# Distributed System Performance Troubleshooting Like You've Been Doing it for Twenty Years

Jon Haddad
Rustyrazorblade Consulting
rustyrazorblade.com

# About me

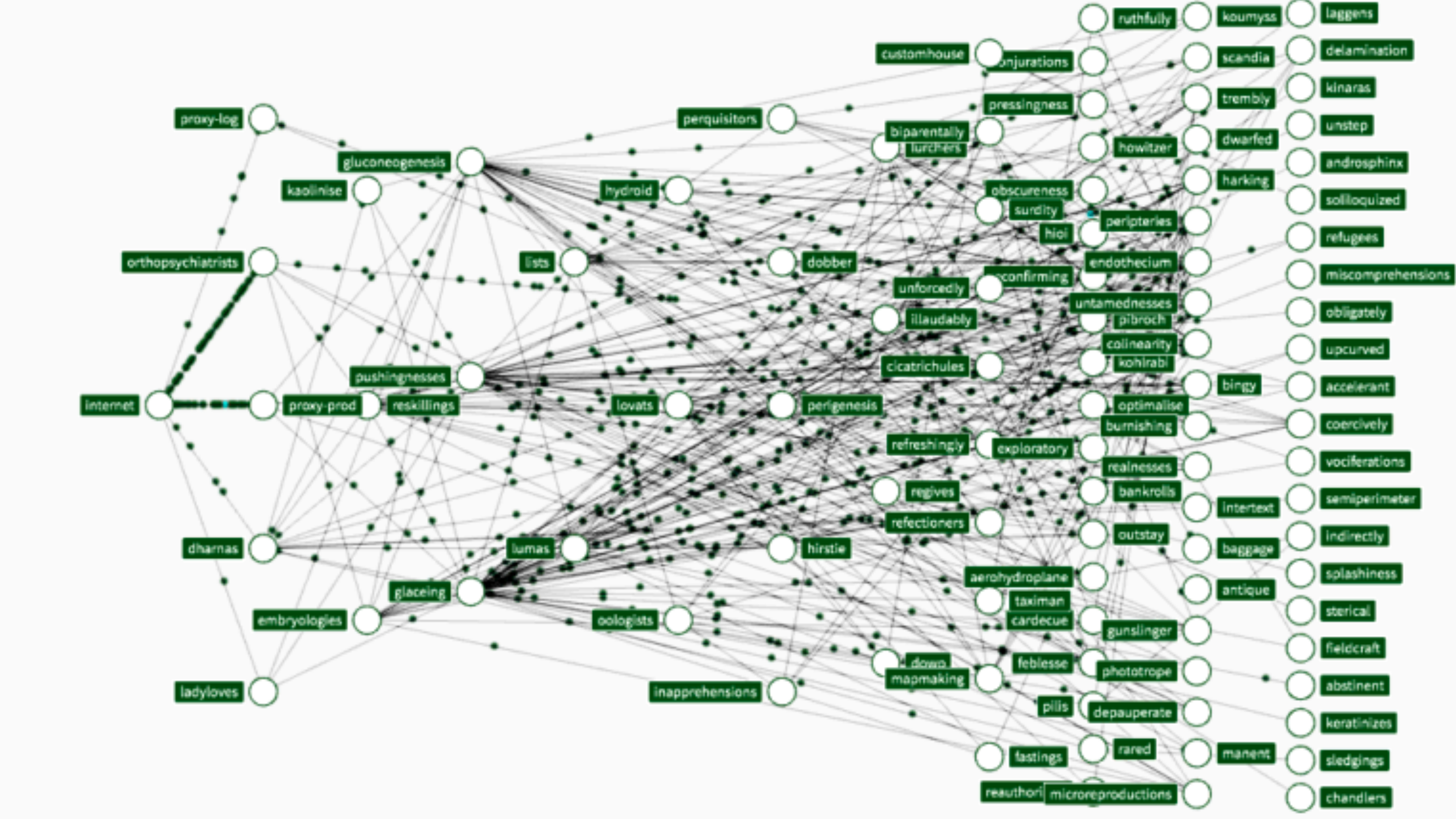
- 20 Year Career
- 30 Years of SW Development
- Apache Cassandra Committer
- Ex-Netflix
- Ex-Apple
- Ex-Lots of other stuff
- Now on my own!

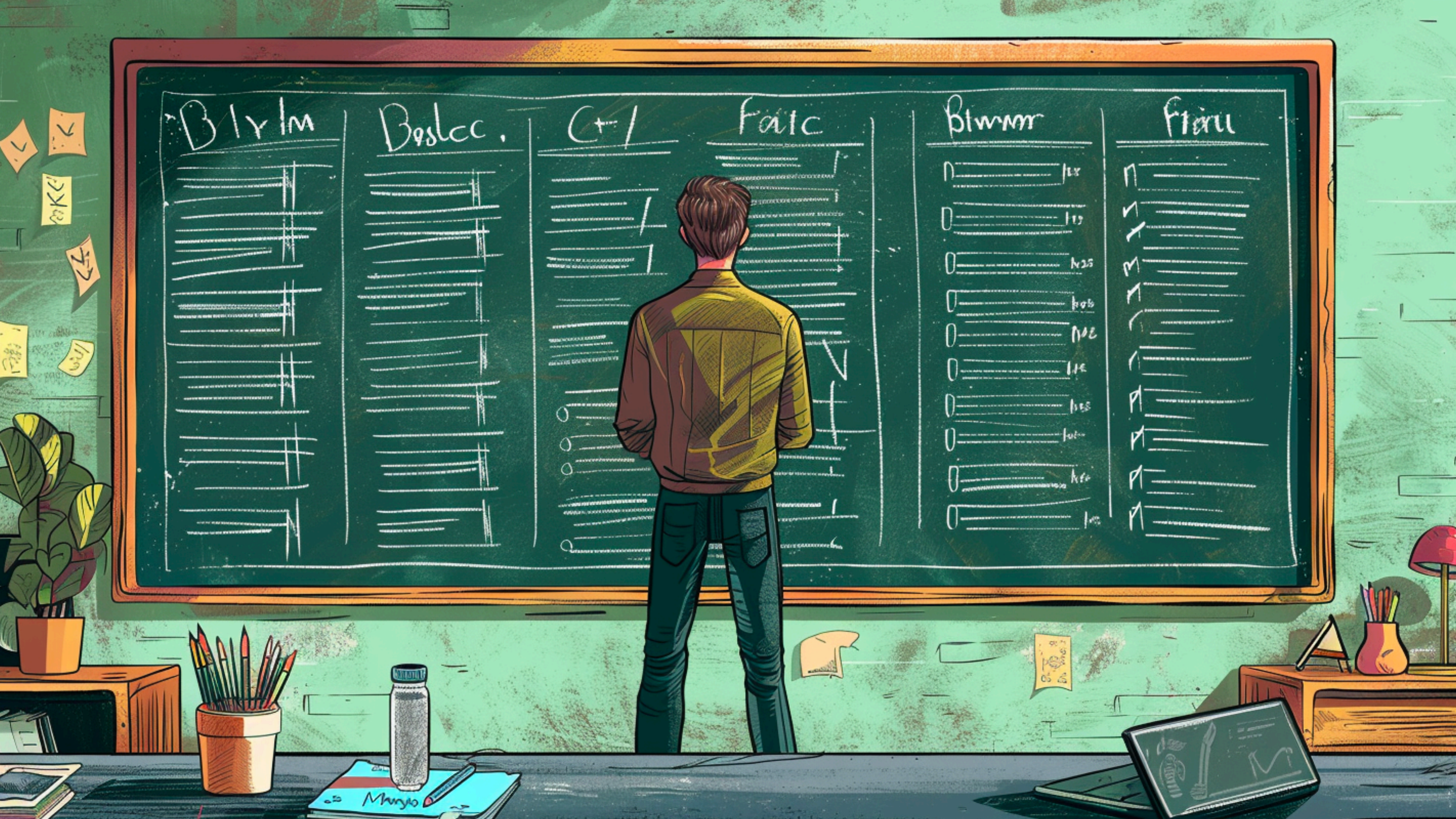
# What do we do when we have a production issue?

