Latency Analysis

ioan.moldovan@tora.com

PyData Cluj-Napoca, meetup #13, 2020.05.28

PDF slides are incomplete (no animations, no footnotes, no presenter notes).

Open latencies.html?presentme=true for a nicer version.

Summary

- What to look at
- How distributions look like
- How to measure
- How to investigate

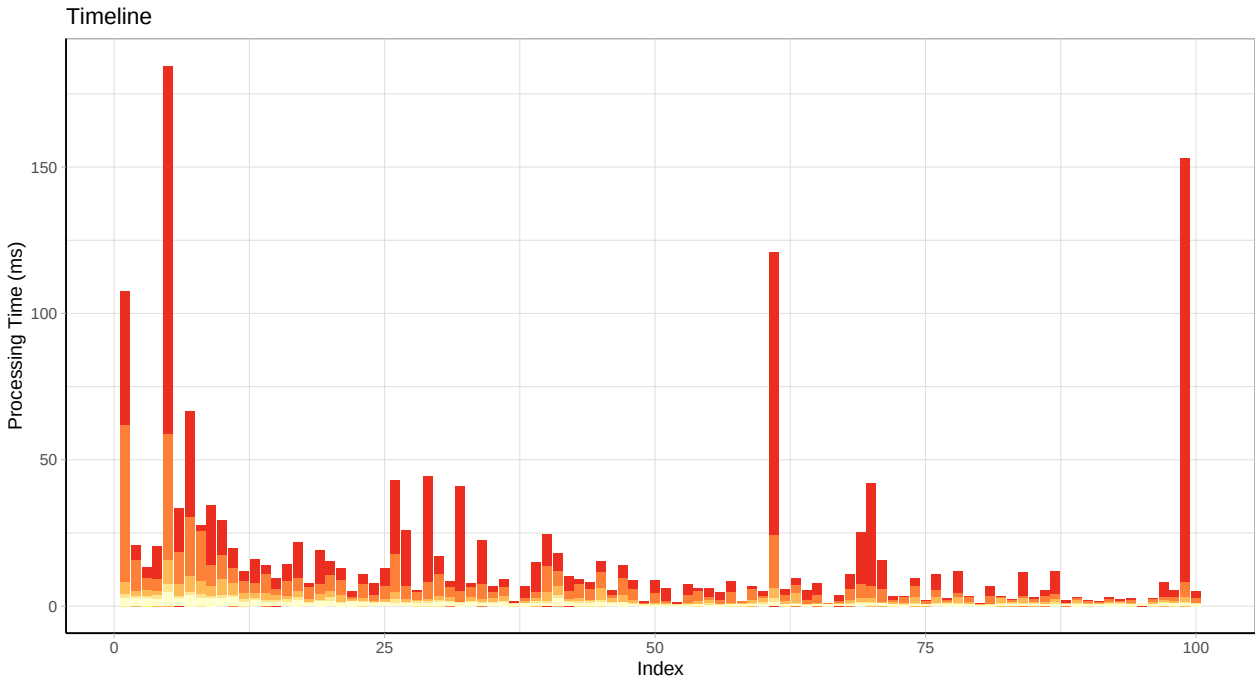
Monitoring

Figure 1:

