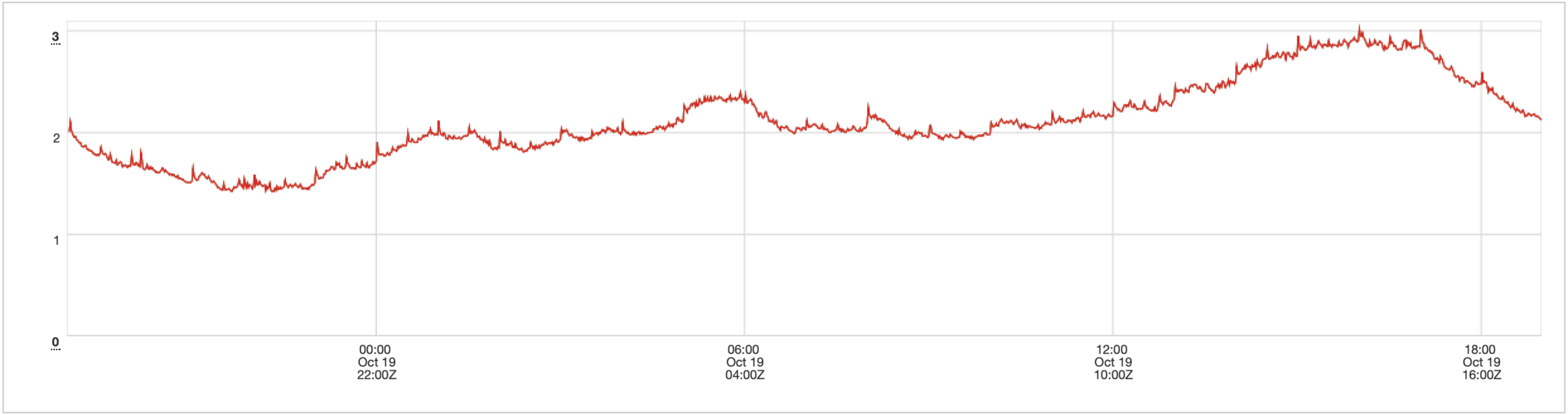


SOCIAL GRAPH: AUTOTUNE - CHARTS



SOCIAL GRAPH: REQUESTS/SEC

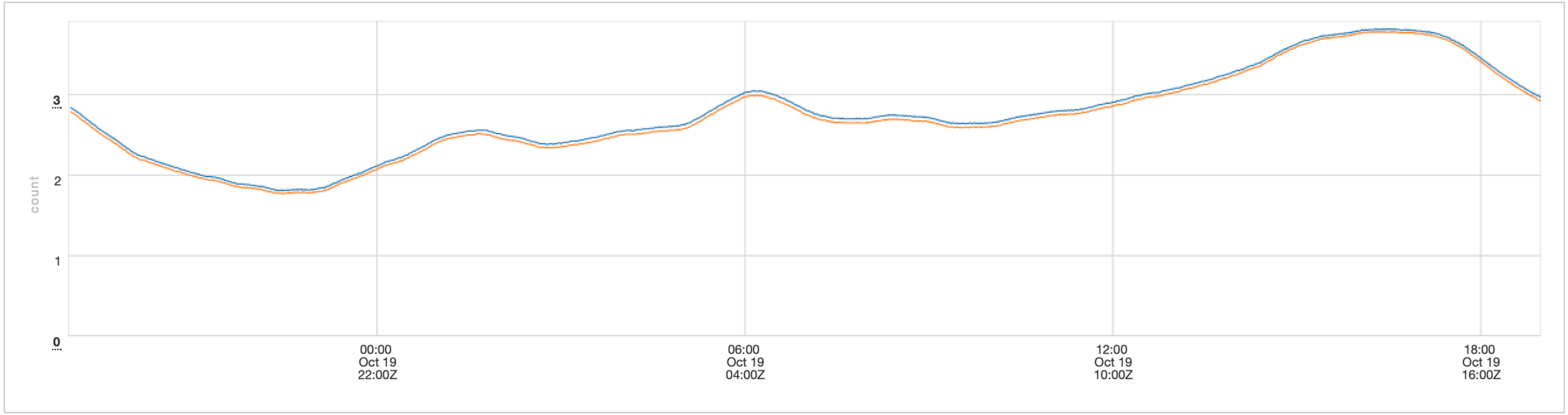
24 hours



C2 Graal Autotune

SOCIAL GRAPH: PARNEW CYCLES

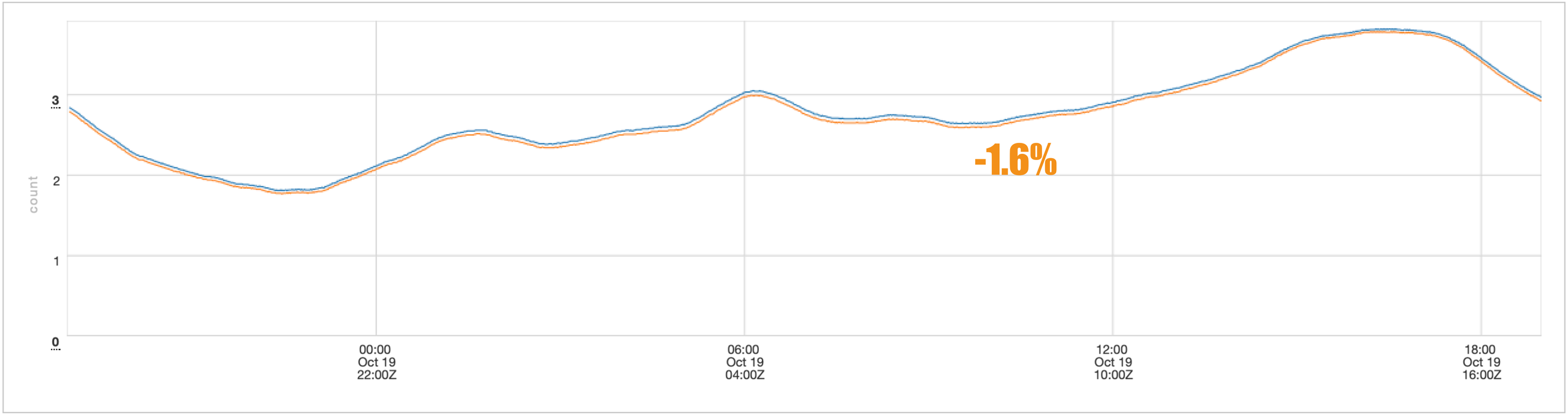
movingavg(60)



