P99 CONF

How to Measure Latency?

Heinrich Hartmann
Principal Engineer @ zalando

Brought to you by SCYLLA

Motivation and Background

- I have been talking about Statistics and Latency for the last years
  State of the Histogram (SLOConf 2021) / Circllhist (paper) / Latency SLOs Done Right (FOSDEM 2019) / Statistics for Engineers (2014..2019)

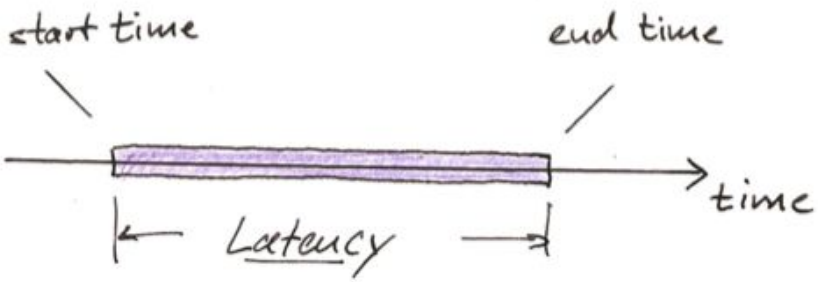
- Inspiration comes from series of talks from ~2013-15
  Gil Tene - How (not) to measure Latency
  Slides (London 2013) / Video (StrangeLoop 2015) / Blog - HighScalability 2015

- On Coordinated Omission - Ivan Prisyazhynyy
  Published two days ago on P99CONF.io

"It's slow" is the hardest problem you will ever debug.

Theo Schlossnagle @postwait

What is Latency?



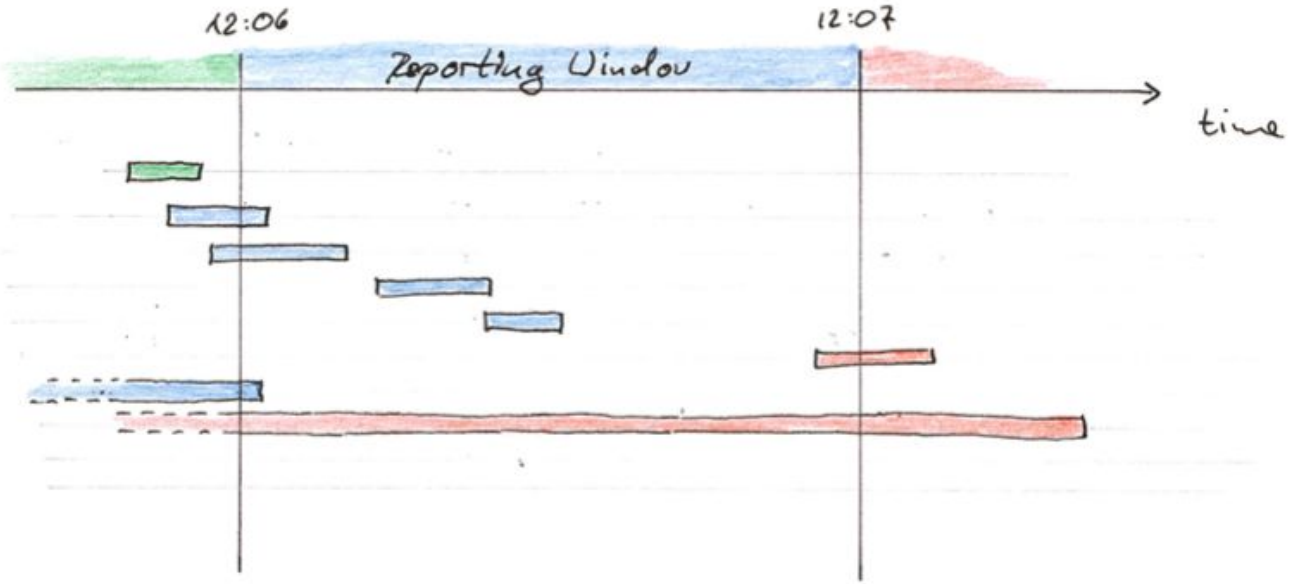
How to Measure Latency?

```
t_start = time.now()

#
# operation you want to measure
#

latency = time.now() - t_start
```

Things to watch out for

- Capture early returns / exceptions
  - Use: try/catch/finally or defer
- Which clock is used?
  - Want: high-resolution, monotonic, system time (e.g. time.monotonic() in Python)
- Measurement Overhead
  - Measuring time takes time (at least 30ns, often >300ns)
  - OK for 0.1ms or more (I/O Latency)
  - Careful for 10us or lower (micro benchmarking)
- Abstracting time measurements in code
  - Write a @timed decorator. Use tracing libraries (@trace)