https://play.grafana.org/

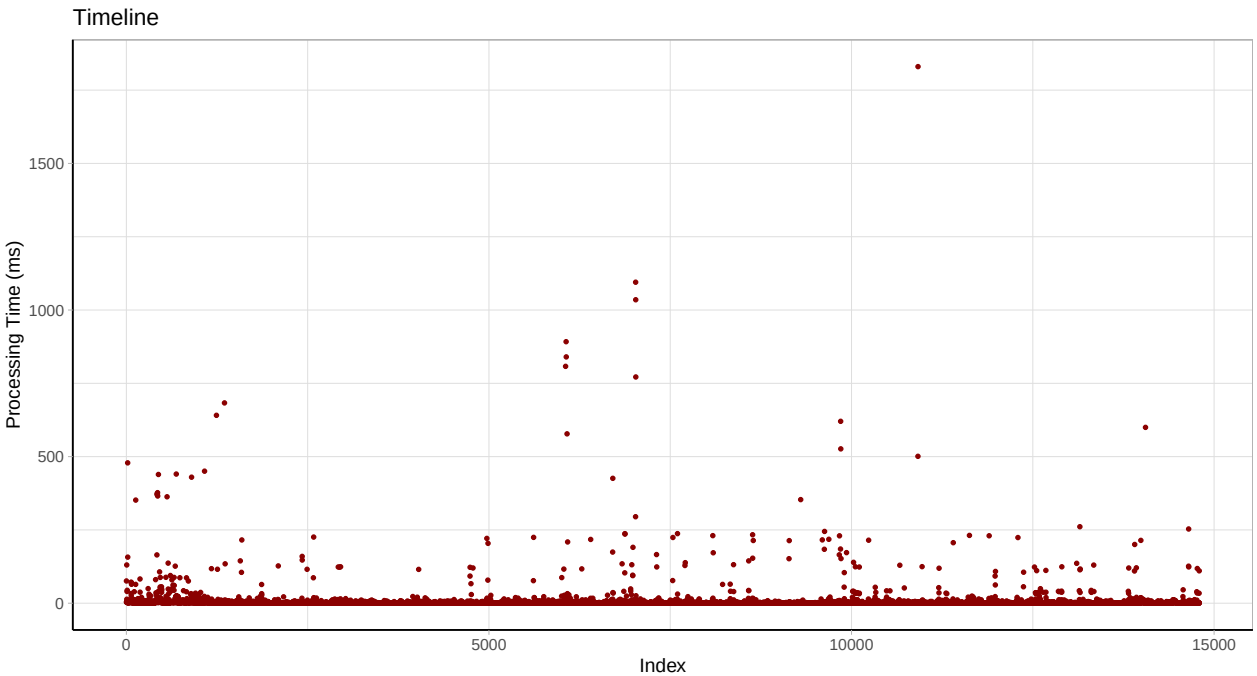
Timelines

NO

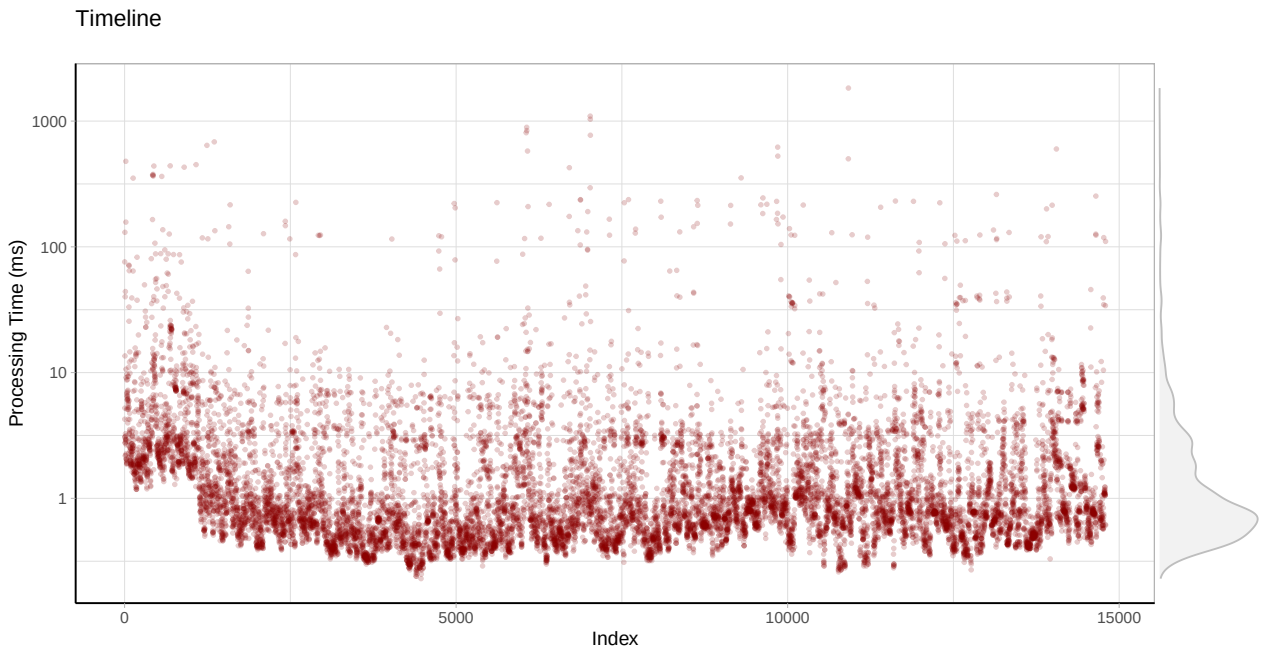


Timelines

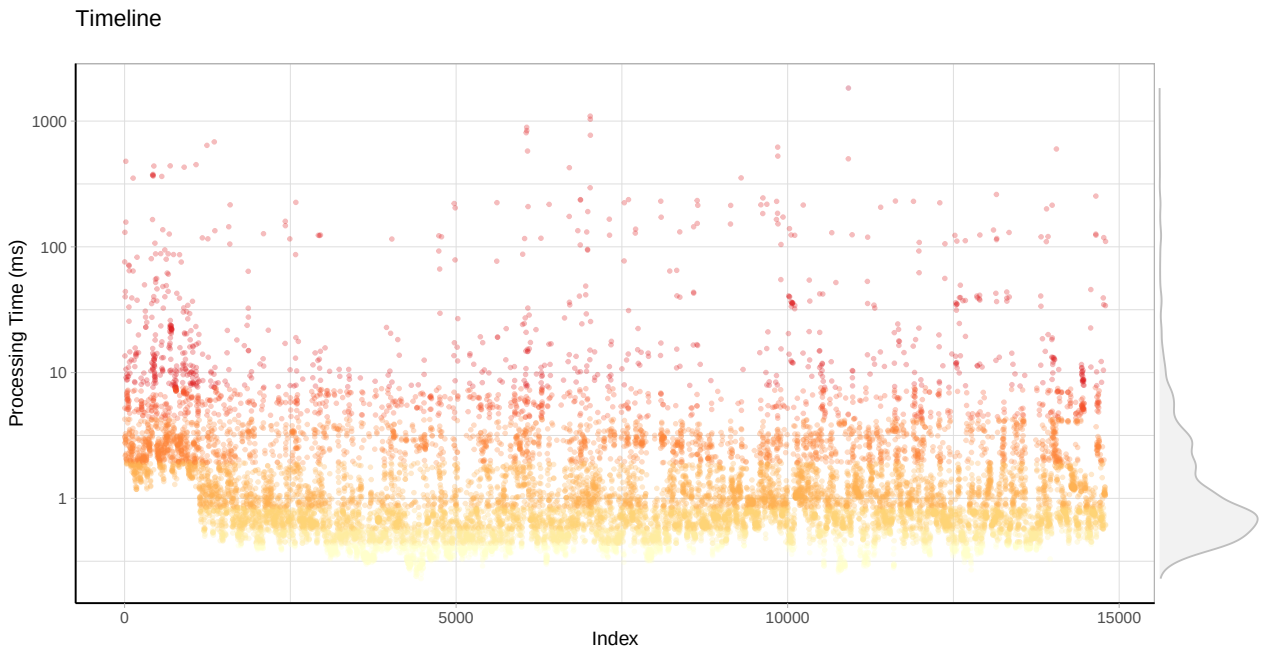
NO



Timelines

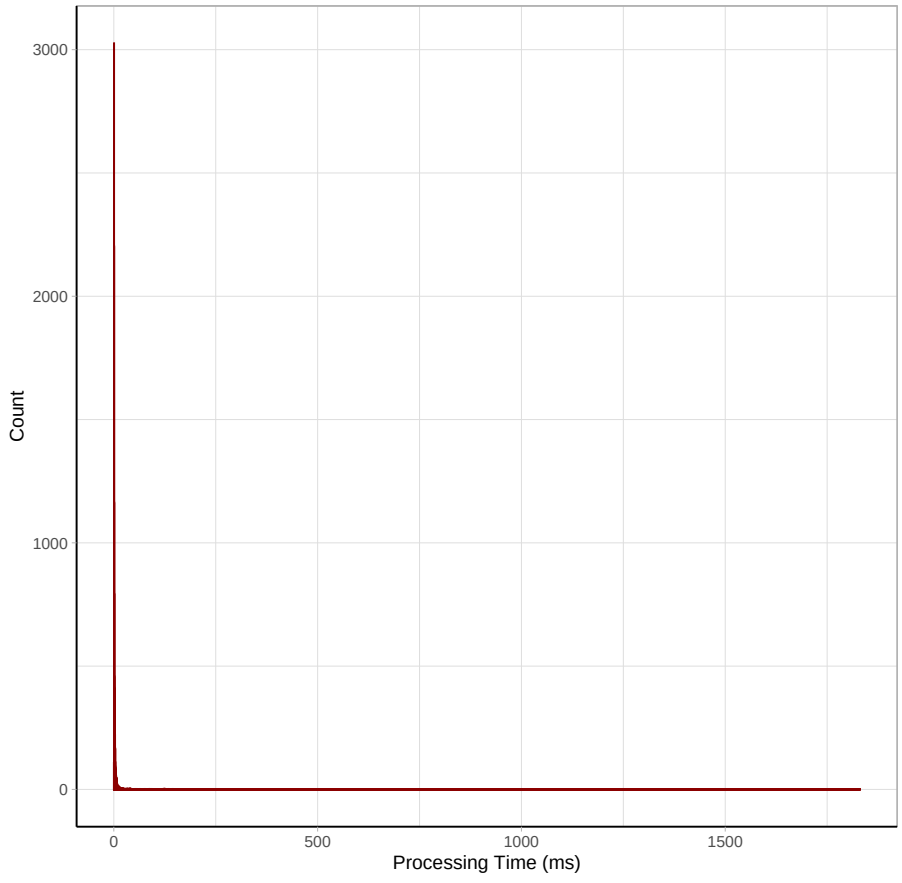

Timeline
Processing Time (ms)
1000
100
10
1
0
5000
10000
15000
Index

Timelines


Timeline
Processing Time (ms)
1000
100
10
1
0
5000
10000
15000
Index

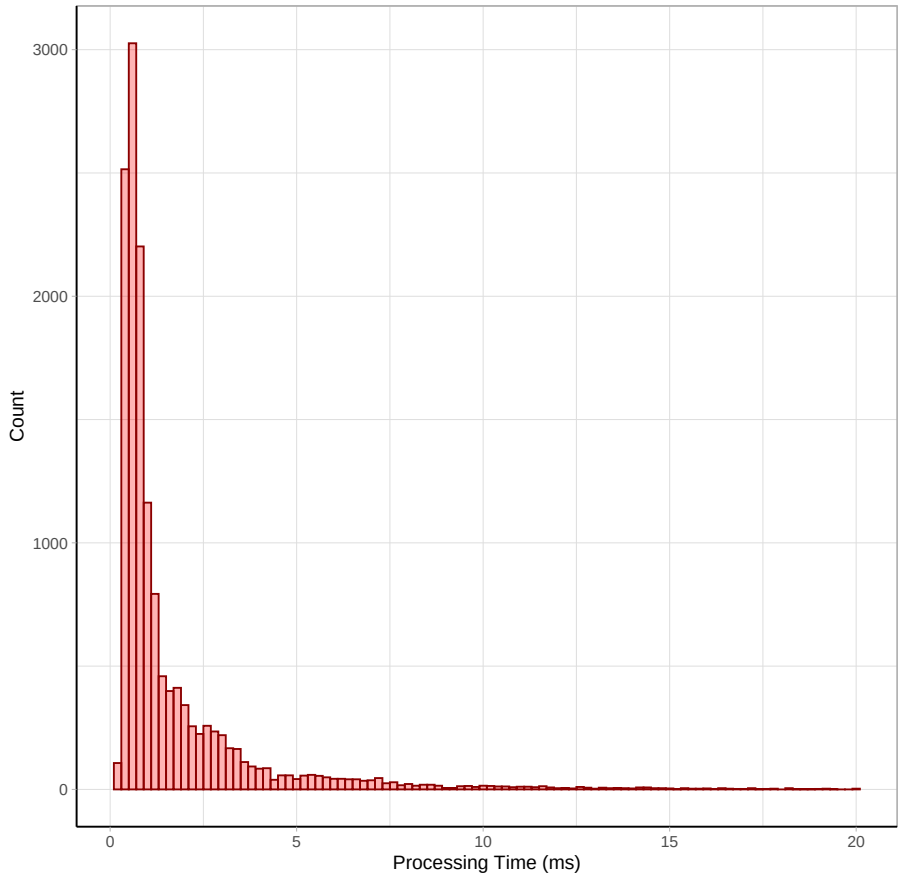
Histograms

NO

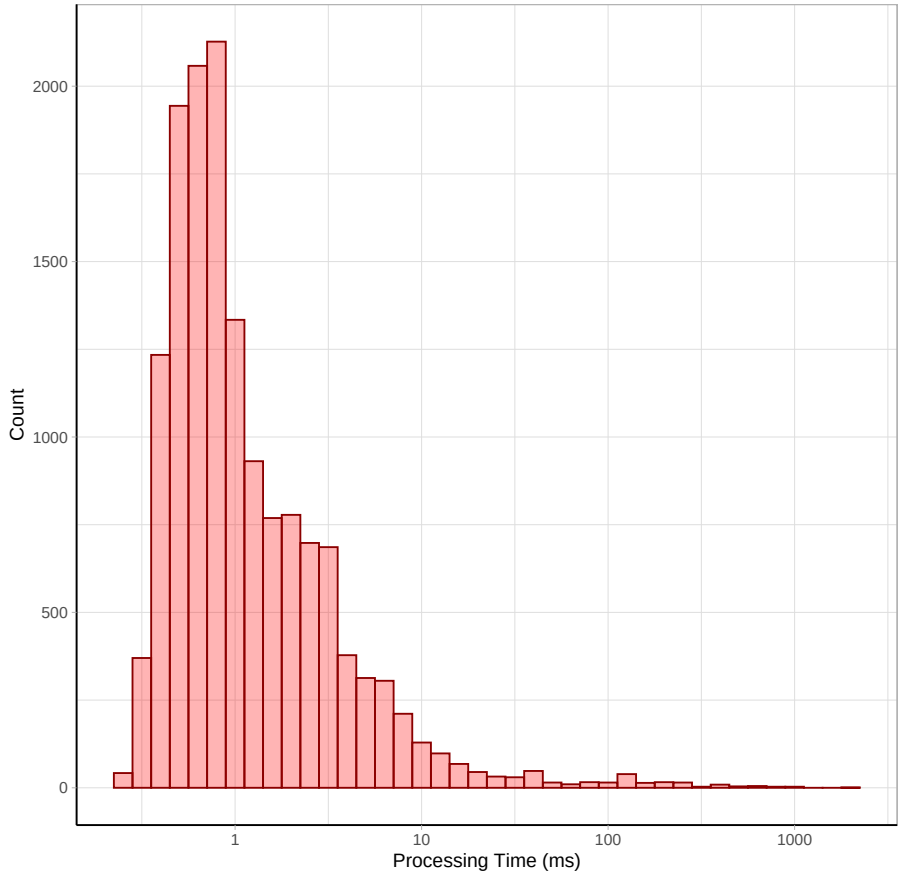


Histograms

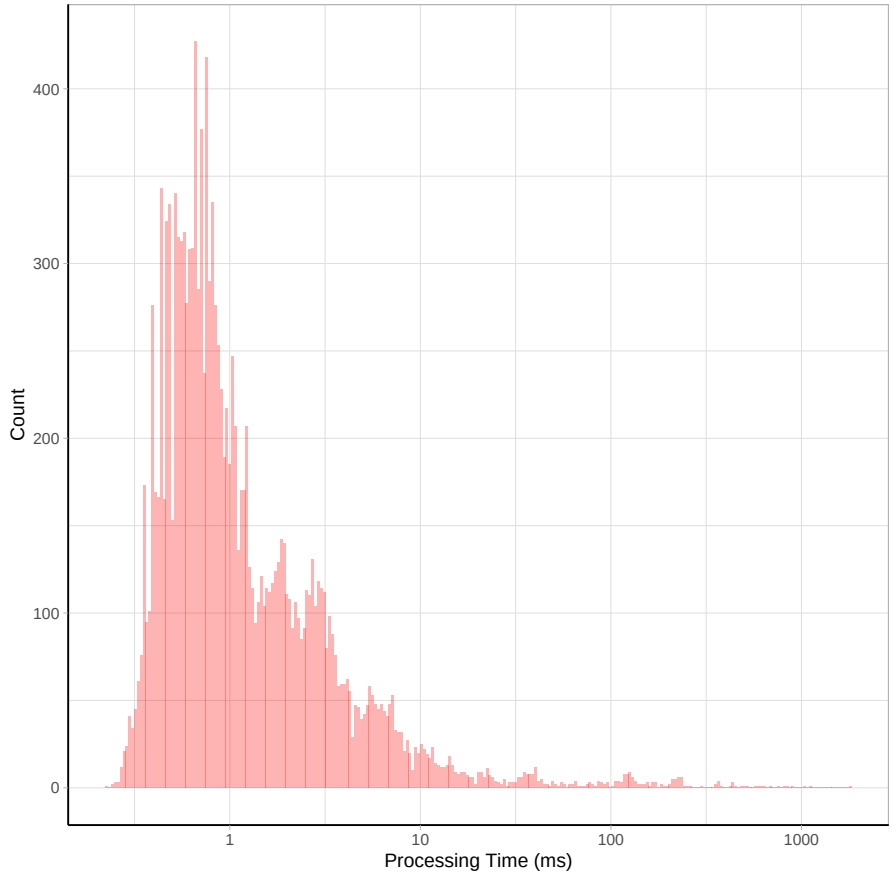
NO



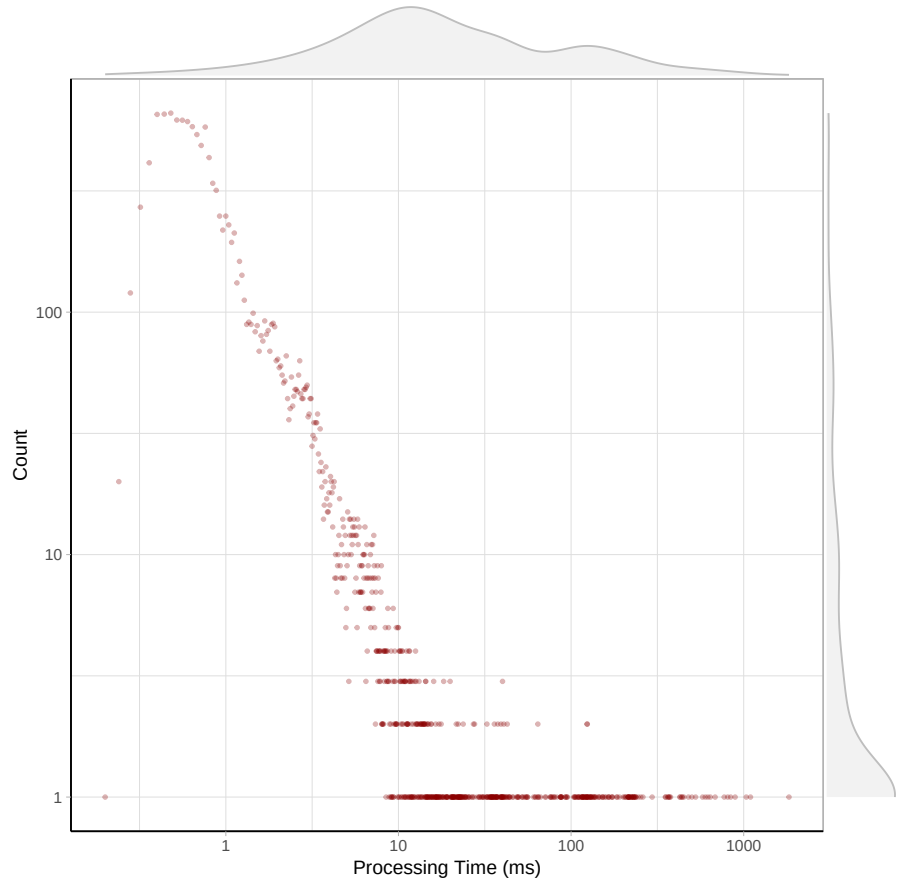
Histograms



Histograms



Histograms



Histograms

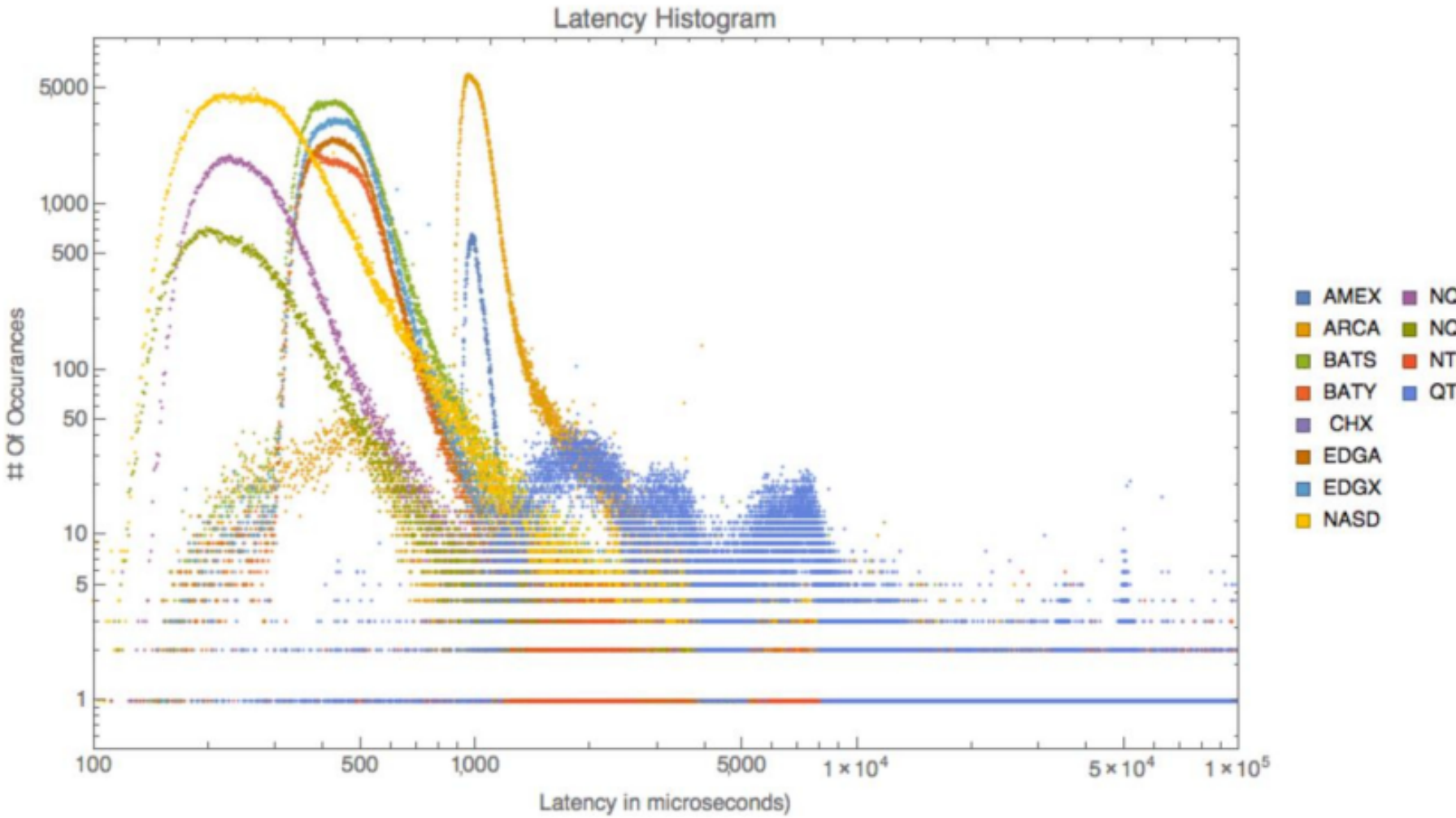
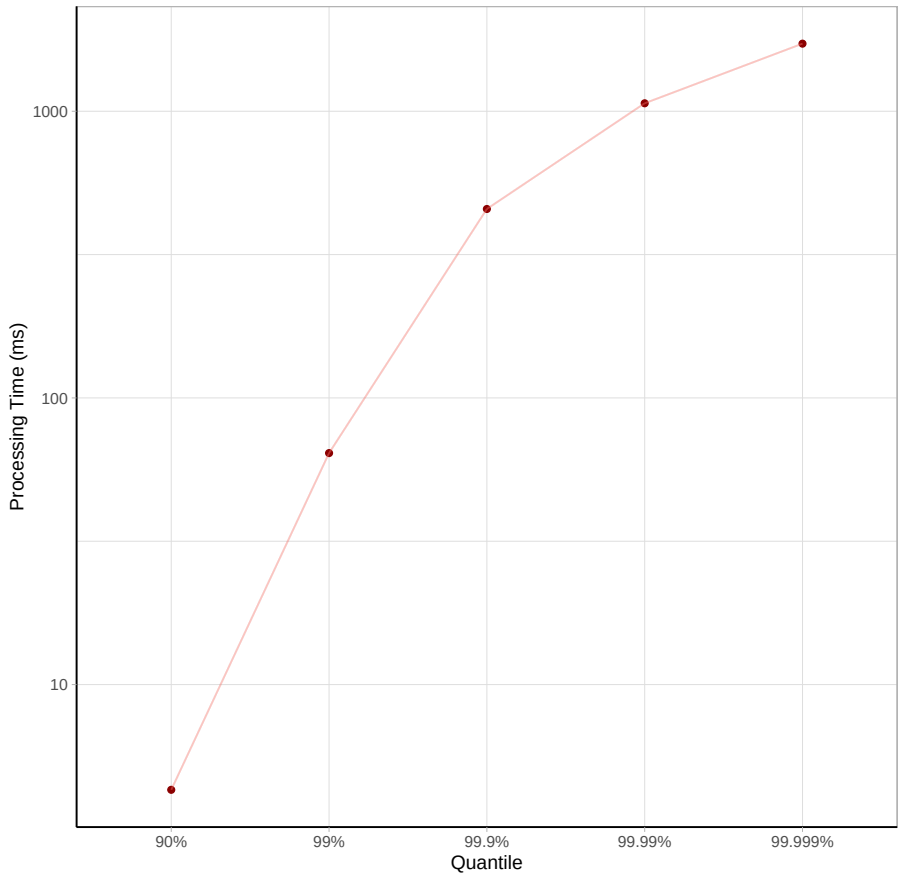


