

# USING METRICS TO MEASURE AND UNDERSTAND YOUR AWS ENVIRONMENT PERFORMANCE

**AWS GLOBAL SUMMIT 2015**

**9 APRIL 2015**

Matt Williams

Evangelist, Datadog

# BEFORE I GET STARTED...

- getting started is easy

# BEFORE I GET STARTED...

- getting started is easy
- getting good is not easy

# BEFORE I GET STARTED...

- getting started is easy
- getting good is not easy
- and it will not be quick

# WHY MEASURE?

- See where you came from

# WHY MEASURE?

- See where you came from
- See how you are doing

# WHY MEASURE?

- See where you came from
- See how you are doing
- See how you can improve

# WHY IS THIS COMPLICATED?

- Elasticity makes this complicated
- Elasticity is the new normal

# SCENARIOS THAT REQUIRE ELASTICITY

- Amateur Sports
- Shopping Events
- Concerts
- Marketing Campaigns
- What is your scenario?

# HOW TO SOLVE THE COMPLEXITY

- Tags are one solution and what Datadog uses

The image illustrates the use of tags in Datadog through three screenshots:

- Email Interface:** Shows an email from Datadog with tags 'dev' and 'support' highlighted in red and grey boxes. A search bar at the top also shows 'dev x support x'.
- File Explorer:** Shows a 'Dropbox' window with a 'Screenshots' folder. A context menu is open over a screenshot, with the option 'Assign tags to "Screenshots...3.42.28"' selected. The 'datadog demo' tag is highlighted in a red box.
- Datadog Instance Management:** Shows a table of instances with a search filter 'DemoTag : mattdemo'. The table lists an instance named 'MattDKR' with the role 'mattdemo' and state 'running'.

Name	DemoTag	Role	Instance ID	Instance Type	Instance State	Status
MattDKR	mattdemo		i-71c9347c	t2.medium	running	✓

# HOW TAGS MAKES THIS EASIER

# TAGS ALLOW FOR AD-HOC AGGREGATION

Monitor all Docker containers running image web

... in region us-west-2 across all availability zones

... and make sure resident set size < 1GB on c3.xl

# TAGS ALLOW FOR AD-HOC AGGREGATION

Monitor all Docker containers running *image* **web**

... in region **us-west-2** across all **availability zones**

... and make sure [ resident set size < 1GB ] on **c3.xl**

# TAGS ALLOW FOR AD-HOC AGGREGATION

Monitor all Docker containers running *image* **web**

... in region **us-west-2** across all **availability zones**

... and make sure [  $RSS > 1.5x \text{AVG}$  ] on **c3.xl**

**WHAT ABOUT CONTEXT?**

# WHAT METRICS TO LOOK AT?

- There isn't a single source of expertise
- This is the hard part

**THERE IS A LOT OF GUIDANCE**

# YOU NEED TO FIND / HIRE / TRAIN AN EXPERT

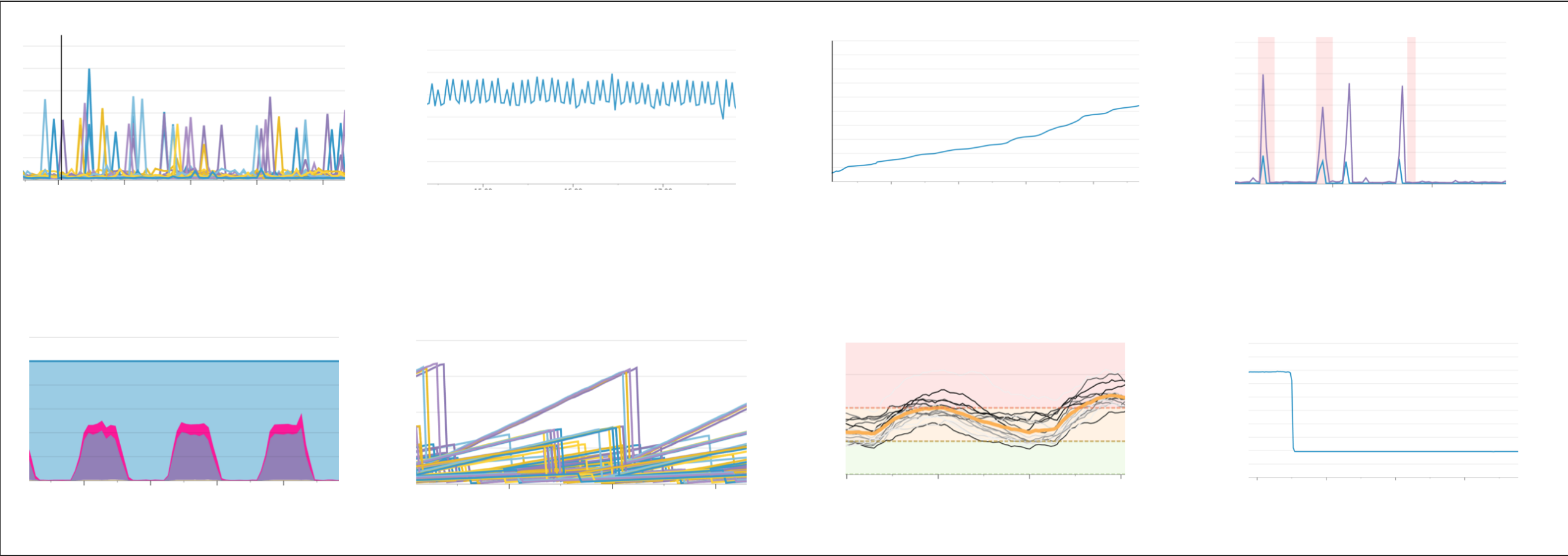
- Your context changes the meaning of metrics