# Blame The Database.
# The End.


NO
NO


ELK-Session-One
Zoom Out
Load
1.0%
State
Up
Cores
4
Memory
8.11 GB
All disks
985 GB
CPU
8.00%
CPU usage
12.500%
10.000%
7.500%
5.000%
2.500%
0%
19:55
20:00
20:05
min max avg current
iowait 0.50% 1.50% 1.09% 1.10%
system 0.40% 0.90% 0.58% 0.70%
user 1.40% 8.10% 3.91% 6.30%
CPU / Memory
12.500%
10.000%
7.500%
5.000%
2.500%
0%
98.1000%
98.0000%
97.9000%
97.8000%
97.7000%
97.6000%
19:55
20:00
20:05
cpu
memory
CPU cores utilization
30.00%
25.00%
20.00%
15.00%
10.00%
5.00%
0%
19:55
20:00
20:05
avg
0 4.077%
1 4.641%
2 3.456%
3 3.720%
Disk Used
16.1%
Network In / Out
100 kBps
80 kBps
60 kBps
40 kBps
19:52
19:54
19:56
19:58
20:00
min
inbound 46.0 kBps
outbound 46.7 kBps


proxy-log
gluconeogenesis
kaolinise
orthopsychiatrists
internet
proxy-prod
pushingnesses
reskillings
dharnas
embryologies
ladyloves
glaceing
perquisitors
hydroid
lists
lovats
lumas
oologists
inapprehensions
perigenesis
hirstie
dobber
customhouse
biparentally
lurchers
pressingness
unforcedly
illaudably
cicatrichules
refreshingly
regives
refectioners
aerohydroplane
taximan
cardecue
feblesse
pills
fastings
reauthori
ruthfully
howitzer
obscureness
surdity
hioi
peripteries
endothecium
untamednesses
pibroch
colinearity
kohlrabi
optimalise
burnishing
realnesses
bankrolls
outstay
gunslinger
phototrope
depauperate
rared
microreproductions
koumyss
scandia
trembly
dwarfed
harking
birgy
intertext
baggage
antique
manent
laggens
delamination
kinaras
unstep
androsphinx
soliloquized
refugees
miscomprehensions
obligately
upcurved
accelerant
coercively
vociferations
semiperimeter
indirectly
splashiness
sterical
fieldcraft
abstinent
keratinizes
sledgings
chandlers



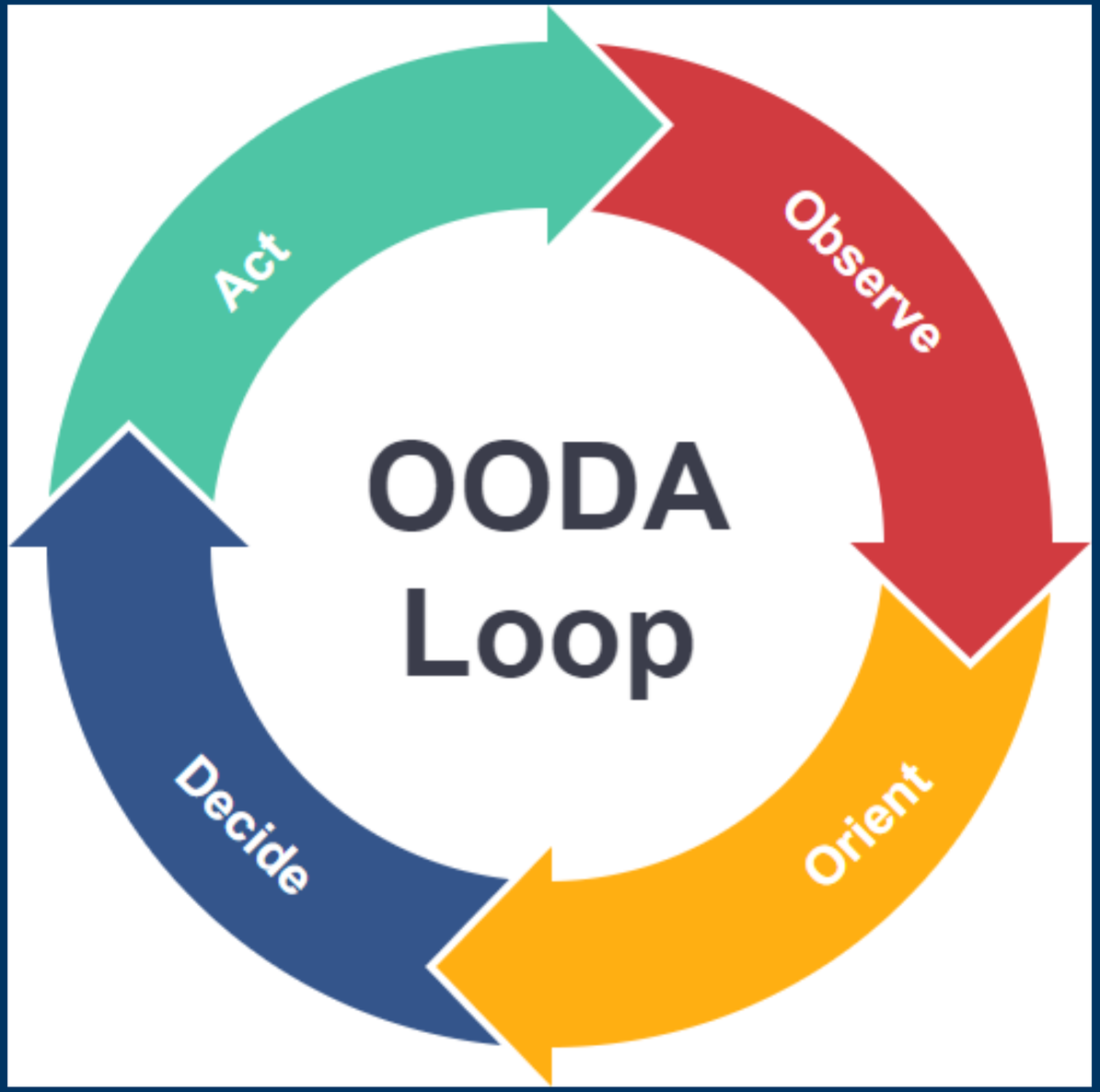
# Guessing isn't effective.