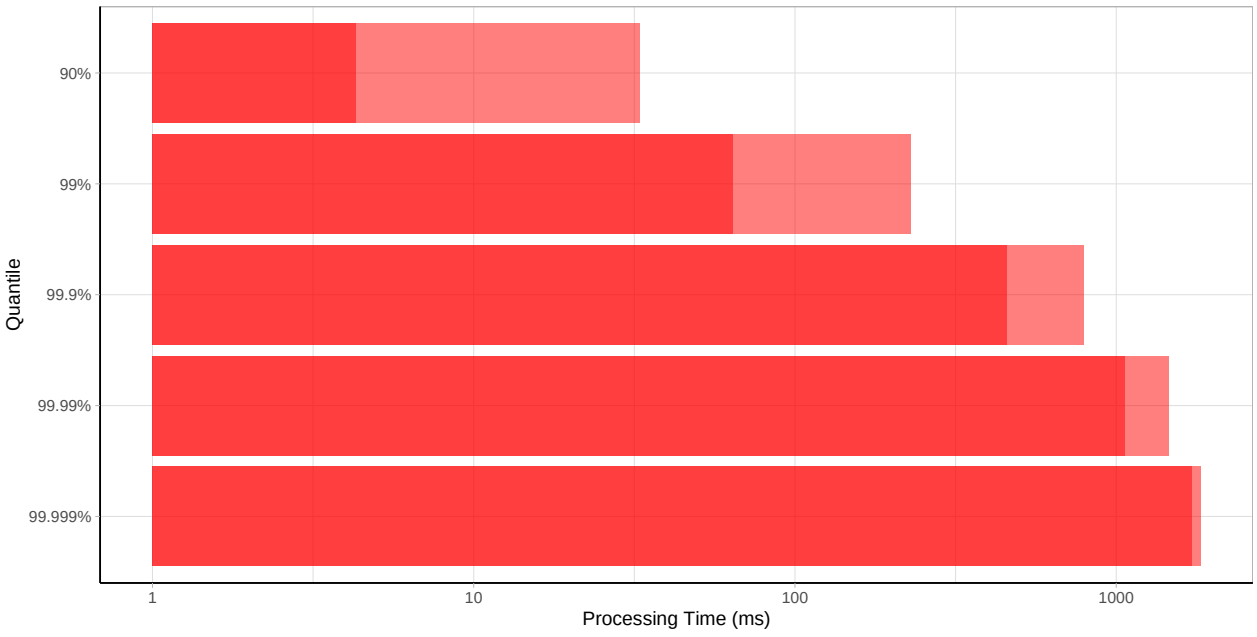
Figure 13. Latency histogram for Apple stock traded on 11 August 2015 by exchange for AAPL (truncated at 10^5 microseconds).

Figure 2:

Quantiles plots



Quantiles plots



Distribution tail plot

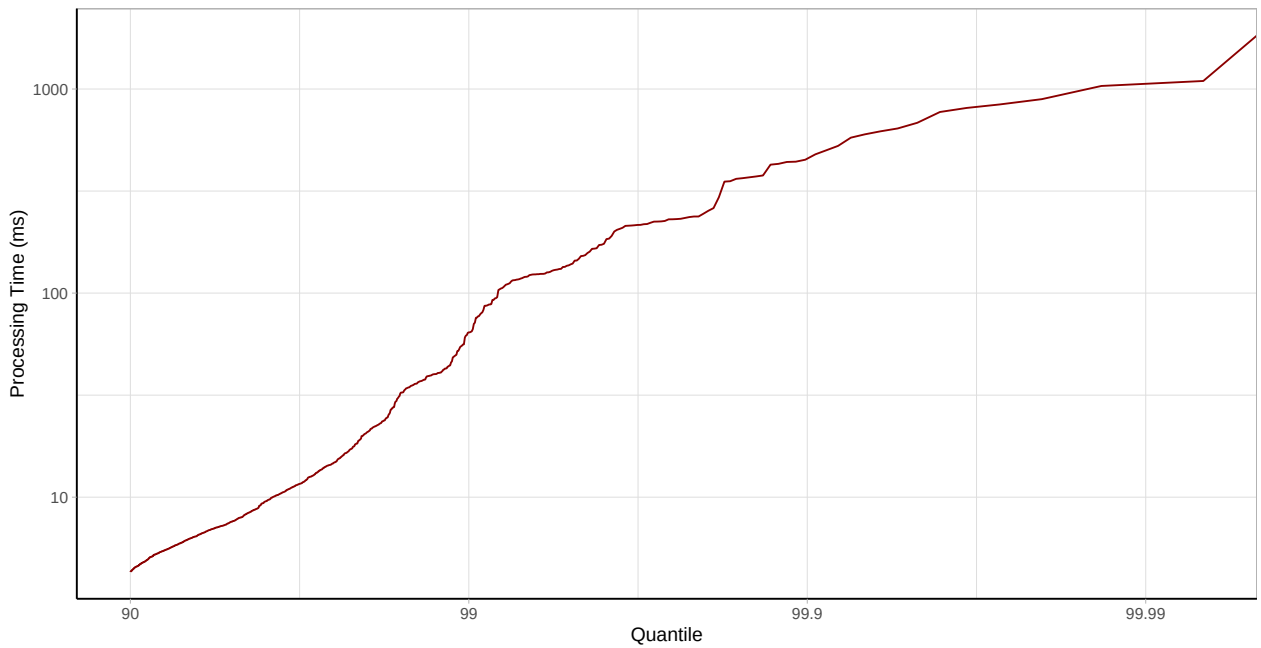


Chart by factors

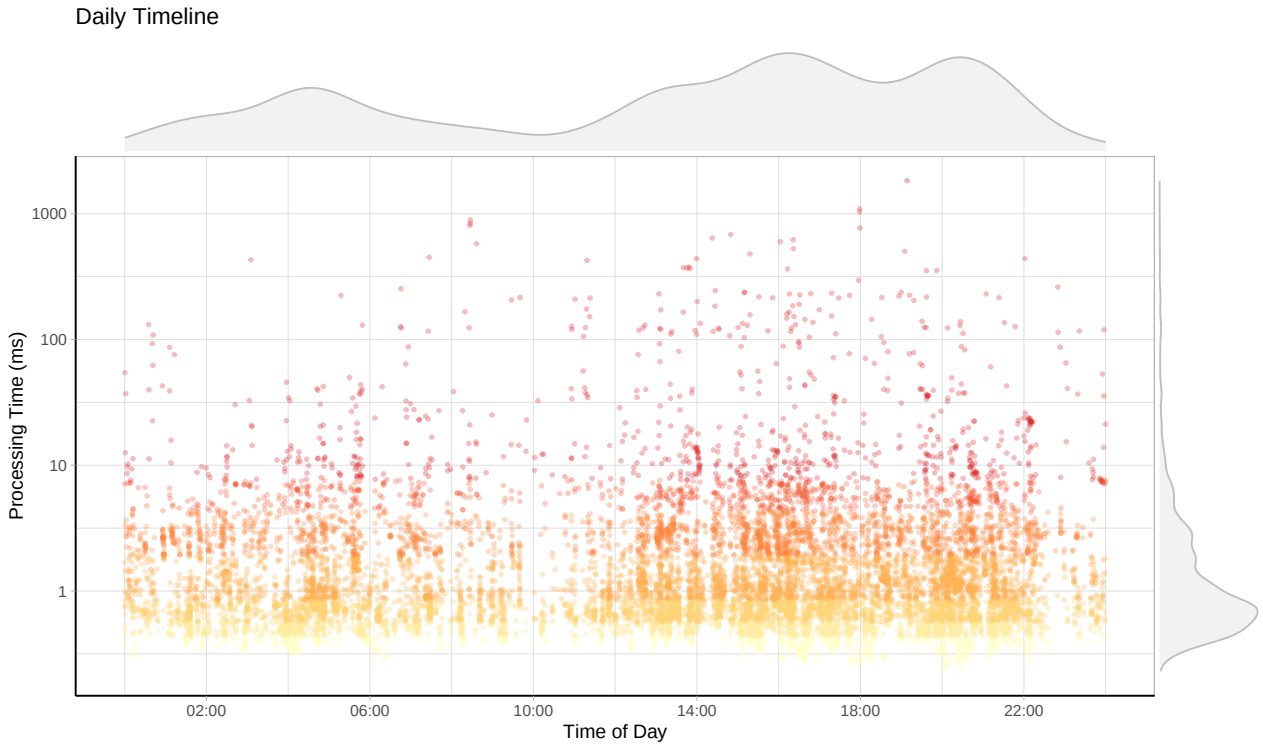
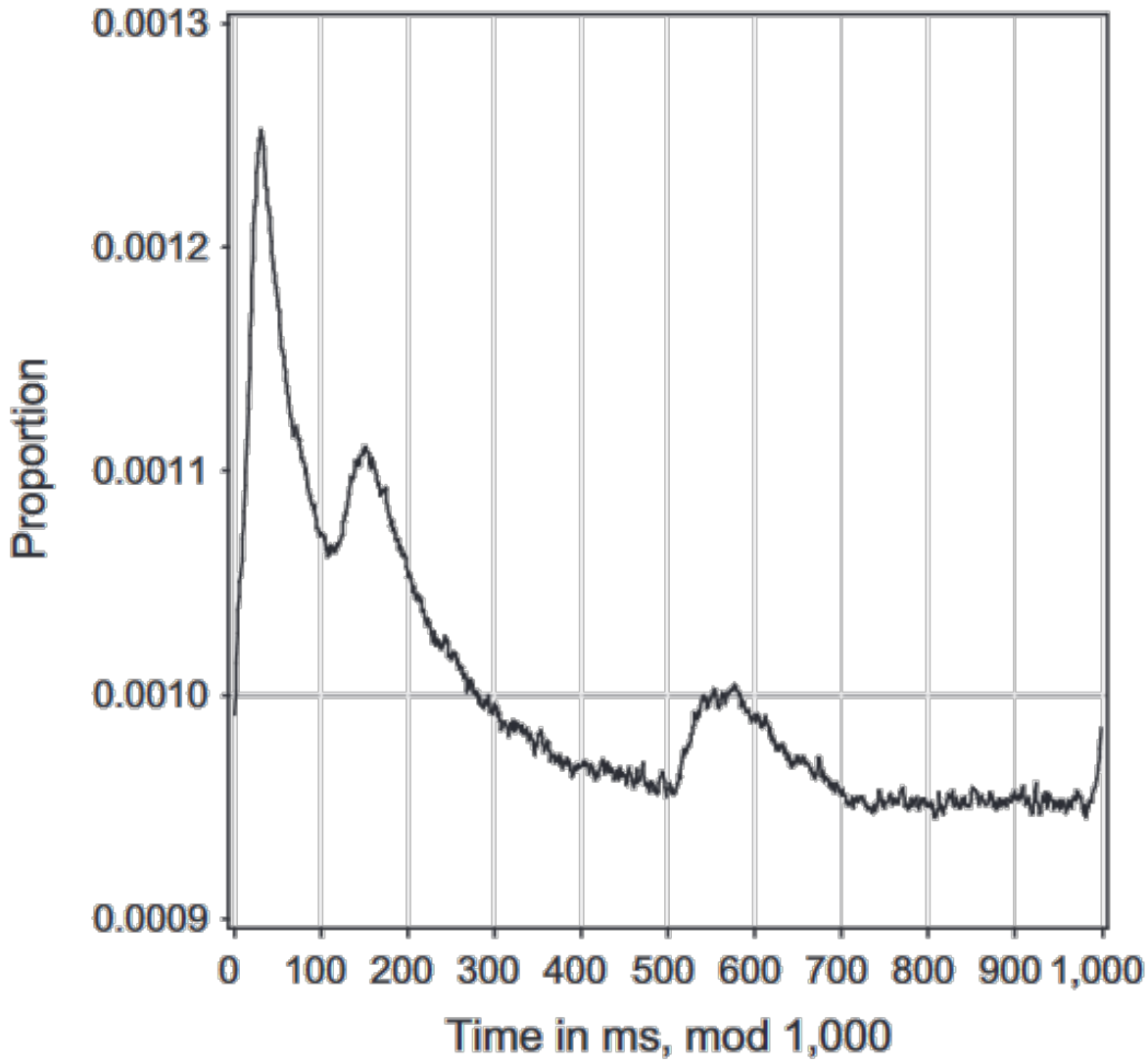