# 3 CATEGORIES OF METRICS

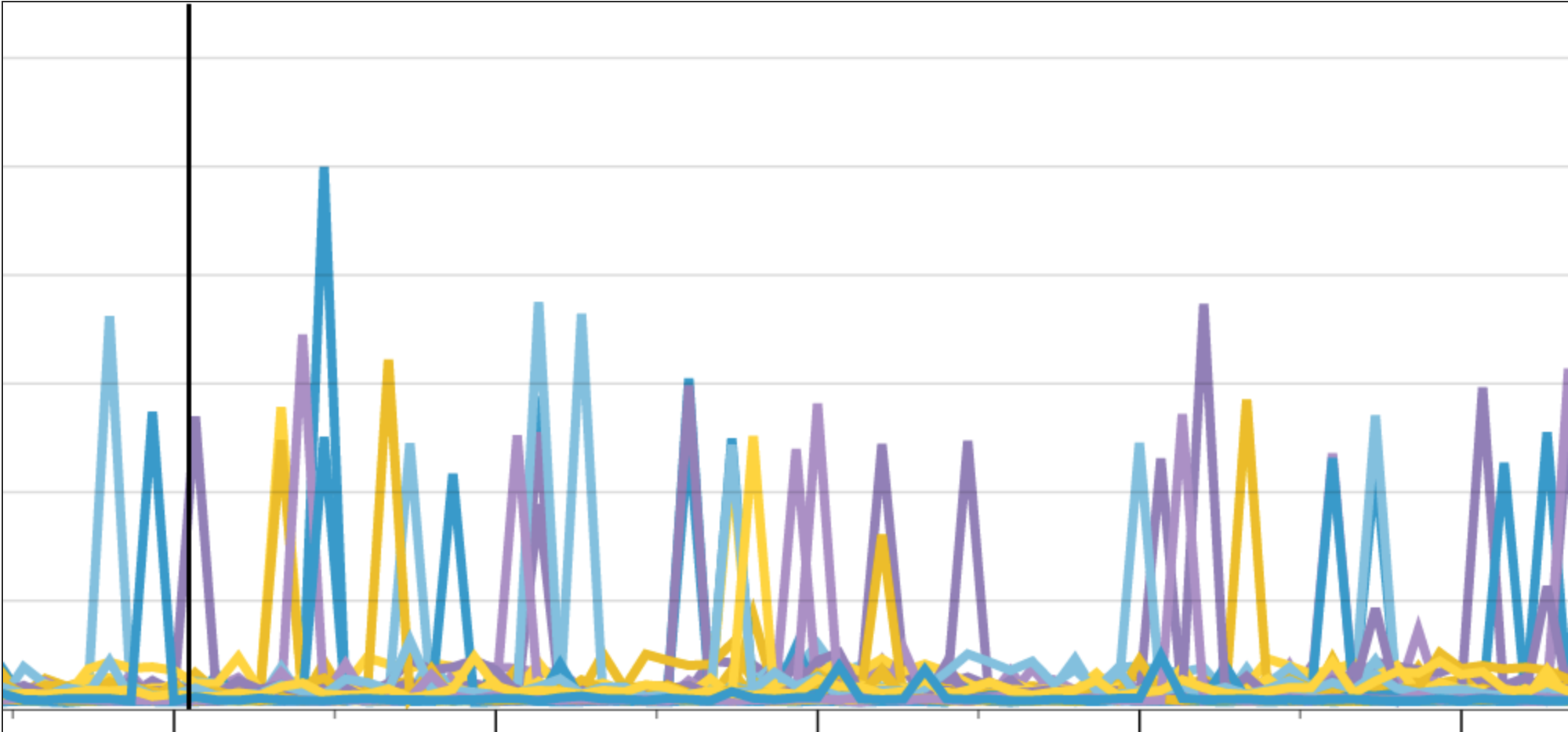
- Utilization - percent/time
- Saturation - wait queue length
- Errors - error count