**We need a methodology!**

**We also need great tools.**

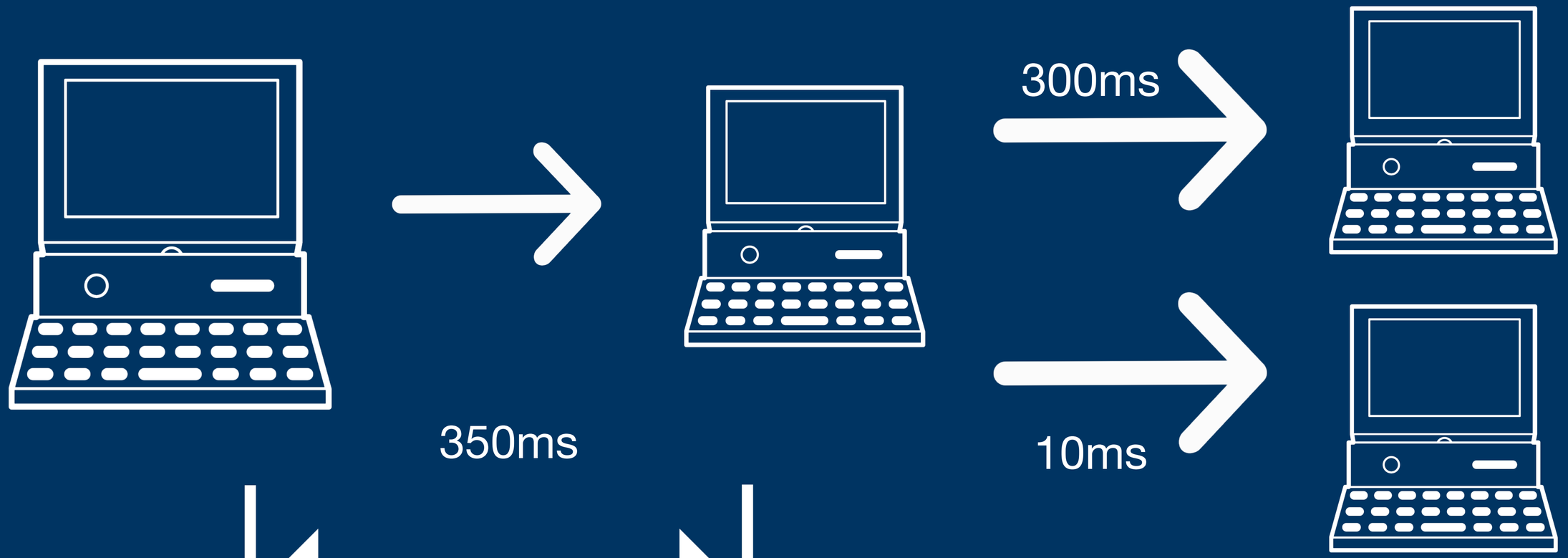
# Let's Start With The OODA Loop

John Boyd

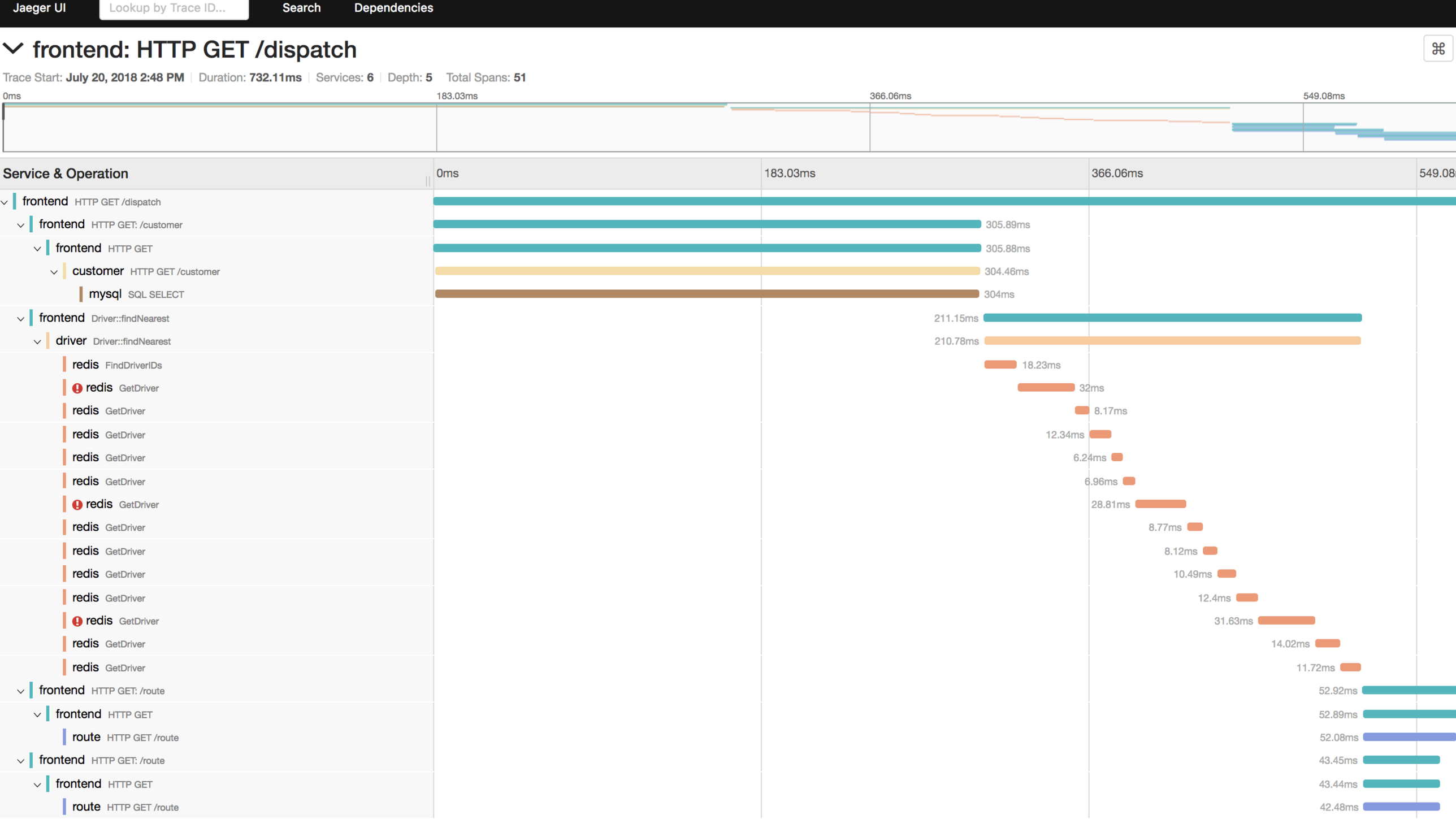


# Observe

# Distributed Tracing




TraceId=<16 byte random>


Jaeger UI
Lookup by Trace ID...
Search
Dependencies
frontend: HTTP GET /dispatch
Trace Start: July 20, 2018 2:48 PM | Duration: 732.11ms | Services: 6 | Depth: 5 | Total Spans: 51
0ms
183.03ms
366.06ms
549.08ms
Service & Operation
0ms
183.03ms
366.06ms
549.08
frontend HTTP GET /dispatch
frontend HTTP GET: /customer
305.89ms
frontend HTTP GET
305.88ms
customer HTTP GET /customer
304.46ms
mysql SQL SELECT
304ms
frontend Driver::findNearest
211.15ms
driver Driver::findNearest
210.78ms
redis FindDriverIDs
18.23ms
redis GetDriver
32ms
redis GetDriver
8.17ms
redis GetDriver
