Capacity Planning

for Web Operations

John Allspaw
Operations Engineering

flickr™

Are you tracking how your servers are performing?

?

Do you know how many servers do you have?

Do you know how much traffic can your servers handle?
(without dying)

Are you tracking how your application is being used ?

monitoring

testing

deployment

forecasting

architecture

metrics

product planning

capex

procurement

monitoring

testing

deployment

go see Adam Jacob's talk!

forecasting

architecture

metrics

product planning

capex

procurement

traditional capacity planning


capacity planning for web

Why capacity planning is important

Hardware* costs $$

(Cloudware costs $$, too)

Having too little is bad (!@#!!) -> ($$$)

Having too much is bad ($$$$!)

* and network, datacenter space, power, etc.

Growth

"Normal"

projected
planned
expected
hoped for

(yay!)

"Instantaneous"

spikes
unexpected
external events
digg, etc.

(yay?)

(omg! wtf!)

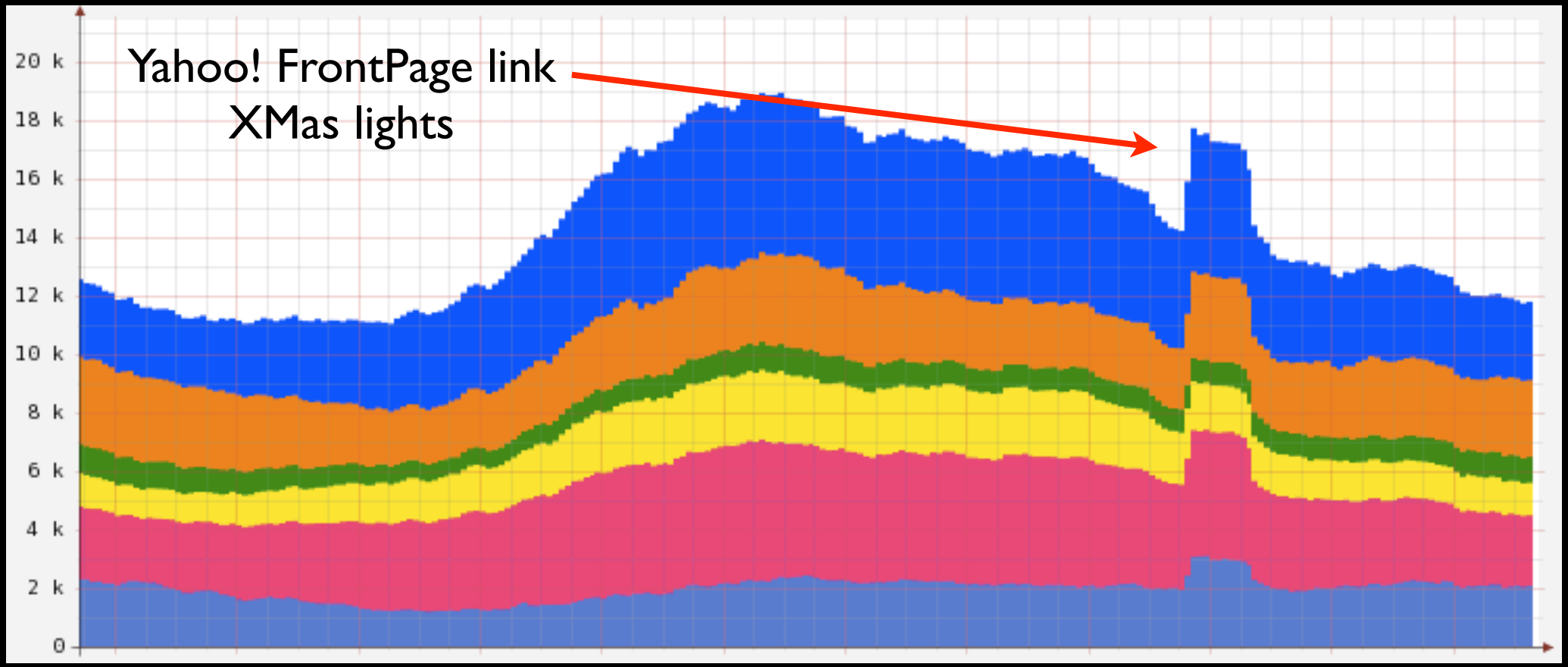
“Normal” growth at Flickr

in a year....

