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
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
Latency Behavior

- Latency: The time it took one operation to happen
- Each operation occurrence has its own latency
- What we care about is how latency behaves
- Behavior is a lot more than “the common case was X”
We like to look at pretty charts...

The “We only want to show good things” chart
A real world, real time example
A real world, real time example
A real world, real time example

So this is a better picture. Right?
Why do we tend to avoid plotting Max latency?

Because no other %ile will be visible on the same chart...
I like to rant about latency...

Saturday, June 21, 2014

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

Question: What's wrong with this picture:

Answer: Everything!
#LatencyTipOfTheDay:

If you are not measuring and/or plotting Max, what are you hiding (from)?
What (TF) does the Average of the 95th percentile mean?
What (TF) does the Average of the 95'lie mean?

Let's 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 any other %'lie
#LatencyTipOfTheDay:

You can't average percentiles.

Period.
Percentiles Matter
Is the 99%’lie “rare”?
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?
<table>
<thead>
<tr>
<th>Site</th>
<th># of requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>amazon.com</td>
<td>190</td>
</tr>
<tr>
<td>kohls.com</td>
<td>204</td>
</tr>
<tr>
<td>jcrew.com</td>
<td>112</td>
</tr>
<tr>
<td>saksfifthavenue.com</td>
<td>109</td>
</tr>
<tr>
<td></td>
<td>--</td>
</tr>
<tr>
<td>nytimes.com</td>
<td>173</td>
</tr>
<tr>
<td>cnn.com</td>
<td>279</td>
</tr>
<tr>
<td></td>
<td>--</td>
</tr>
<tr>
<td>twitter.com</td>
<td>87</td>
</tr>
<tr>
<td>pinterest.com</td>
<td>84</td>
</tr>
<tr>
<td>facebook.com</td>
<td>178</td>
</tr>
<tr>
<td></td>
<td>--</td>
</tr>
<tr>
<td>google.com</td>
<td>31</td>
</tr>
<tr>
<td>(yes, that simple noise-free page)</td>
<td></td>
</tr>
<tr>
<td>google.com</td>
<td>76</td>
</tr>
<tr>
<td>search for &quot;http requests per page&quot;</td>
<td></td>
</tr>
<tr>
<td>Site</td>
<td># of requests</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>amazon.com</td>
<td>190</td>
</tr>
<tr>
<td>kohls.com</td>
<td>204</td>
</tr>
<tr>
<td>jcrew.com</td>
<td>112</td>
</tr>
<tr>
<td>saksfifthavenue.com</td>
<td>109</td>
</tr>
<tr>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>nytimes.com</td>
<td>173</td>
</tr>
<tr>
<td>cnn.com</td>
<td>279</td>
</tr>
<tr>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>twitter.com</td>
<td>87</td>
</tr>
<tr>
<td>pinterest.com</td>
<td>84</td>
</tr>
<tr>
<td>facebook.com</td>
<td>178</td>
</tr>
<tr>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>google.com (yes, that simple noise-free page)</td>
<td>31</td>
</tr>
<tr>
<td>google.com search for &quot;http requests per page&quot;</td>
<td>76</td>
</tr>
</tbody>
</table>
#LatencyTipOfTheDay:

MOST page loads will experience the 99% 'lie server response
Which HTTP response time metric is more “representative” of user experience?

The 95%’lie or the 99.9%’lie
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% ‘lie of http requests?

  Answer: ~0.003%

- How may of our users will experience at least one response that is longer than the 99.9% ‘lie?

  Answer: ~18%
Gauging user experience

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

- What http response percentile will be experienced by the 95%’ile of users?
  Answer: ~99.97%

- What http response percentile will be experienced by the 99%’ile of users
  Answer: ~99.995%
#LatencyTipOfTheDay:

Median Server Response Time:
The number that 99.99999999999% of page views can be worse than
Why don’t we have response time or latency stats with multiple 9s in them???
Why don’t we have response time or latency stats with multiple 9s in them???

You can’t average percentiles...

And you also can’t get an hour’s 99.999%’lie out of lots of 10 second interval 99%’lie reports...
You can’t average percentiles...

Why don’t we have response time or latency stats with multiple 9s in them???

Check out HdrHistogram

It lets you have nice things....
Latency “wishful thinking”

- We know how to compute averages & std. deviation, etc.
- Wouldn’t it be nice if latency had a normal distribution?
- The average, 90%’lie, 99%’lie, std. deviation, etc. can give us a “feel” for the rest of the distribution, right?
- If 99% of the stuff behaves well, how bad can the rest be, really?
The real world: latency distribution

Latency by Percentile Distribution

Latency (msec)

Percentile
The real world: latency distribution

Latency by Percentile Distribution

Latency (msec)

Percentile

0% 90% 99% 99.9%

0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
The real world: latency distribution

Latency by Percentile Distribution

Latency (msec)

Percentile

0% 90% 99% 99.9% 99.9% 99.99% 99.999% 99.9999%
Dispelling standard deviation

Latency by Percentile Distribution

Percentile

Latency (msec)

0% 90% 99% 99.9% 99.99% 99.999% 99.9999%

A B C D E F
Dispelling standard deviation

Mean = 0.06 msec
Std. Deviation (σ) = 0.21msec

99.999% = 38.66msec
~184 σ (!!!) away from the mean

In a normal distribution,
the 99.999%’ile falls within 4.5 σ

These are NOT normal distributions
The coordinated omission problem

An accidental conspiracy...