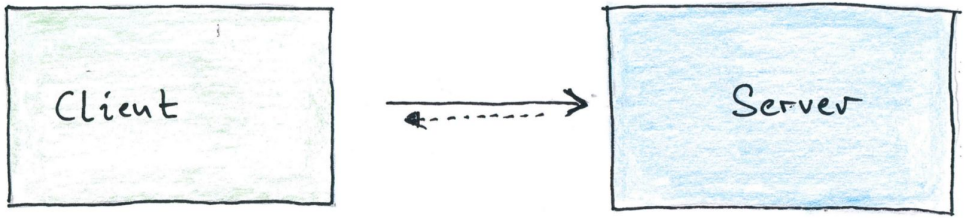
Measuring Latency over Time



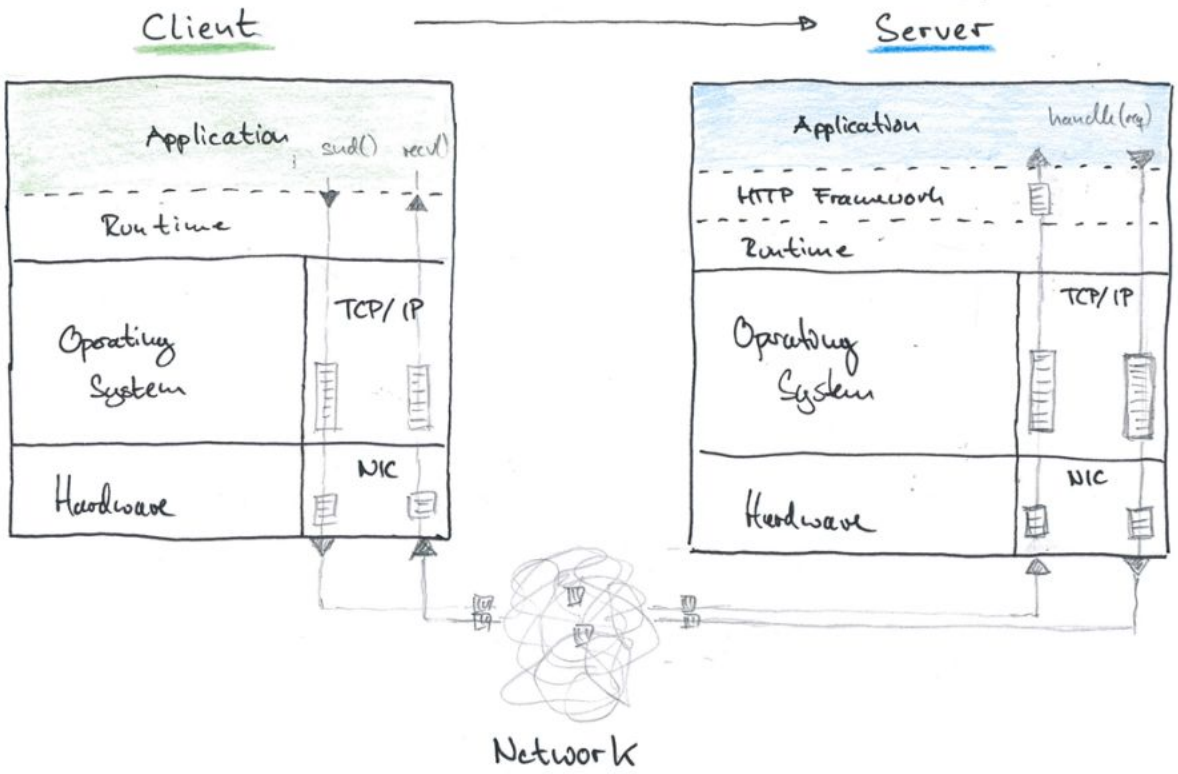
The End

Where to Measure Latency?