12.34ms
redis GetDriver
6.24ms
redis GetDriver
6.96ms
redis GetDriver
28.81ms
redis GetDriver
8.77ms
redis GetDriver
8.12ms
redis GetDriver
10.49ms
redis GetDriver
12.4ms
redis GetDriver
31.63ms
redis GetDriver
14.02ms
redis GetDriver
11.72ms
frontend HTTP GET: /route
52.92ms
frontend HTTP GET
52.89ms
route HTTP GET /route
52.08ms
frontend HTTP GET: /route
43.45ms
frontend HTTP GET
43.44ms
route HTTP GET /route
42.48ms

JAEGER UI | Search | Compare | System Architecture | Monitor | Source: Sock Shop Tracing | Lookup by Trace ID... | About Jaeger

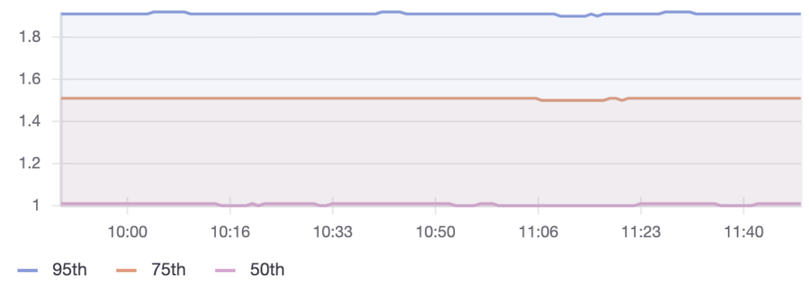
## Choose service

payment

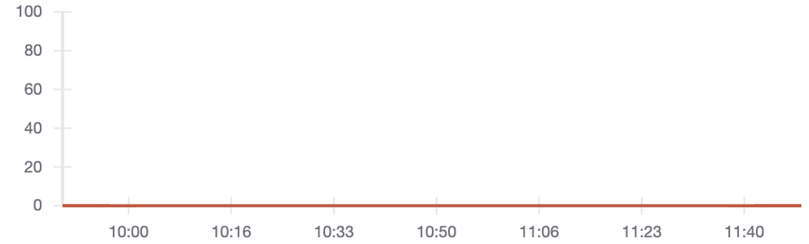
Aggregation of all "payment" metrics in selected timeframe. View all traces

Last 2 hour

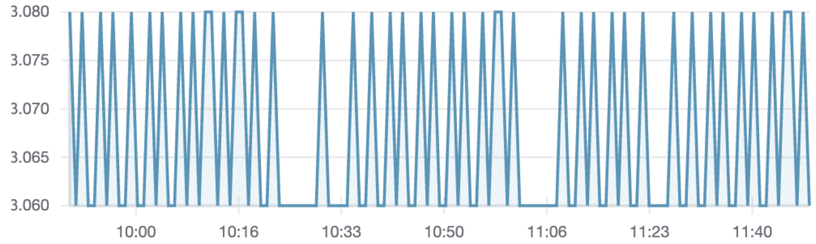
### Latency (ms)