C2 Graal

SOCIAL GRAPH: PARNEW CYCLES

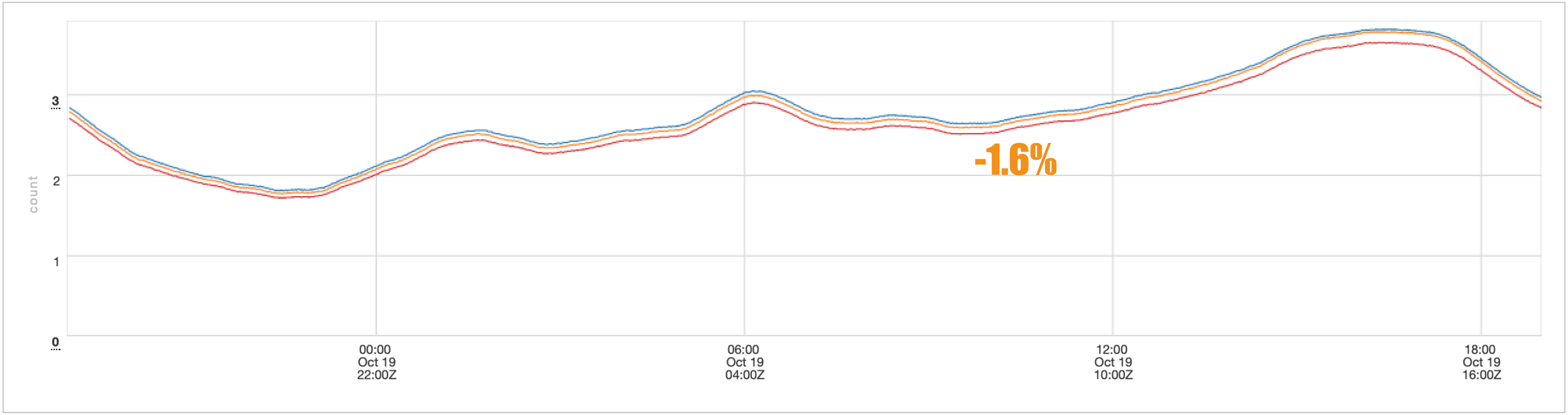
movingavg(60)



C2 Graal

SOCIAL GRAPH: PARNEW CYCLES

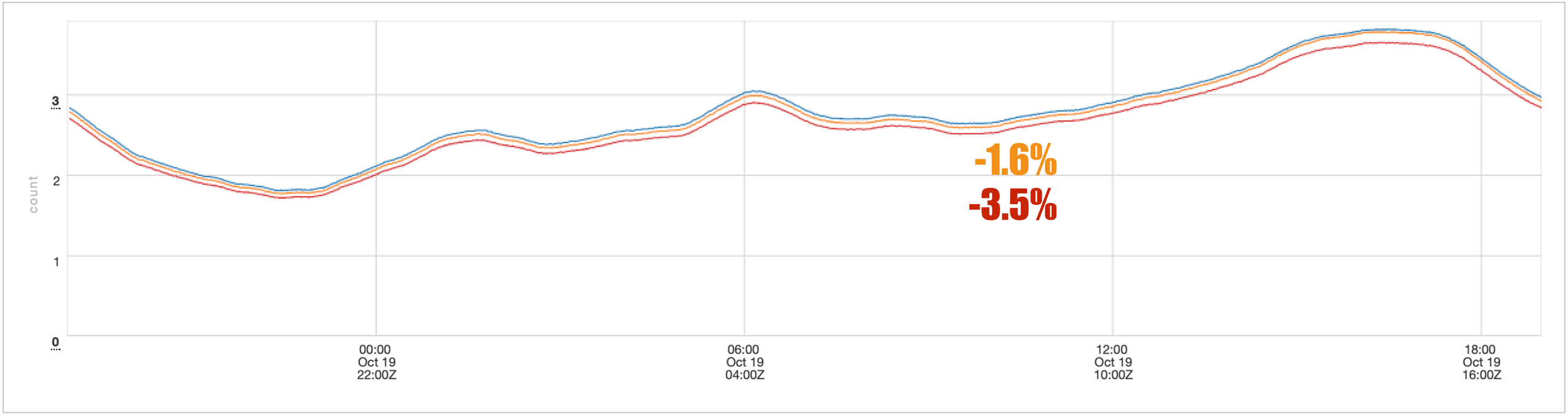
movingavg(60)