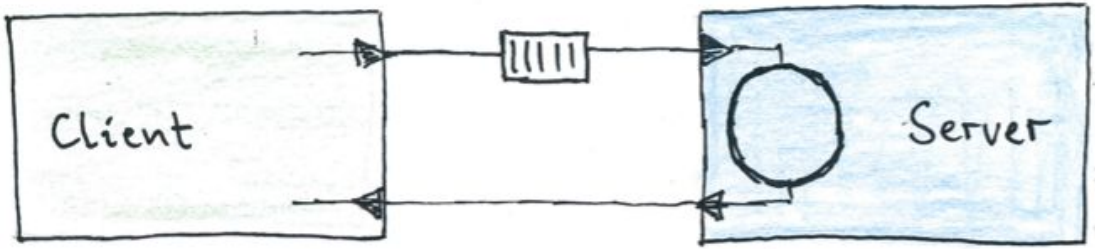
Hidden Queues



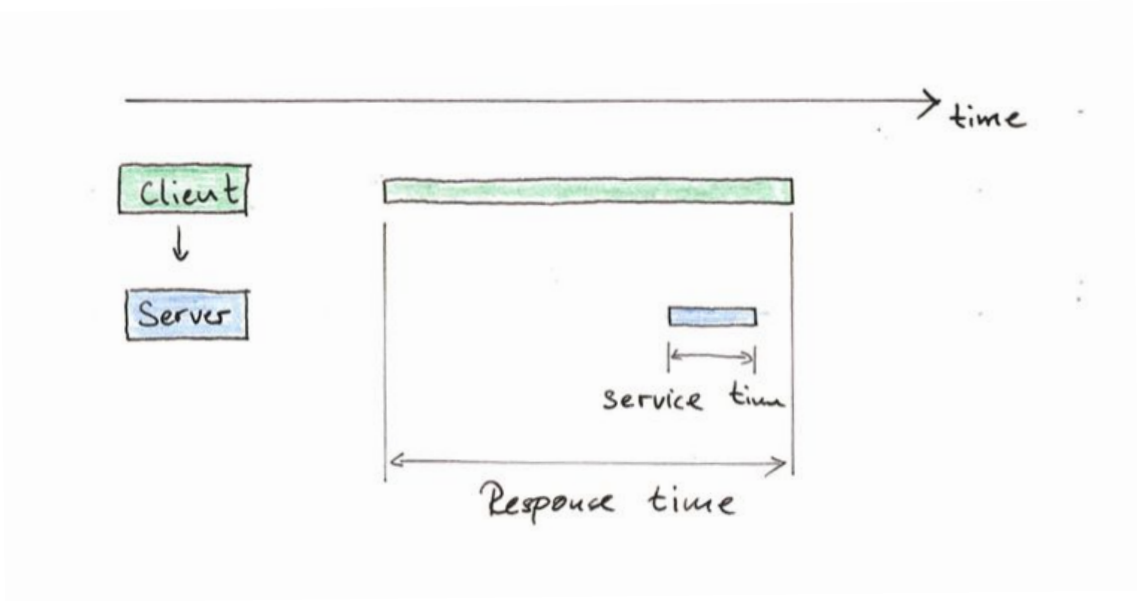
Hidden Queues



A practical Queuing Model



Response Time vs. Service Time

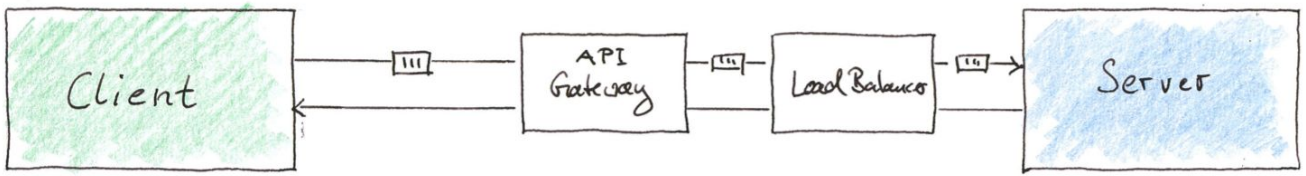


Response Time vs. Service Time

Response Time

Service Time

Where to Measure Latency?



+ Most meaningful

- Hard to implement

- Only get Service Time

+ Easy to implement

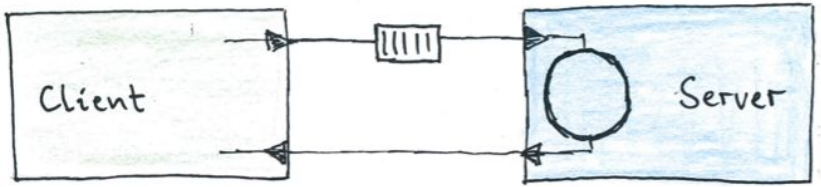
You can't measure Response Time on the Server.

SAD BUT TRUE

Request Time vs. Service Time

An Experiment

Simulation Setup



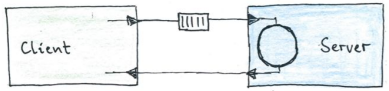
- 10 workers
- 10ms service time
- 1K rps capacity