### Error rate (%)




### Request rate (req/s)

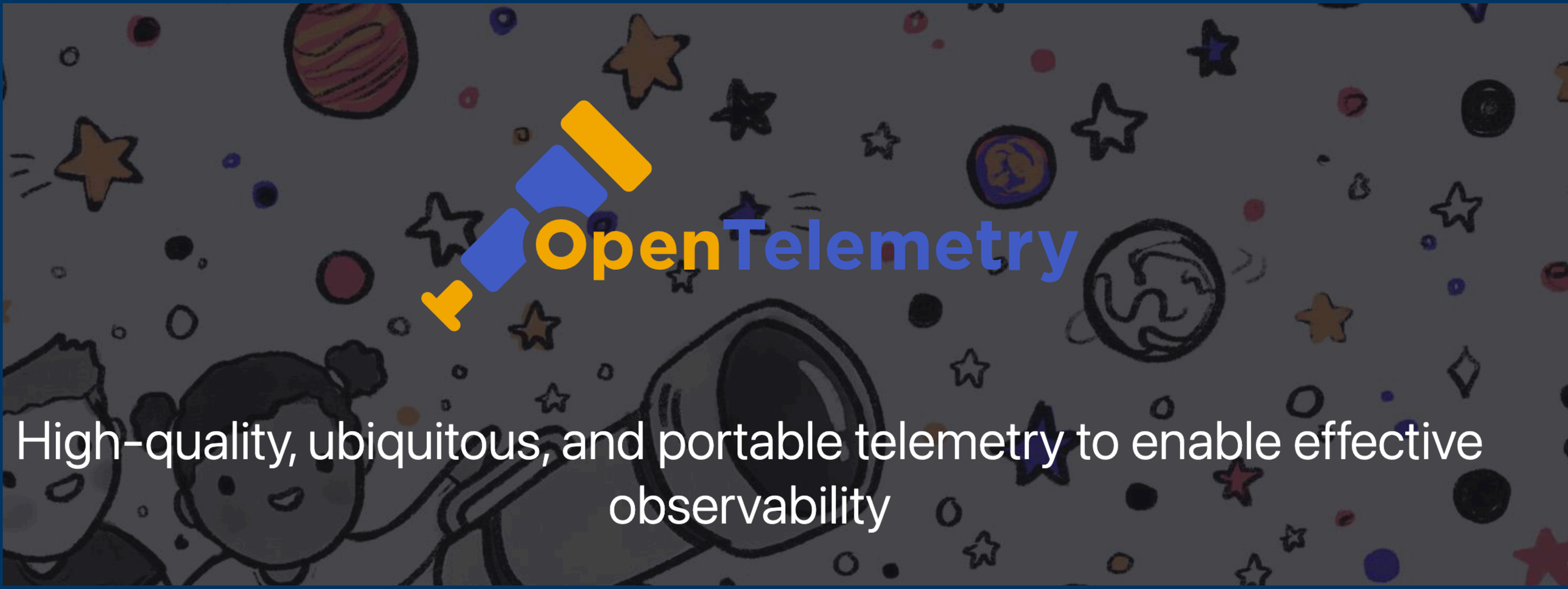



## Operations metrics under payment Over the last 2 hour

Search operation

| Name | P95 Latency | Request rate | Error rate | Impact 1 |
|---|---|---|---|---|
| payment | 1.92ms | 1.56 req/s | < 0.1% | |
| authorize payment | 1.9ms | 0.83 req/s | < 0.1% | |
| health check | 1.9ms | 0.67 req/s | No Data | |

1 | 20 / page




High-quality, ubiquitous, and portable telemetry to enable effective observability

# Structured, Contextual Logging

```
127.0.0.1 alice Alice [06/May/2021:11:26:42
+0200] "GET / HTTP/1.1" 200 3477
```

```
{
  "ip_address": "127.0.0.1",
  "user_identifier": "alice",
  "user_authentication": "Alice",
  "timestamp": "06/May/2021:11:26:42 +0200",
  "request_method": "GET",
  "request_url": "/",
  "protocol": "HTTP/1.1",
  "status_code": 200,
  "response_size": 3477
}
```

# We've Narrowed Down the Problem.

# Now What?

# Narrow it further.

# Dashboards!

CockroachDB

JB

# Transactions

Search transactions

Filters (0)

1-10 of 557 transactions

Last cleared Nov

| TRANSACTIONS | STATEMENTS | RETRIES | EXECUTION COUNT | ROWS AFFECTED | LATENCY |
|---|---|---|---|---|---|
| SELECT FROM customer | 5 | 0 | 13k | 6 | 3.1 s |
| SELECT FROM customer | 5 | 1 | 10k | 6 | 3.7 s |
| SELECT FROM customer | 5 | 0 | 10k | 6 | 3.8 s |
| INSERT INTO history | 4 | 1 | 9k | 4 | 3.2 s |
| INSERT INTO history | 4 | 1 | 7k | 4 | 3.8 s |
| INSERT INTO history | 4 | 1 | 7k | 4 | 3.9 s |
| UPDATE order | 24 | 0 | 2k | 150 | 181.1 ms |
| SELECT FROM order_line | 2 | 0 | 2k | 2 | 178.8 ms |
| UPDATE order | 24 | 0 | 2k | 150 | 185.5 ms |
| SELECT FROM order_line | 2 | 0 | 2k | 2 | 201.0 ms |

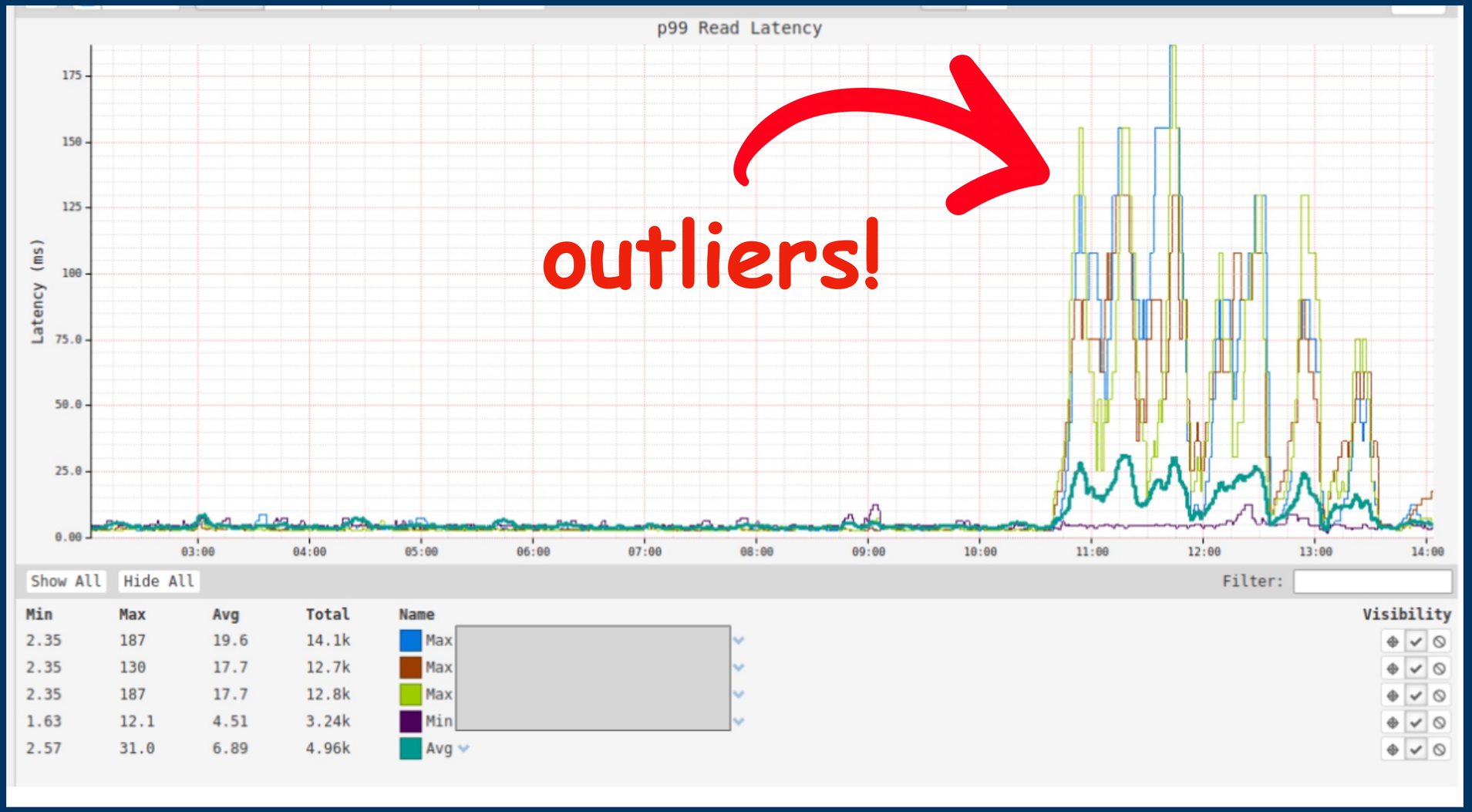
# Narrow it further.

