

*Slow is the new down.*

---

# Understanding Slowness

---

When shit goes wrong,  
the gloves come off.





*The many faces of*

# Theo Schlossnagle

 @postwait

CEO Circonus





---

# Goals

---

- ❖ Approach an understanding of your architecture,
- ❖ Convert this understanding into a strategic plan
- ❖ Develop logistics for diagnosis
- ❖ Discuss discipline around remediation



*The first step*

---

# Understand

---

Build a map

Build two

“If you don’t have a good map of your architecture, Dora will whoop you.”  
-Theo





*How you'd like to think of*

# Your architecture

Elegant  
Beautiful in its simplicity  
Robust  
Resilient





*When in actuality*

---

# Your Architecture

---

*is*

Organically grown  
Cancerous tumors  
Disaster waiting to happen  
Hella complicated

*of which you are*

Inexplicably proud



Photograph courtesy of Herman Rhoids





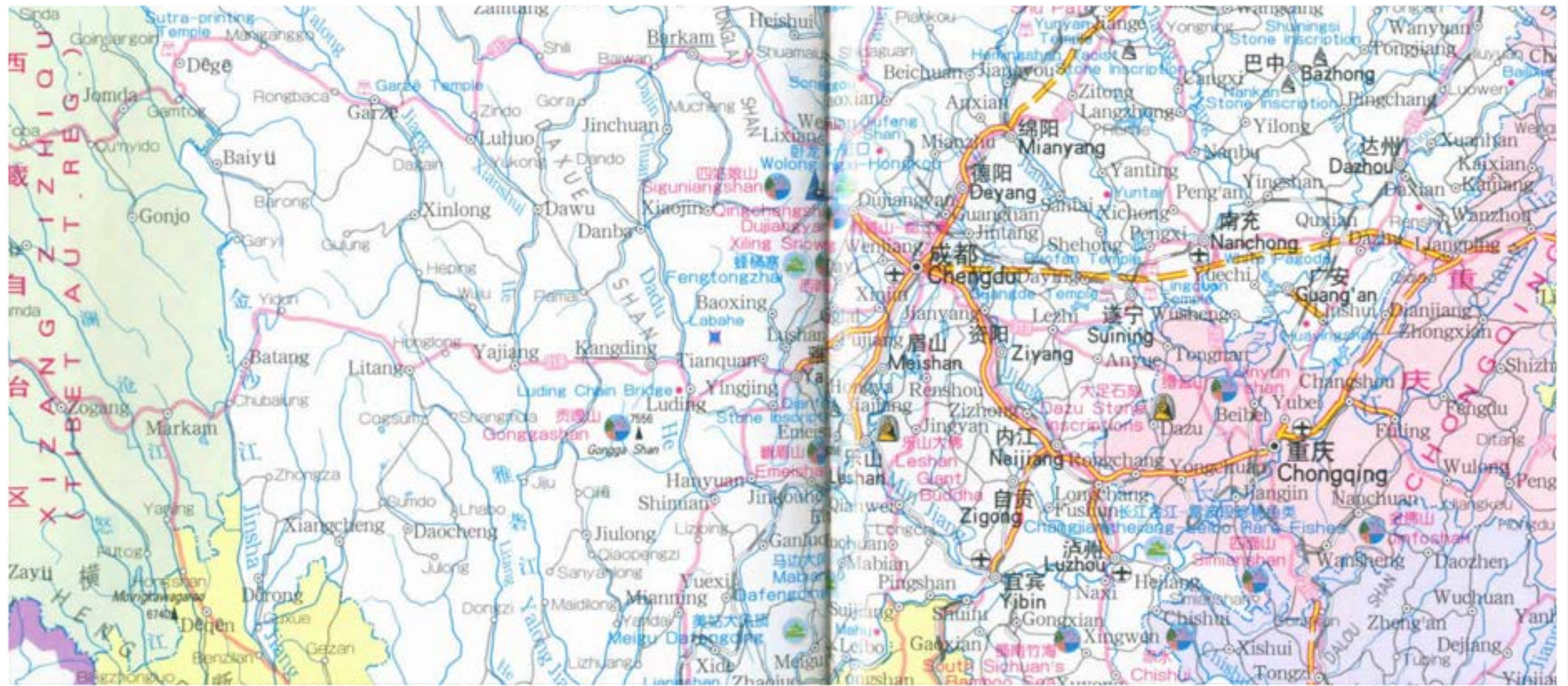
Map #1

# High-level map

Architectural components  
Connectedness  
Data flow







Map #2

# Low-level map

Component versions  
Component languages  
OS/NICs/HBAs  
Location  
Switches/Routers/FW  
Connected Service details







*Develop*

# Strategic Plan

There are 2 types useful SREs:

Spanning several boundaries

Spanning all boundaries



*You can't play ball without bases.*

---

## Who's on first?

---

Establish who is responsible for each component in each context.