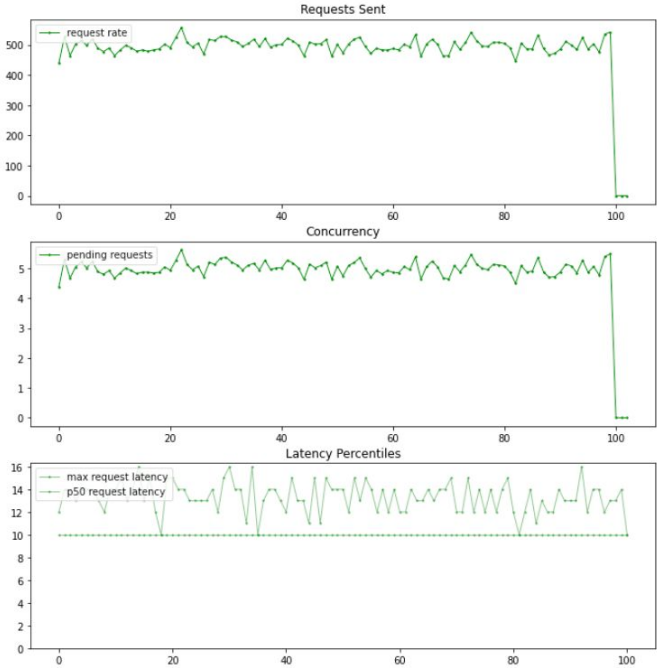
Metrics

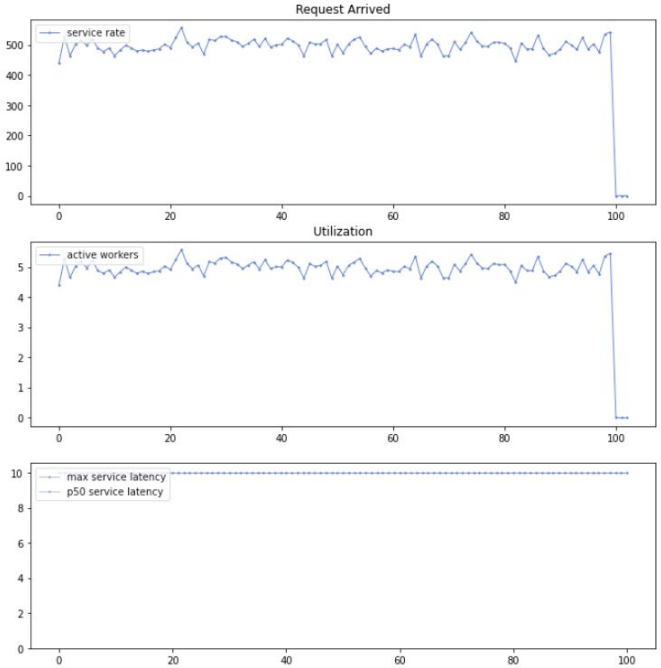
- Request Rate (~ constant)
- Concurrency (Active Requests)
- Response Time
- Arrival Rate
- Concurrency (Active Workers)
- Service Time (constant)