4x increase in photo requests/sec

2.2x increase in uploads/day

3x increase in database queries/sec



“Instantaneous”

“Instantaneous” coping

- Disabling “heavier” features on the site

- Cache aggressively or serve stale data

- Bake dynamic pages into static ones

capacity != performance

Making something fast doesn't necessarily make it *last*

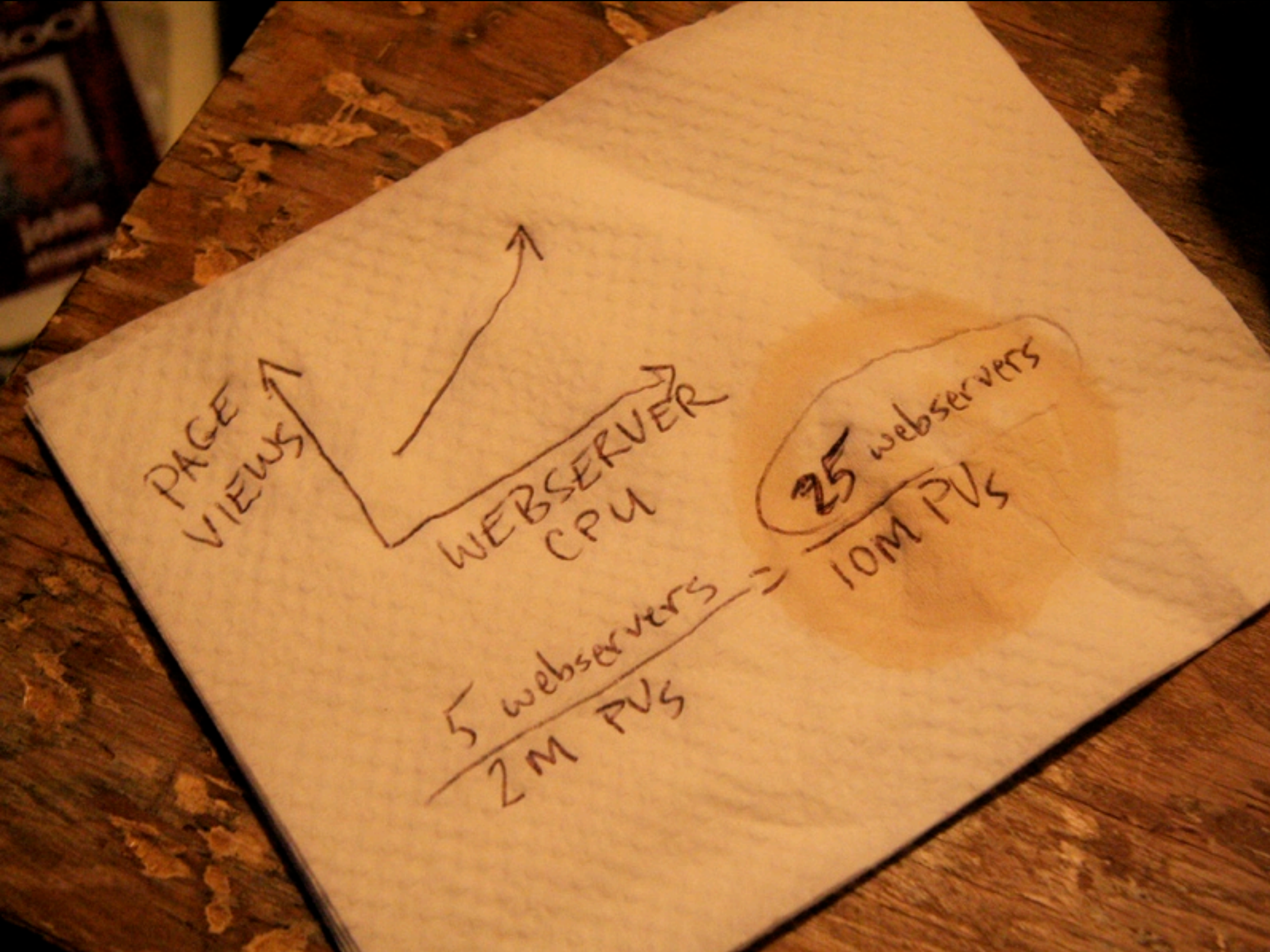
Performance tuning = good, just don't count on it