# WHAT IS GOOD VS BAD?

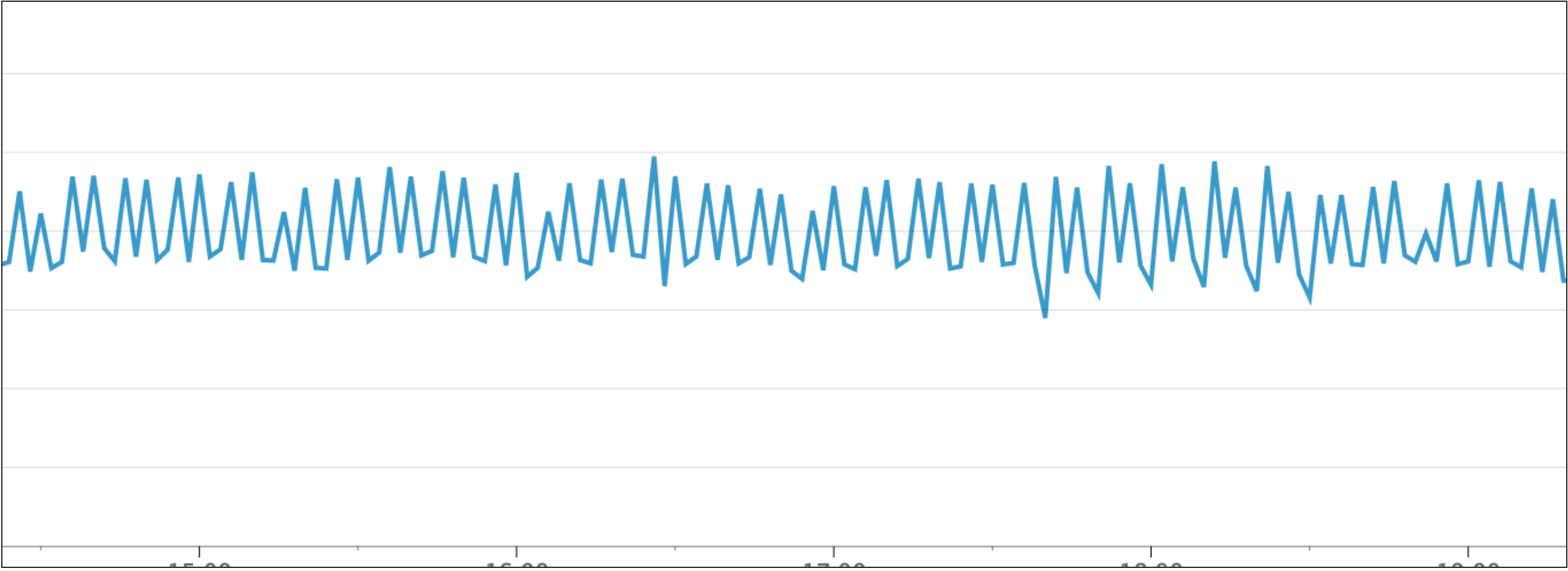
First learn the patterns



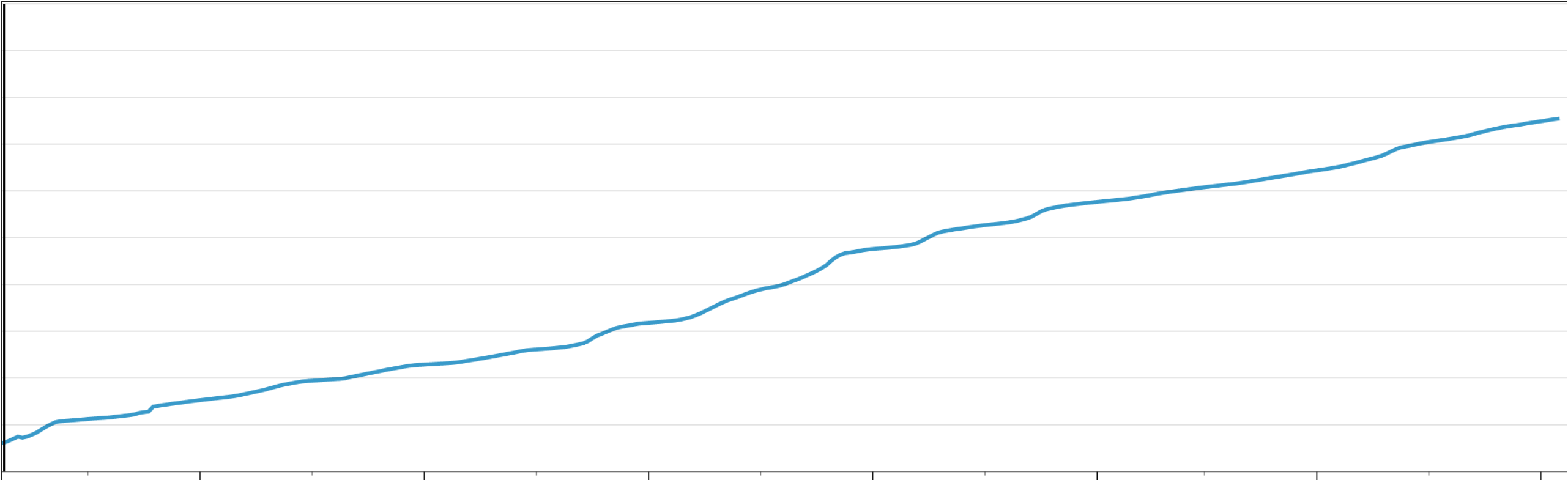
# SPIKY



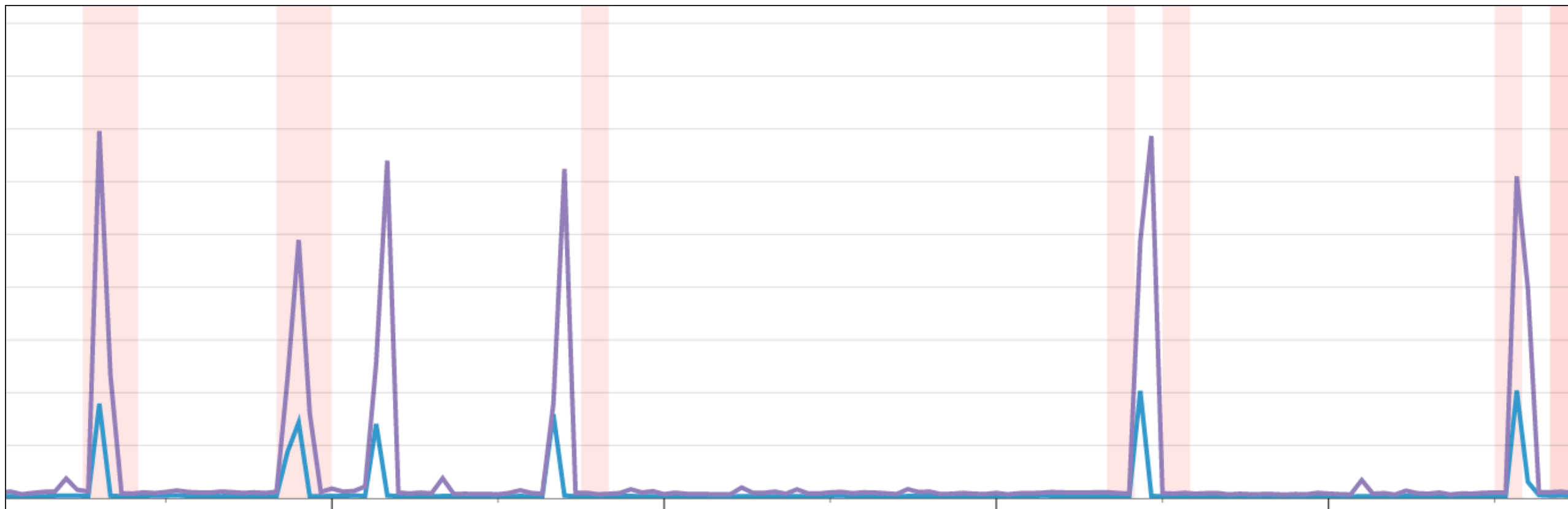
