

How I stopped Worrying and Learned to Love Misery

Gil Tene, CTO & co-Founder, Azul Systems
@giltene





Gil Tene, CTO & co-Founder, Azul Systems
@giltene

About me: Gil Tene

- co-founder, CTO @Azul Systems
- Have been working on “think different” GC approaches since 2002
- A Long history building Virtual & Physical Machines, Operating Systems, Enterprise apps, etc...
- I also depress people by pulling the wool up from over their eyes...



* working on real-world trash compaction issues, circa 2004

Understanding Latency and Application Responsiveness

Gil Tene, CTO & co-Founder, Azul Systems
@giltene



(Micro?)services and Response Time Behavior

Gil Tene, CTO & co-Founder, Azul Systems
@giltene



How NOT to Measure Latency

Gil Tene, CTO & co-Founder, Azul Systems
@giltene



The “Oh S@%#!” talk

Gil Tene, CTO & co-Founder, Azul Systems
@giltene

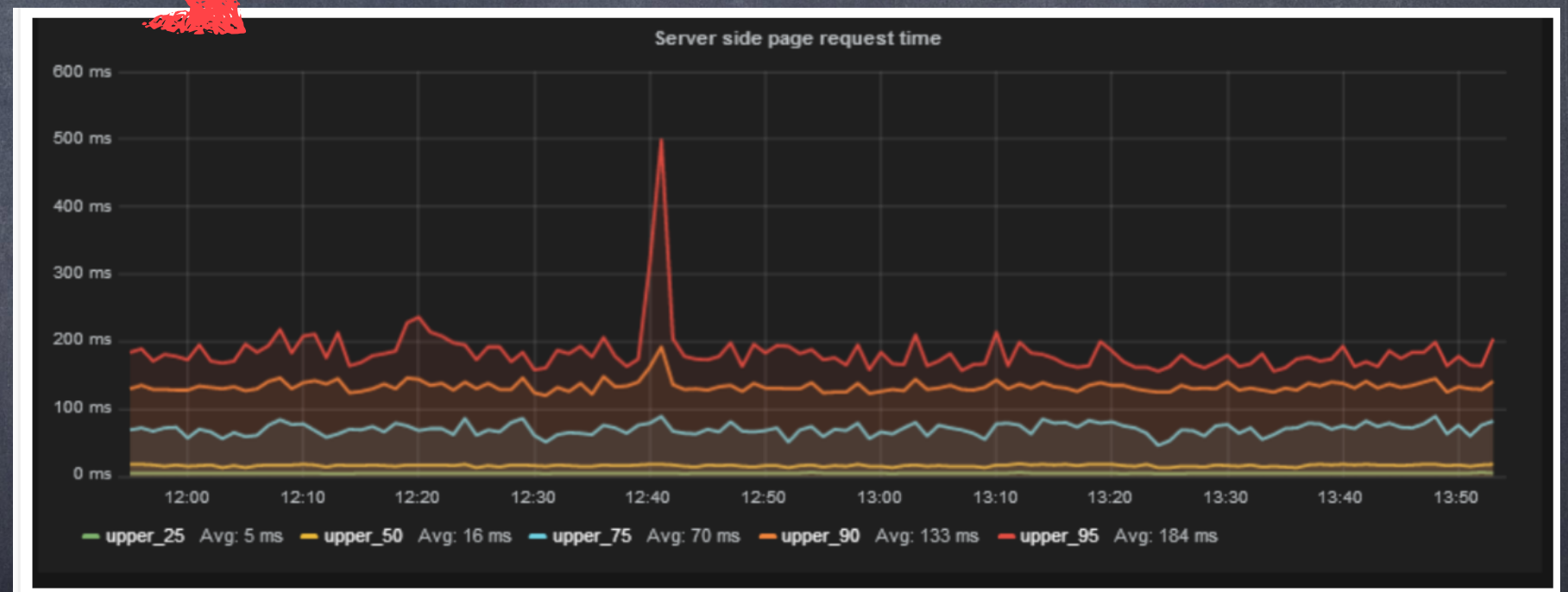




Remember: all I'm offering is the truth.

We like to look at pretty charts...