code @ https://github.com/HeinrichHartmann/libq

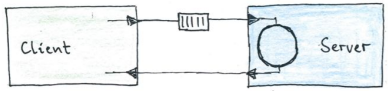
Queuing System at 50% Capacity

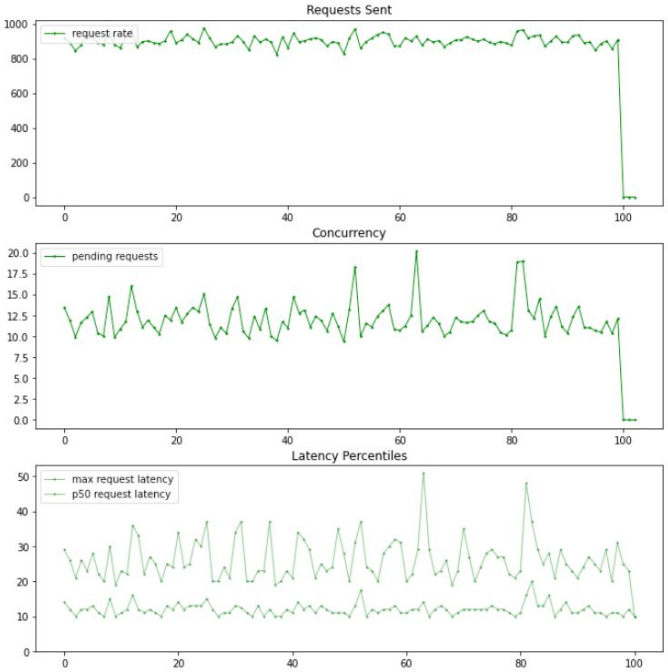


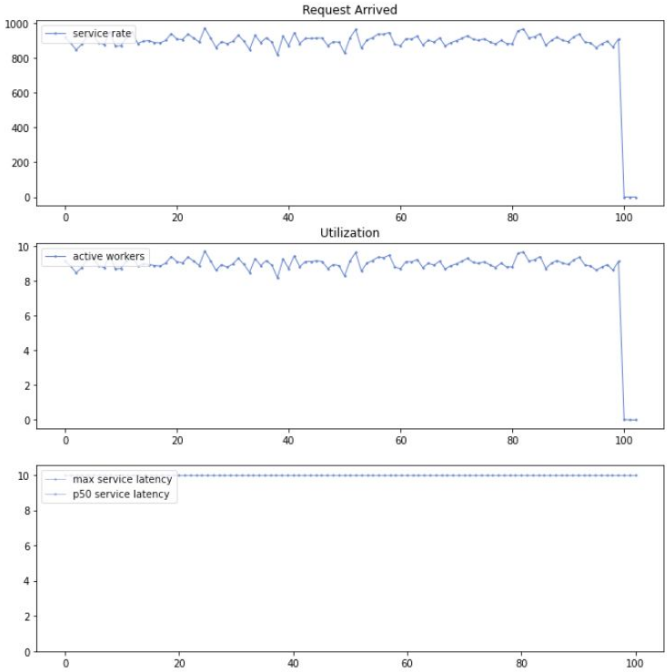




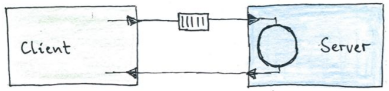
Queuing System at 90% Capacity

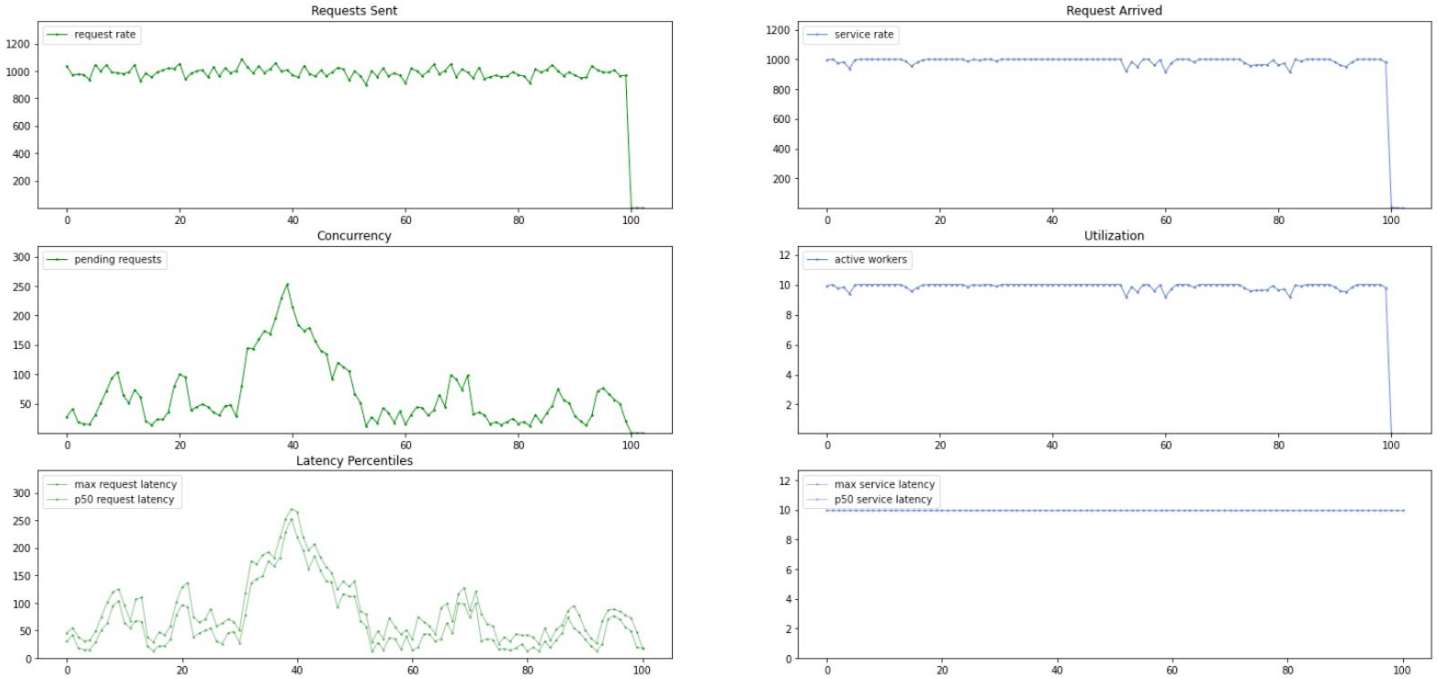




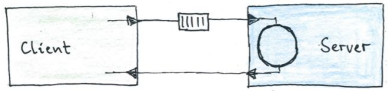


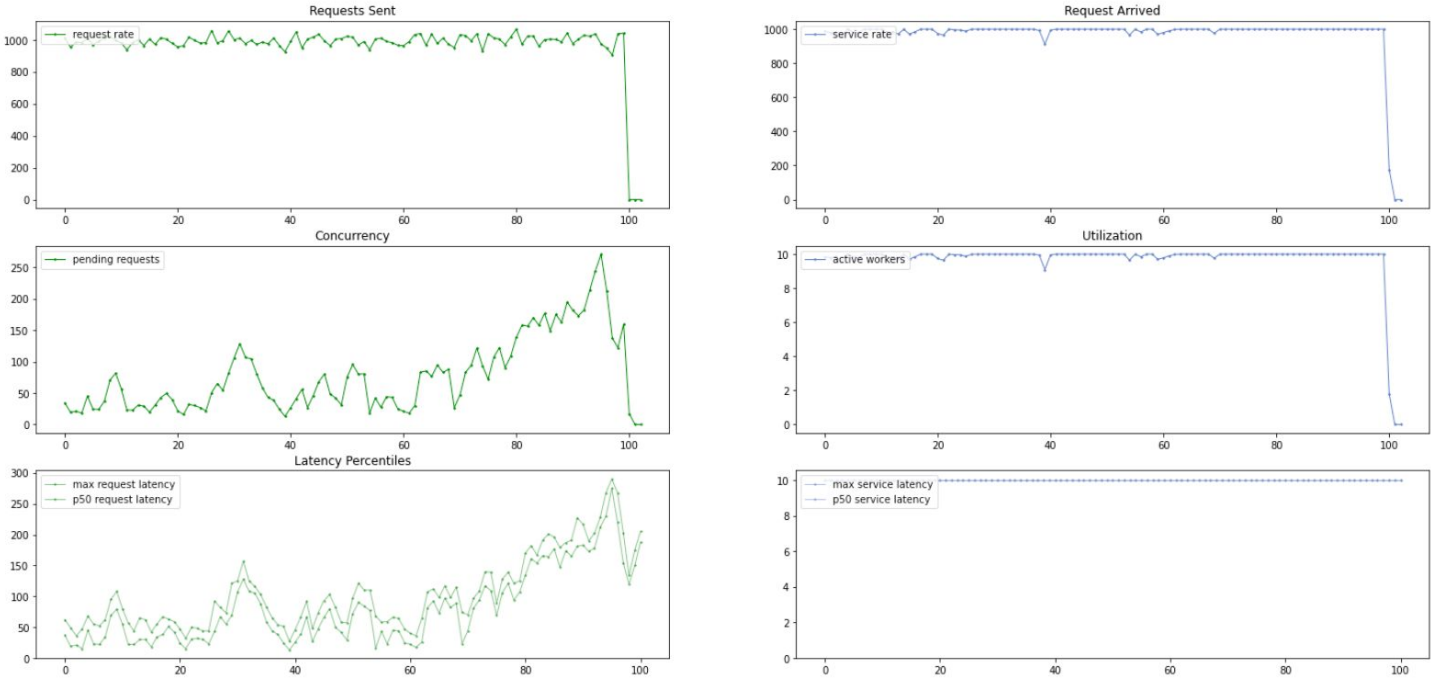