# Even More Dashboards!

# Common Mistakes

- Averaging Percentiles
  - "average p99 latency is X"
- Showing feel good metrics
  - p75, average

# Make It Easy


p99 Read Latency
Latency (ms)
175
150
125
100
75.0
50.0
25.0
0.00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
outliers!
Show All
Hide All
Filter:
Min
Max
Avg
Total
Name
Visibility
2.35
187
19.6
14.1k
Max
2.35
130
17.7
12.7k
Max
2.35
187
17.7
12.8k
Max
1.63
12.1
4.51
3.24k
Min
2.57
31.0
6.89
4.96k
Avg

# USE Method

# Utilization

*the average time that the resource was busy servicing work*

*a percent over a time interval. eg, "one cpu is running at 90% utilization".*

# Saturation

*the degree to which the resource has extra work which it can't service, often queued*

*as a queue length. eg, "the CPUs have an average run queue length of four".*

# Errors

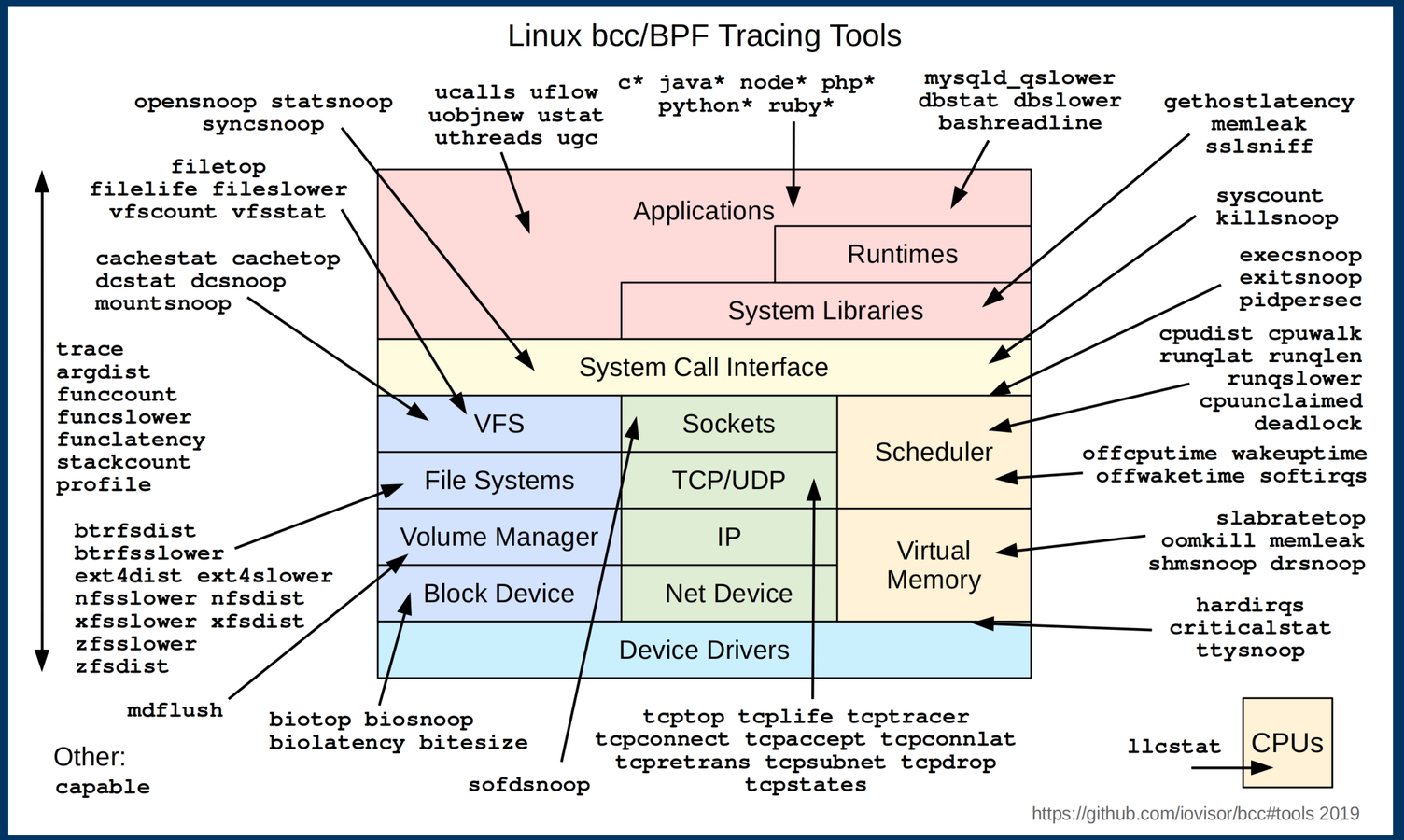
*the count of error events*

*scalar counts. eg, "this network interface has had fifty late collisions".*

```
root@ubuntu-vm:~# iostat -dmc 2
Linux 5.15.0-84-generic (ubuntu-vm)       09/24/2023          _aarch64_  (2 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.77    0.00    5.38   42.82    0.00   51.03

Device             tps    MB_read/s    MB_wrtn/s    MB_dscd/s    MB_read    MB_wrtn    MB_dscd
dm-0              0.00         0.00         0.00         0.00          0          0          0
dm-1          12165.50        47.52         0.00         0.00         95          0          0
loop0             0.00         0.00         0.00         0.00          0          0          0
loop1             0.00         0.00         0.00         0.00          0          0          0
loop2             0.00         0.00         0.00         0.00          0          0          0
loop3             0.00         0.00         0.00         0.00          0          0          0
sr0               0.00         0.00         0.00         0.00          0          0          0
vda           12165.50        47.52         0.00         0.00         95          0          0
```

```
root@ubuntu-vm:~# mpstat -P ALL 2
Linux 5.15.0-84-generic (ubuntu-vm)       09/24/2023          _aarch64_  (2 CPU)

03:12:50 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
03:12:52 AM  all    0.78    0.00    4.40   43.26    0.00    0.00    0.00    0.00    0.00   51.55
03:12:52 AM    0    1.03    0.00    4.62   39.49    0.00    0.00    0.00    0.00    0.00   54.87
03:12:52 AM    1    0.52    0.00    4.19   47.12    0.00    0.00    0.00    0.00    0.00   48.17

03:12:52 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
03:12:54 AM  all    0.78    0.00    4.91   42.89    0.00    0.00    0.00    0.00    0.00   51.42
03:12:54 AM    0    1.03    0.00    5.15   44.33    0.00    0.00    0.00    0.00    0.00   49.48
03:12:54 AM    1    0.52    0.00    4.66   41.45    0.00    0.00    0.00    0.00    0.00   53.37
```