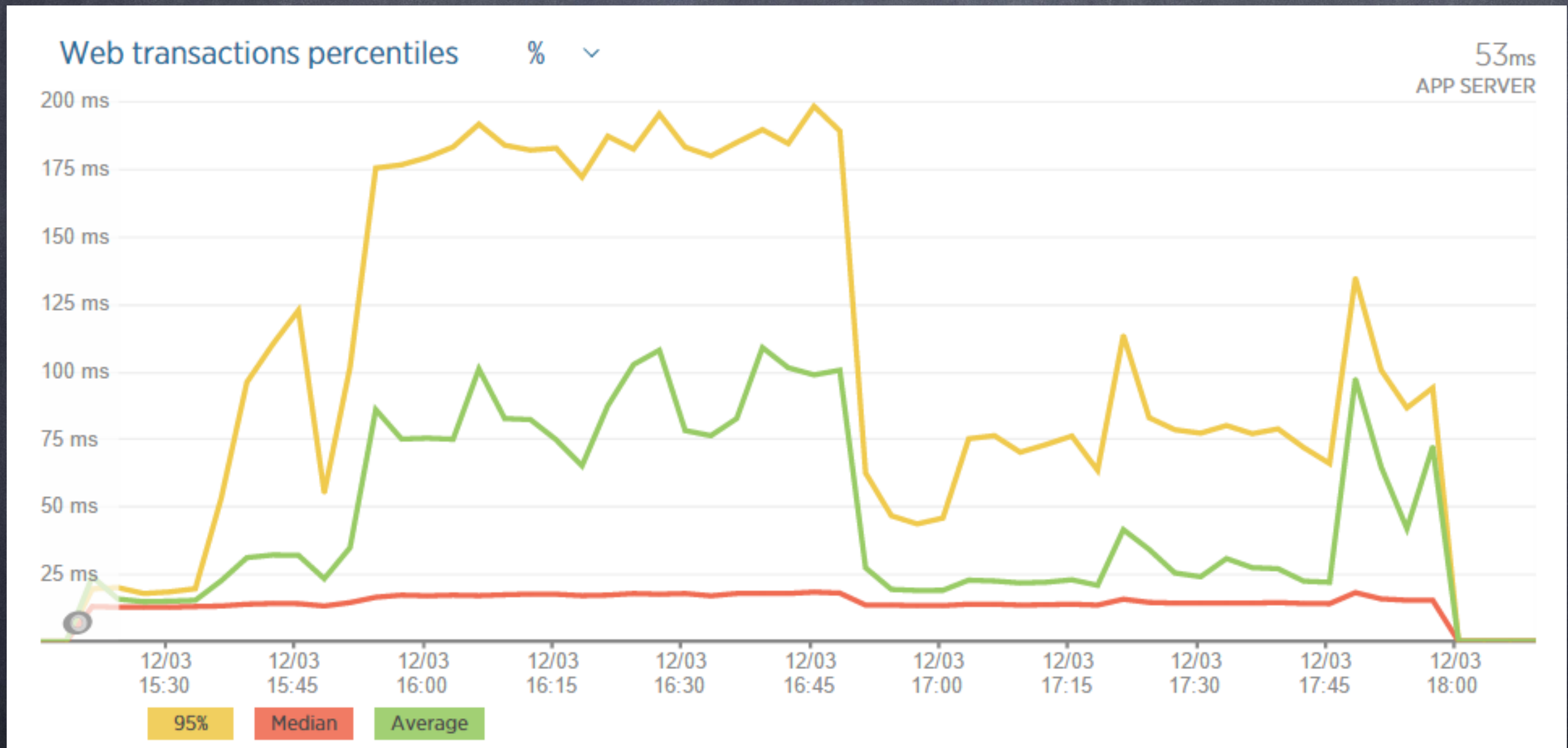
95%'lie



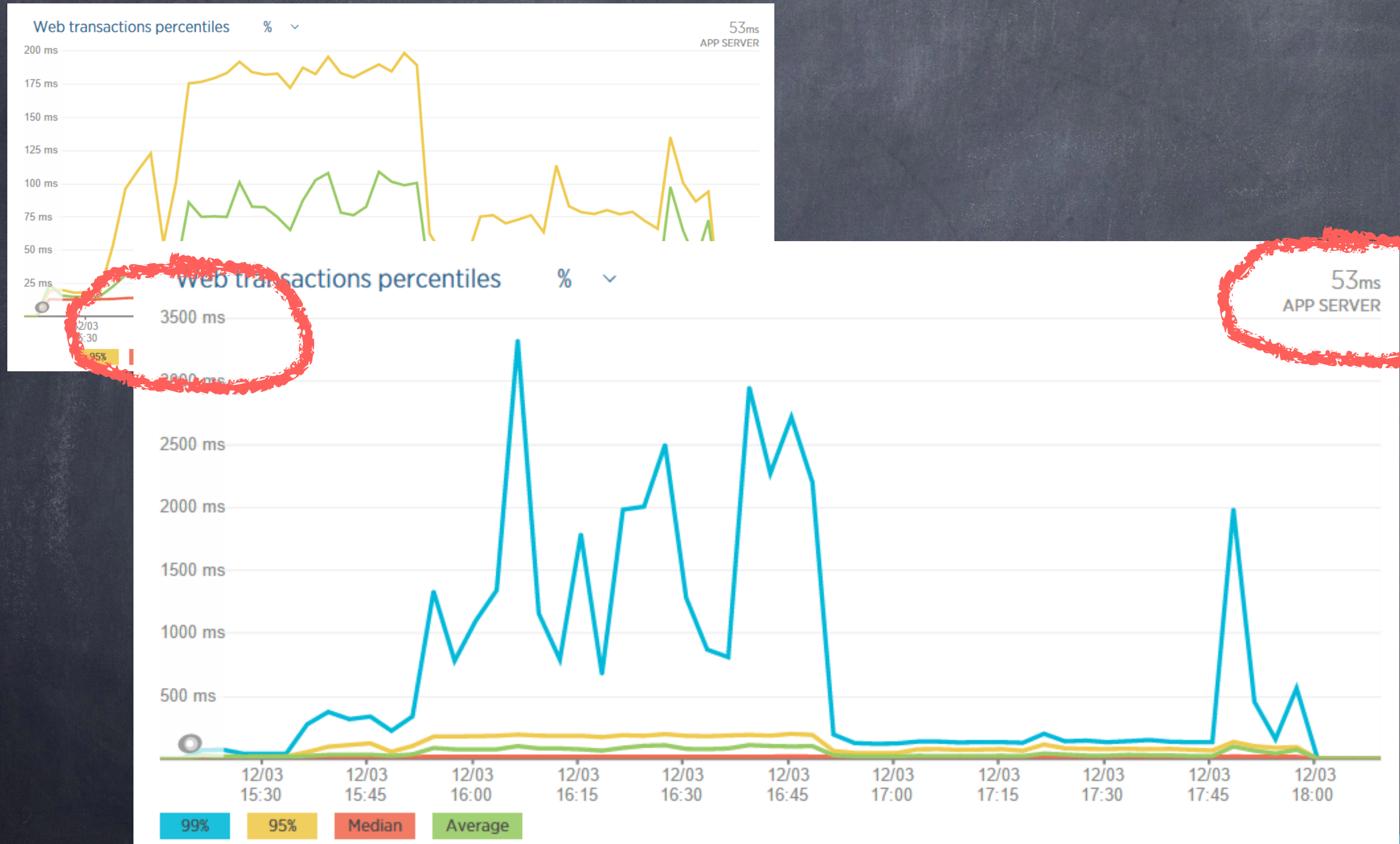
The “We only want to show good things” chart



A real world, real time example



A real world, real time example



Latency Tip Of The Day

This blog is meant to capture short (as in tweet-able) tips and observations about latency: Its understanding, its measurement, and maybe even its improvement. In the blog, I may expand on specific tip entries. Many times in rant form. Comments telling me how wrong I am are welcome. [See more here.](#)

About Me



 Gil Tene

CTO and co-founder
of Azul Systems.

[View my complete
profile](#)

Blog Archive

▼ 2014 (8)

▼ June (8)

[#LatencyTipOfTheDay: Median
Server Response Time: ...](#)

[#LatencyTipOfTheDay: MOST
page loads will experien...](#)

[#LatencyTipOfTheDay: Q:
What's wrong with this pic...](#)

Wednesday, June 18, 2014

#LatencyTipOfTheDay: Average (def): a random number that falls somewhere between the maximum and 1/2 the median. Most often used to ignore reality.

Averages are silly things when it comes to latency. I've yet to meet an application that actually has a use for, or a valid business or technical requirement for an average latency. Short of the ones that are contractually required to produce and report on this silly number, of course. Contracts are good reasons to measure things.

So I often wonder why people measure averages, or require them at all. Let alone use averages as primary indicators of behavior, and as a primary target for tuning, planning, and monitoring.

My opinion is that this fallacy comes from a natural tendency towards "Latency wishful thinking". We really wish that latency behavior exhibited one of those nice bell curve

About Me



Gil Tene

CTO and co-founder
of Azul Systems.

[View my complete
profile](#)

Blog Archive

▼ **2014** (8)

▼ **June** (8)

#LatencyTipOfTheDay: Median
Server Response Time: ...

#LatencyTipOfTheDay: MOST
page loads will experien...

#LatencyTipOfTheDay: Q:
What's wrong with this pic...

#LatencyTipOfTheDay: If you
are not measuring and/...

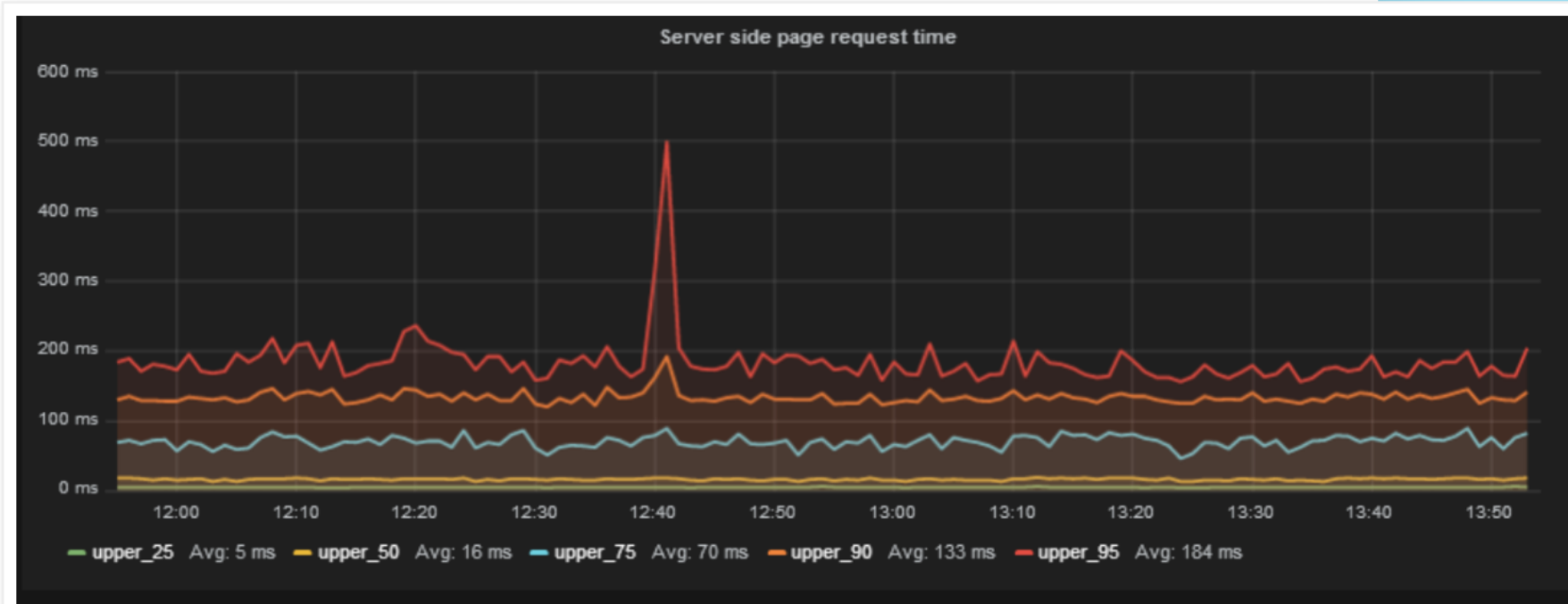
#LatencyTipOfTheDay :
Measure what you need to
mon...

#LatencyTipOfTheDay: Average
(def): a random numbe...