# STEADY



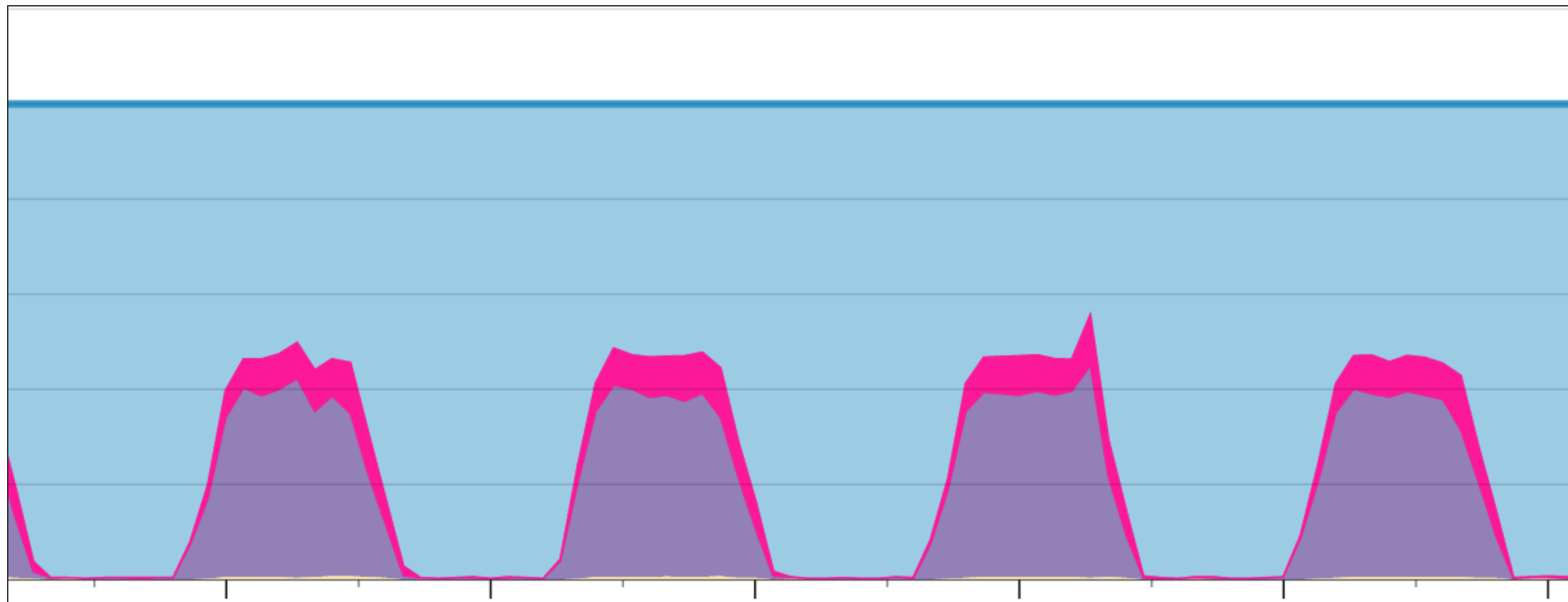
# COUNTER



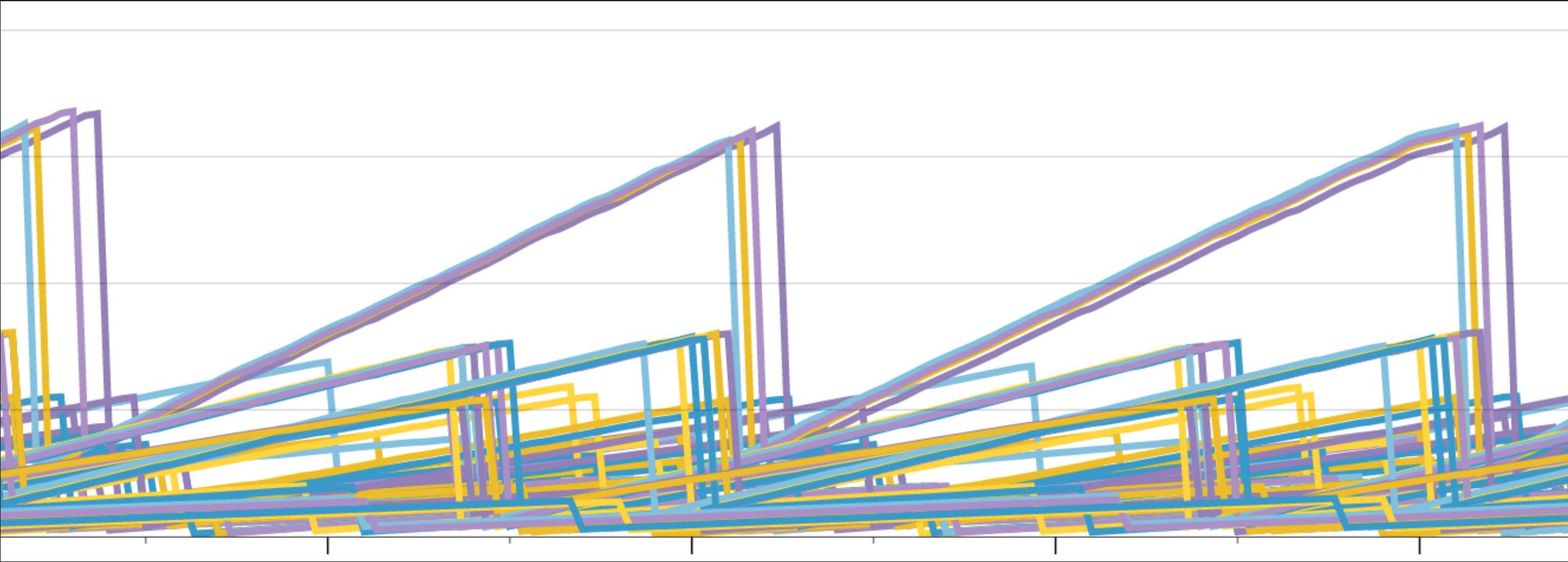
# BURSTY



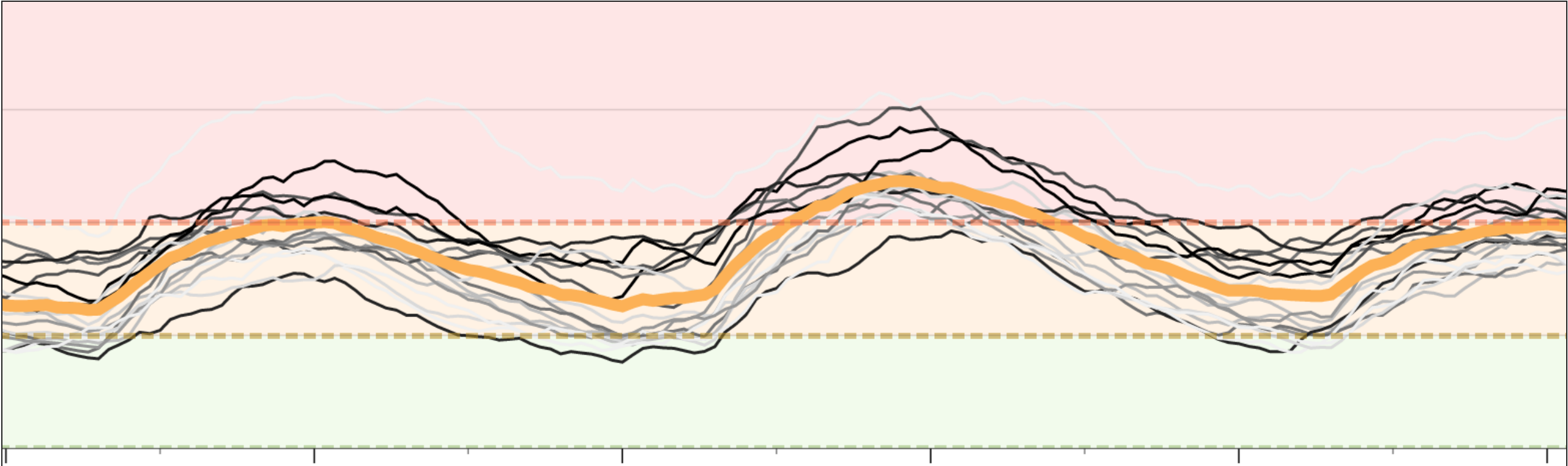