Chart by factors



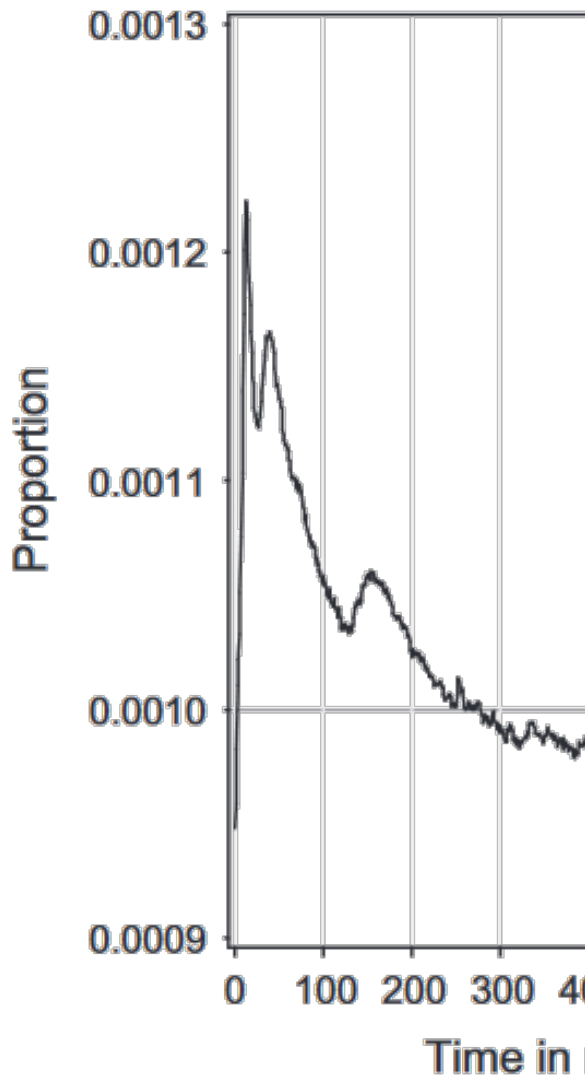


Figure 2:

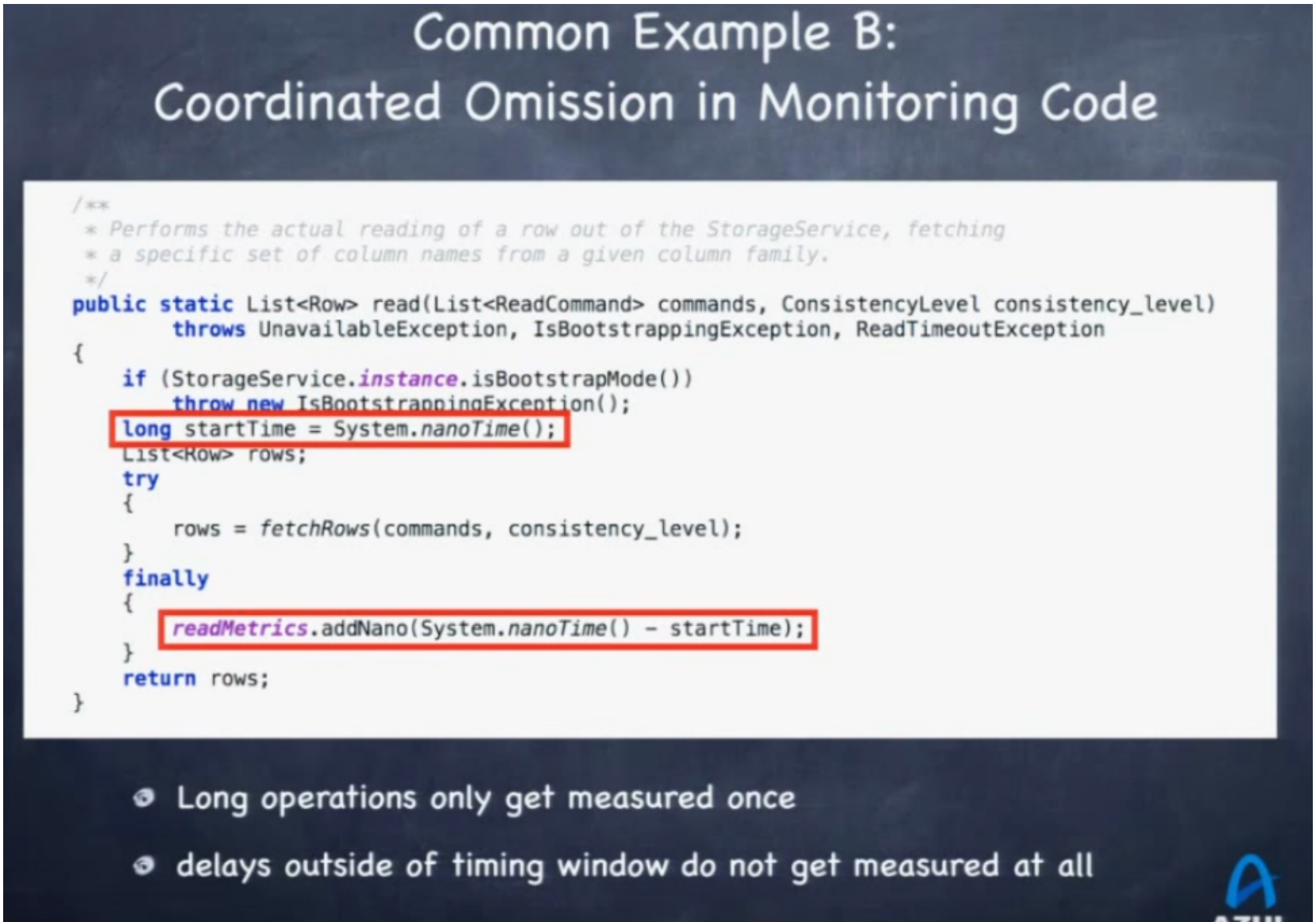
Take a step back

What are we measuring?

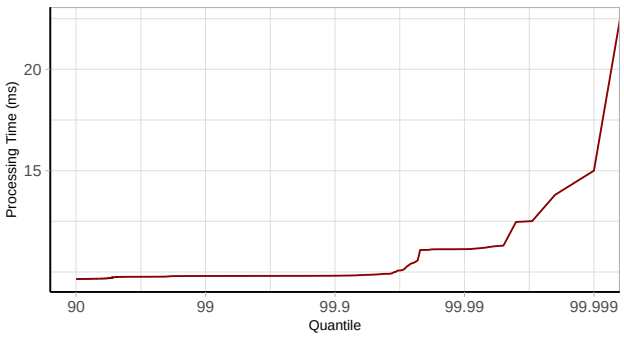
- processing times
- roundtrip times

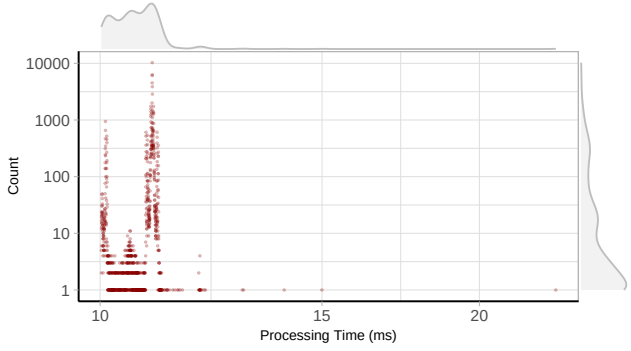
Coordinated omission

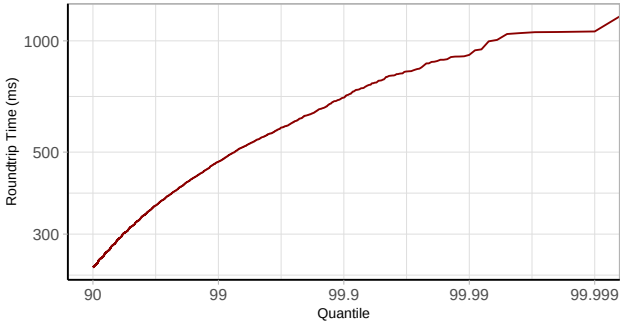
Gil Tene - “How NOT to Measure Latency”https://www.youtube.com/watch?v=lJ8ydIuPFeU&t=15m50

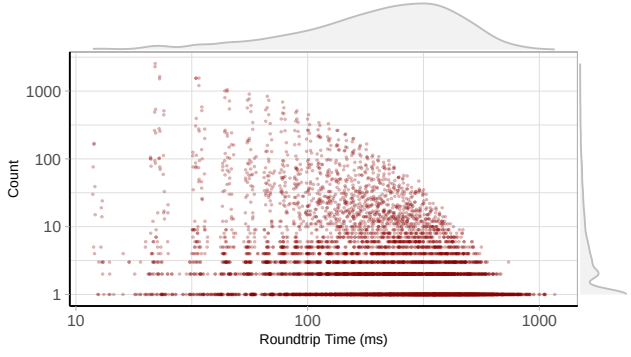


Coordinated omission





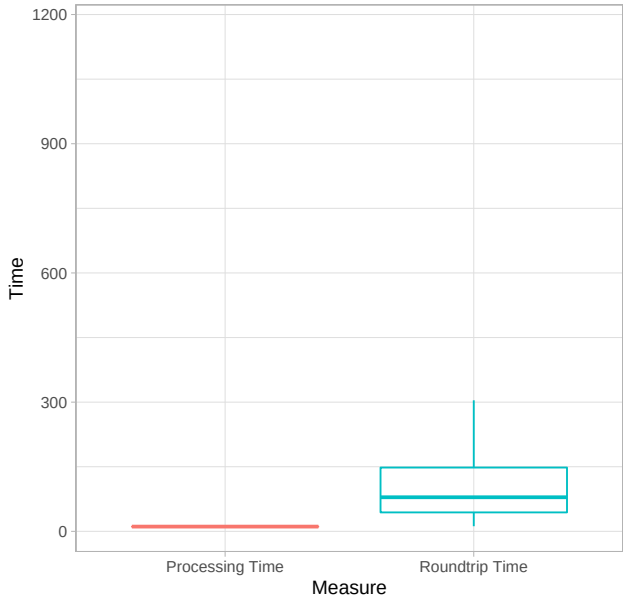




Why not averages and few percentiles?

NO

	Processing Time	Roundtrip Time
Average	10.92	112.62
Stdev	0.25	99.48
50%	11	79.13
75%	11.01	148.12
90%	11.03	243.98
95%	11.09	313.89
99%	11.12	471.01



What does the 99th percentile mean?

Whatsapp (2018)

~65 billion messages per day ~500M daily active users Assuming iid delays and avg # requests per user, 99th percentile slowness will affect (1 - 0.99^(65/0.5)) = 72% of the users

Depending on the number of requests per client:

	1	10	100	1000	1000	10000	1000000
95	5%	40.13%	99.41%	>99%	>99%	>99%	>99%
99	1%	9.56%	63.4%	>99%	>99%	>99%	>99%
99.9	0.1%	1%	9.52%	63.23%	63.23%	>99%	>99%
99.99	0.01%	0.1%	1%	9.52%	9.52%	63.21%	>99%
99.999	<0.01%	0.01%	0.1%	1%	1%	9.52%	>99%
99.9999	<0.01%	<0.01%	0.01%	0.1%	0.1%	1%	63.21%

Grafana dashboard again

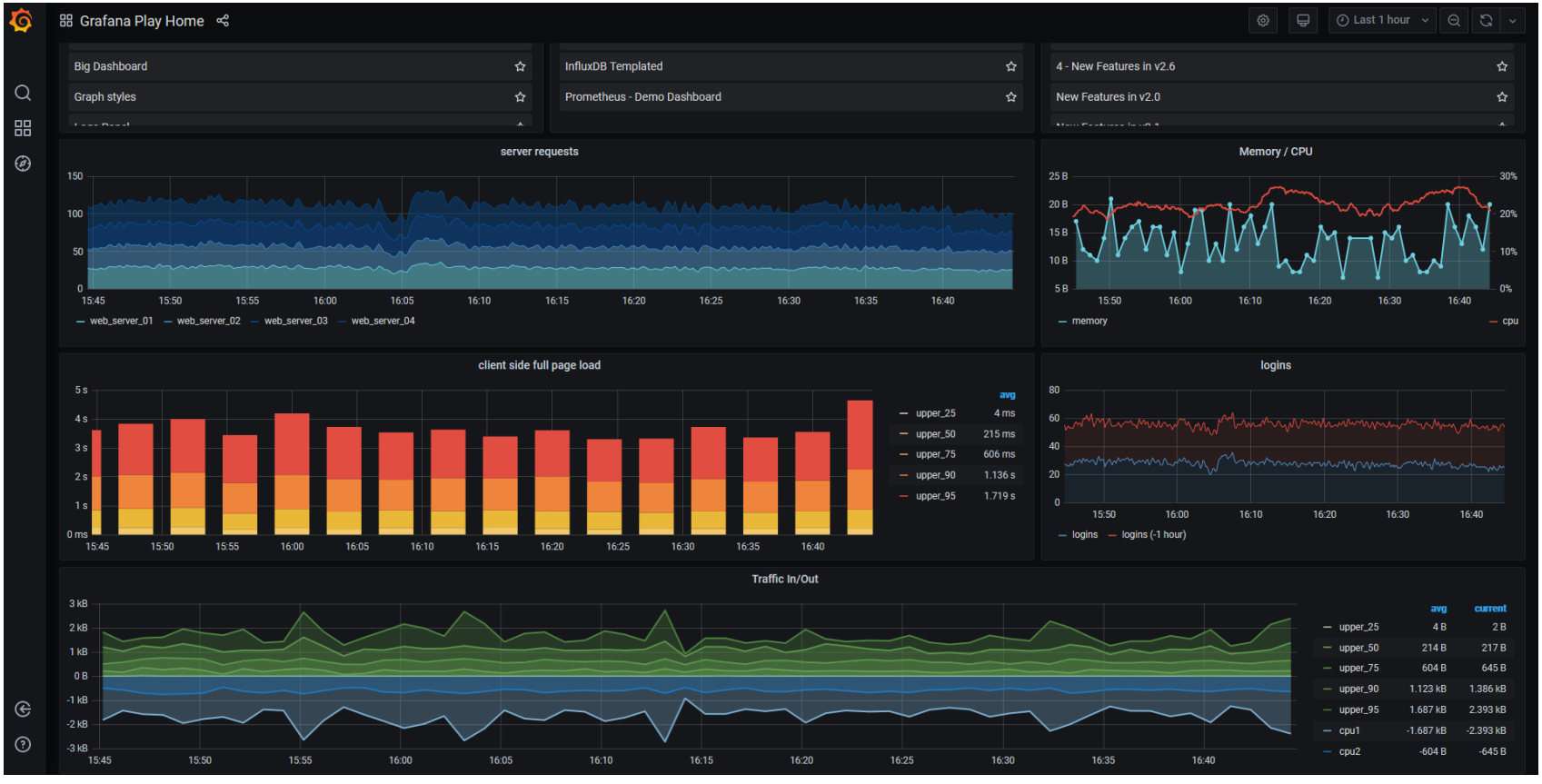
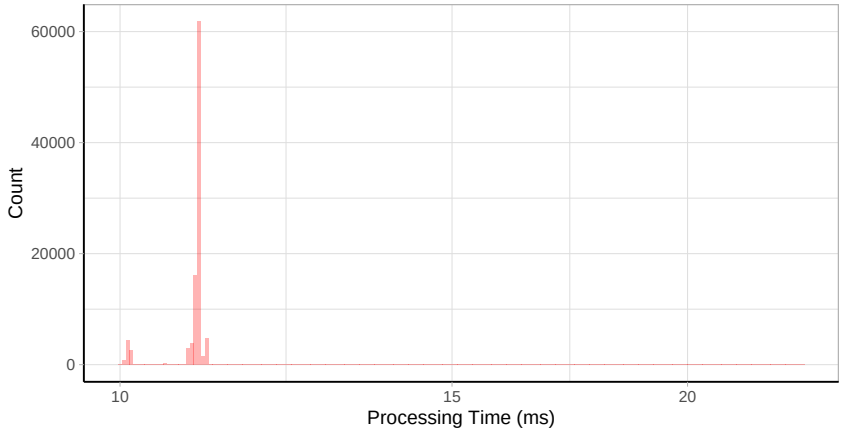
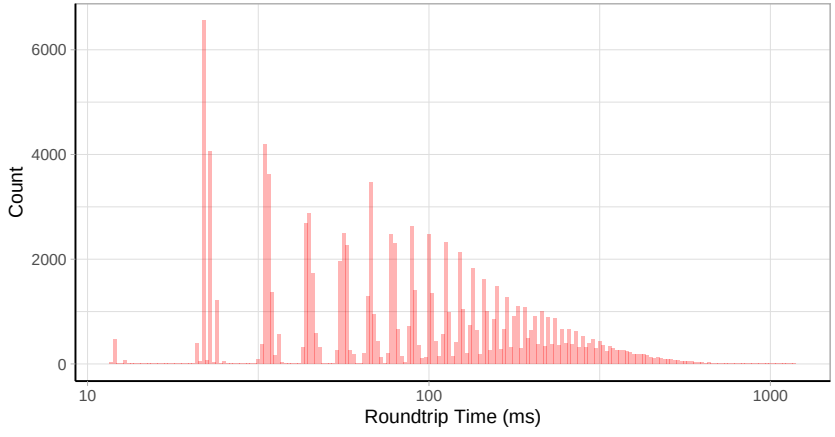