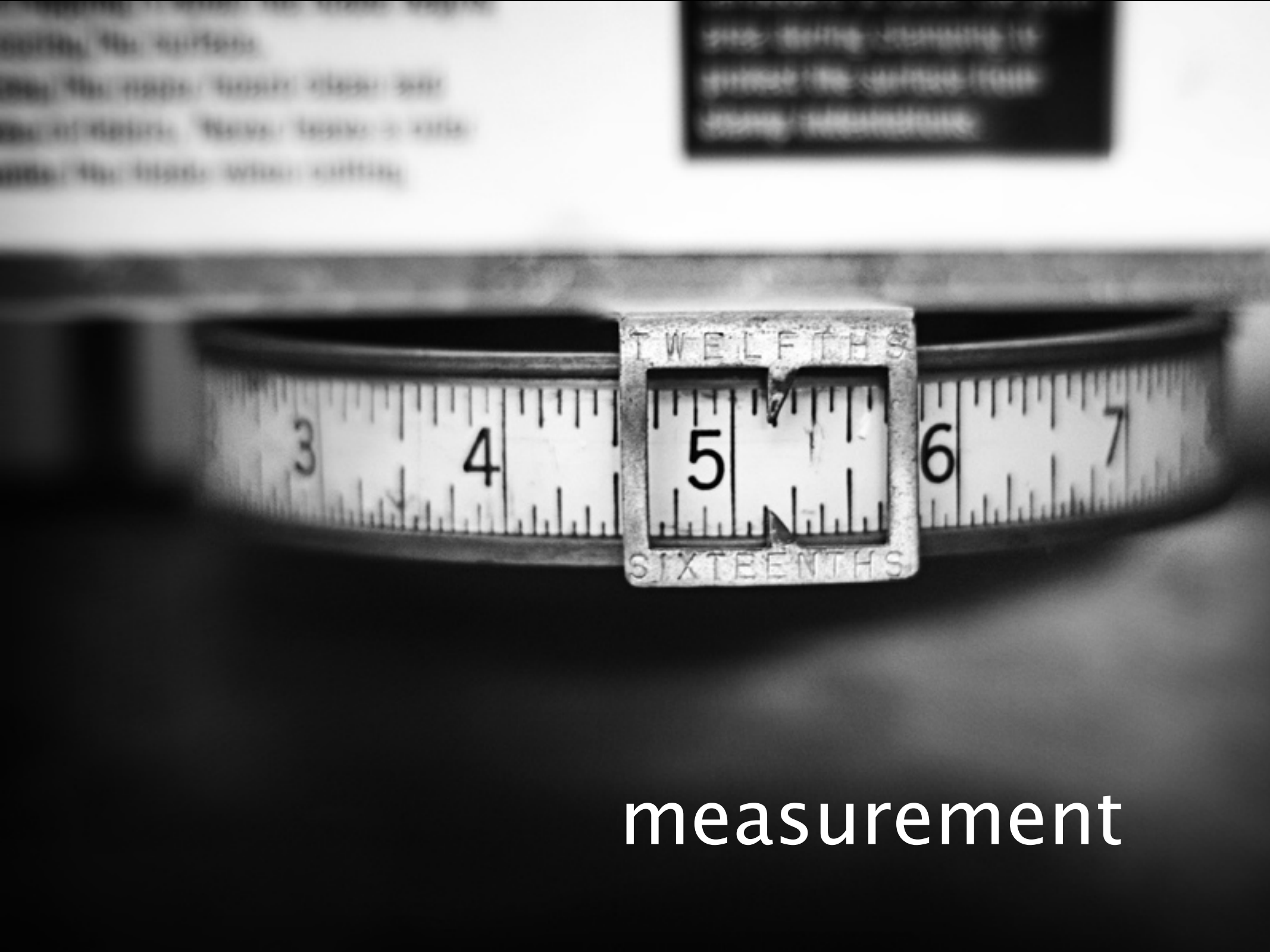
Accept (for now) the performance you have, not the performance you *wished* you had, or you *think* you might have later


Stewart:
"Allspaw!!!! OMG!!!"
How many servers will we need next year?!
(we need to tell finance by 2pm today)

“Ah, just buy twice as much as we need”

2 x (how much we need) = ?


PAGE VIEWS
WEBSERVER CPU
5 webservers
2M PVs
=
25 webservers
10M PVs



measurement

Good capacity measurement tools can...