# BINARY



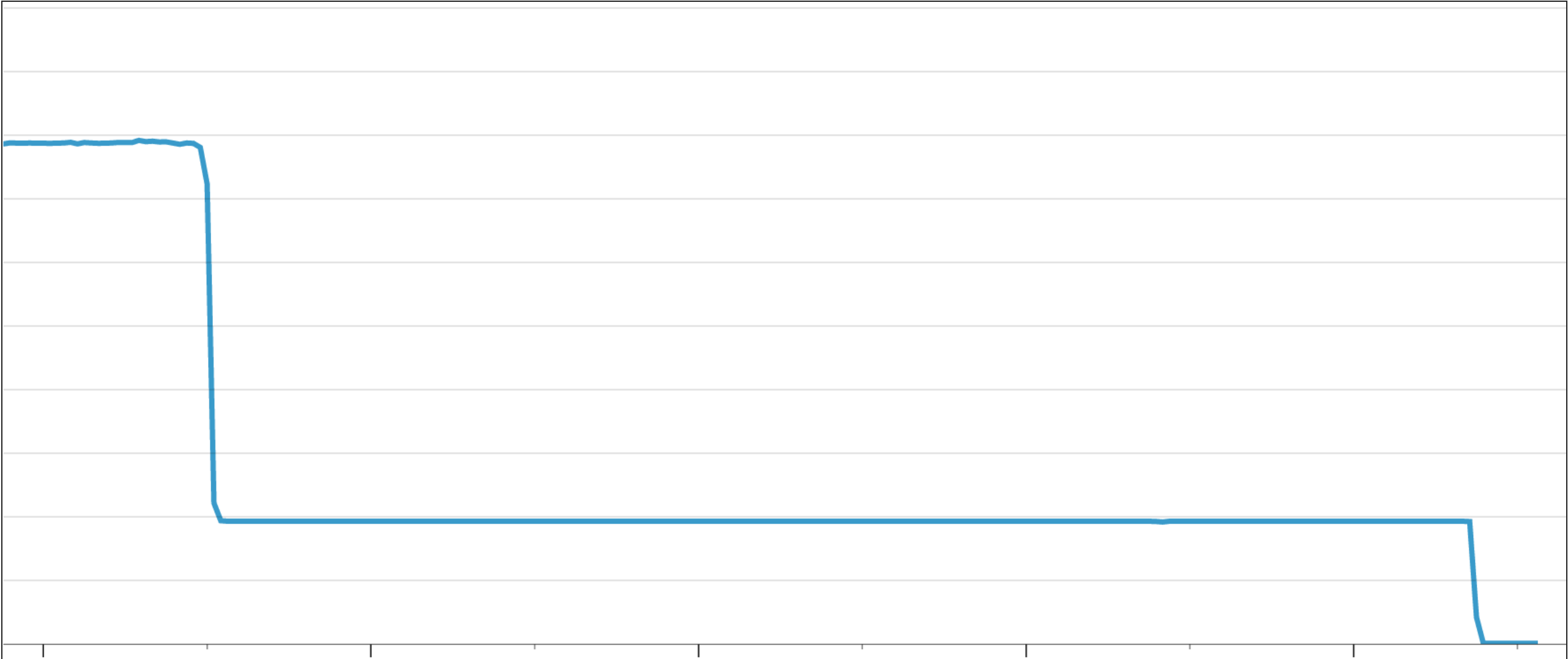
# CLASSIC SAWTOOTH



# CYCLIC



# STAIRY



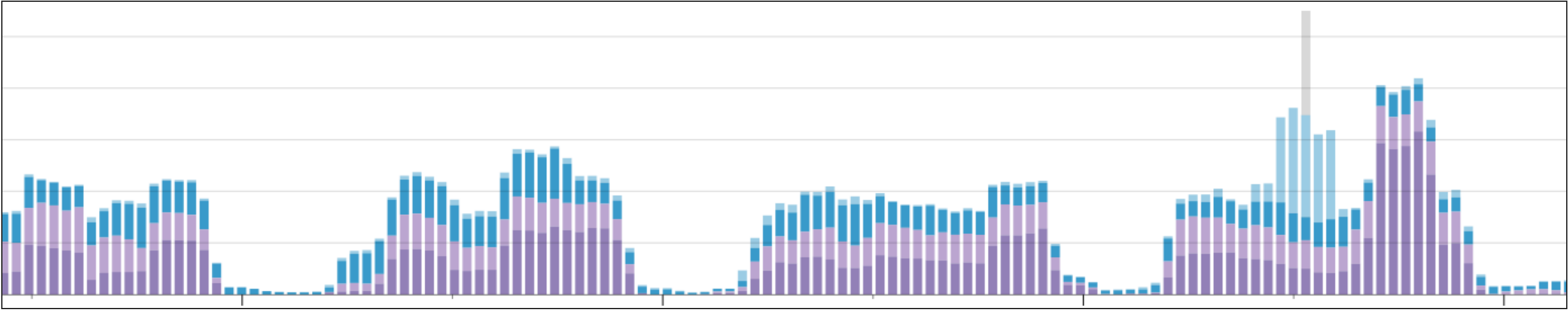
# ARE THE ANOMALIES THE FOCUS...