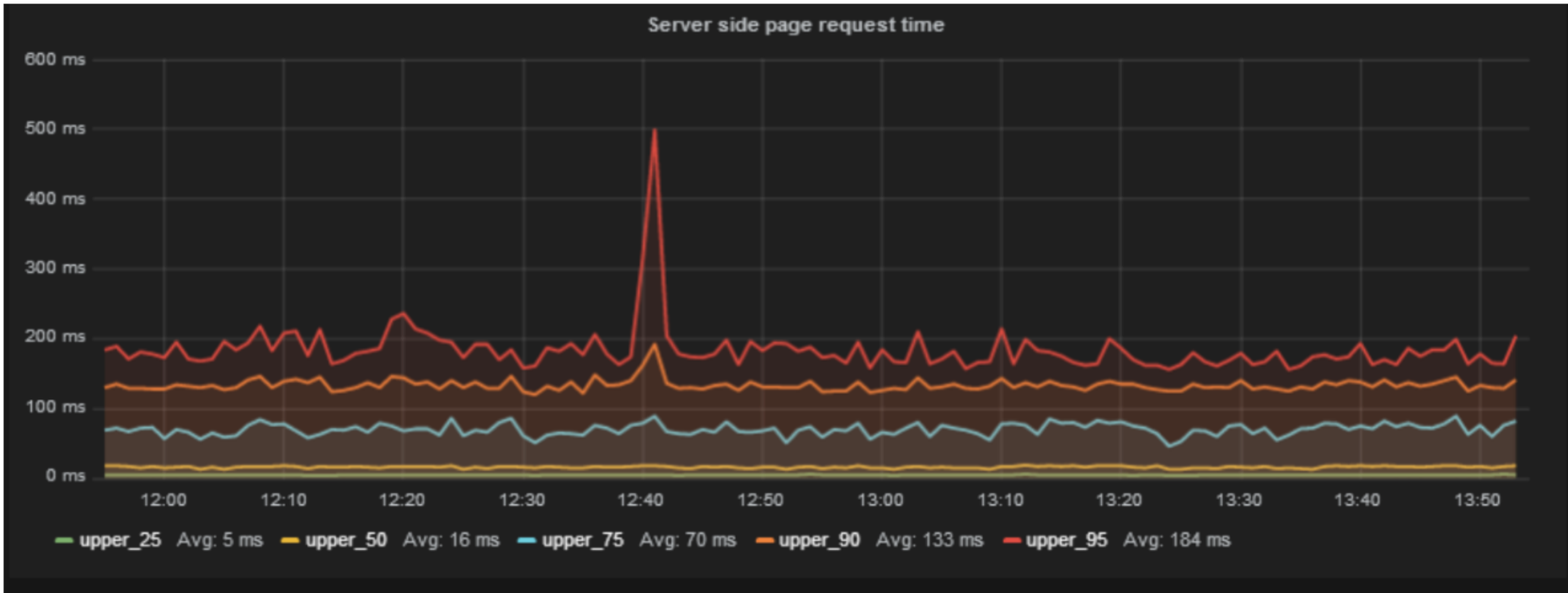
Saturday, June 21, 2014

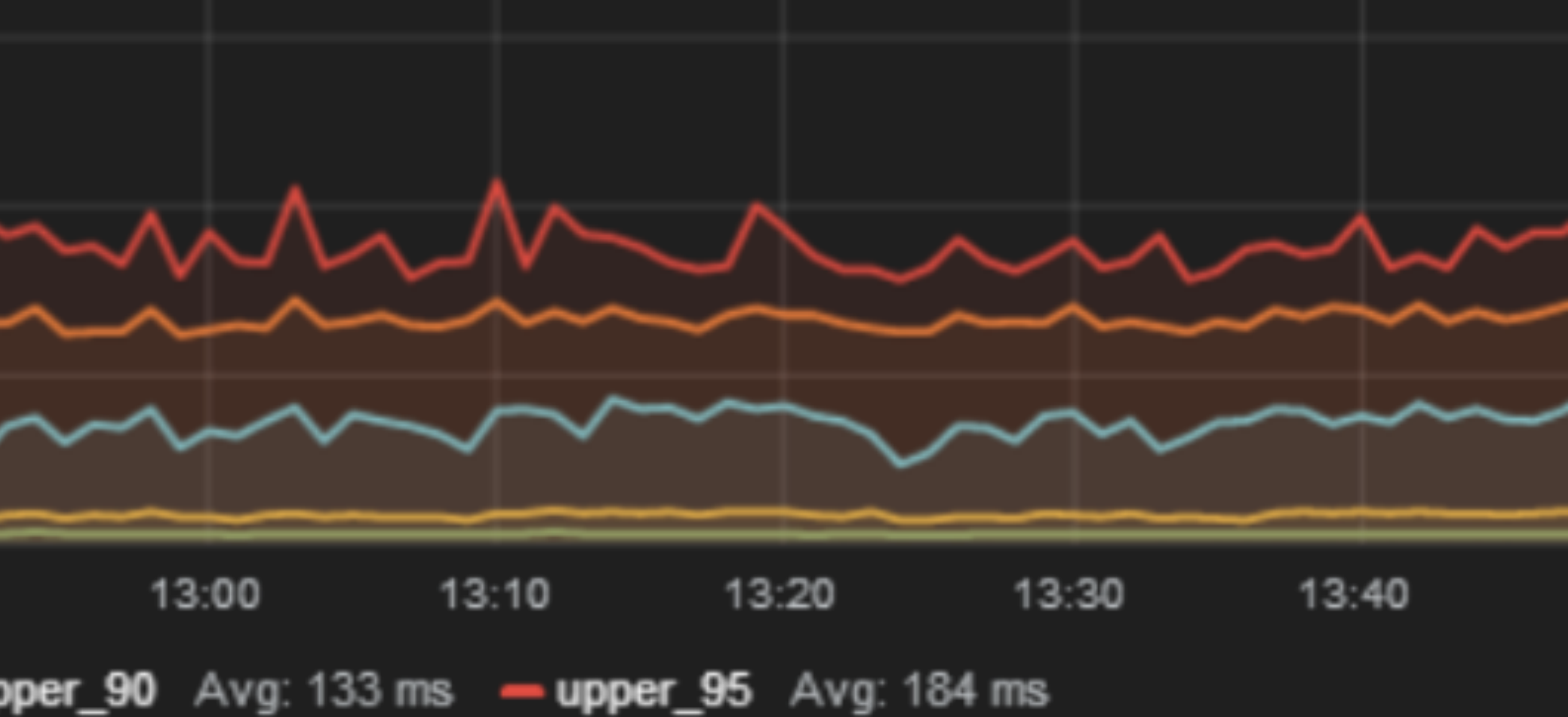
#LatencyTipOfTheDay: Q: What's wrong with this picture? A: Everything!

Question: What's wrong with this picture:



Answer: Everything!





What (TF) does the Average
of the 95%'lie mean?

What (TF) does the Average of the 95%'ile mean?

- Lets do the same with 100%'ile; Suppose we a set of 100%'ile values for each minute:

[1, 0, 3, 1, 601, 4, 2, 8, 0, 3, 3, 1, 1, 0, 2]

“The average 100%'ile over the past 15 minutes was 42”

- Same nonsense applies to averaging any other %'ile

#LatencyTipOfTheDay:

You can't average percentiles.
Period.

99%'lie: a good indicator, right?

What are the chances of a single web page view experiencing >99%'lie latency of:

- A single search engine node?
- A single Key/Value store node?
- A single Database node?
- A single CDN request?
- A single (micro)service call?

Site	# of requests
amazon.com	190
kohls.com	204
jcrew.com	112
saksfifthavenue.com	109
--	--
nytimes.com	173
cnn.com	279
--	--
twitter.com	87
pinterest.com	84
facebook.com	178
--	--
google.com (yes, that simple noise-free page)	31
google.com search for "http requests per page"	76

Site	# of requests	page loads that would experience the 99% th ile [(1 - (.99 ^ N)) * 100%]
amazon.com	190	85.2%
kohls.com	204	87.1%
jcrew.com	112	67.6%
saksfifthavenue.com	109	66.5%
--	--	--
nytimes.com	173	82.4%
cnn.com	279	93.9%
--	--	--
twitter.com	87	58.3%
pinterest.com	84	57.0%
facebook.com	178	83.3%
--	--	--
google.com (yes, that simple noise-free page)	31	26.7%
google.com search for "http requests per page"	76	53.4%

#LatencyTipOfTheDay:

MOST page loads will experience the
99%ile server response

Gauging user experience

Example: If a typical user session involves 5 page loads, averaging 40 resources per page.

- How many of our users will NOT experience something worse than the 95%ile of http requests?

Answer: ~0.003%

- How many of our users will experience at least one response that is longer than the 99.9%ile?

Answer: ~18%

#LatencyTipOfTheDay:

Median Server Response Time:
The number that 99.99999999999999% of
page views can be worse than

The coordinated omission problem

An accidental conspiracy...