The lie in the 99%’lies
The coordinated omission problem

Common Example A (load testing):
- each “client” issues requests at a certain rate
- measure/log response time for each request

So what’s wrong with that?
- works **only** if ALL responses fit within interval
- implicit “automatic back off” coordination
Coordinated Omission in Monitoring Code

Long operations only get measured once

delays outside of timing window do not get measured at all
How bad can this get?

System easily handles 100 requests/sec
System Stalled for 100 Sec
Responds to each in 1msec

How would you characterize this system?

Overall Average response time is ~25 sec.

~50%‘ile is 1 msec  ~75%‘ile is 50 sec  99.99%‘ile is ~100sec

Avg. is 1 msec over 1st 100 sec
Avg. is 50 sec. over next 100 sec
Measurement in practice

System easily handles 100 requests/sec
Responds to each in 1 msec

What actually gets measured?
Overall Average is 10.9 msec (!!!)

50%‘ile is 1 msec  
75%‘lie is 1 msec  
99.99%‘lie is 1 msec

(should be ~50sec)  
(should be ~100 sec)
Proper measurement

System easily handles 100 requests/sec
Responds to each in 1msec

System Stalled for 100 Sec

10,000 results Varying linearly from 100 sec to 10 msec

10,000 results @ 1 msec each

~50%‘ile is 1 msec  ~75%‘ile is 50 sec  99.99%‘ile is ~100sec
Proper measurement

System easily handles 100 requests/sec
Responds to each in 1msec
10,000 results varying linearly from 100 sec to 10 msec

~50%‘ile is 1 msec  ~75%‘ile is 50 msec  99.99%‘ile is 100 msec
"Better" can look "Worse"

System easily handles 100 requests/sec
Responds to each in 1 msec
10,000 @ 1 msec
50%'ile is 1 msec
75%'lie is 2.5 msec
(stalked shows 1 msec)

System Slowed for 100 Sec
Still easily handles 100 requests/sec
Responds to each in 5 msec
10,000 @ 5 msec
99.99%'lie is ~5 msec
(stalked shows 1 msec)
System easily handles 100 requests/sec

Responds to each in 1msec

10,000 results varying linearly from 100 sec to 10 msec

1 result @ 100 sec

10,000 results @ 1 msec each

~50%‘ile is 1 msec

~75%‘ile is 50 msec

99.994%‘ile is 100 msec

“Correction”: “Cheating Twice”

Coordinated Omission

1 msec

9,999 additional results @ 1 msec each

“correction”
Response Time vs. Service Time
Service Time vs. Response Time
Coordinated Omission

*Usually*

makes something that you *think* is a *Response Time* metric

only represent

the *Service Time* component
Response Time vs. Service Time @2K/sec
Response Time vs. Service Time @20K/sec
Response Time vs. Service Time @60K/sec
Response Time vs. Service Time @80K/sec
Response Time vs. Service Time @90K/sec
How “real” people react

Kelly Sommers @kellabyte
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.
Service Time, 90K/s vs 80K/s
Response Time, 90K/s vs 80K/s
Response Time, 90K/s vs 80K/s (to scale)
Latency doesn’t live in a vacuum
Sustainable Throughput: The throughput achieved while safely maintaining service levels
Comparing behavior under different throughputs and/or configurations
Comparing response time or latency behaviors
System A @90K/s & 85K/s vs. System B @90K/s & 85K/s

Wrong Place to Look: They both “suck” at >85K/sec
System A 85K/s vs. System B 85K/s

Looks good, but still the wrong place to look
System A @40K/s vs. System B @40K/s

More interesting...
What can we do with this?
System A @10K/s vs. System B @40K/s

E.g. if “99%’ile < 5msec” was a goal: System B delivers similar 99%’ile and superior 99.9%’ile+ while carrying 4x the throughput
System A @2K/s vs. System B @20K/s

E.g. if “99.9%’ile < 10msec” was a goal:
System B delivers similar 99%’ile and 99.9%’ile while carrying 10x the throughput
System A @2k thru 80k
System A @2k thru 70k
System B @20k thru 70k
System A & System B @2k thru 70k
System A & System B
10K/s thru 60K/s

Lots of conclusions can be drawn from the above...
E.g. System B delivers a consistent 100x reduction in the rate of occurrence of >20msec response times.
System A: 200-1400 msec stalls

System B drawn to scale

---

<table>
<thead>
<tr>
<th>Latency Metric</th>
<th>System A</th>
<th>System B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency mean</td>
<td>30.6 (0.7)</td>
<td>0.6 (0.5)</td>
</tr>
<tr>
<td>Latency median</td>
<td>0.5 (0.5)</td>
<td>0.5 (0.5)</td>
</tr>
<tr>
<td>Latency 95th percentile</td>
<td>244.4 (1.1)</td>
<td>1.0 (0.9)</td>
</tr>
<tr>
<td>Latency 99th percentile</td>
<td>537.4 (2.0)</td>
<td>2.7 (1.9)</td>
</tr>
<tr>
<td>Latency 99.9th percentile</td>
<td>1052.2 (8.4)</td>
<td>13.3 (3.8)</td>
</tr>
<tr>
<td>Latency max</td>
<td>1314.9 (1312.8)</td>
<td>110.6 (28.2)</td>
</tr>
</tbody>
</table>

Response Time  Service time
A simple visual summary

This is Your Load on System A

This is Your Load on System B

Any Questions?
Any Questions?

http://www.azulsystems.com