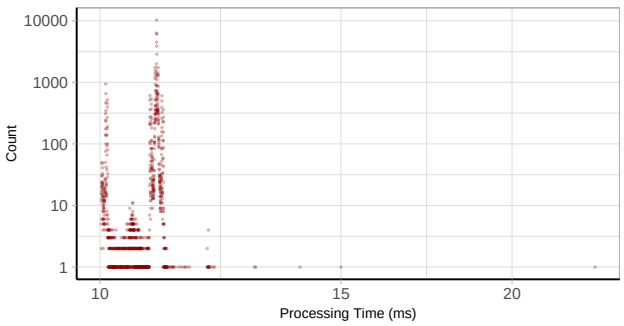
Figure 6:

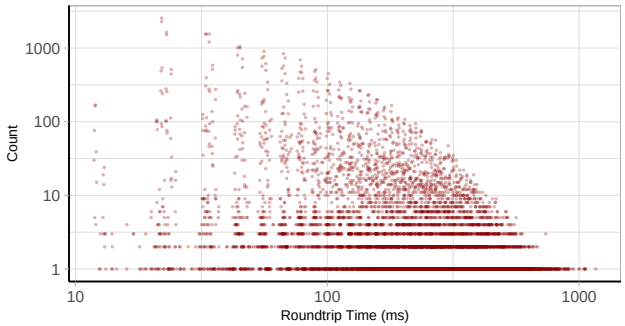
Visualize the distribution

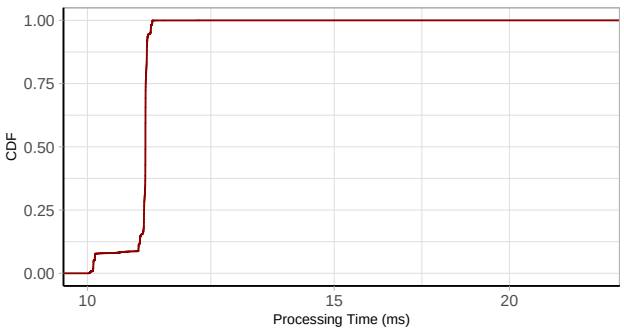


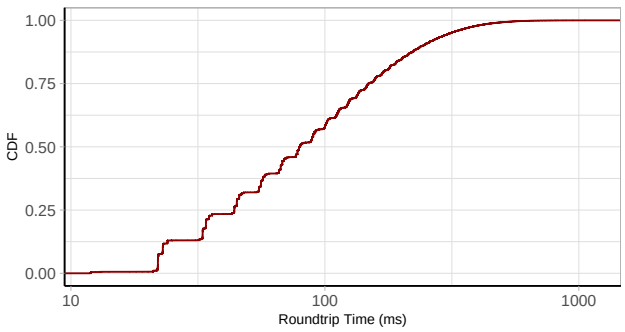


Visualize the distribution



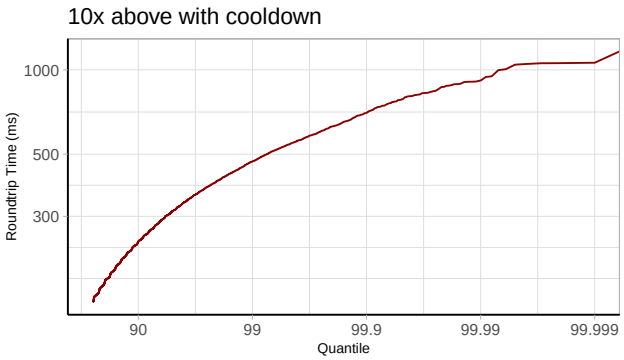


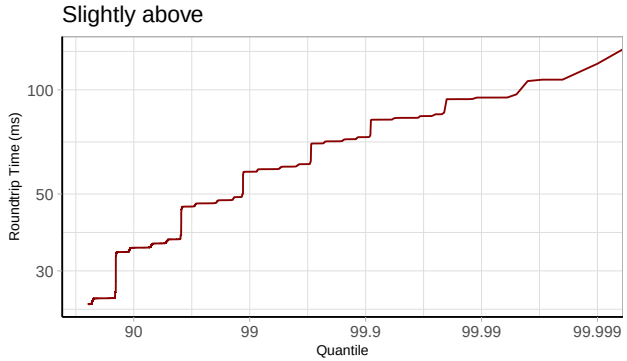


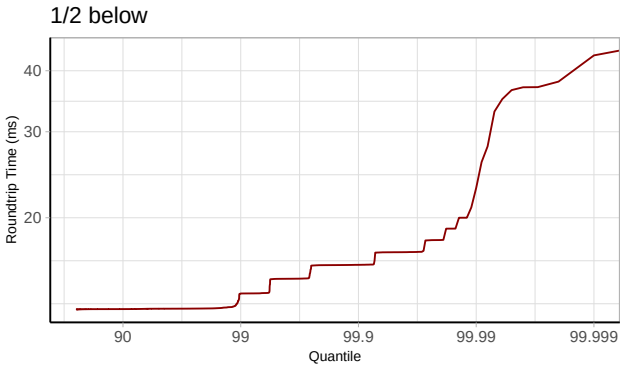


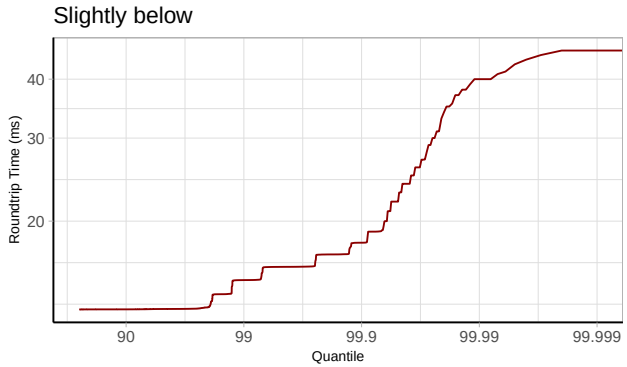
Distribution tails

Throughput



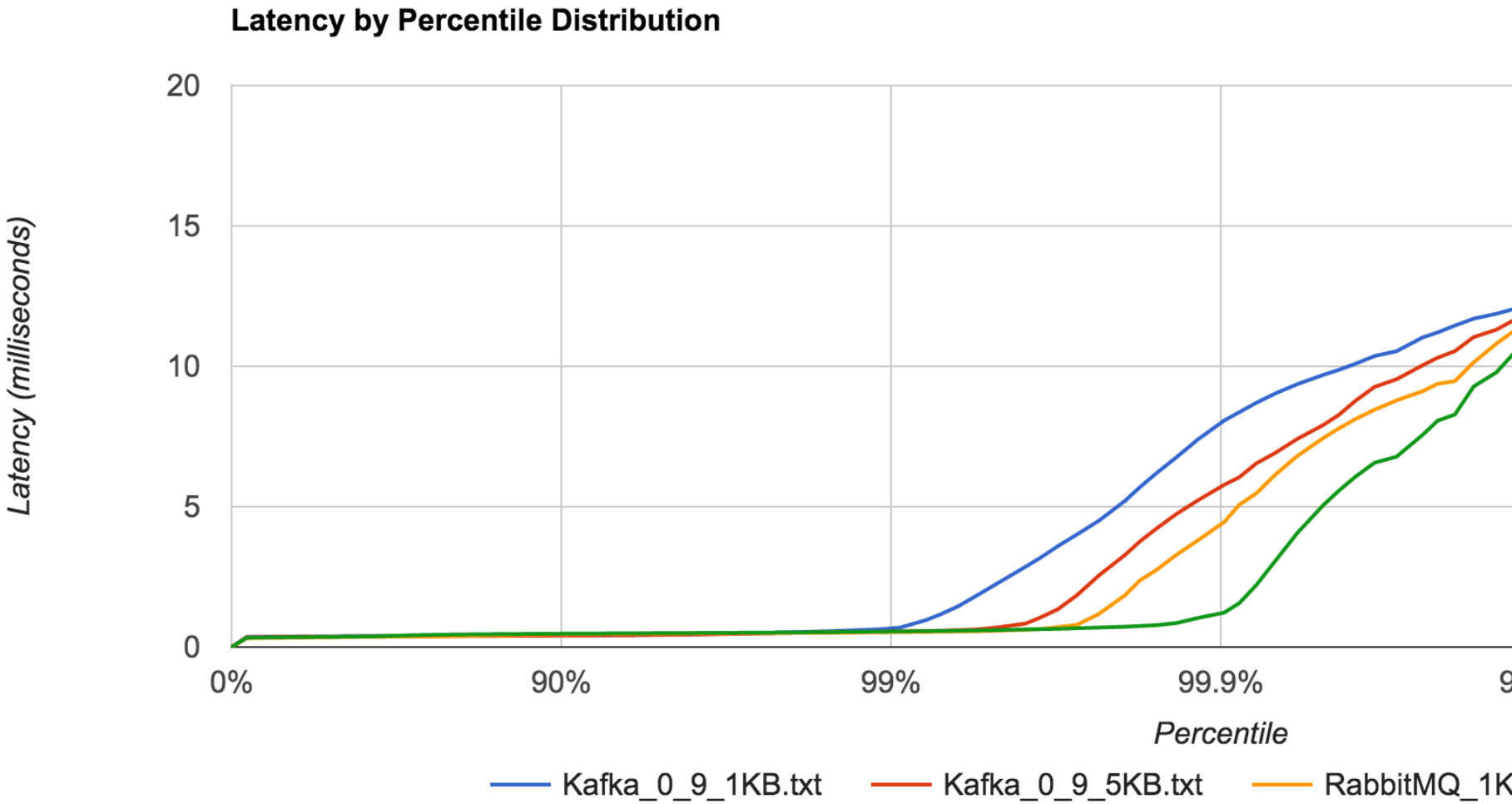






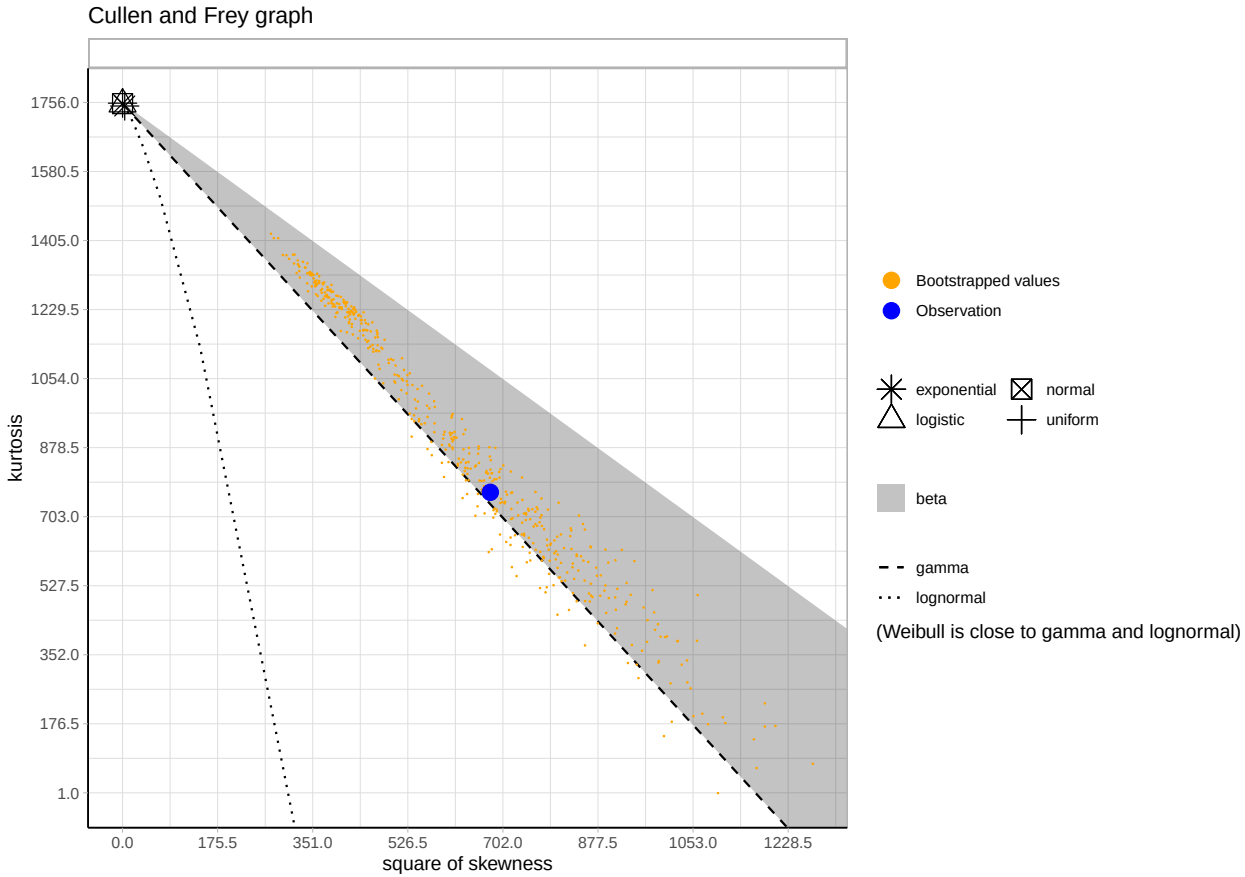
Comparing performance