Establish who is responsible when that person fails (upward).

Establish who is responsible when that person needs help (upward and downward)



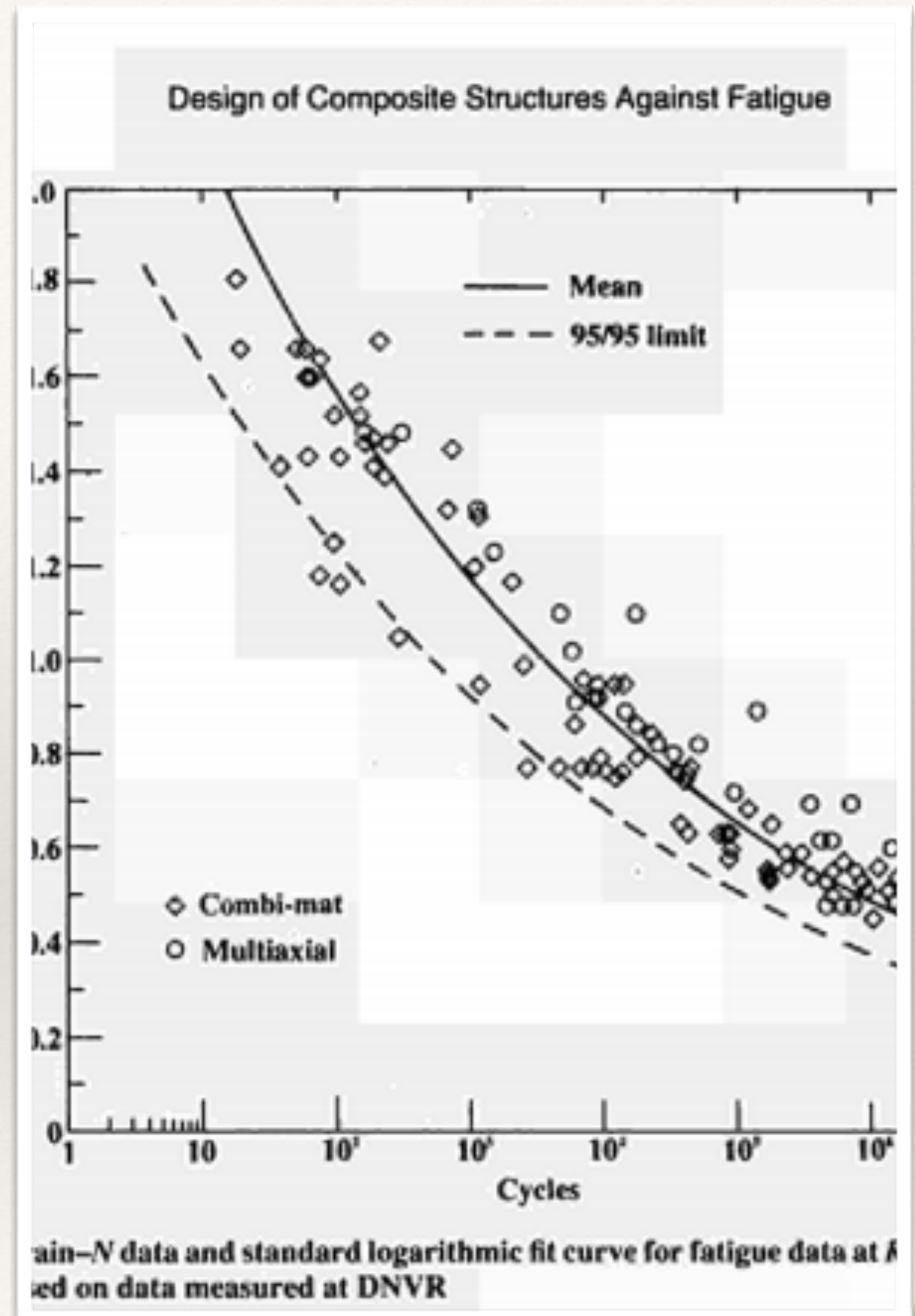


*Nothing will ever be “broken”  
if it isn’t expected to “work.”*

# Expectations

Set expectations for  
breakages and slowdowns.