C2 Graal Autotune

SOCIAL GRAPH: PARNEW CYCLES

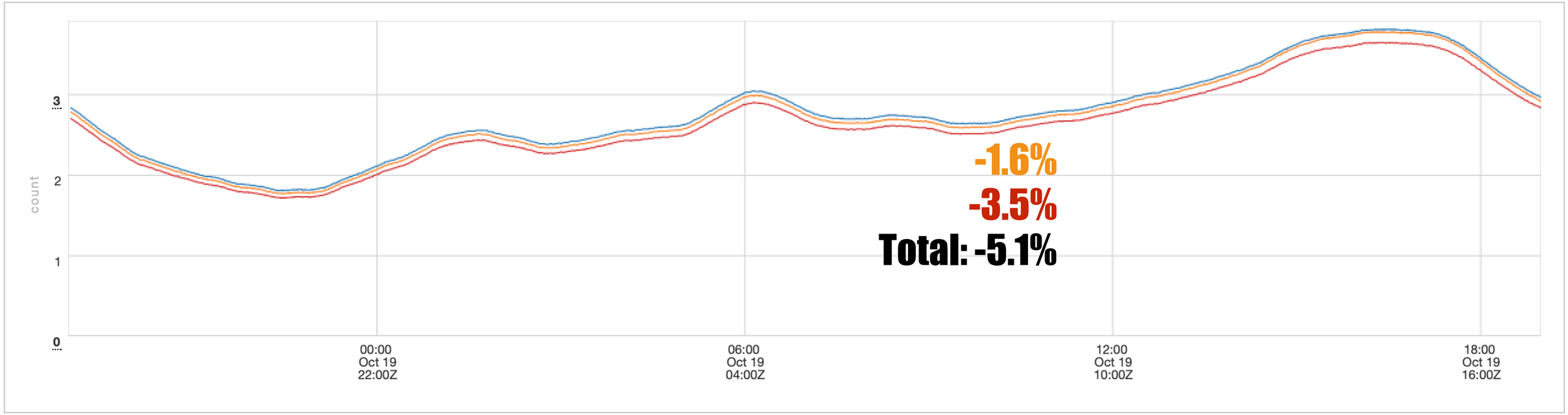
movingavg(60)



C2 Graal Autotune

SOCIAL GRAPH: PARNEW CYCLES

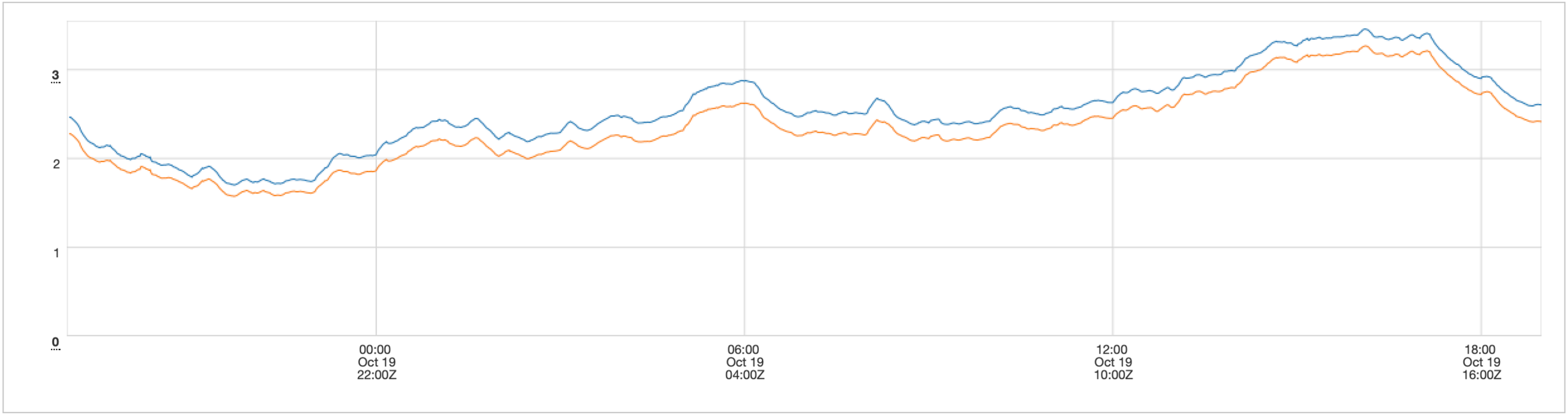
movingavg(60)



C2 Graal Autotune

SOCIAL GRAPH: USER CPU TIME

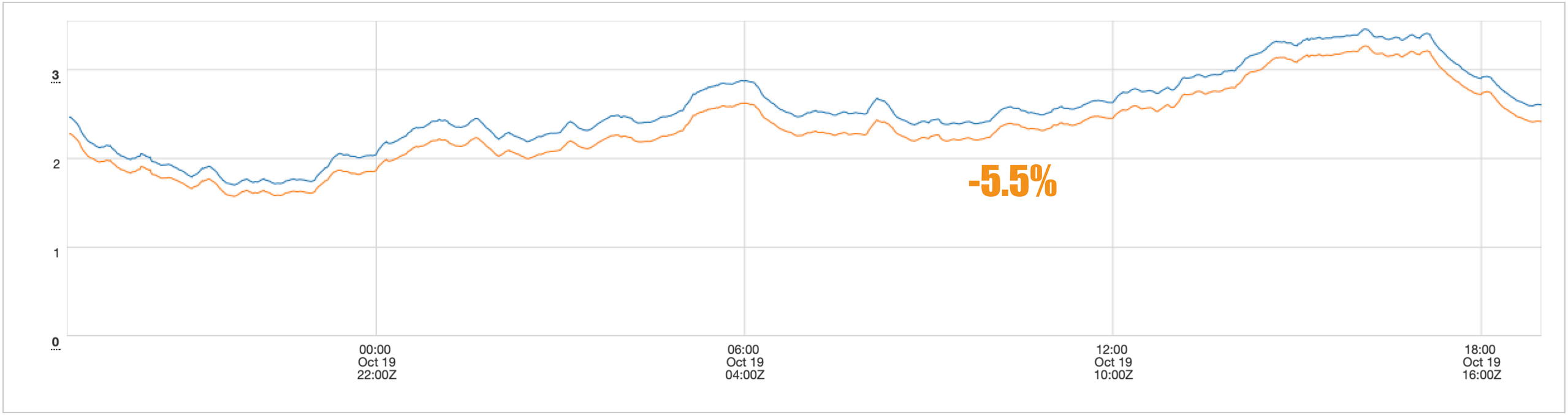
movingavg(10)