Easily measure and record **any** number that changes over time

Easily compare metrics to **any** other metrics from anywhere else

(import/export)

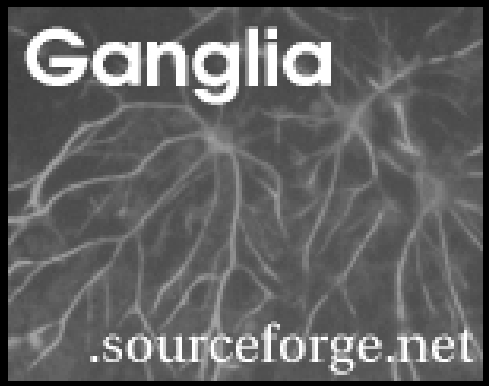
Easily make graphs

good tools are out there

cacti.net

munin.projects.linpro.no

hyperic.com



ganglia.info

good tools are out there

cacti.net

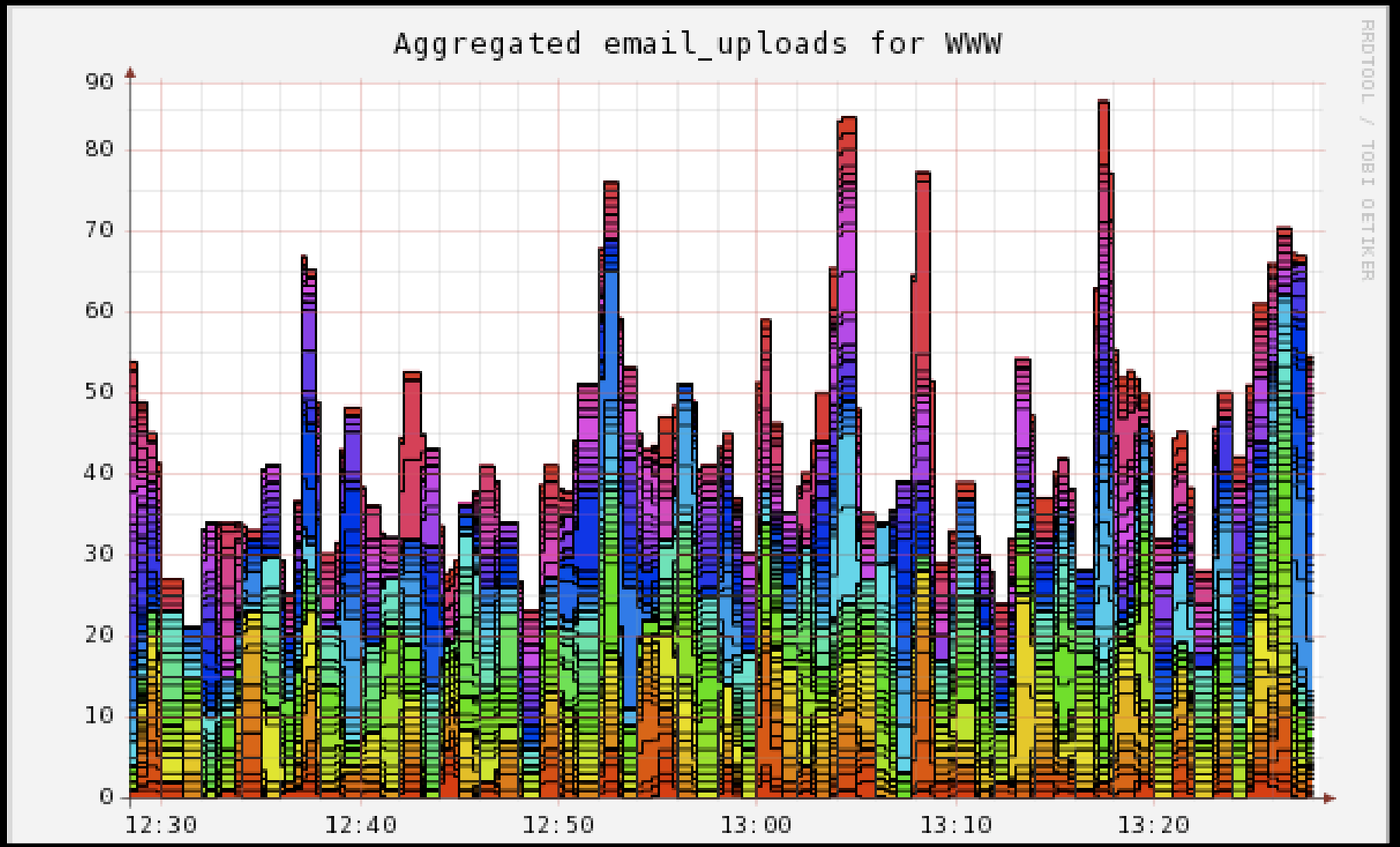
munin.projects.linpro.no

hyperic.com

Flickr uses

ganglia.info

photo uploads via email per minute