What you build will break,  
understanding under what stress is your  
job as an engineer.



*Parts are parts.*

---

## Ø tech loyalty

---

Constructing a solution from parts.

Parts are replaceable.

Have a list of replacement vendors of part alternates.

If you design a solution relying on a part available only from a single vendor, you have accomplished lock-in.



Photograph courtesy of Jason Ilagan  
<https://www.flickr.com/photos/thepen/428014152>







*When things are broken (or slow)*

# Logistics matter

Observability

Tool parity

Safety harnesses





*You cannot improve  
what you cannot measure*

---

# Measure

---

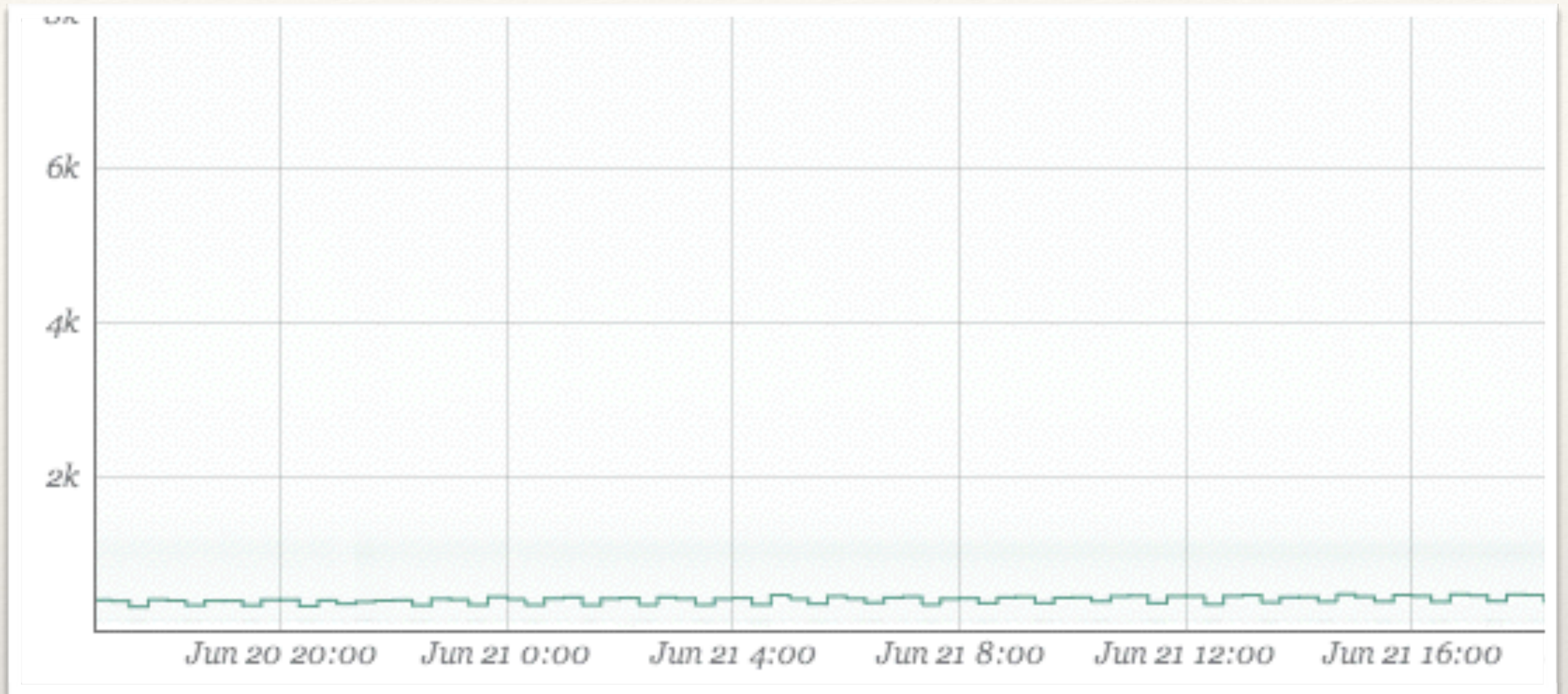
Cut once



Rear Admiral Grace Murray Hopper 1906-1992







*The one beast you cannot slay:*

# Latency

You must subdue it

First you must understand it





Averages are for chumps

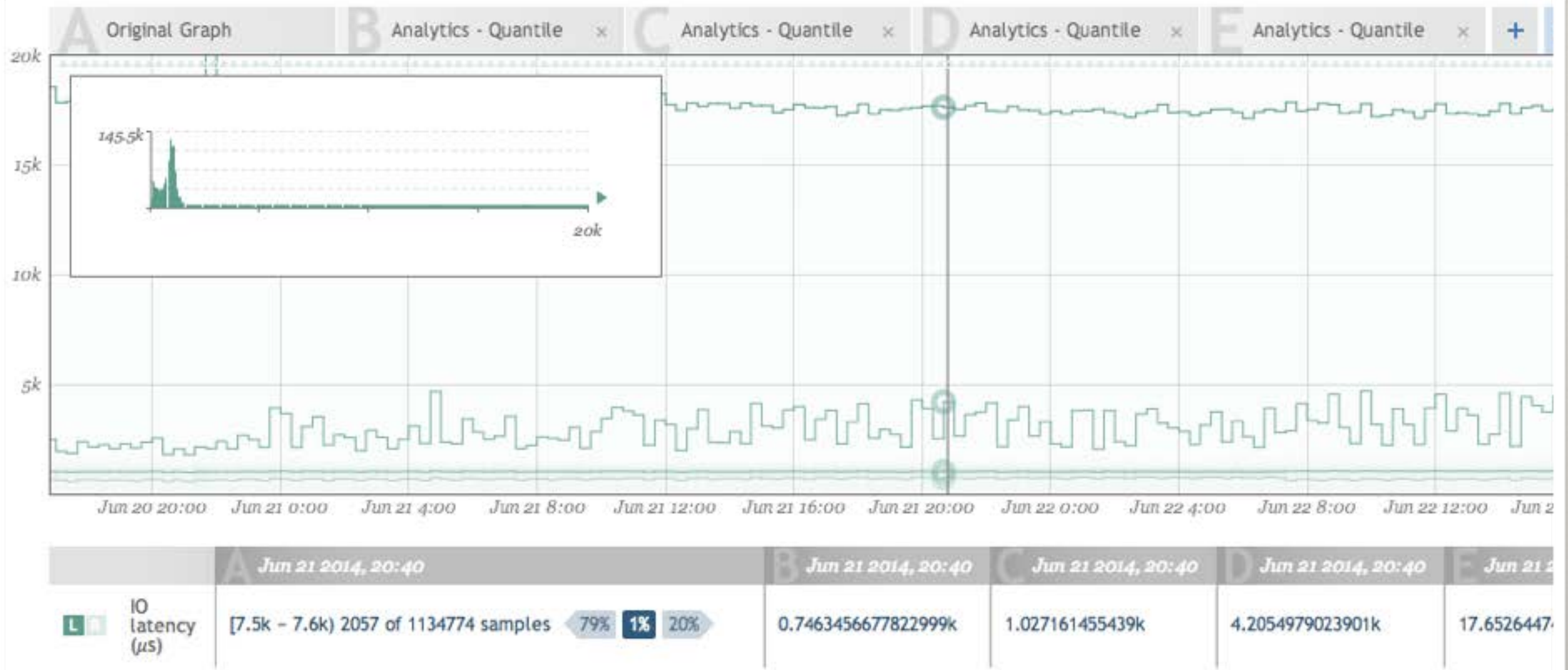
# Histograms over Aggregations

Reducing many observations  $S$  to  $N$  values ( $\forall |N| \ll |S|$ ) is the definition of lossy.

or... “you don’t know shit”