C2 Graal

SOCIAL GRAPH: USER CPU TIME

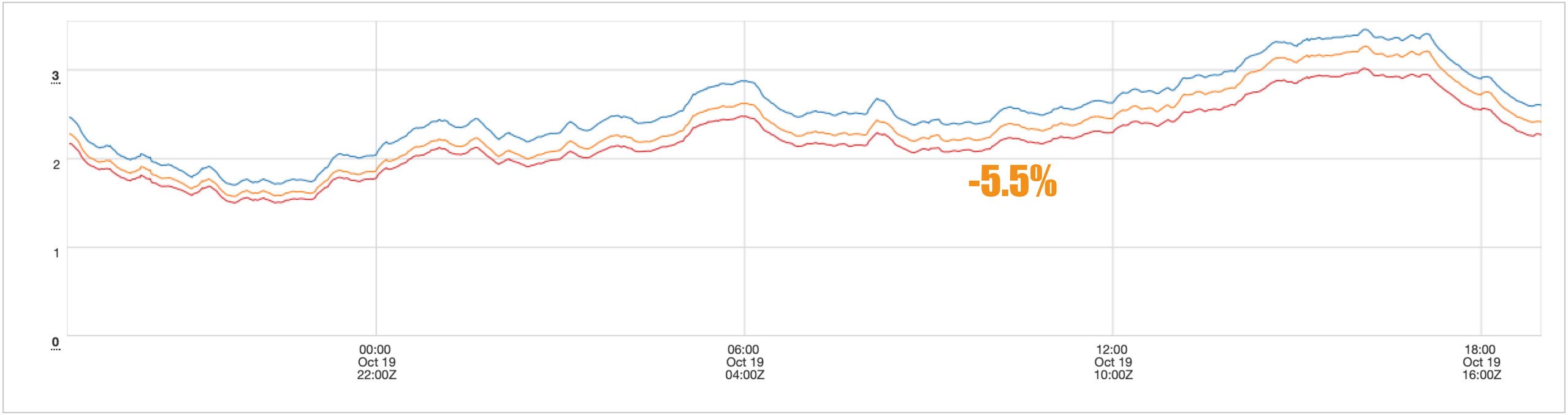
movingavg(10)



C2 Graal

SOCIAL GRAPH: USER CPU TIME

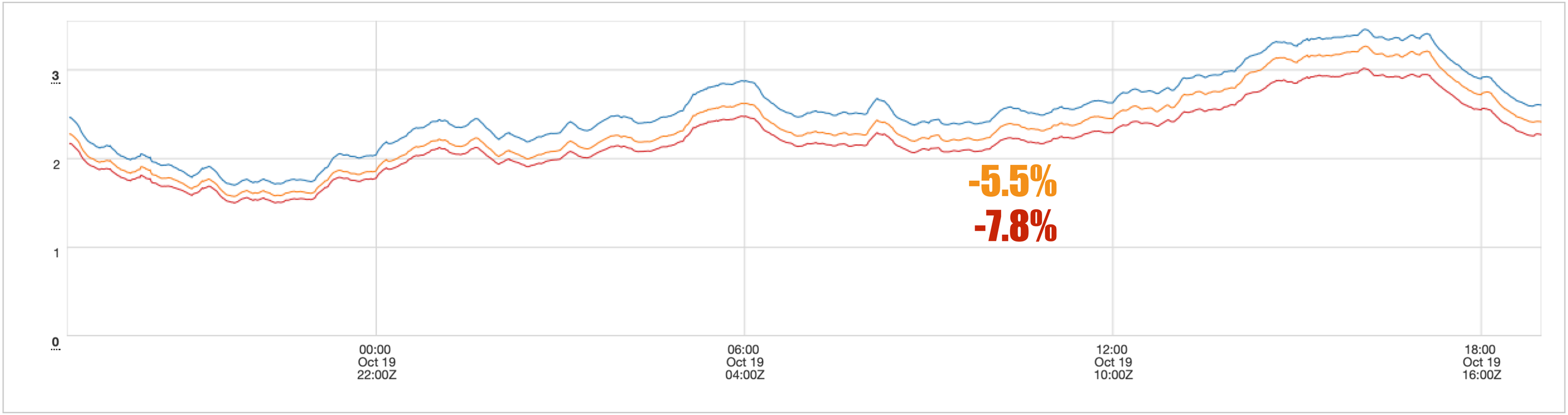
movingavg(10)