Comparing messaging systems performance

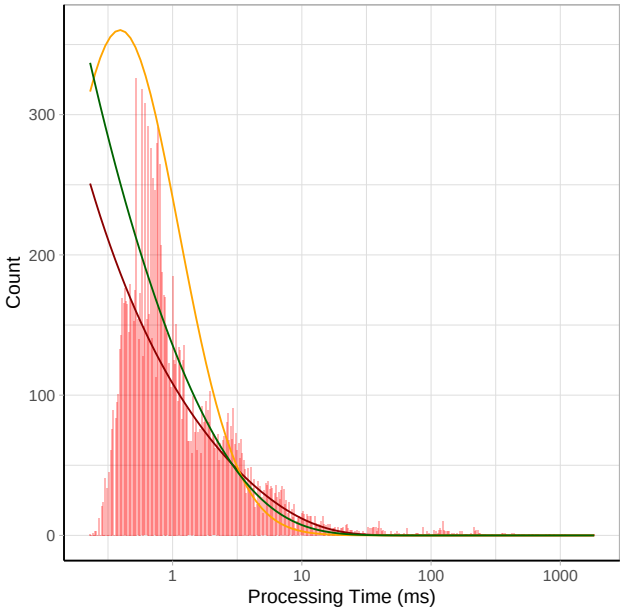


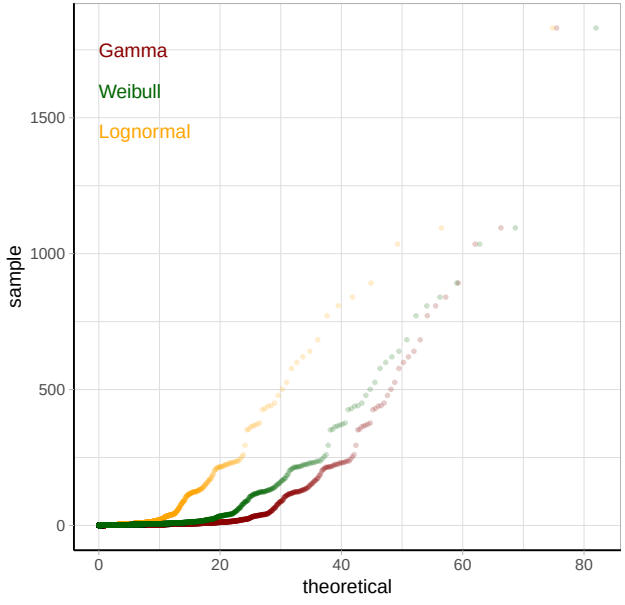
How distributions look like

Example 1: processing times

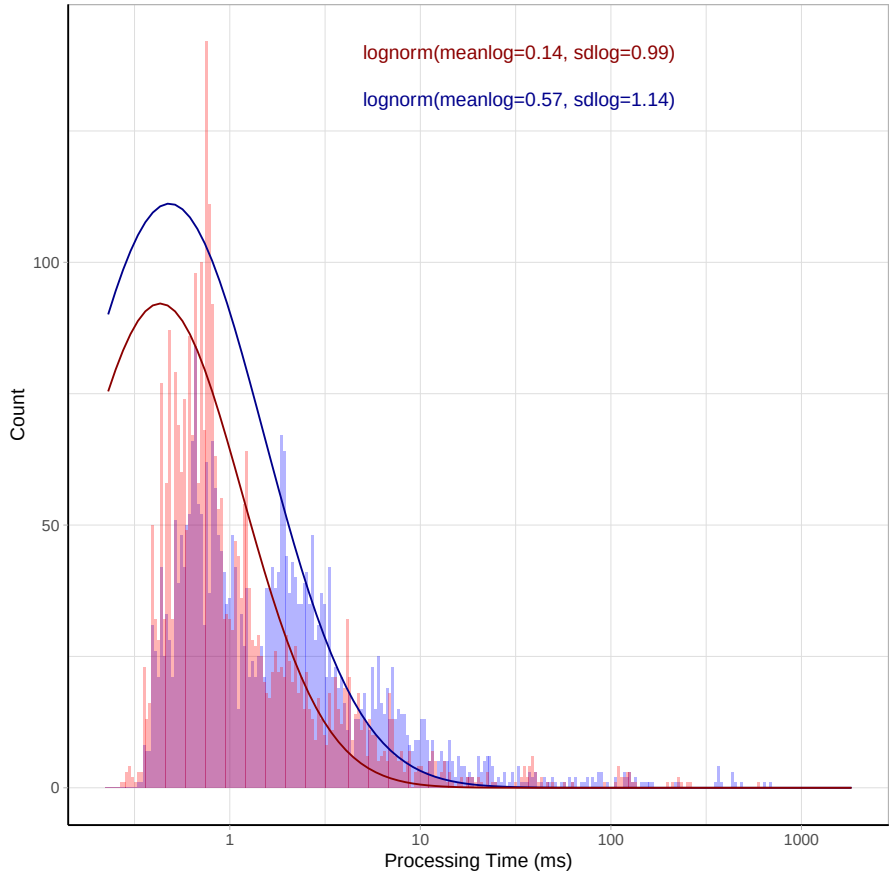


How distributions look like

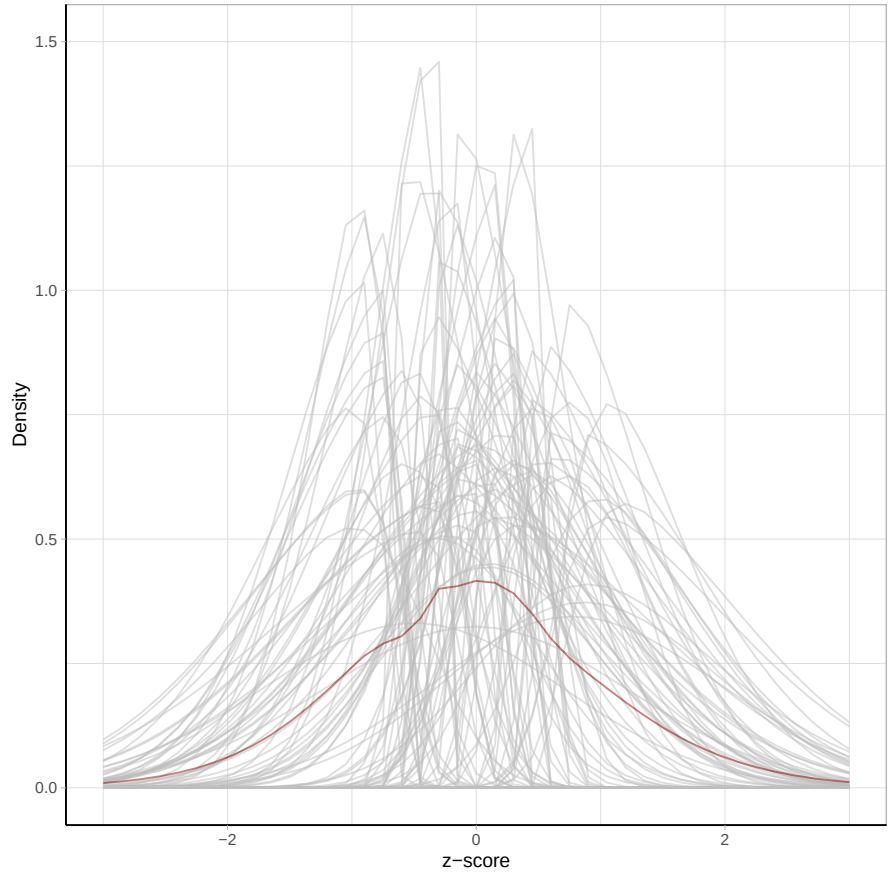




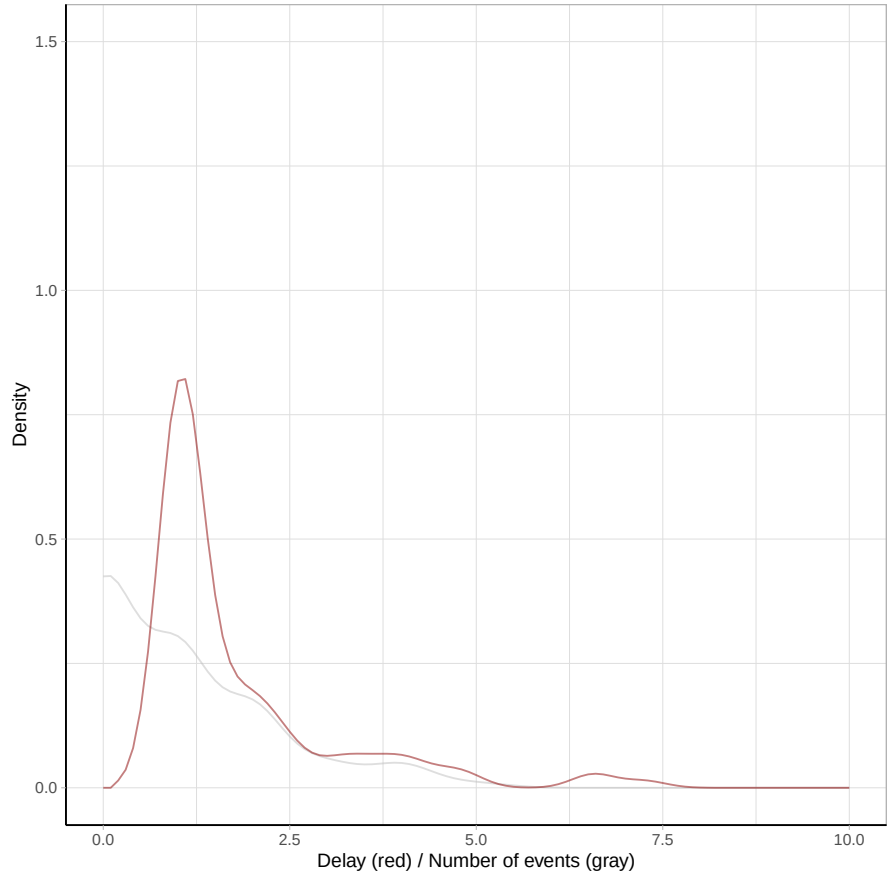
Compare distributions



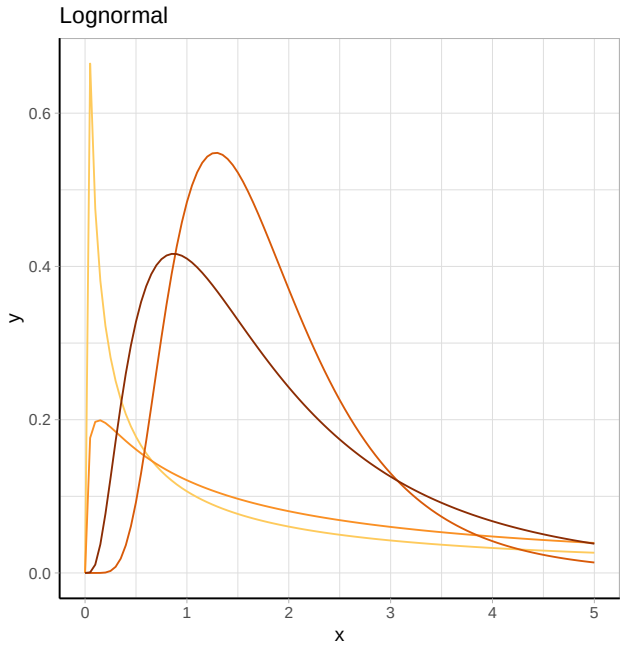
How distributions arise

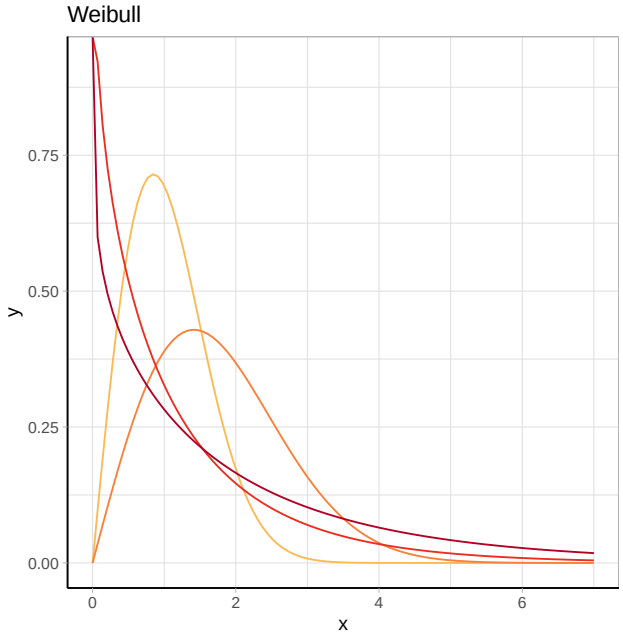


How distributions arise

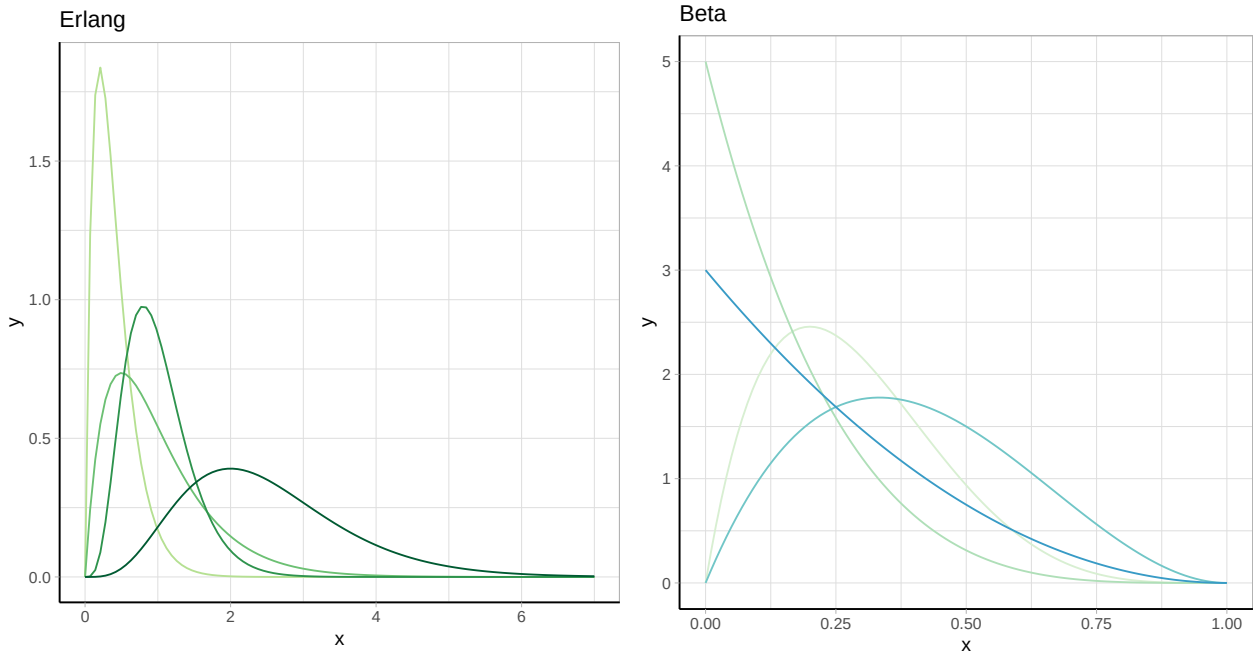


Examples of distributions