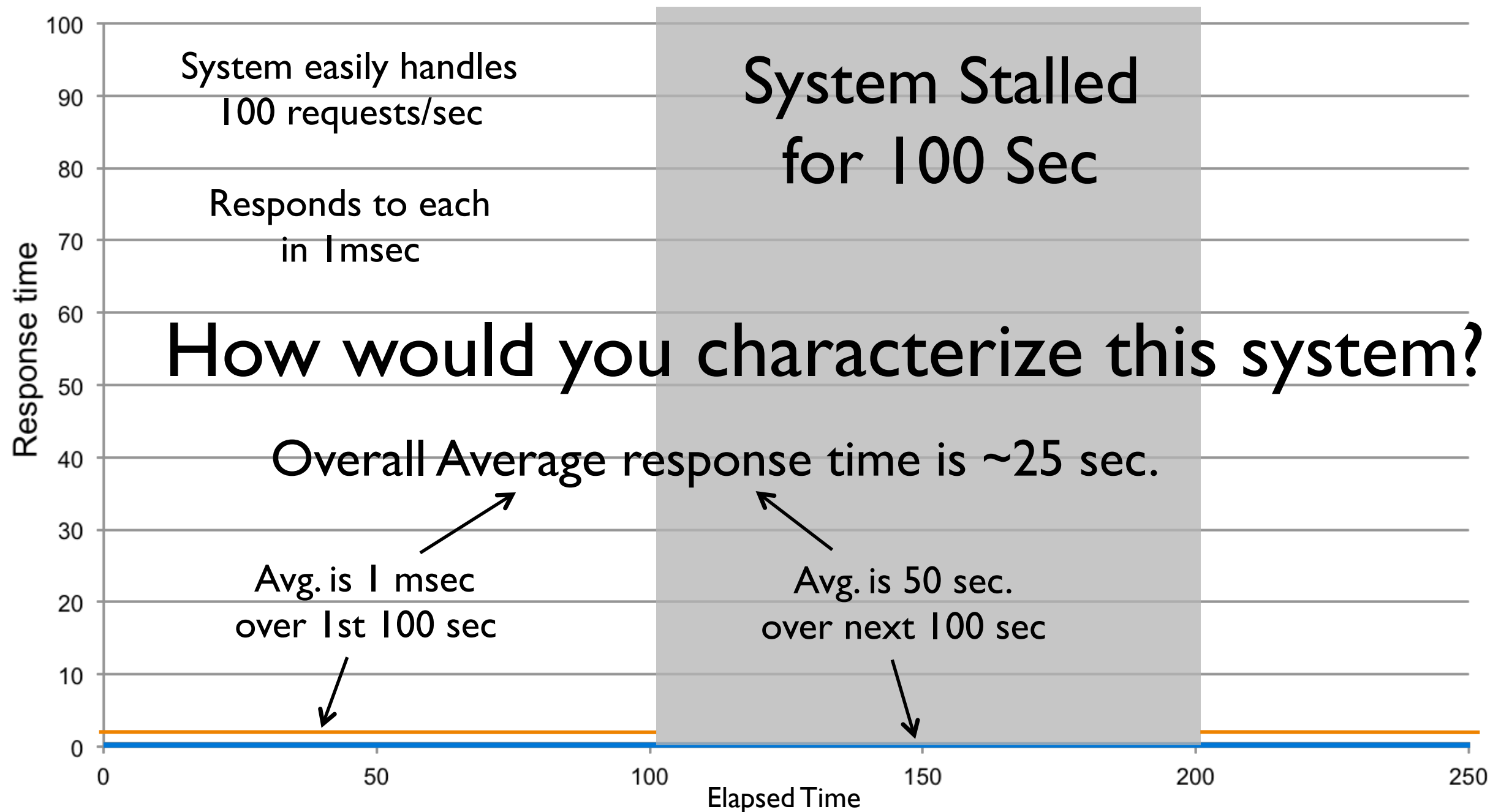
The lie in the 99%'lies

Coordinated Omission in Monitoring Code

```
/**
 * Performs the actual reading of a row out of the StorageService, fetching
 * a specific set of column names from a given column family.
 */
public static List<Row> read(List<ReadCommand> commands, ConsistencyLevel consistency_level)
    throws UnavailableException, IsBootstrappingException, ReadTimeoutException
{
    if (StorageService.instance.isBootstrapMode())
        throw new IsBootstrappingException();
    long startTime = System.nanoTime();
    List<Row> rows;
    try
    {
        rows = fetchRows(commands, consistency_level);
    }
    finally
    {
        readMetrics.addNano(System.nanoTime() - startTime);
    }
    return rows;
}
```

- Long operations only get measured once
- delays outside of timing window do not get measured at all

How bad can this get?