C2 Graal Autotune

SOCIAL GRAPH: USER CPU TIME

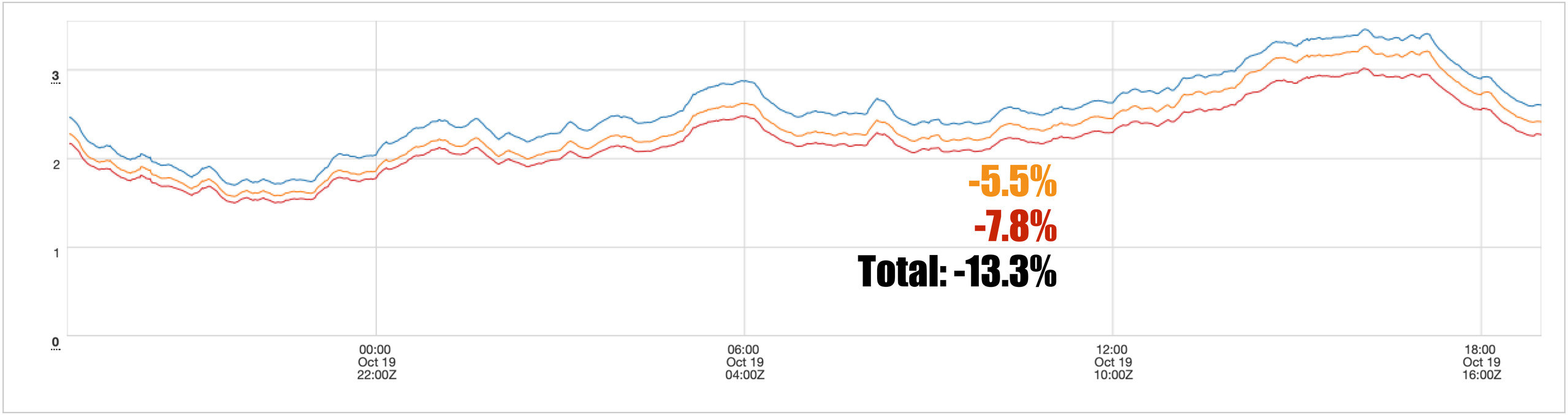
movingavg(10)



C2 Graal Autotune

SOCIAL GRAPH: USER CPU TIME

movingavg(10)



C2 Graal Autotune

Questions?

Q: Did you try the same with C2?

A: Well… yeah!

WHICH PARAMETERS?

```
intx MaxInlineLevel                    = 9

intx MaxInlineSize                     = 35

intx InlineSmallCode                   = 2000
```

WHAT RANGES?

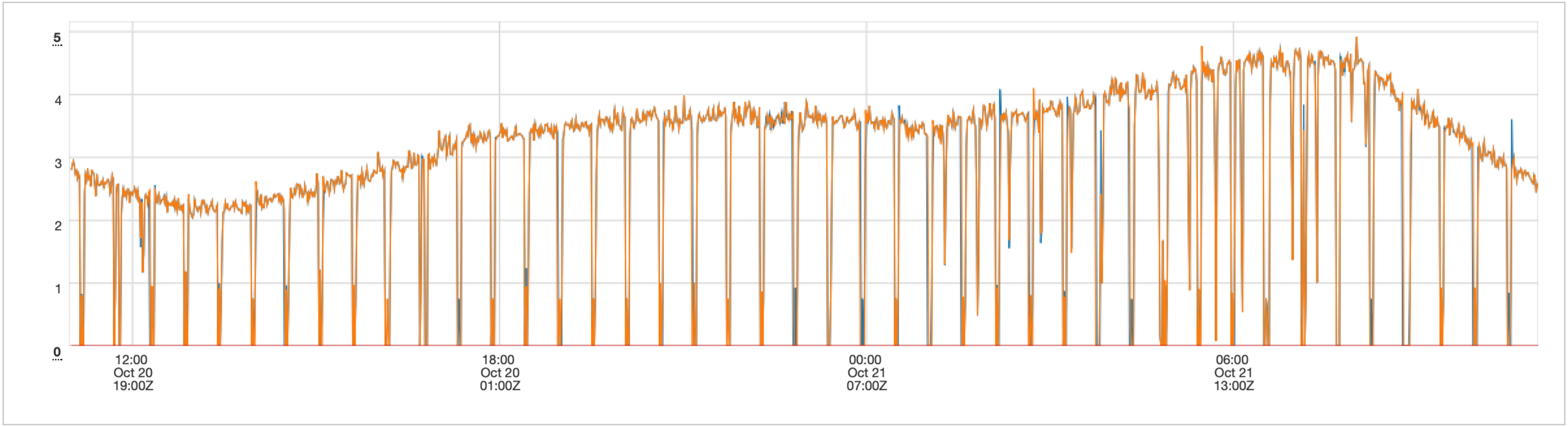
```
 {
   "name" : "XX:MaxInlineLevel",
   "whetlabName" : "MaxInlineLevel",
   "parameterType": "integer",
   "min" : "5",
   "max" : "20"
 },

{
  "name" : "XX:MaxInlineSize",
  "whetlabName" : "MaxInlineSize",
  "parameterType": "integer",
  "min" : "10",
  "max" : "50"
},
```

```
{
  "name" : "XX:InlineSmallCode",
  "whetlabName" : "InlineSmallCode",
  "parameterType": "integer",
  "min" : "500",
  "max" : "4000"
}
```