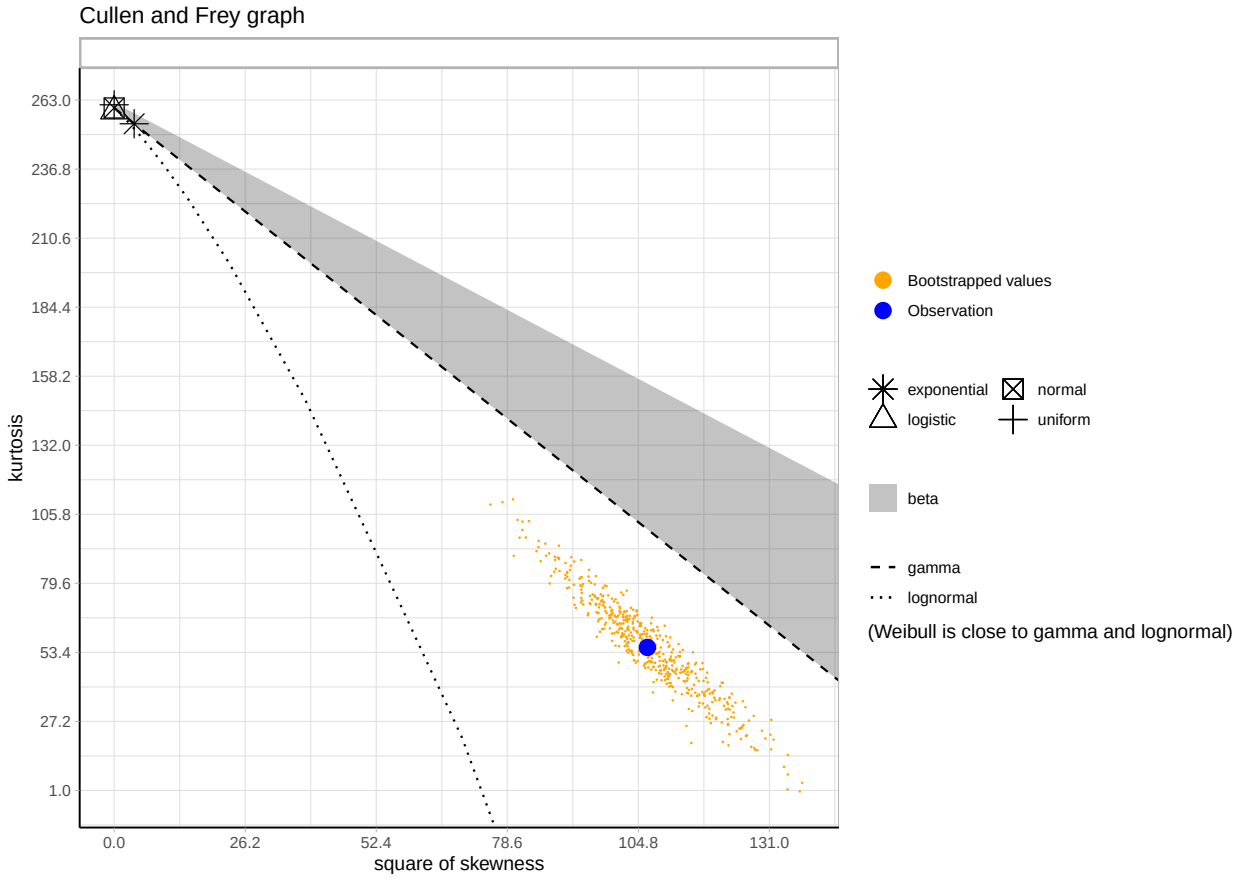


Examples of distributions



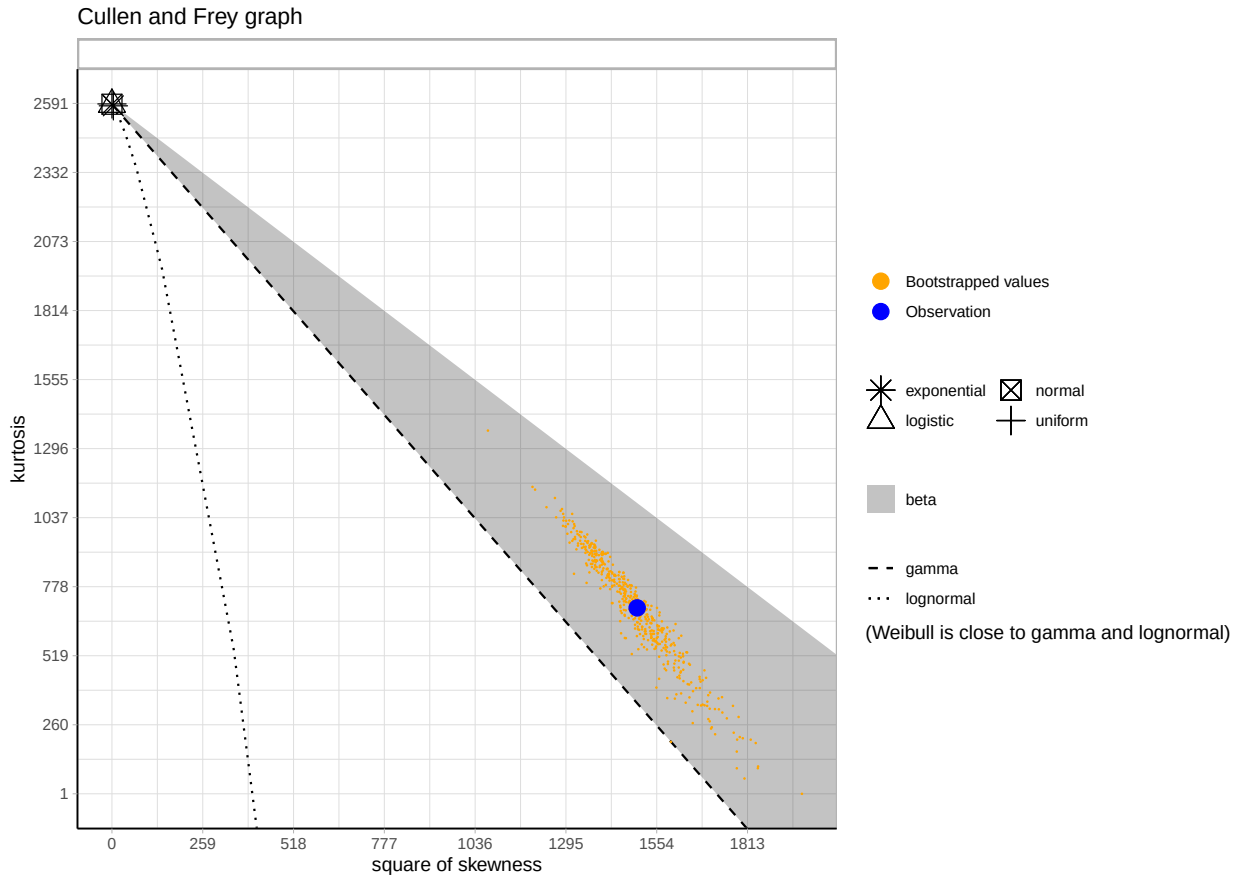
Types of distributions

Example 2: roundtrip times, single consumer, localhost, constant small processing time



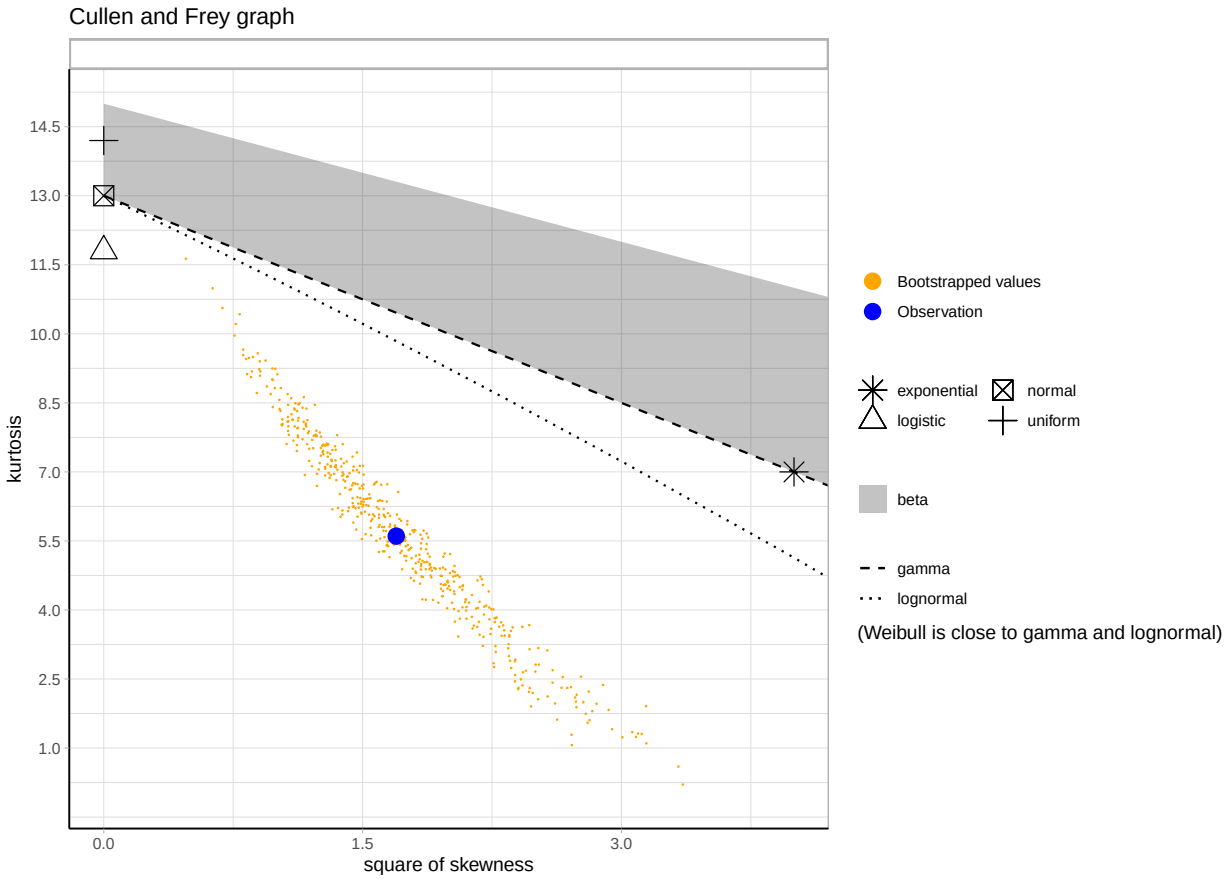
Types of distributions

Example 3: roundtrip times, single consumer, internet (short distance), constant small processing time



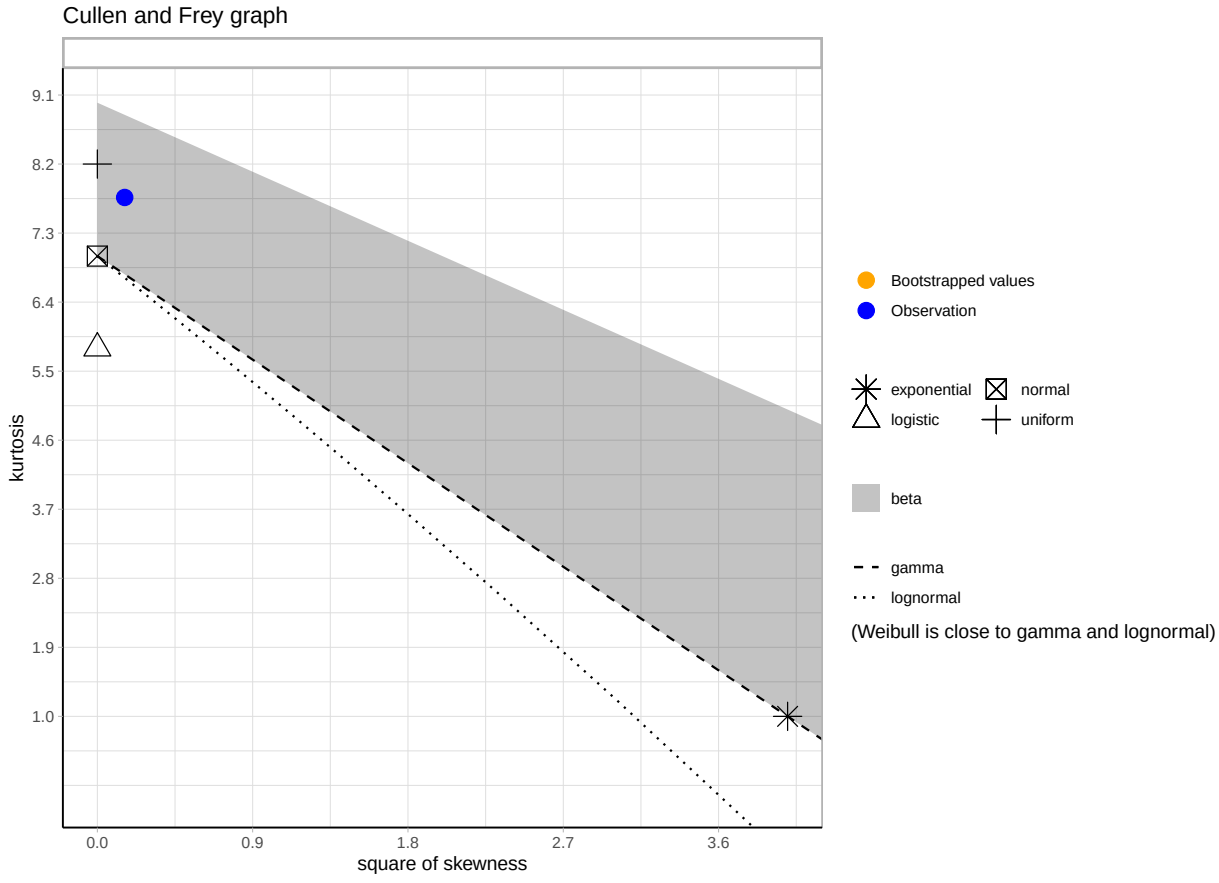
Types of distributions

Example 4: roundtrip times, single consumer, internet (long distance), no processing time (instant reply)