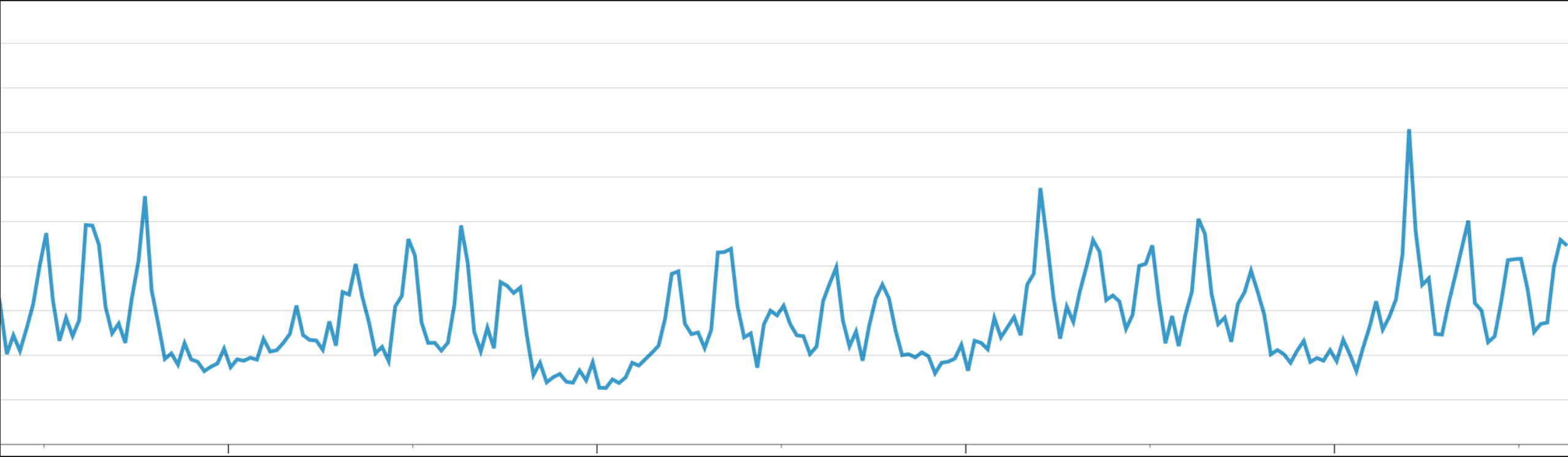
# ARE THE ANOMALIES THE FOCUS, OR SHOULD THEY BE IGNORED



# FIGURE OUT YOUR CYCLES



# A MONTH OF CLOUDTRAIL

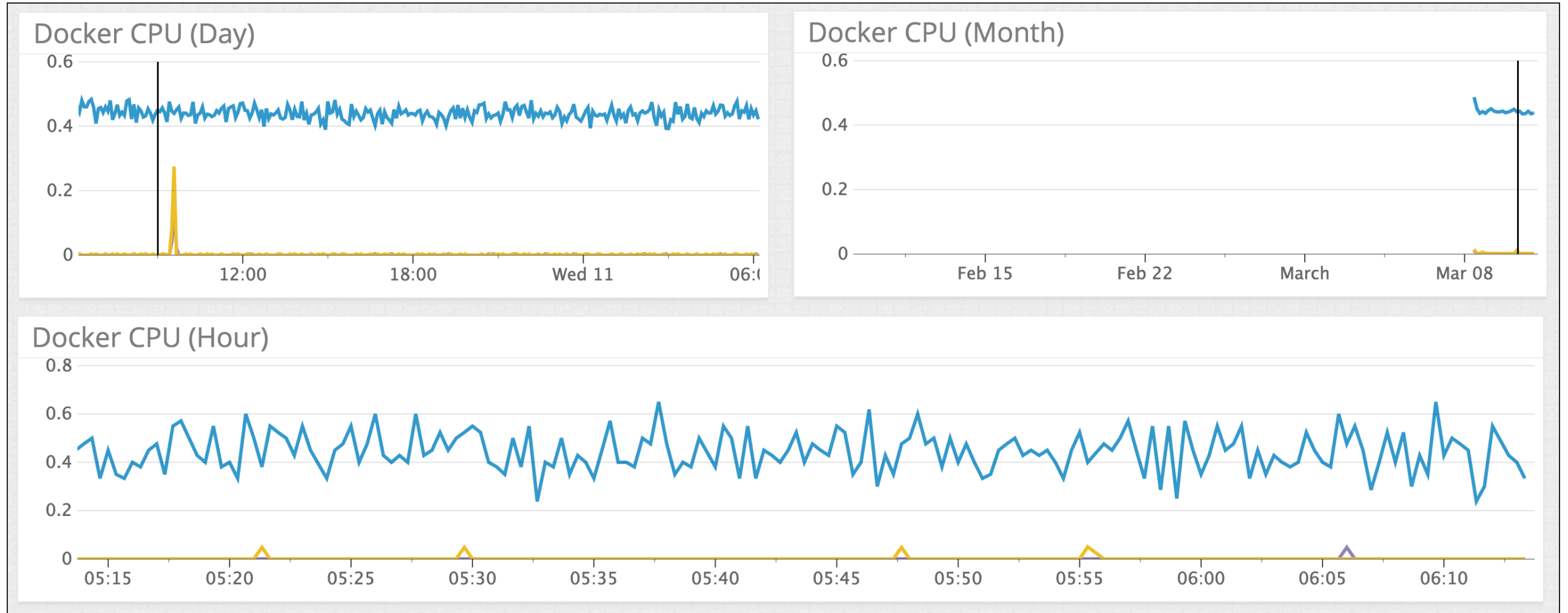


# GET THE SCALES RIGHT

Table 2.2 Example Time Scale of System Latencies

Event	Latency	Scaled
1 CPU cycle	0.3 ns	1 s
Level 1 cache access	0.9 ns	3 s
Level 2 cache access	2.8 ns	9 s
Level 3 cache access	12.9 ns	43 s
Main memory access (DRAM, from CPU)	120 ns	6 min
Solid-state disk I/O (flash memory)	50–150 $\mu$ s	2–6 days
Rotational disk I/O	1–10 ms	1–12 months
Internet: San Francisco to New York	40 ms	4 years
Internet: San Francisco to United Kingdom	81 ms	8 years
Internet: San Francisco to Australia	183 ms	19 years
TCP packet retransmit	1–3 s	105–317 years
OS virtualization system reboot	4 s	423 years
SCSI command time-out	30 s	3 millennia
Hardware (HW) virtualization system reboot	40 s	4 millennia
Physical system reboot	5 m	32 millennia