# bcc-tools & bpftrace



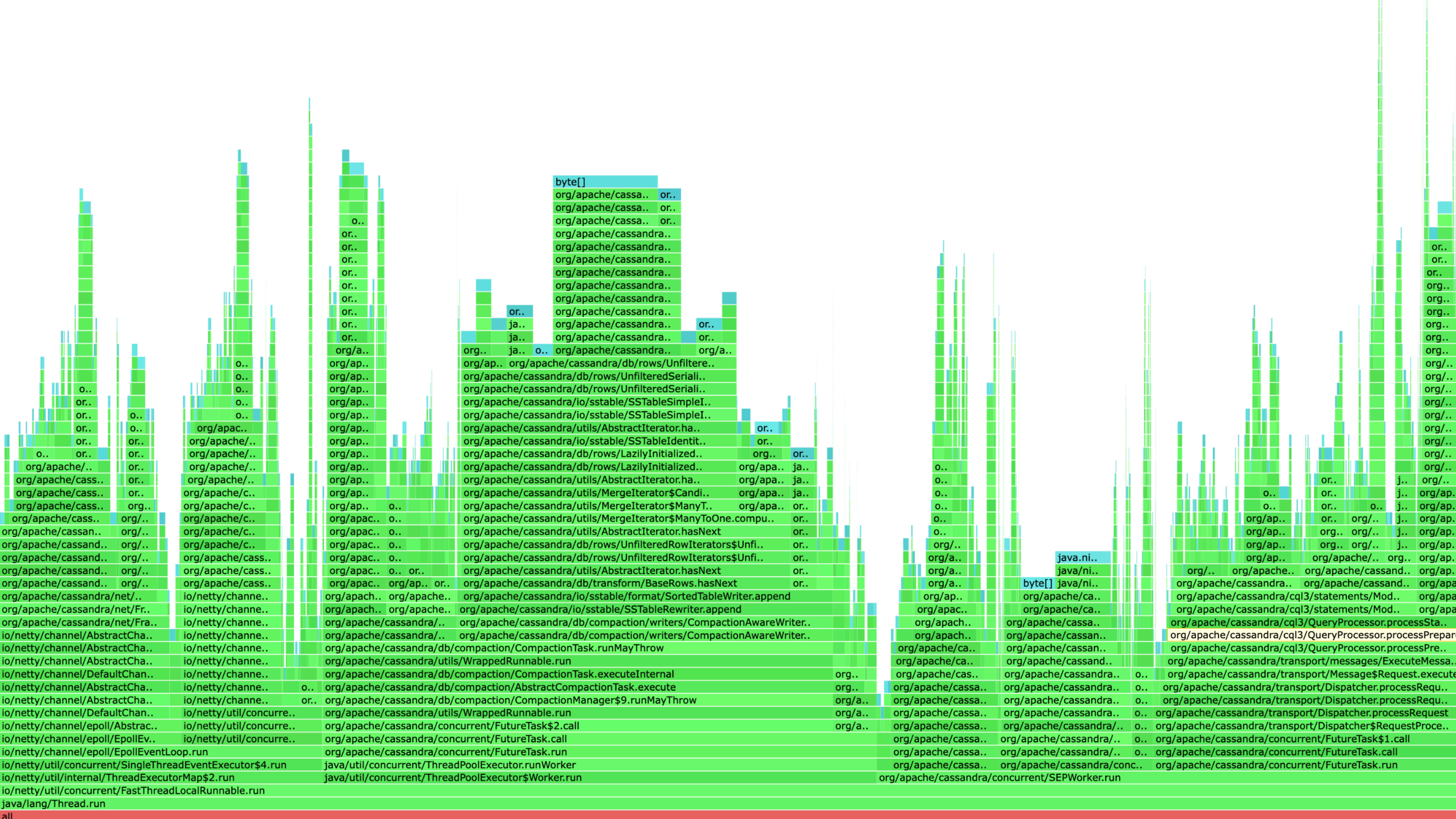
```
$ xfsdist 1 10 -m
Tracing XFS operation latency... Hit Ctrl-C to end.

23:51:18:

operation = 'read'
msecs               : count     distribution
0 -> 1          : 110925   |****************************************|

23:51:19:

operation = 'read'
msecs               : count     distribution
0 -> 1          : 30715    |****************************************|
2 -> 3          : 73       |                                        |
4 -> 7          : 19       |                                        |
8 -> 15         : 1        |                                        |
```

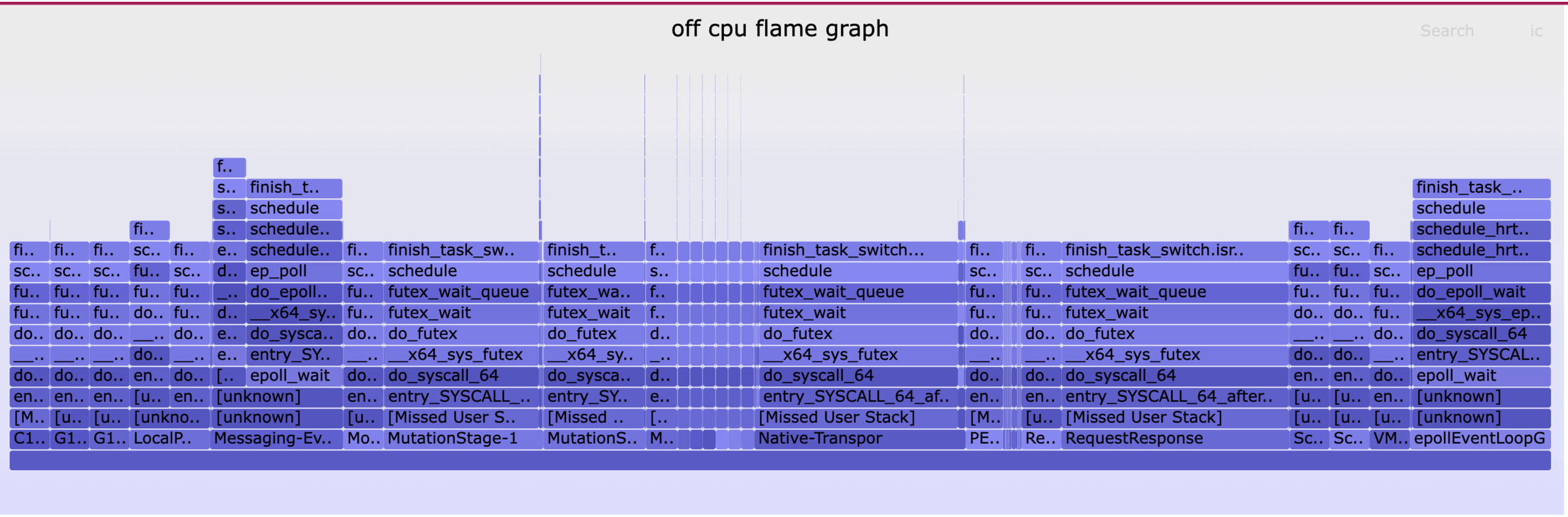
```
root@ubuntu-vm:~# cachestat 2
    HITS   MISSES  DIRTIES HITRATIO   BUFFERS_MB  CACHED_MB
       0    24016        0    0.00%           31        709
       0    24288        0    0.00%           31        677
       0    23686        0    0.00%           31        705
       0    22041        0    0.00%           31        664
       0    20342        0    0.00%           31        680
       0    22785        0    0.00%           31        705
       0    22714        0    0.00%           31        666
       0    22904        0    0.00%           31        692
       0    22805        0    0.00%           31        654
       0    22782        0    0.00%           31        679
       0    22999        0    0.00%           31        705
       0    22851        0    0.00%           31        667
       0    22758        0    0.00%           31        692
```

```
$ root@ubuntu-vm:~# biolatency 2
Tracing block device I/O... Hit Ctrl-C to end.

     usecs               : count     distribution
         0 -> 1          : 0        |                                        |
         2 -> 3          : 0        |                                        |
         4 -> 7          : 0        |                                        |
         8 -> 15         : 0        |                                        |
        16 -> 31         : 0        |                                        |
        32 -> 63         : 4093     |**********                              |
        64 -> 127        : 15175    |****************************************|
       128 -> 255        : 250      |                                        |
       256 -> 511        : 108      |                                        |
       512 -> 1023       : 44       |                                        |
      1024 -> 2047       : 17       |                                        |
      2048 -> 4095       : 9        |                                        |
      4096 -> 8191       : 3        |                                        |
      8192 -> 16383      : 4        |                                        |
     16384 -> 32767      : 3        |                                        |
     32768 -> 65535      : 1        |                                        |
```