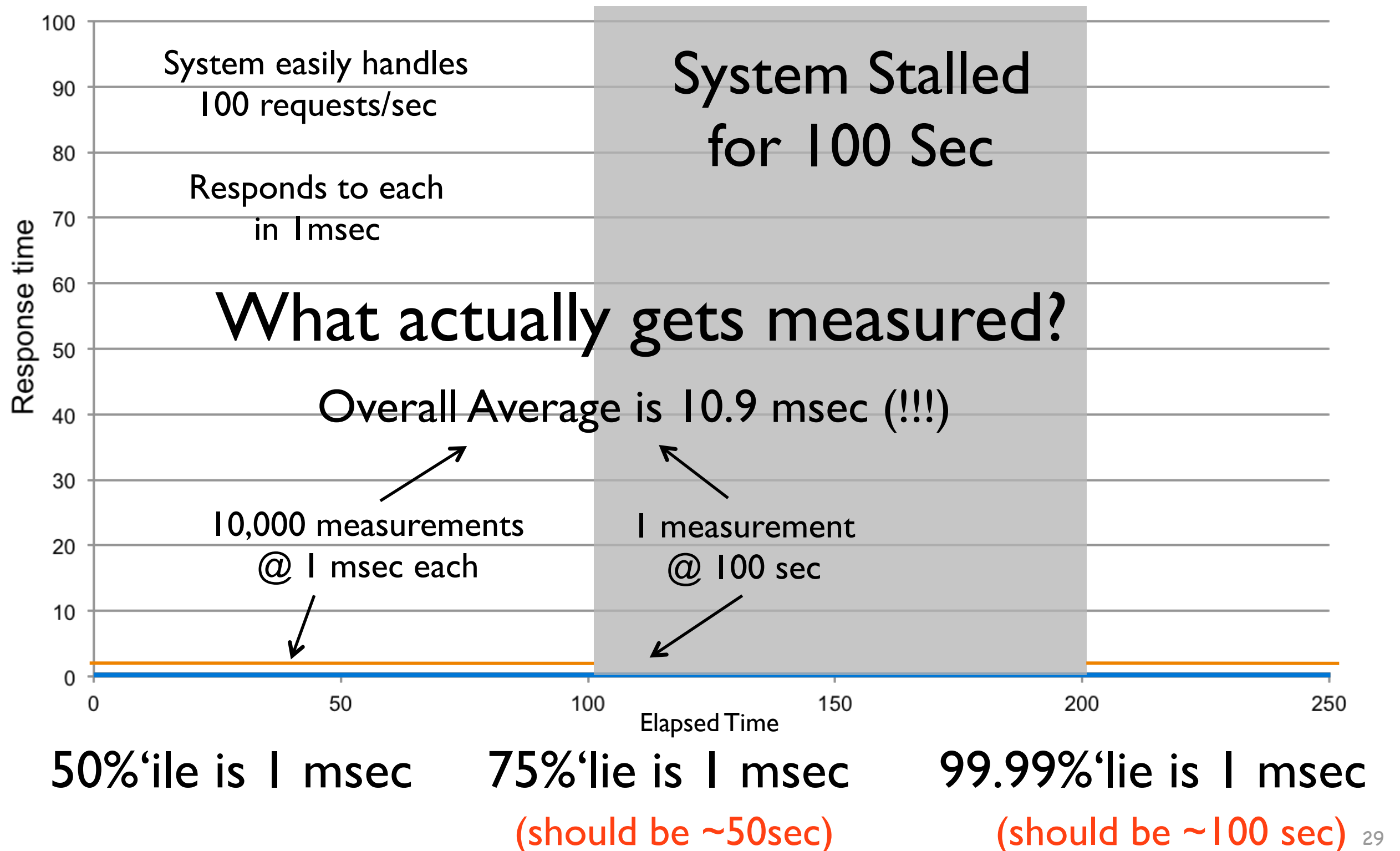


~50%ile is 1 msec

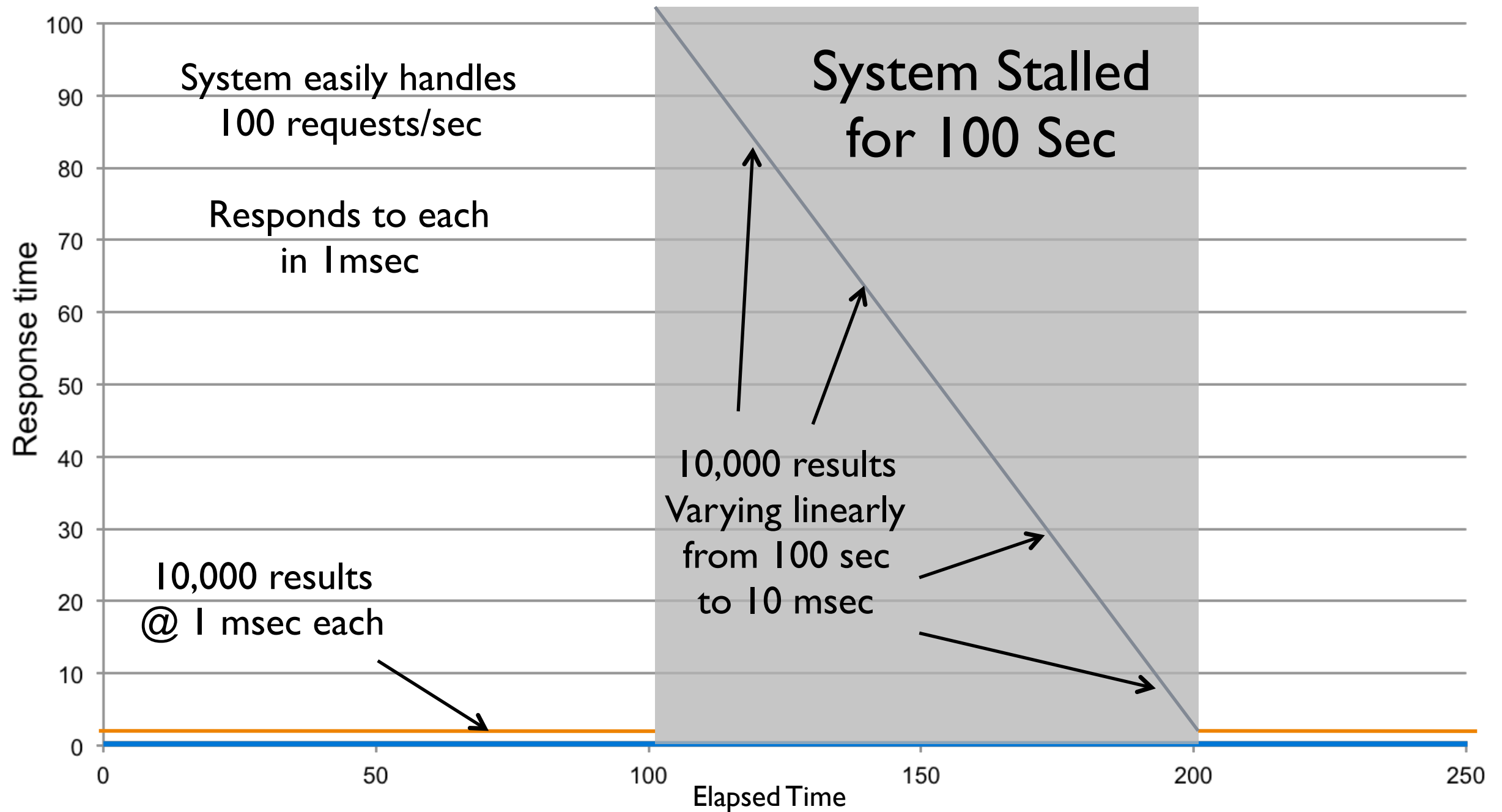
~75%ile is 50 sec

99.99%ile is ~100sec

Measurement in practice



Proper measurement



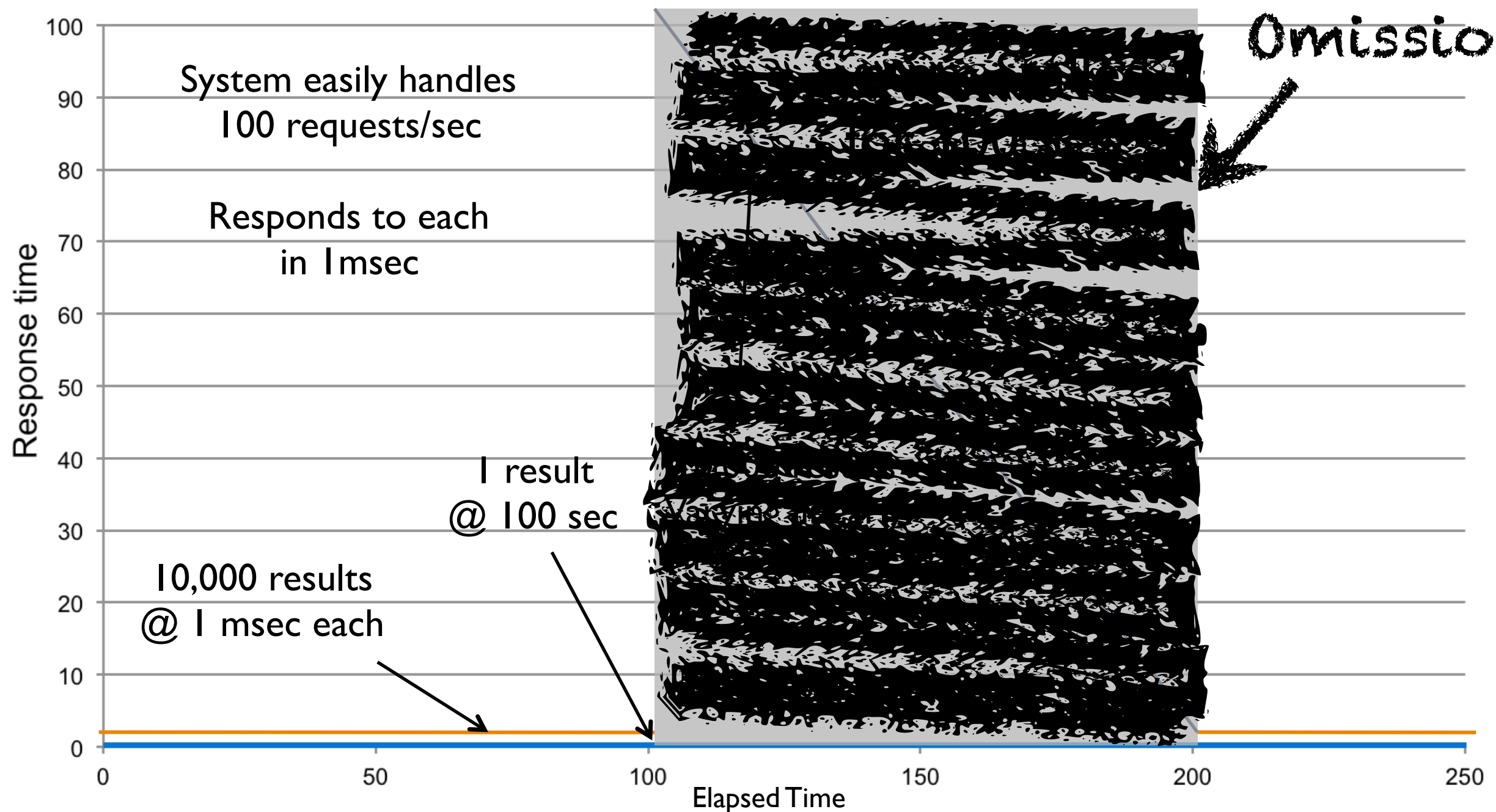
~50%ile is 1 msec

~75%ile is 50 sec

99.99%ile is ~100sec

Proper measurement

Coordinated

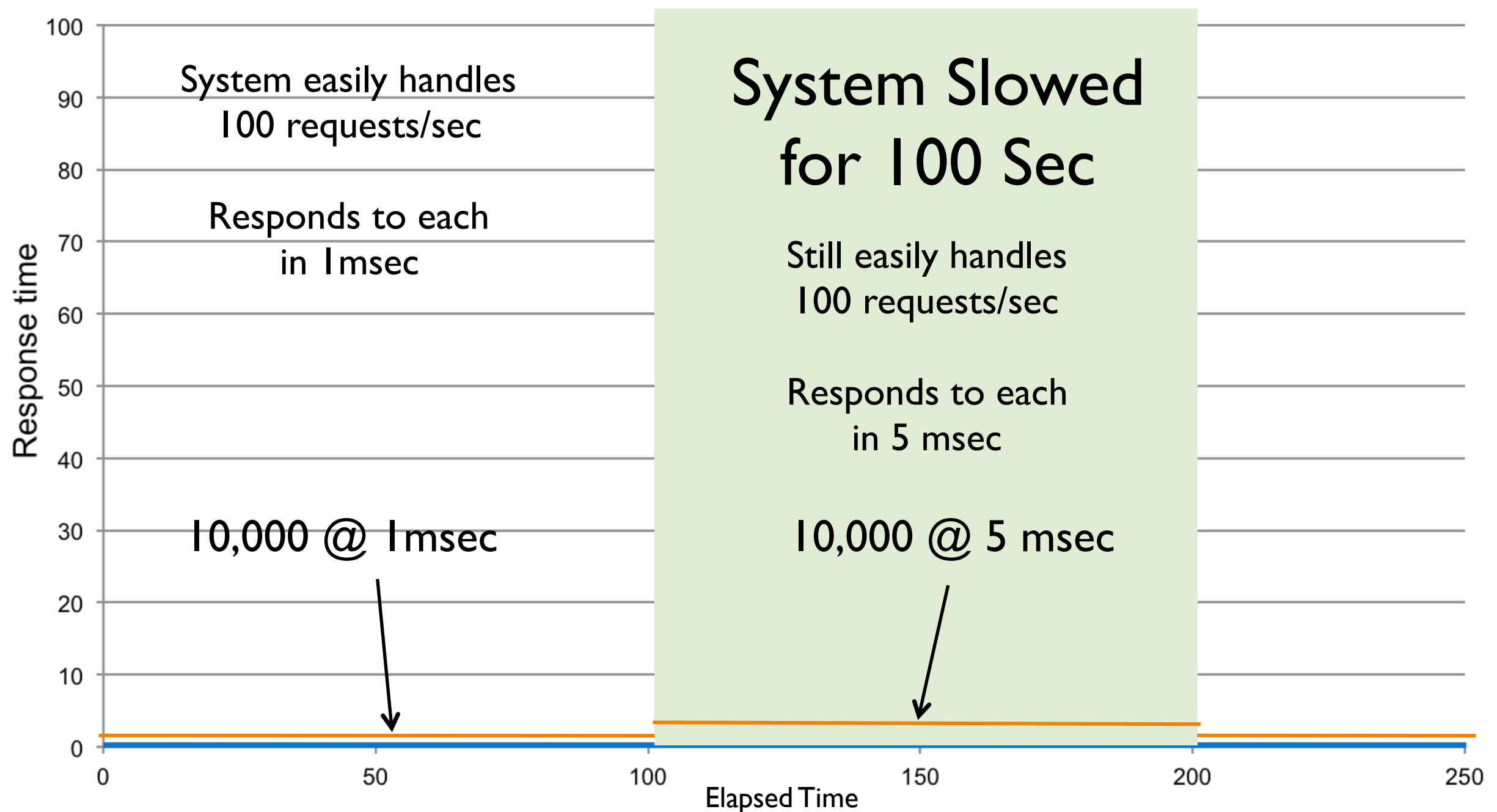


~50%ile is 1 msec

~75%ile is ~~50 sec~~
1 msec

99.99%ile is ~~100 sec~~
1 msec

"Better" can look "Worse"