Queuing System at 99% Capacity



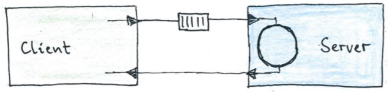


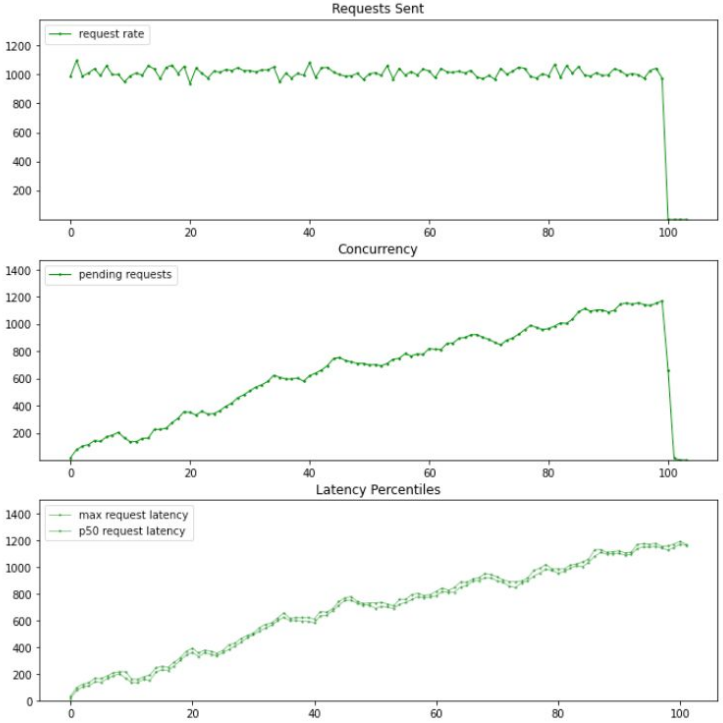
Queuing System at 100% Capacity

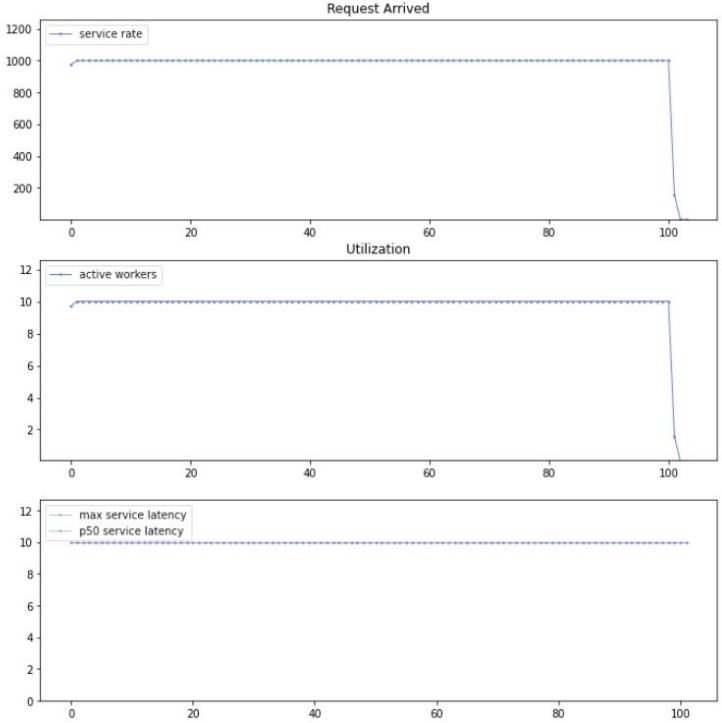




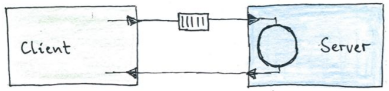
Queuing System at 101% Capacity

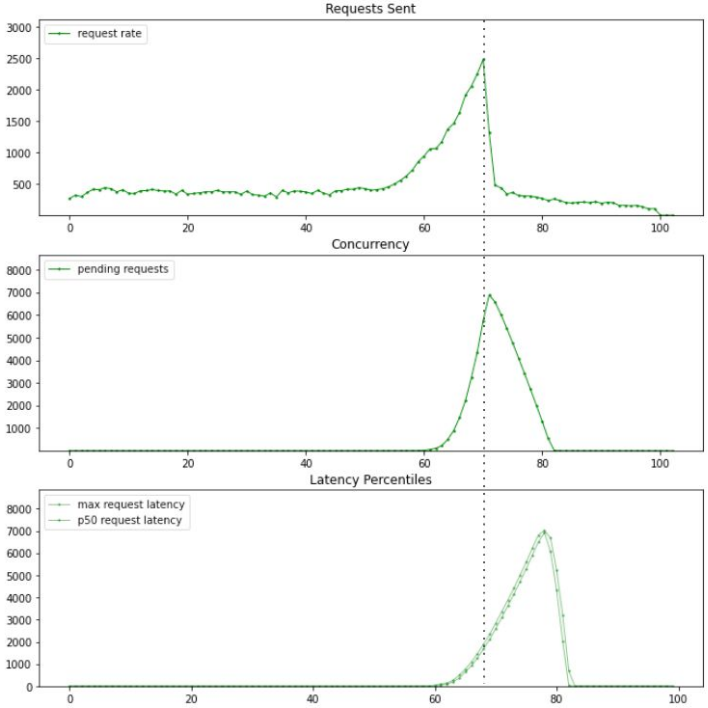


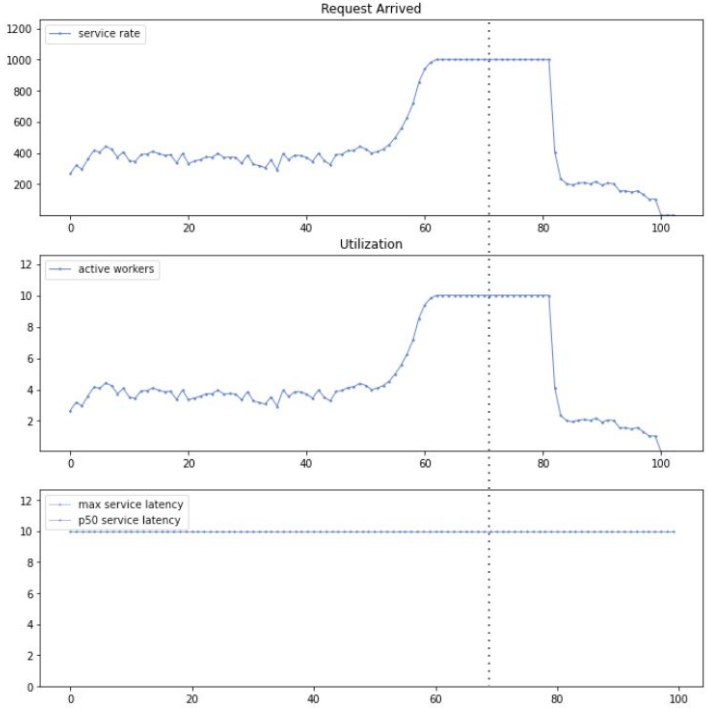




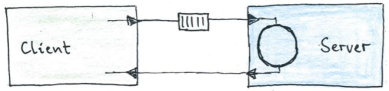