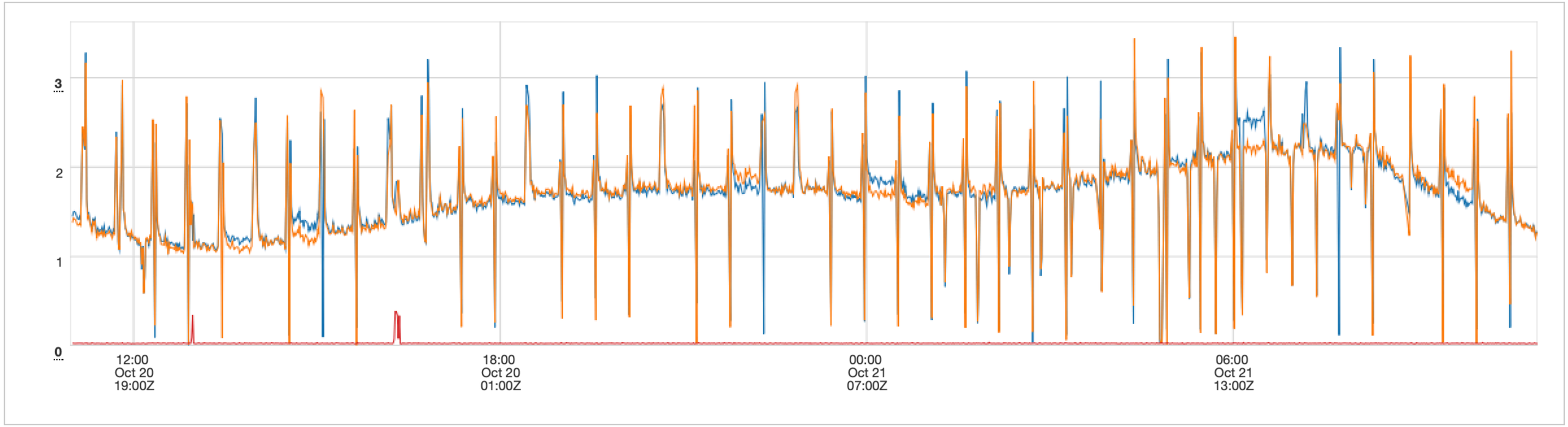
TWEET SERVICE: REQUESTS/SEC (C2)

24 hours



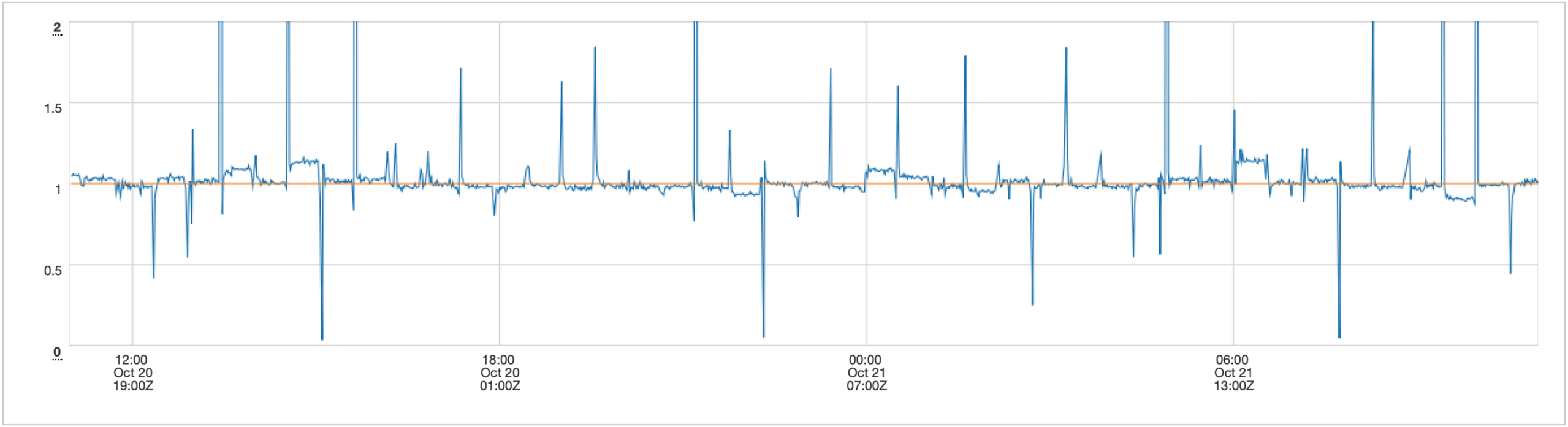
experiment ontrol

TWEET SERVICE: USER CPU TIME (C2)



experiment ontrol

TWEET SERVICE: USER CPU TIME - RATIO (C2)



experiment ontrol

TWEET SERVICE: AUTOTUNE RESULTS (C2)

63 results, 3 parameters. View description

Table | Charts

Best | Completed | Constraint Violated | In Progress

Add Result | Select all | Deselect all

Status	Result ID	Objective ↑	AutotuneOutcome	InlineSmallCode	MaxInlineLevel	MaxInlineSize	is_valid	isValid	Start
●	316069	1.0510		3688	16	32	true		10/21
●	316067	1.0380		1500	14	26	true		10/21
●	316045	1.0350		3305	17	41	true		10/20/
●	316035	1.0329		1335	18	39	true		10/20/
●	316057	1.0295		2758	16	50	true		10/21/1
●	316041	1.0291		2867	19	26	true		10/20/

(500-4000) (5-20) (10-50)

TWEET SERVICE: AUTOTUNE RESULTS (C2)