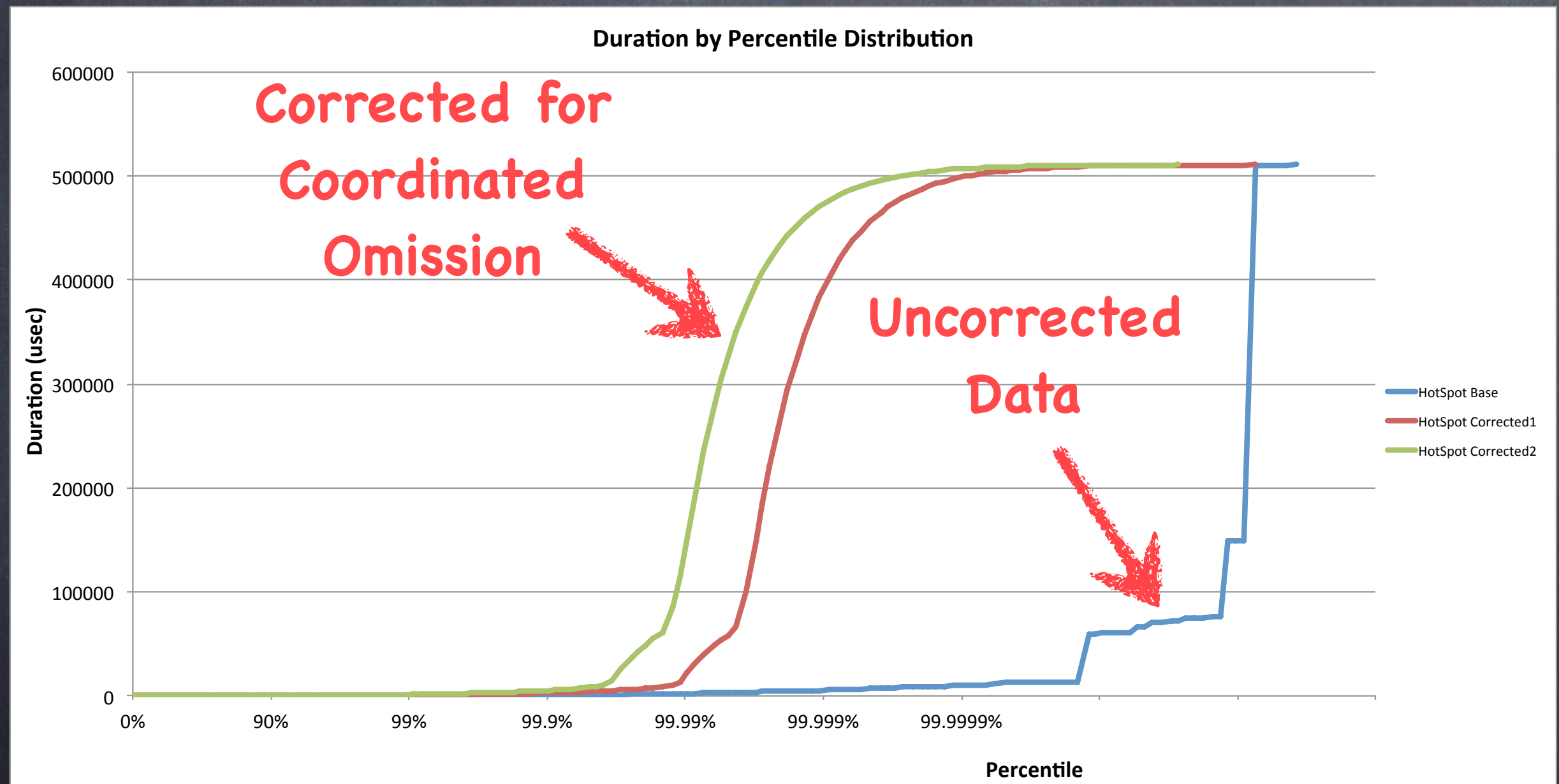


50%ile is 1 msec

75%ile is 2.5msec
(stalled shows 1 msec)

99.99%ile is ~5msec
(stalled shows 1 msec)