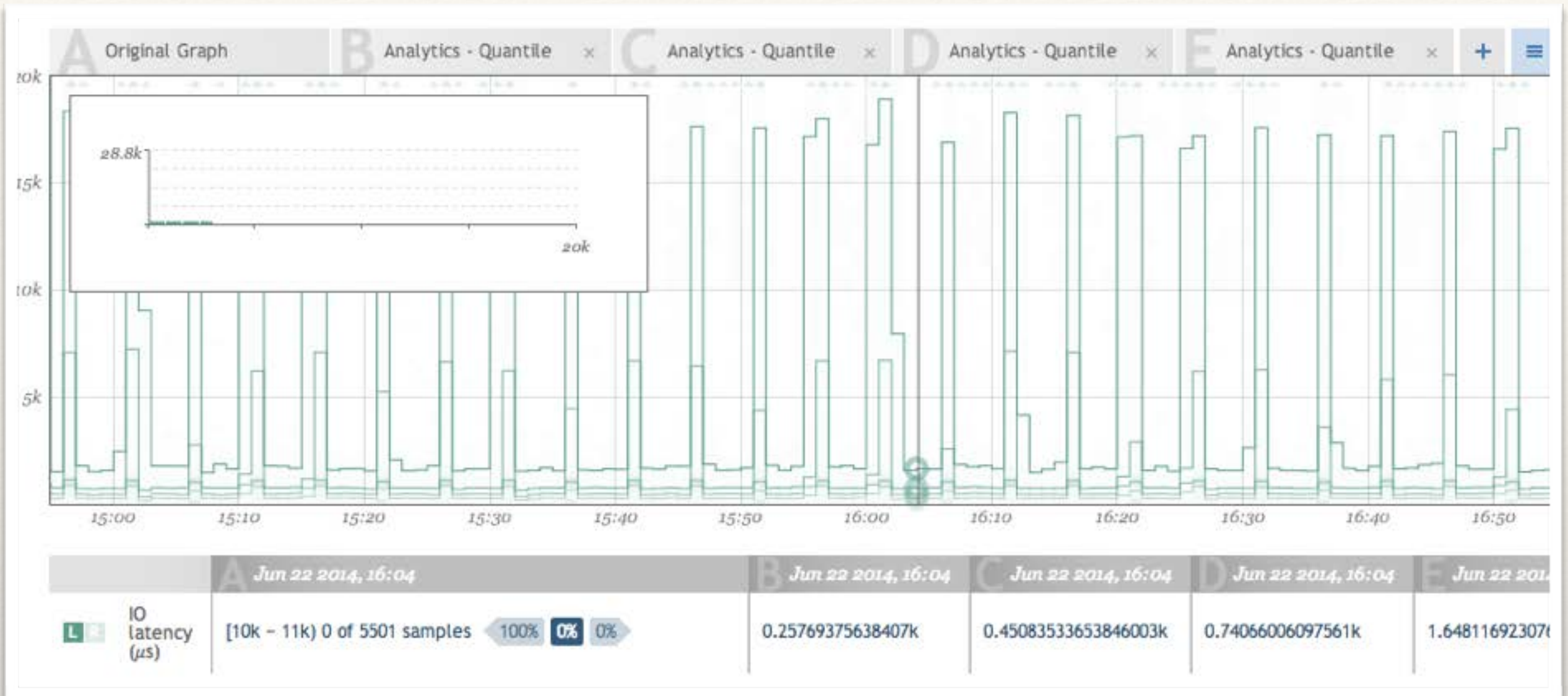
*Exploring quantiles is simple and can provide increased understanding.*

# Quantiles

Time-series histograms are a lot of information to digest.

Moving quantiles can often provide much more insight.