63 results, 3 parameters. View description

Table | Charts

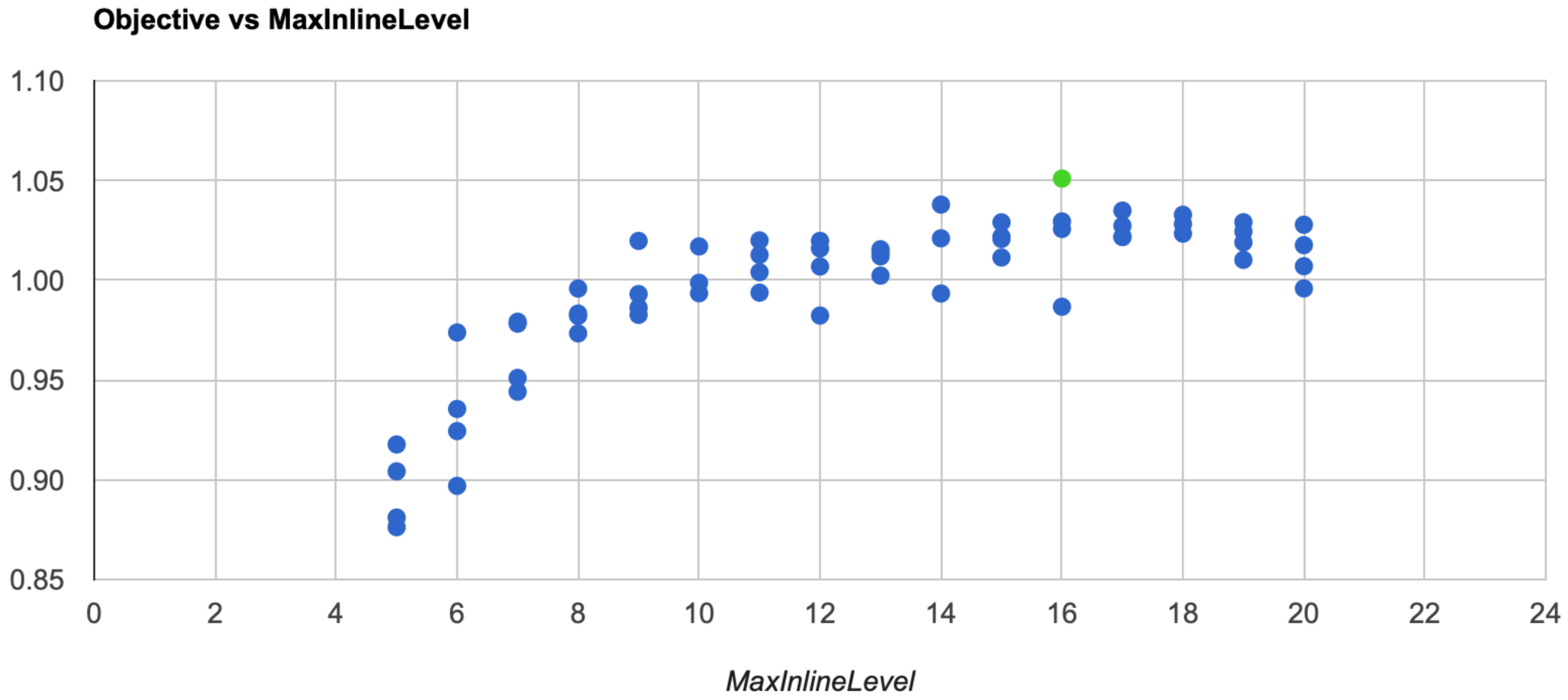
Best | Completed | Constraint Violated | In Progress

Add Result | Select all | Deselect all

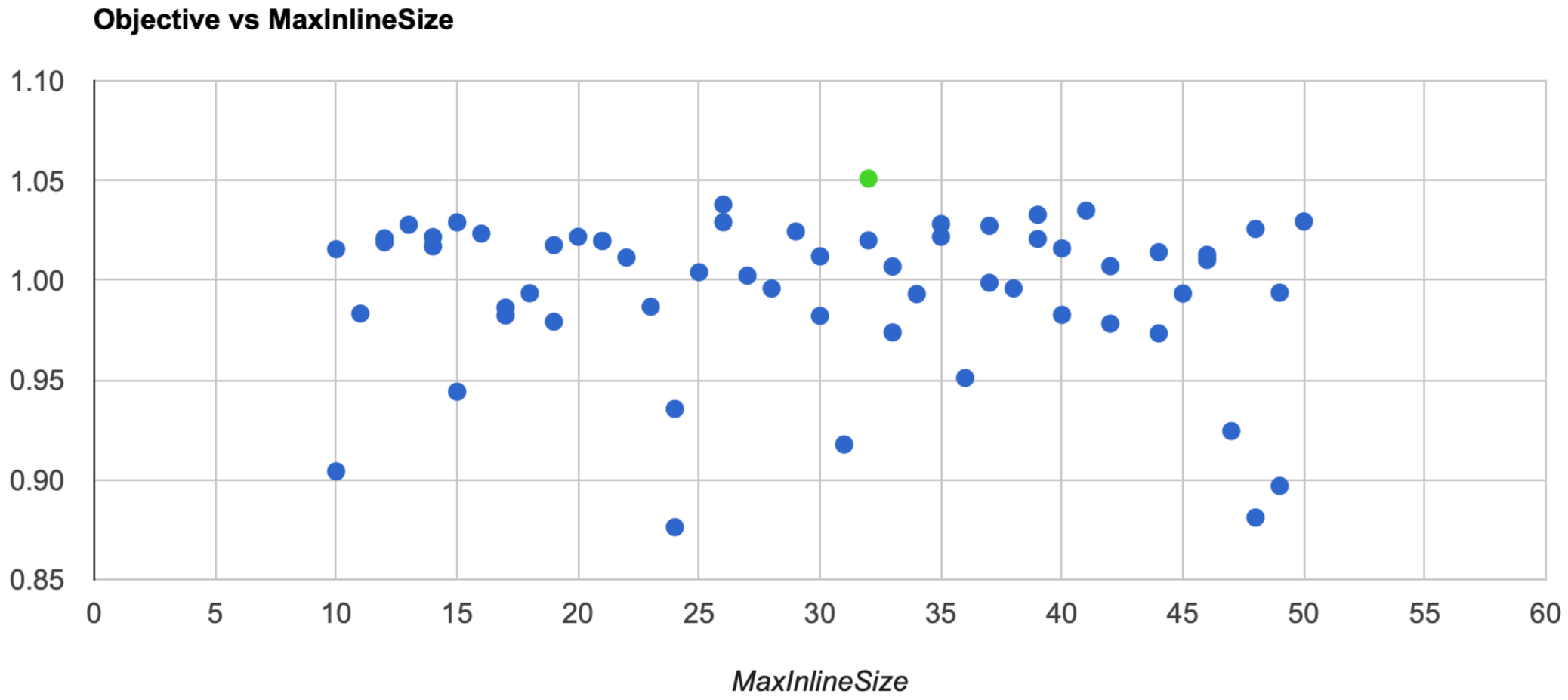
Status	Result ID	Objective ↓	AutotuneOutcome	InlineSmallCode	MaxInlineLevel	MaxInlineSize	is_valid	isValid	Start
	316033			2211	14	28			10/20
	316094			2265	10	43			10/21
	316077	0.87622		3469	5	24	true		10/21/1
	316049	0.88104		2539	5	48	true		10/20/1
	316032	0.89694		3961	6	49	true		10/20/1
	316083	0.90421		1609	5	10	true		10/21/1