# COMBINE TIME SCALES TO FIND PATTERNS



# DIFFERENT INTERVALS

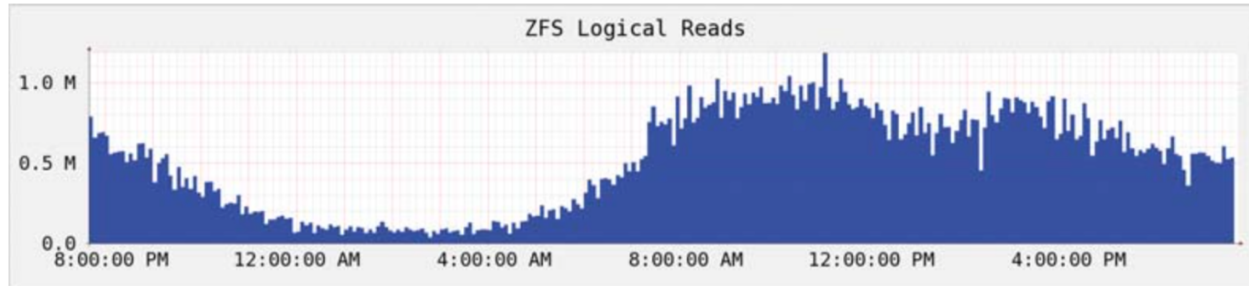


Figure 2.23 Monitoring activity: one day

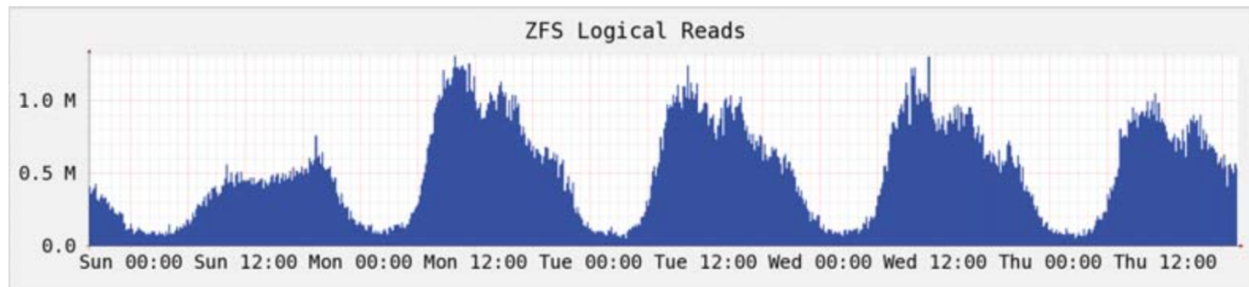


Figure 2.24 Monitoring activity: five days

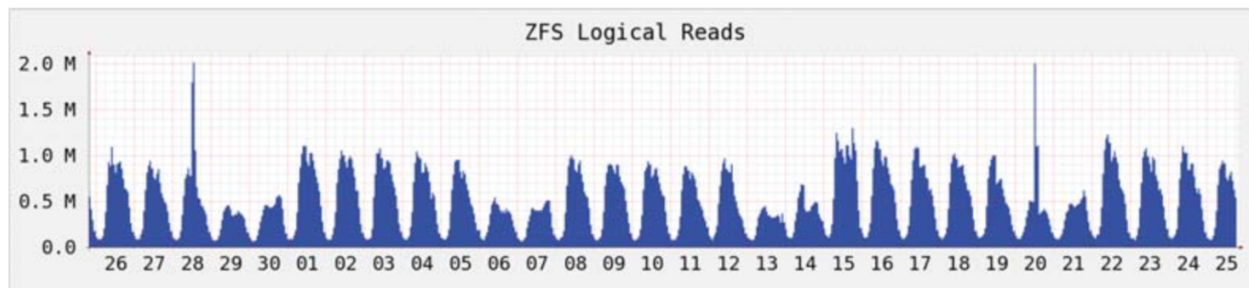
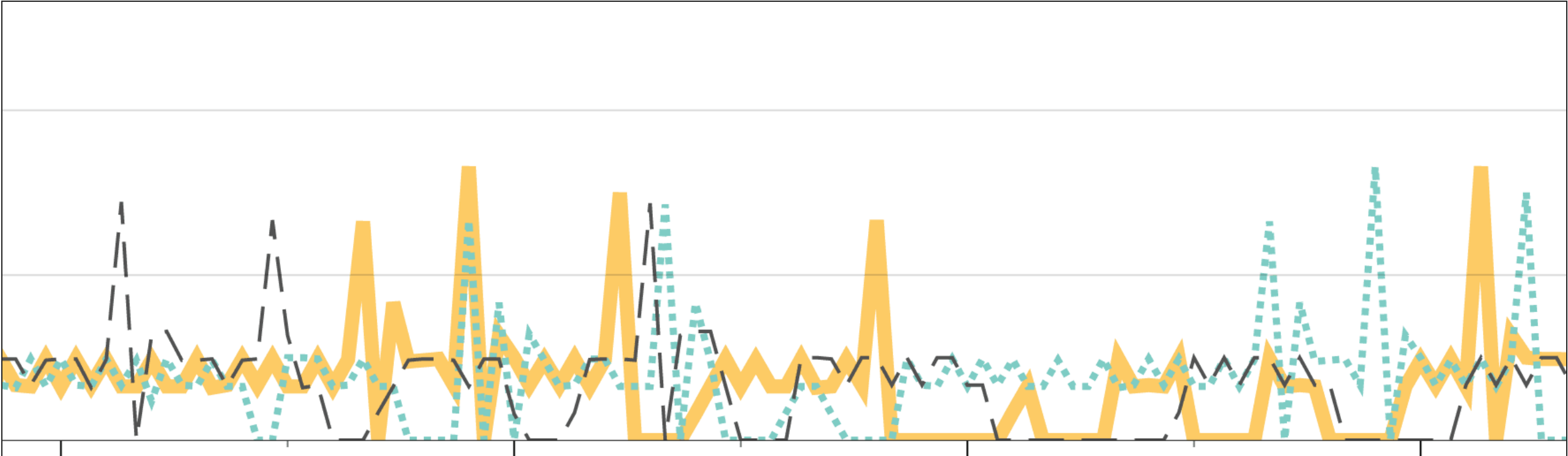


Figure 2.25 Monitoring activity: 30 days

# COMBINE SCALES TO FIND PATTERNS



# **SPECIFIC METRICS TO CONSIDER WITH AWS**

It's impossible to say what the best metrics are in all cases.