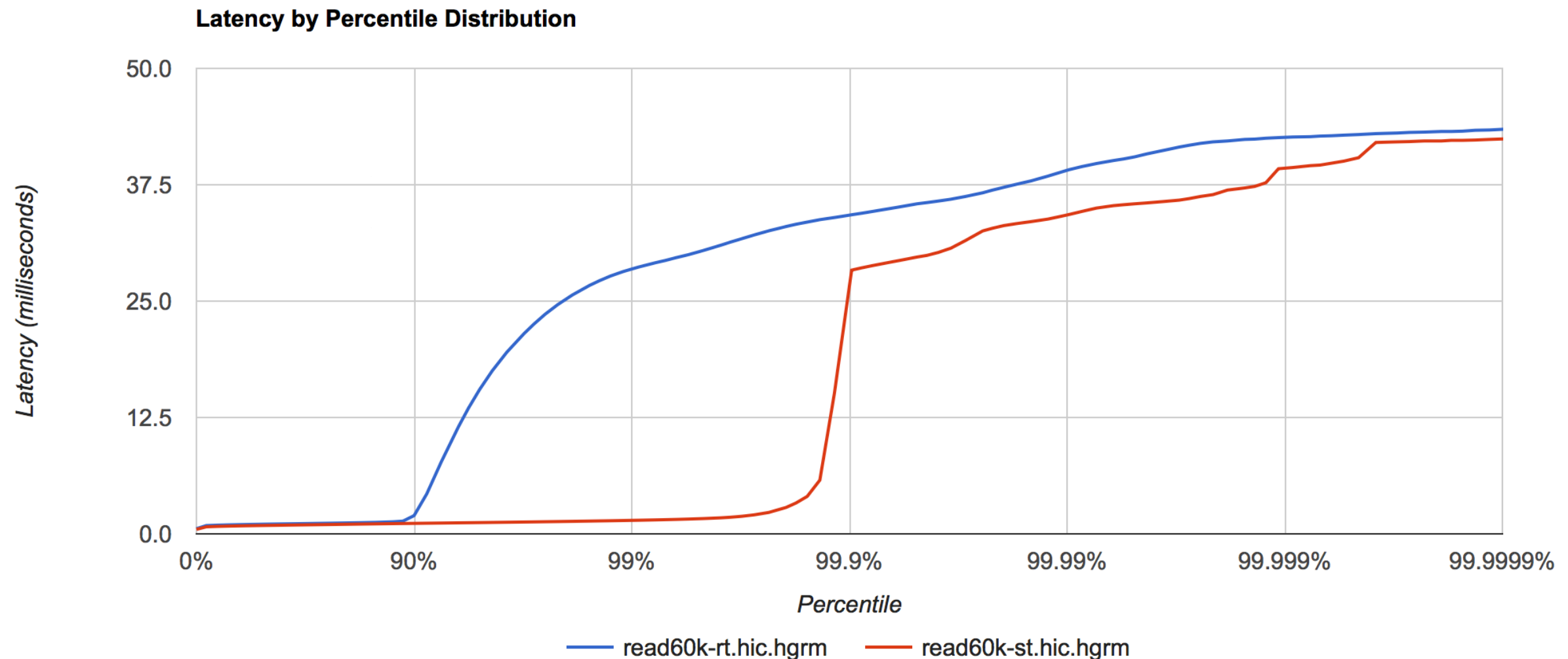
Real World Coordinated Omission effects



Service Time vs. Response Time



Response Time vs. Service Time @60K/sec



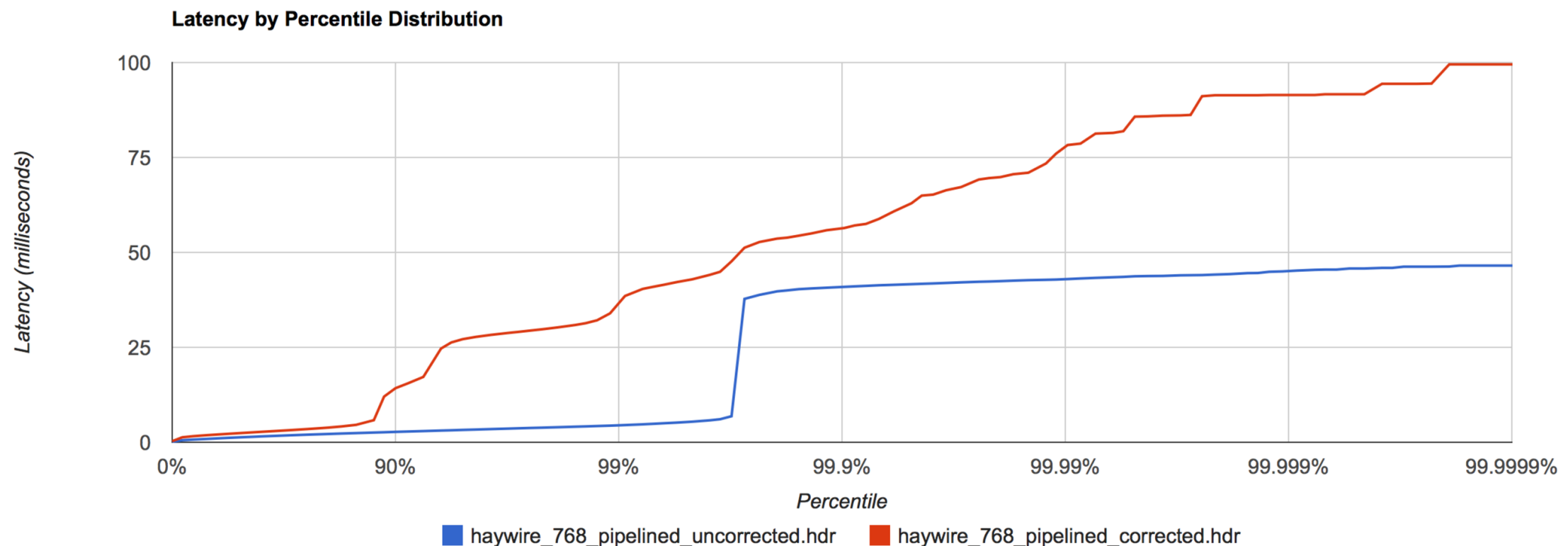
How “real” people react



Kelly Sommers @kellabyte

2d

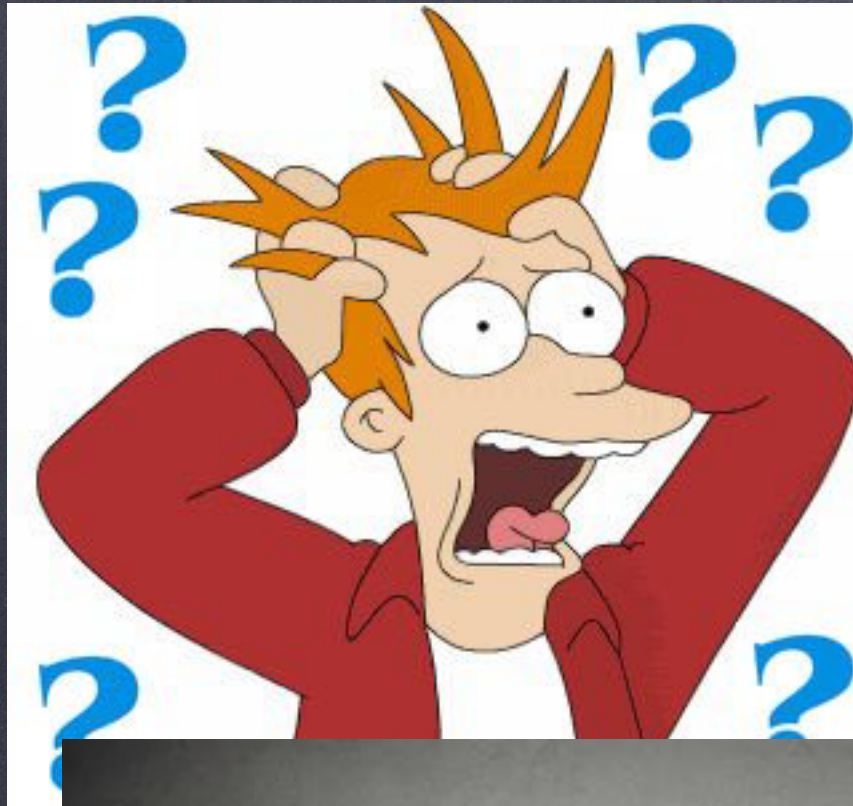
LOL at how badly we all benchmark. Blue is how most of us are benchmarking, Red is the actual truth i.imgur.com/HYoWEu6.png



Leandro Pereira @lafp

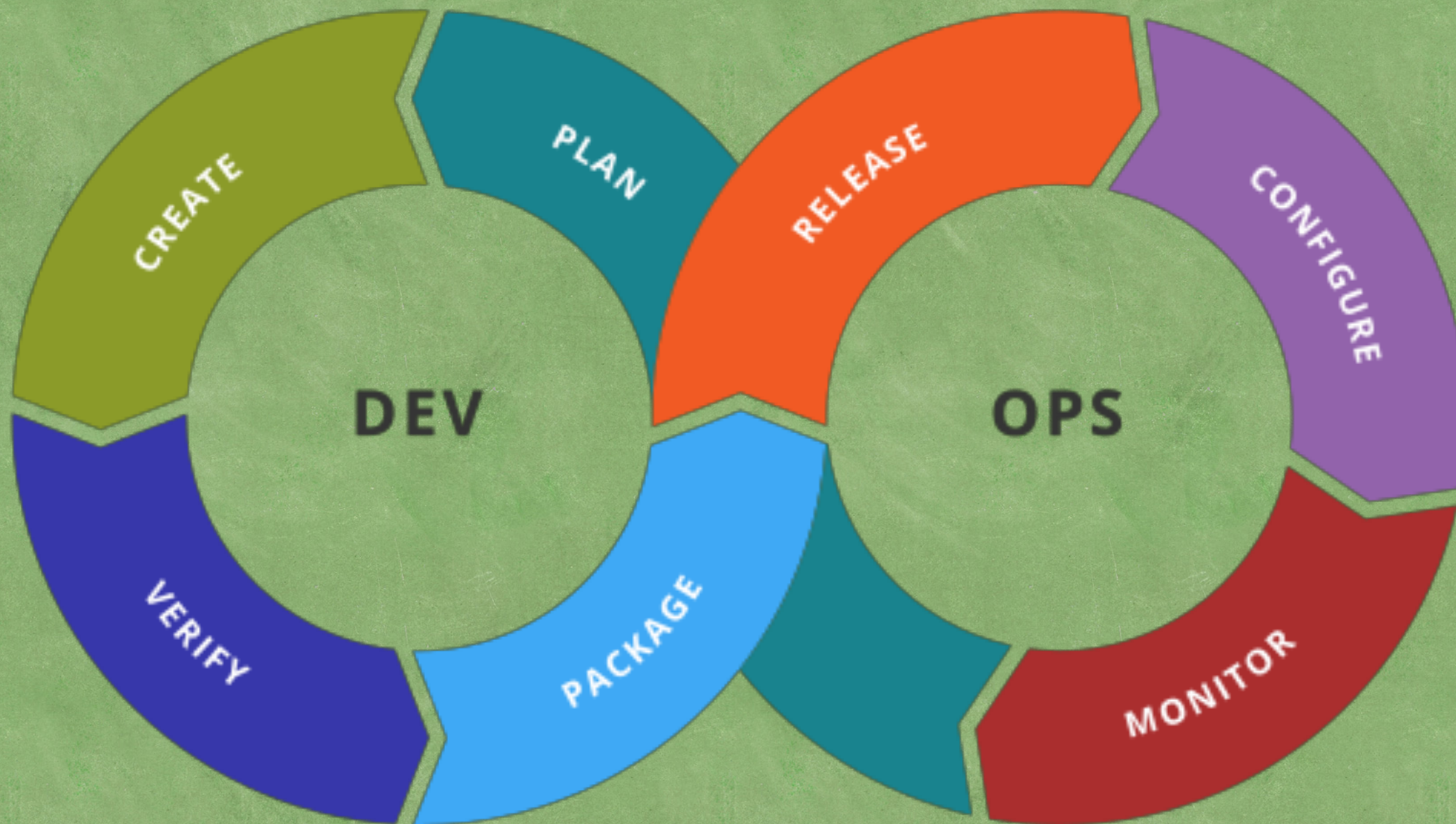


@kellabyte Blue, you believe in whatever you want to believe. Red, you wake up in Wonderland and see how deep the rabbit hole goes.



How does the world
even keep working ?!?!?





OPS

MONITOR



WHY ARE
BLACK CATS
BAD LUCK?











Retries

Misses

Timeouts

Failed
Queries

Angry
Phone Calls

Abandoned
Shopping Carts

Loss of
Engagement



It is nonsense to ask:
“was my 99%’lie a timeout?”