*Remember that you're consolidating time.*

# Granular data

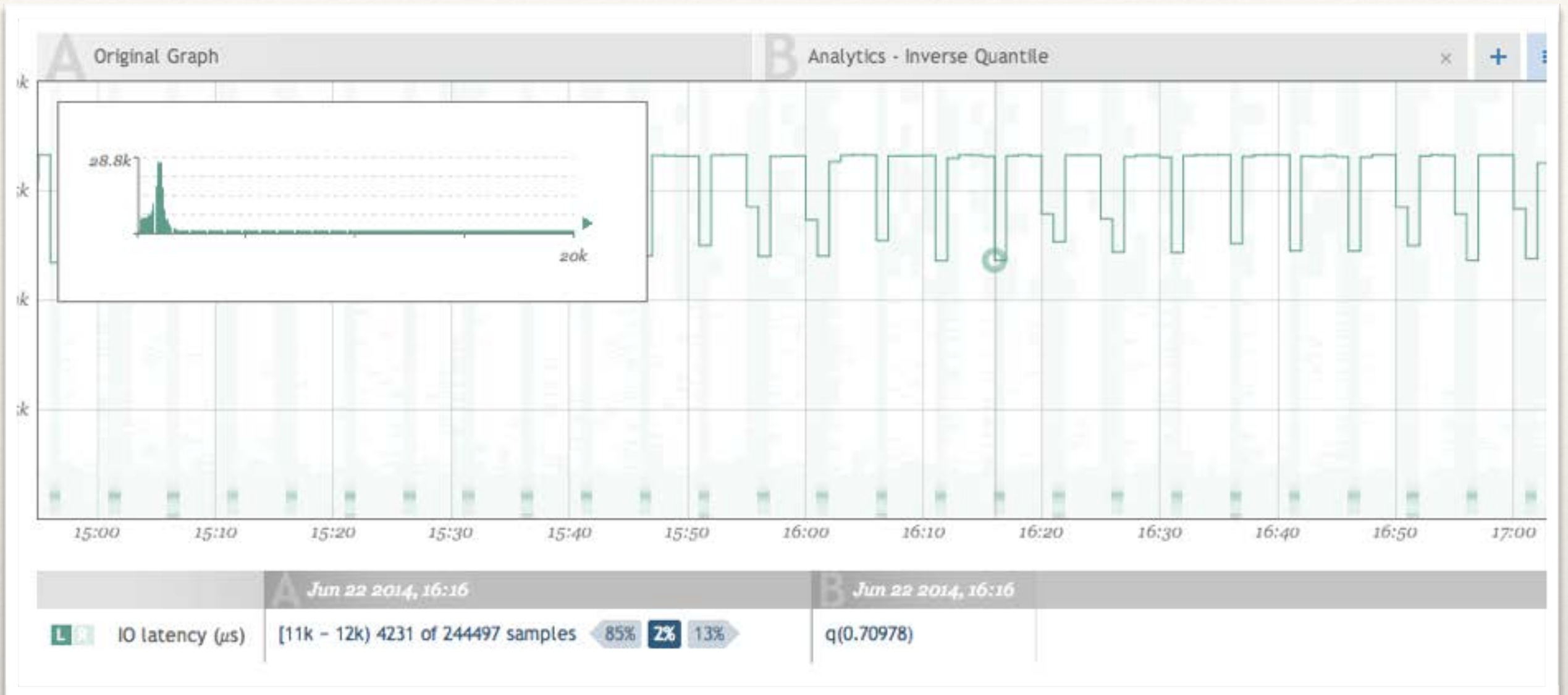
Time consolidation is needed.

It can be misleading.

Ask good statistical questions.







*Knowing your  $q(0.99)$  is “too high” is one thing...*

# Work backwards

Work backwards.

At what quantile are you?





mvalue: <http://www.brendangregg.com/FrequencyTrails/modes.html>

# Understand Workloads





*man(1) is a tool's tool.*

---

# Tools

---

Tools do not a master craftsman make.

Regardless, know your damn tools.

There are three types of tools.



Photograph courtesy of James Bowe  
<https://www.flickr.com/photos/jamesbowe/7164489201>







*Tool type #1*

# Observation

Taking measurements.

Inspecting state.

Inspecting conversation.







*Tool type #2*

# Synthesis

Synthesizes something to  
enable the use of tool type #1







*Tool type #3*

# Manipulation

Changing state.

Used for testing hypotheses.





*Favorite tools*

---

# Martial Arts

---

#1

- DTrace
- truss/ktrace/strace
- tcpdump/snoop
- mdb/gdb/dbx/lldb
- sar/mpstat/iostat/vmstat

#2

- curl

#3

- vi/echo
- sysctl/mdb(-w)
- DTrace(-w)



Photograph courtesy of Republic of Korea  
<https://www.flickr.com/photos/koreanet/6099430458>







*Lorem Ipsum Dolor Indeed*

---

# Anecdotes

This one time at band camp





*Latency*

---

I'm huge in Japan

---

Latency for a hot landing page jumps from around 300ms to around 450ms.

No changes in latency to other regions.





*Latency*

---

# Scrub in or go home

---

Latency for disk writes radically change behavior.

It's as if we have a new workload.