A Hockey Stick

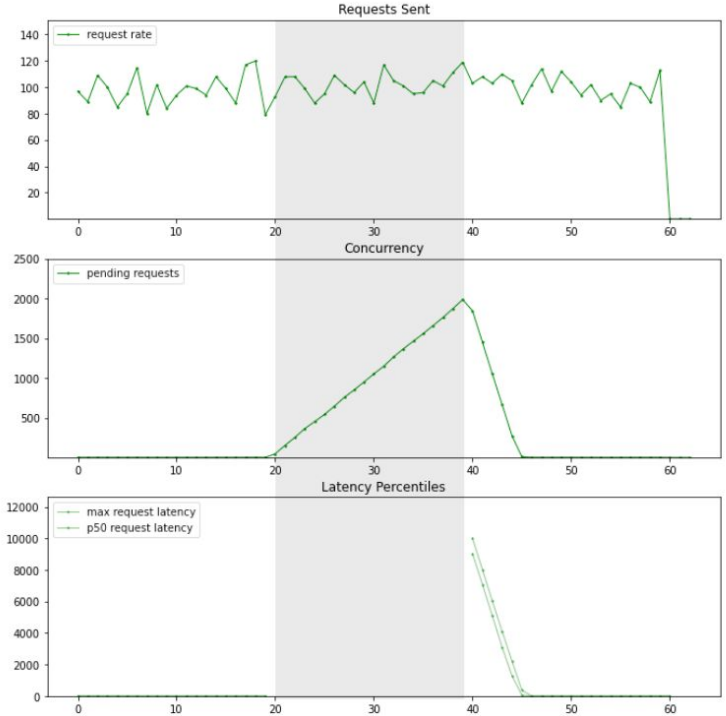


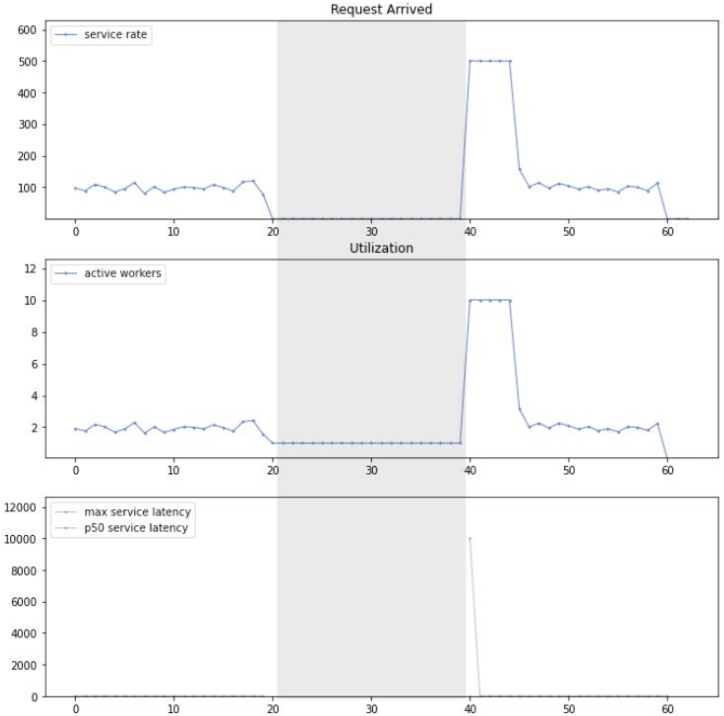




A Stalled System







Coordinated Omission in Load Testing

Def. Coordination between Load Generator (Client) and Server that leads to confusing Service Time with Response times.

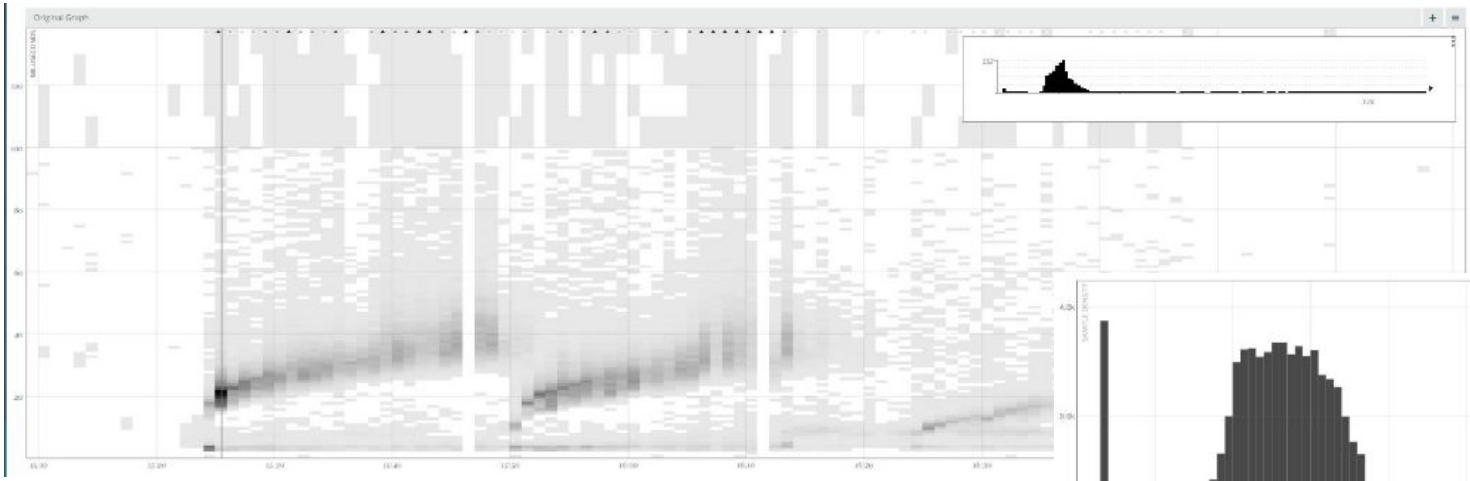
Examples

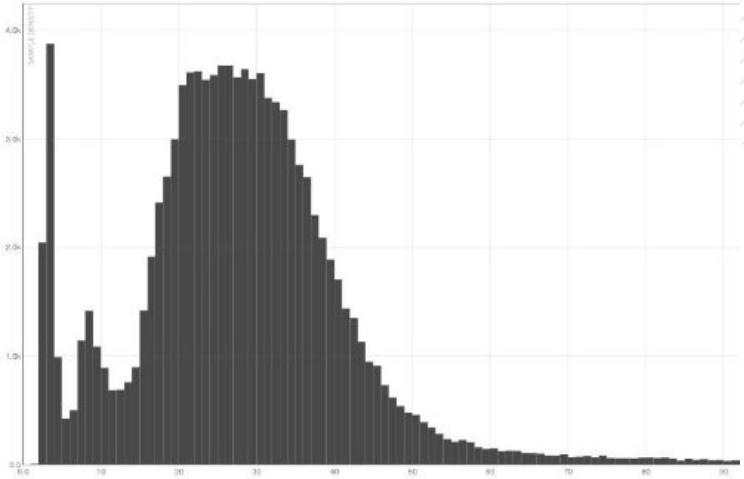
- Client backs off when server is falling behind
- Client is stalled when Server is stalled

This is surprisingly common (cf. *Gil's talk*, Ivan's blog)

How to Analyze Latency Data?

Best Practice: Histogram (Metrics)





More Details:
- Latency SLOs done right @ FOSDEM 2019
- State of The Histogram @ SLOConf 2021 [slides] [video]

P99 CONF

Thank you!

Further Reading

- HeinrichHartmann.com/latency
- @HeinrichHartmann