(500-4000) (5-20) (10-50)

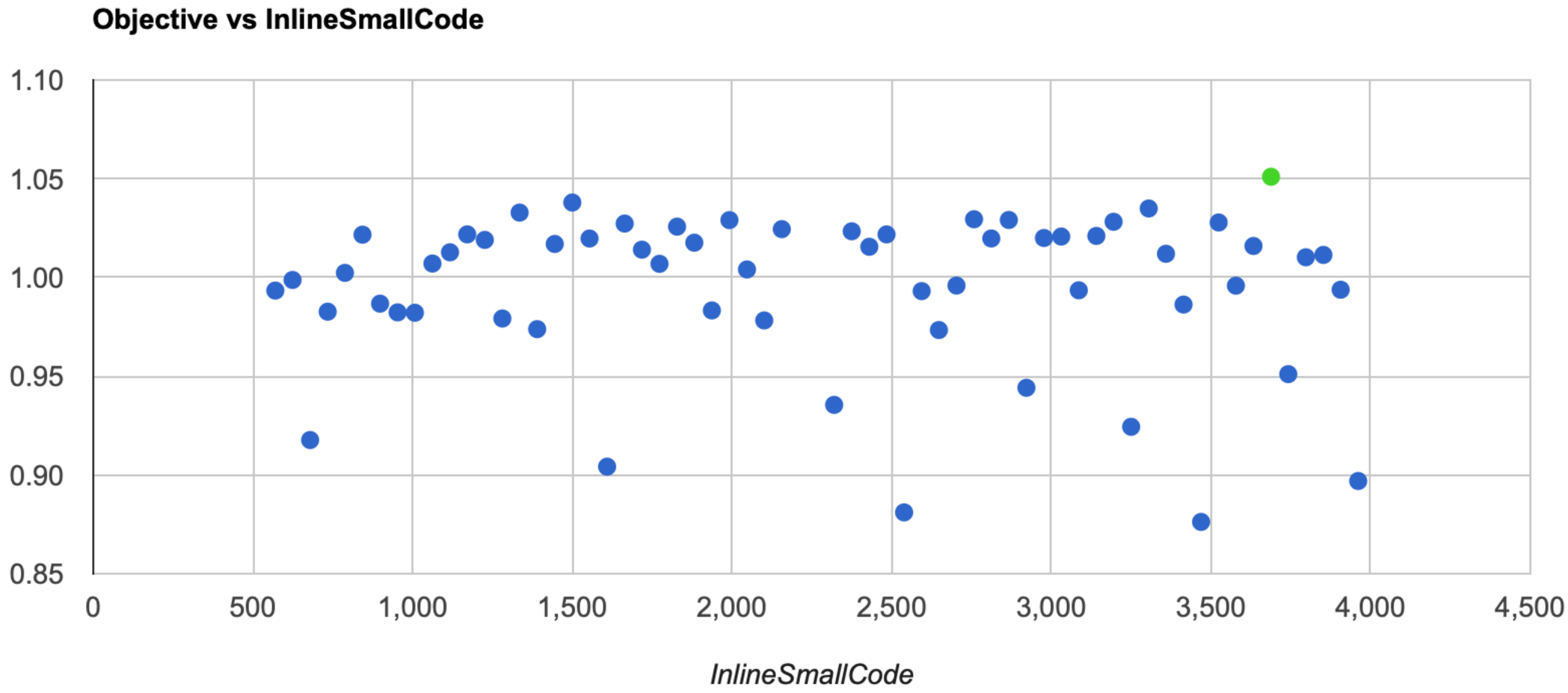
TWEET SERVICE: AUTOTUNE - CHARTS



TWEET SERVICE: AUTOTUNE - CHARTS



TWEET SERVICE: AUTOTUNE - CHARTS



SUMMARY

```
-XX:+UnlockExperimentalVMOptions -XX:+EnableJVMCI -XX:+UseJVMCICompiler
```

#TWITTERVMTEAM