Types of distributions

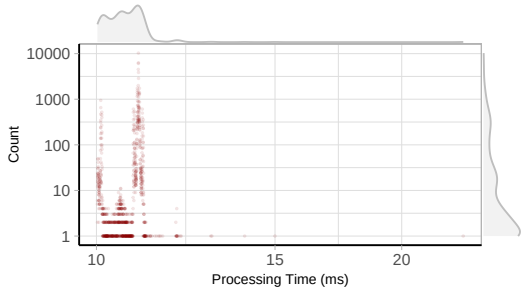
Example 5: roundtrip times, single consumer, localhost, heavy server-side calculations based on a random uniform input

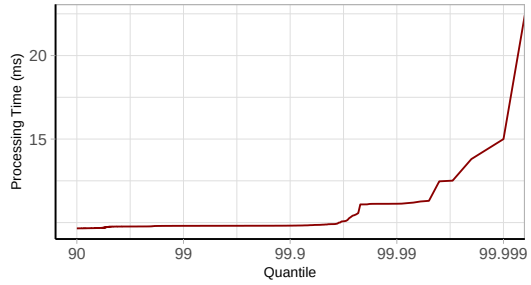


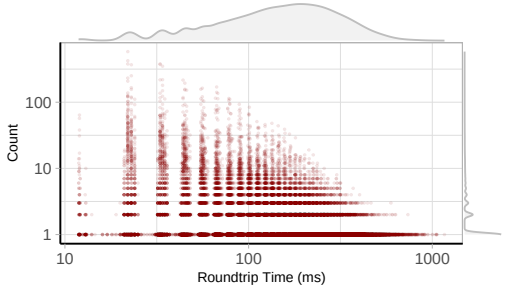
How to measure

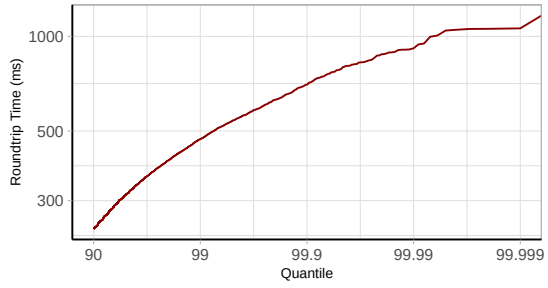
- log everything (if possible)
- HdrHistogram
- add timestamps to packets and chain them
- use high resolution timers (if suitable)
- build time series with timestamps, counts, events

How to report latencies





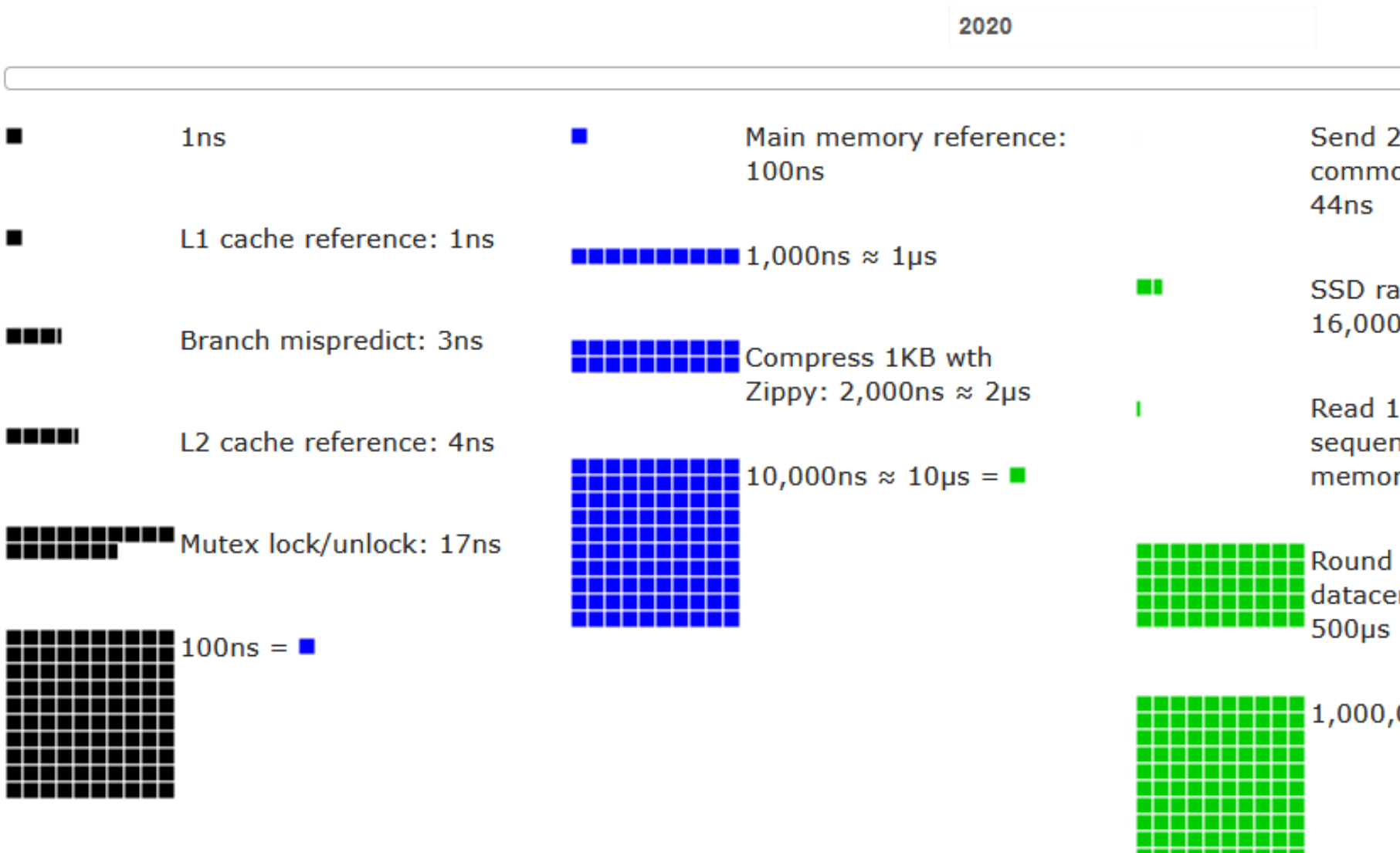




	Average	Stdev	MAD	50%	75%	90%	95%	99%	99.9%	99.99%	99.999%	99.9999%	Max
Processing Time	10.92	0.25	0.02	11	11.01	11.03	11.09	11.12	11.13	12	15	22.2	23
Roundtrip Time	112.62	99.48	67.05	79.13	148.12	243.98	313.89	471.01	702.01	916.3	1060.24	1154.45	1164.91

What numbers mean

https://colin-scott.github.io/personal_website/research/interactive_latency.html



What clocks to use

https://www.python.org/dev/peps/pep-0418/

Examples of clock resolution on x86_64:

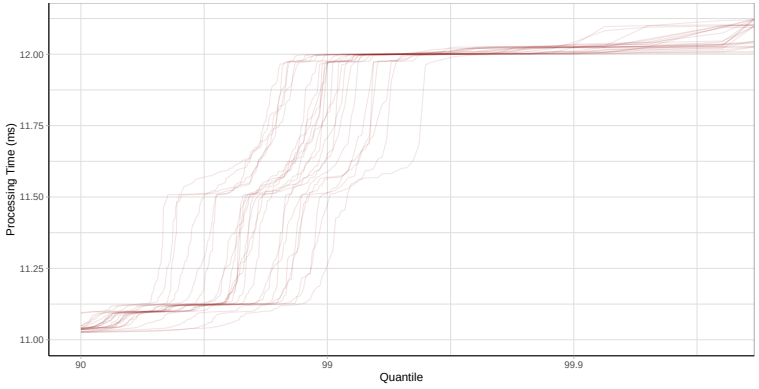
Name	Operating system	OS Resolution	Python Resolution
QueryPerformanceCounter	Windows Seven	10 ns	10 ns
CLOCK_HIGHRES	SunOS 5.11	2 ns	265 ns
CLOCK_MONOTONIC	Linux 3.0	1 ns	322 ns
CLOCK_MONOTONIC_RAW	Linux 3.3	1 ns	628 ns
CLOCK_BOOTTIME	Linux 3.3	1 ns	628 ns
mach_absolute_time()	Mac OS 10.6	1 ns	3 µs
CLOCK_MONOTONIC	FreeBSD 8.2	11 ns	5 µs
CLOCK_MONOTONIC	OpenBSD 5.0	10 ms	5 µs
CLOCK_UPTIME	FreeBSD 8.2	11 ns	6 µs
CLOCK_MONOTONIC_COARSE	Linux 3.3	1 ms	1 ms
CLOCK_MONOTONIC_COARSE	Linux 3.0	4 ms	4 ms
GetTickCount64()	Windows Seven	16 ms	15 ms

Clock synchronization

- measure on the same machine (when possible)
- use time synchronization services (NTP, Chrony, etc.)
- triangulate different sources
- sample times with a ping-like service
- account for clock desynchronization

Investigate and improve

- "how?": measure
- compare distribution tail over time
- know your throughput. improve. scale horizontally or collapse when not possible
- offline profiling reveals some of the problems, make sure you have enough info also from prod logs
- extract critical scenarios from prod logs. stress tests vs historical replays

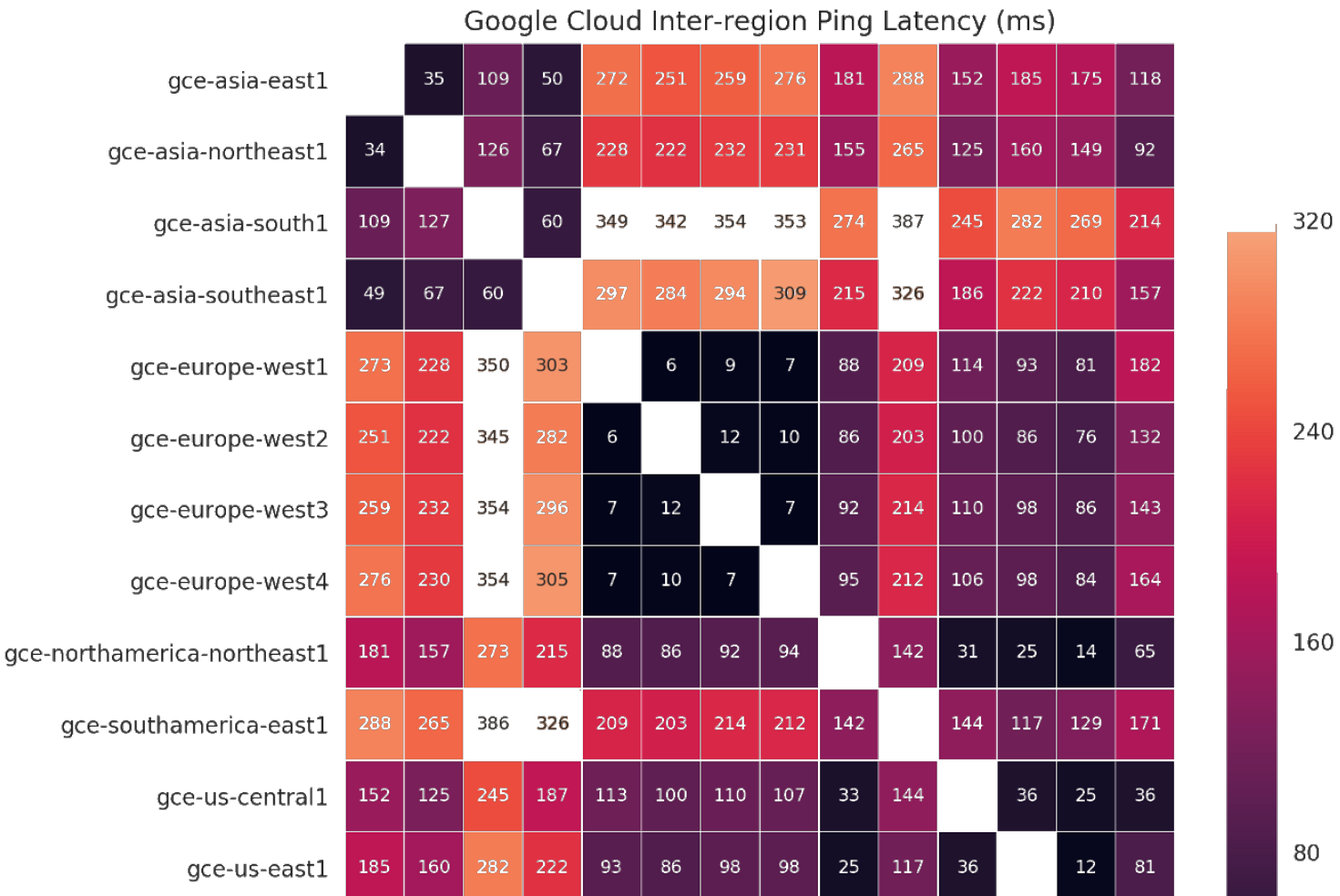


Investigate and improve

- your system is not a black box, you can debug it: find the factors that change your latencies
- build time series with factors throughout the day
- assess their importance
- find the dimensions on which delays cluster
- prioritize consumer tasks

Investigate and improve

- source of delays are stochastic processes, try to identify and understand them
- split delays into multiple transport and processing stages
- reduce the number of rountrips needed to build a result



Investigate and improve

- target the critical path and historical or theoretical cases of failure
- check outliers, major failures may have a chain of things that went wrong
- multimodal distributions are a sign of multiple paths, try to identify the split factor
- drifts from parametric distributions

Thank you