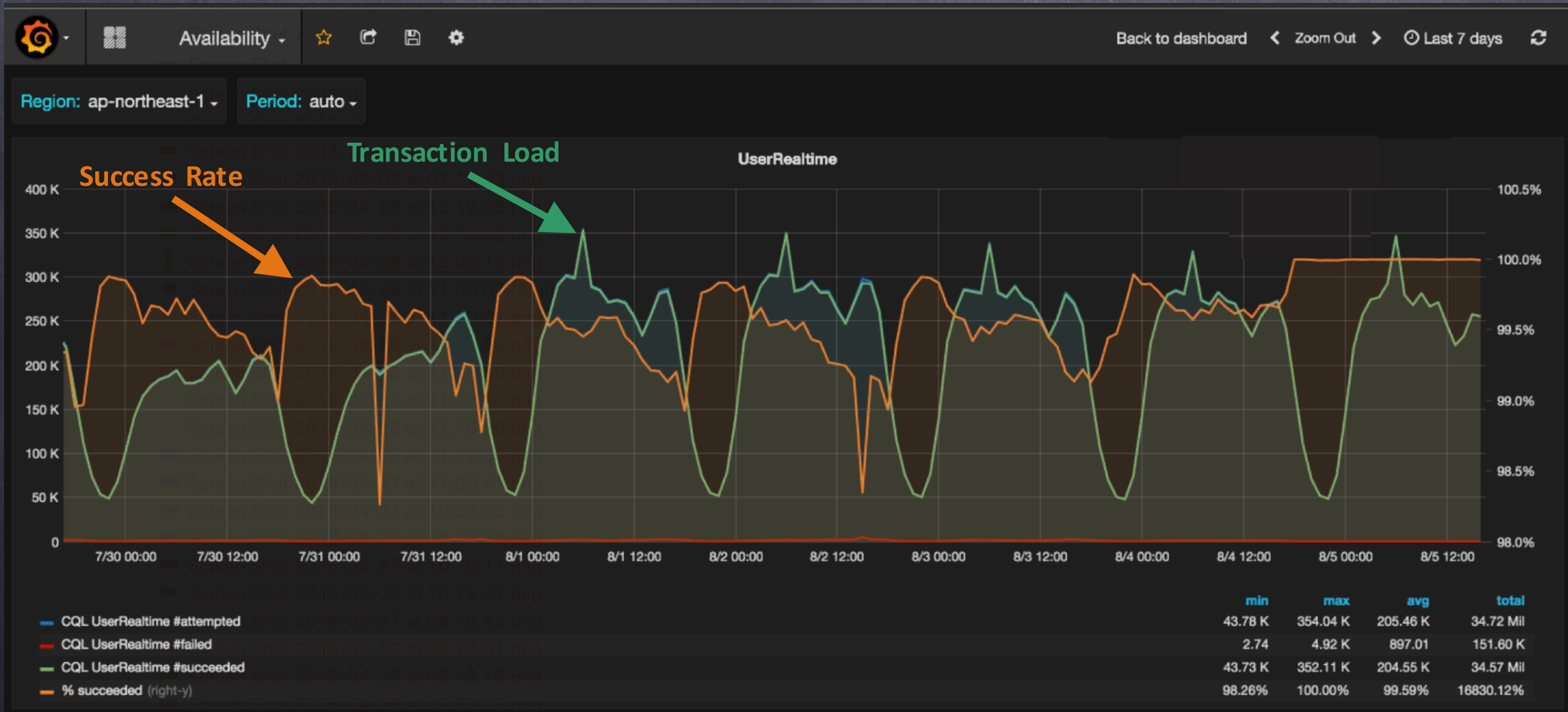


Instead Ask:

“How many timeouts
did I have?”

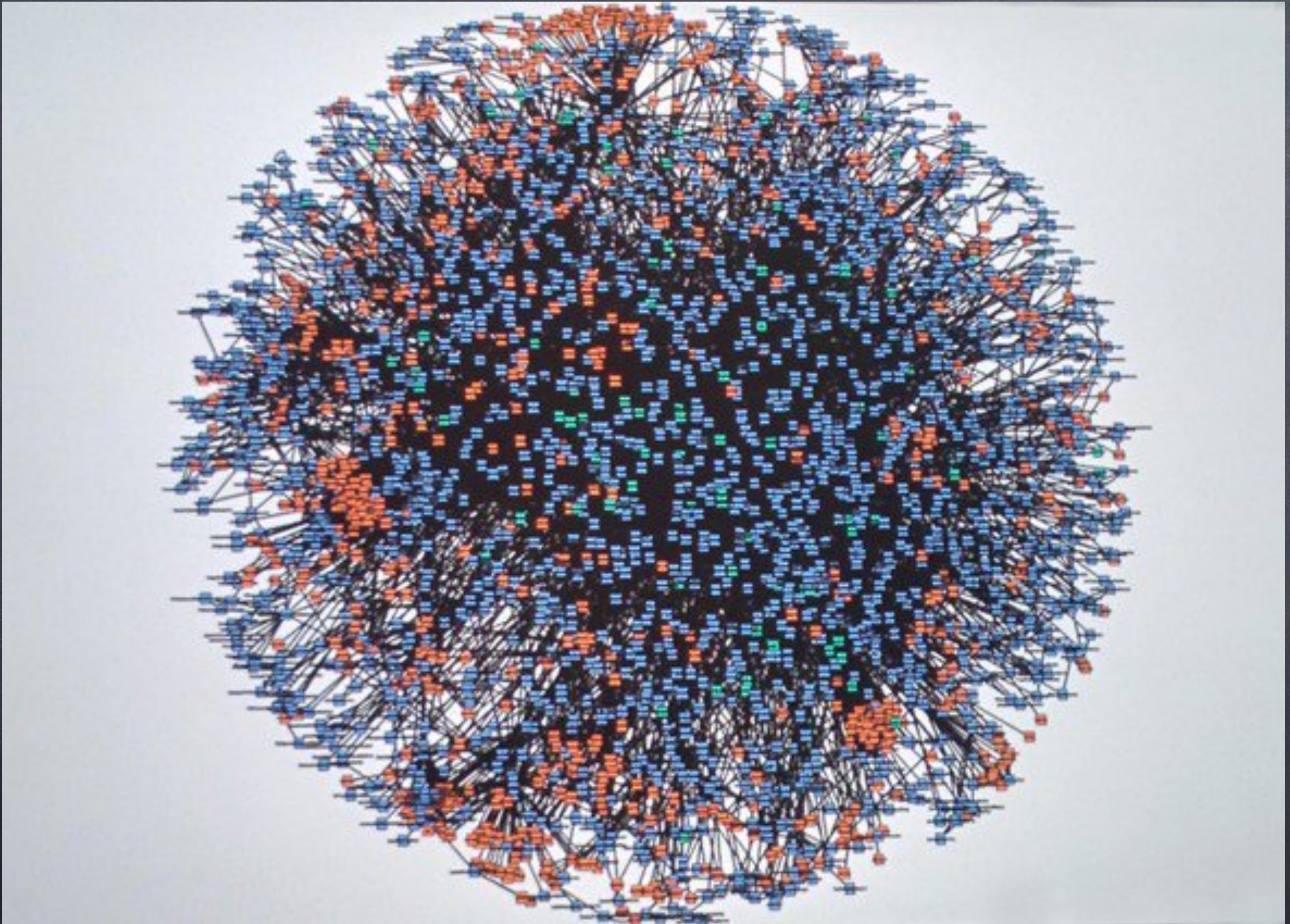
“What % of my
requests timed out?”

Looking at misery





Source: "Architecting for Speed: How agile innovators accelerate growth through microservices", Jesper Nordstrom & Tomas Betzholtzblog, 3gamma.com



Source: "Microservices at Amazon", Alan Ho (@karlunho), apigee developer blog, image from circa 2009

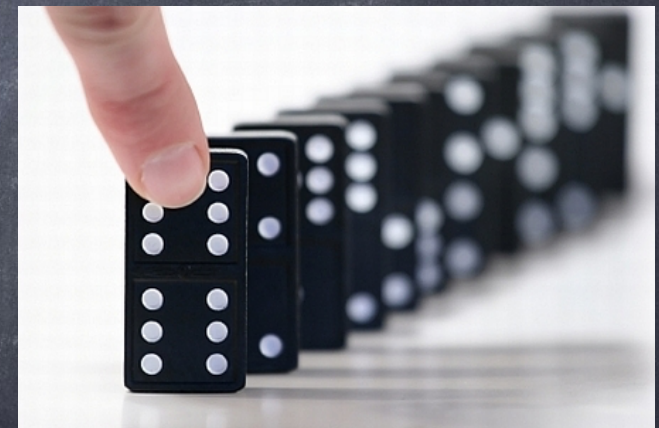
This is a *Microservice*



This is a Microservices Architecture



Focus on measuring Misery



Focus on measuring Misery



Don't
Worry



Love
Misery

@giltene

final #LatencyTipOfTheDay:
Don't aim to make a %'ile acceptable
Instead, aim for an acceptable %
... of misery

Don't
Worry



Love
Misery