We do not have a new workload.

... we do have a new workload.



Photograph courtesy of Phalinn Ooi  
<https://www.flickr.com/photos/umjanedoan/497411169>





*Latent effect*

---

# Hitting the wall

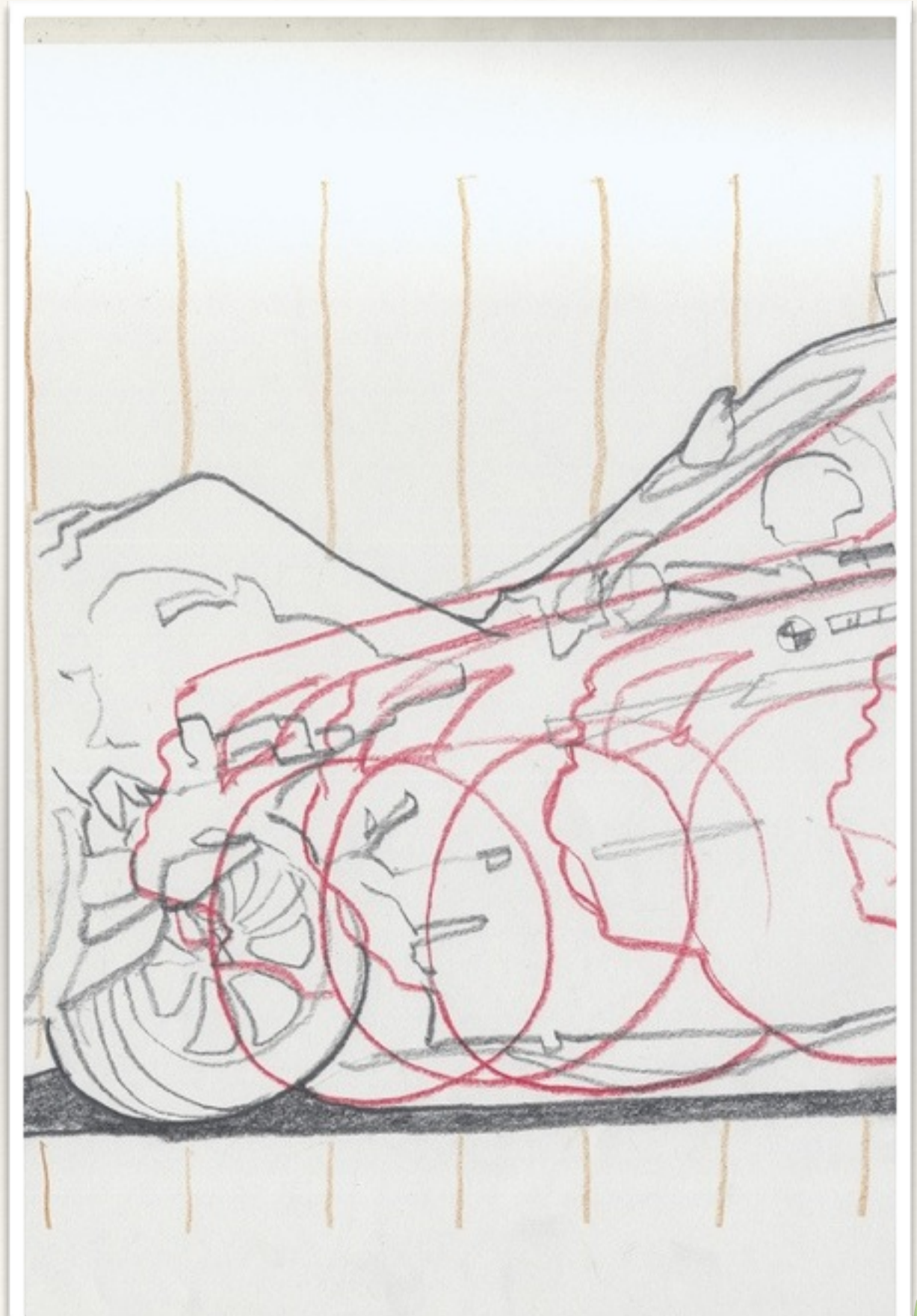
---

Disk I/O latency goes to hell at 3pm.

Turns out disk throughput is plateaued.

No change in configuration near 3pm.

Oops, I tripped at 10am.



# Thank You