It depends on YOUR workload

# AWS EC2

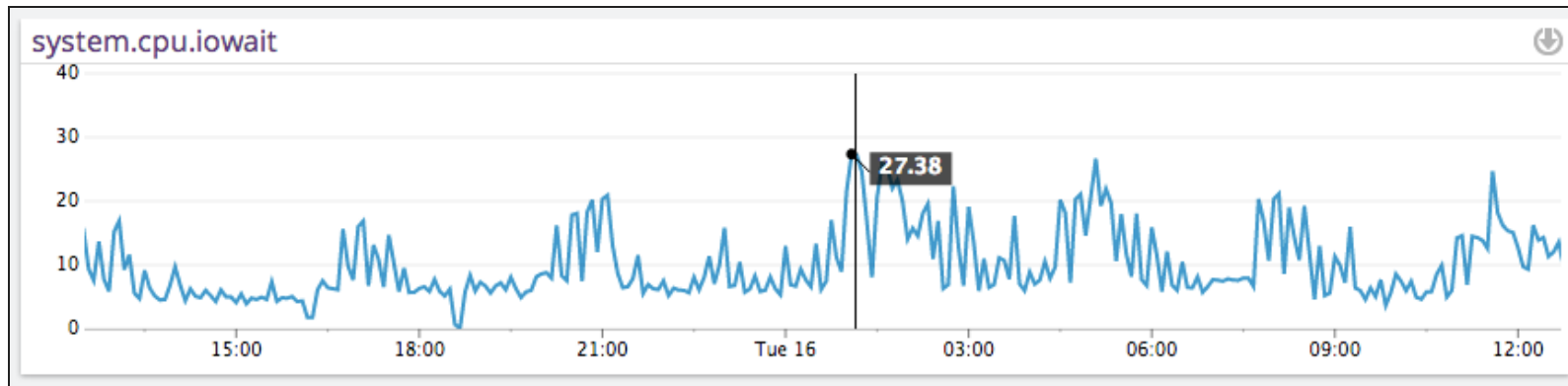
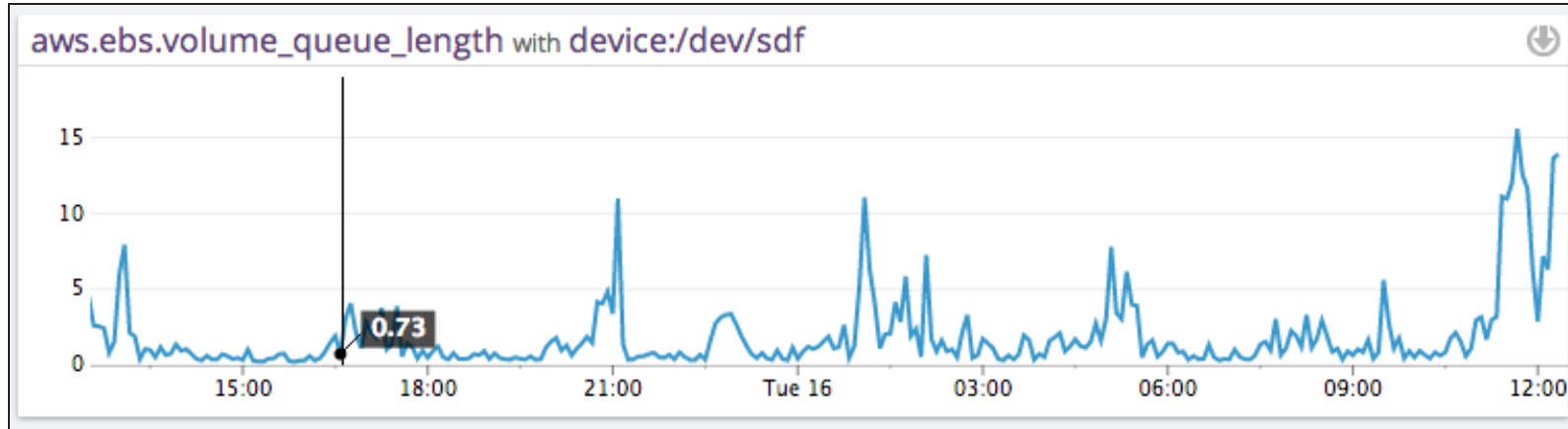
- system.cpu.stolen
- system.cpu.idle
- system.load.norm.5 \*
- system.mem.pct\_usable \*
- system.disk.pct\_usable \*
- aws.ec2.network\_in/out
- aws.ec2.disk\_read/write\_ops

# AWS EC2

(<https://www.datadoghq.com/2013/08/understanding-aws-stolen-cpu-and-how-it-affects-your-apps/>)

# AWS EBS PROVISIONED IOPS

- `aws.ebs.volume_queue_length` (should never go over 1/100)



(<https://www.datadoghq.com/2013/07/aws-ebs-provisioned-iops-getting-optimal-performance/>)

# AWS ELASTIC LOAD BALANCER

- healthy\_host\_count, latency
- HTTPCode\_ELB\_5XX
- surge\_queue\_length
- spill\_over\_count
- backend\_connection\_errors

(<https://www.datadoghq.com/2013/11/key-aws-elb-monitoring-metrics/>)

# SPECIFIC SQS METRICS

- number\_of\_messages\_sent (NOM\_sent), NOM\_received
- NOM\_deleted (throughput)
- sent\_message\_size (cost)
- approximate\_NOM\_visible (backlog)
- approximate\_NOM\_not\_visible (worked on)

(<https://www.datadoghq.com/2014/08/monitor-amazon-sqs-message-traffic-datadog/>)

# HOW TO GET STARTED

# QUESTIONS

# THANK YOU

Matt Williams

Evangelist, Datadog

[matt.williams@datadoghq.com](mailto:matt.williams@datadoghq.com) (mailto:matt.williams@datadoghq.com)

<http://datadoghq.com> (http://datadoghq.com)

[@technovangelist](http://twitter.com/technovangelist) (http://twitter.com/technovangelist)