# Profiling is great.

# Flame Graphs Are Amazing


byte[]
org/apache/cassandra/db/rows/Unfiltere..
org/apache/cassandra/db/rows/UnfilteredSeriali..
org/apache/cassandra/db/rows/UnfilteredSeriali..
org/apache/cassandra/io/sstable/SSTableSimpleI..
org/apache/cassandra/io/sstable/SSTableSimpleI..
org/apache/cassandra/utils/AbstractIterator.ha..
org/apache/cassandra/io/sstable/SSTableIdentit..
org/apache/cassandra/db/rows/LazilyInitialized..
org/apache/cassandra/db/rows/LazilyInitialized..
org/apache/cassandra/utils/AbstractIterator.ha..
org/apache/cassandra/utils/MergeIterator$Candi..
org/apache/cassandra/utils/MergeIterator$ManyT..
org/apache/cassandra/utils/MergeIterator$ManyToOne.compu..
org/apache/cassandra/utils/AbstractIterator.hasNext
org/apache/cassandra/db/rows/UnfilteredRowIterators$Unfi..
org/apache/cassandra/db/rows/UnfilteredRowIterators$Unfi..
org/apache/cassandra/utils/AbstractIterator.hasNext
org/apache/cassandra/db/transform/BaseRows.hasNext
org/apache/cassandra/io/sstable/format/SortedTableWriter.append
org/apache/cassandra/io/sstable/SSTableRewriter.append
org/apache/cassandra/db/compaction/writers/CompactionAwareWriter..
org/apache/cassandra/db/compaction/writers/CompactionAwareWriter..
org/apache/cassandra/db/compaction/CompactionTask.runMayThrow
org/apache/cassandra/utils/WrappedRunnable.run
org/apache/cassandra/db/compaction/CompactionTask.executeInternal
org/apache/cassandra/db/compaction/AbstractCompactionTask.execute
org/apache/cassandra/db/compaction/CompactionManager$9.runMayThrow
org/apache/cassandra/utils/WrappedRunnable.run
org/apache/cassandra/concurrent/FutureTask$2.call
org/apache/cassandra/concurrent/FutureTask.call
org/apache/cassandra/concurrent/FutureTask.run
java/util/concurrent/ThreadPoolExecutor.runWorker
java/util/concurrent/ThreadPoolExecutor$Worker.run
org/apache/cassandra/concurrent/SEPWorker.run
java/lang/Thread.run
all


off cpu flame graph
Search
ic
f..
s..
finish_t..
finish_task_..
s..
schedule
schedule
fi..
s..
schedule..
fi..
fi..
schedule_hrt..
fi..
fi..
fi..
sc..
fi..
e..
schedule..
fi..
finish_task_sw..
finish_t..
f..
finish_task_switch...
fi..
fi..
finish_task_switch.isr..
sc..
sc..
fi..
schedule_hrt..
sc..
sc..
sc..
fu..
sc..
d..
ep_poll
sc..
schedule
schedule
s..
schedule
sc..
sc..
schedule
fu..
fu..
sc..
ep_poll
fu..
fu..
fu..
fu..
fu..
_..
do_epoll..
fu..
futex_wait_queue
futex_wa..
f..
futex_wait_queue
fu..
fu..
futex_wait_queue
fu..
fu..
fu..
do_epoll_wait
fu..
fu..
fu..
do..
fu..
d..
__x64_sy..
fu..
futex_wait
futex_wait
f..
futex_wait
fu..
fu..
futex_wait
do..
do..
fu..
__x64_sys_ep..
do..
do..
do..
__..
do..
e..
do_sysca..
do..
do_futex
do_futex
d..
do_futex
do..
do..
do_futex
__..
__..
do..
do_syscall_64
__..
__..
__..
do..
__..
e..
entry_SY..
__..
__x64_sys_futex
__x64_sy..
_..
__x64_sys_futex
__..
__..
__x64_sys_futex
do..
do..
__..
entry_SYSCAL..
do..
do..
do..
en..
do..
[..
epoll_wait
do..
do_syscall_64
do_sysca..
d..
do_syscall_64
do..
do..
do_syscall_64
en..
en..
do..
epoll_wait
en..
en..
en..
[u..
en..
[unknown]
en..
entry_SYSCALL_..
entry_SY..
e..
entry_SYSCALL_64_af..
en..
en..
entry_SYSCALL_64_after..
[u..
[u..
en..
[unknown]
[M..
[u..
[u..
[unkno..
[unknown]
[u..
[Missed User S..
[Missed ..
[..
[Missed User Stack]
[M..
[u..
[Missed User Stack]
[u..
[u..
[u..
[unknown]
C1..
G1..
G1..
LocalP..
Messaging-Ev..
Mo..
MutationStage-1
MutationS..
M..
Native-Transpor
PE..
Re..
RequestResponse
Sc..
Sc..
VM..
epollEventLoopG

# Wrap it up buddy

- Keep narrowing down the problem
- Distributed Tracing For High Level
- Use Structured Logging To Build a Complete Picture Across Services
- Analyze Performance Problems With eBPF and Flame Graphs

# Thank You!