hour

application metrics

your stuff, not just system stuff

photos uploaded (and processed) per minute

average photo processing time per minute

average photo size

disk space consumed per day

user registrations per day

etc etc etc

your stuff, not just system stuff

photos uploaded (and processed) per minute

average photo processing time per minute

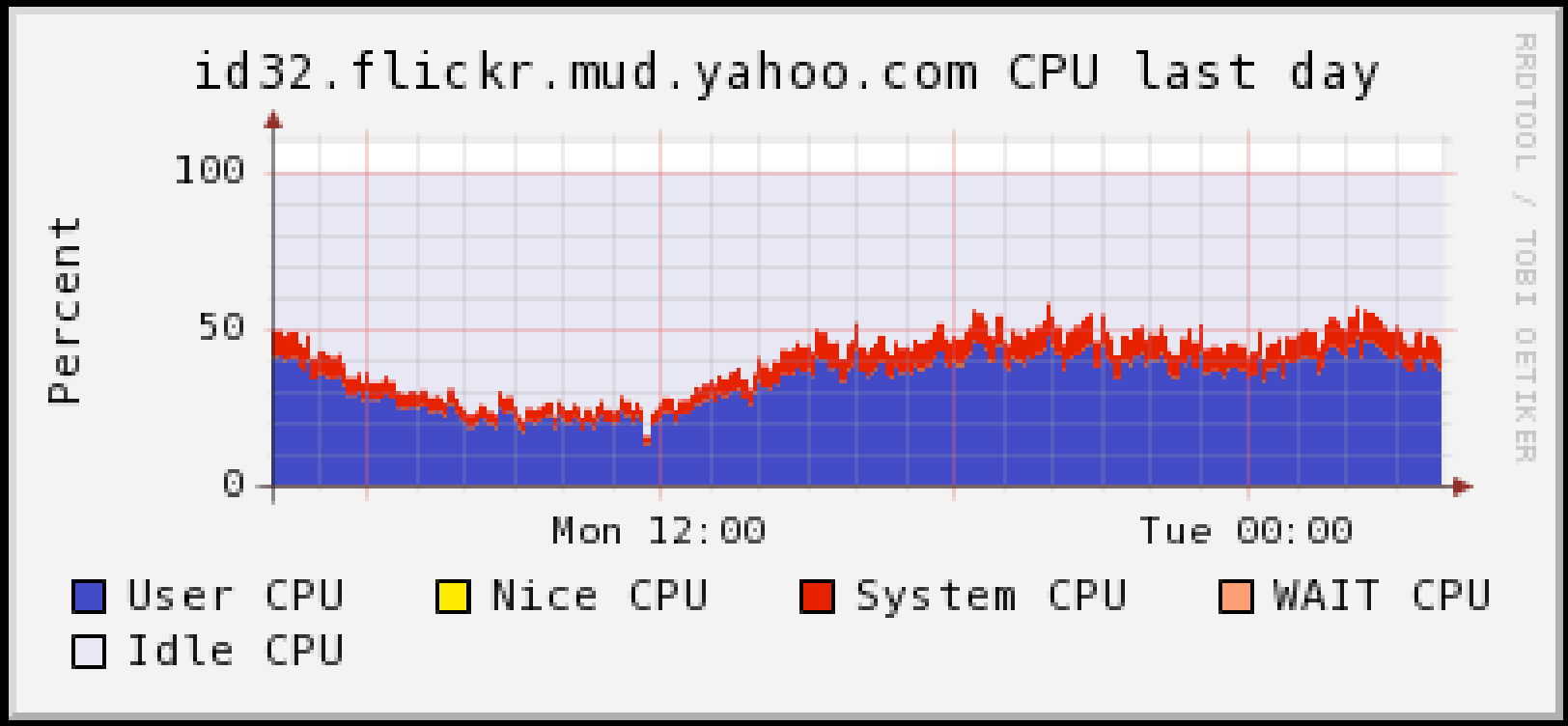
average photo size

disk space consumed per day

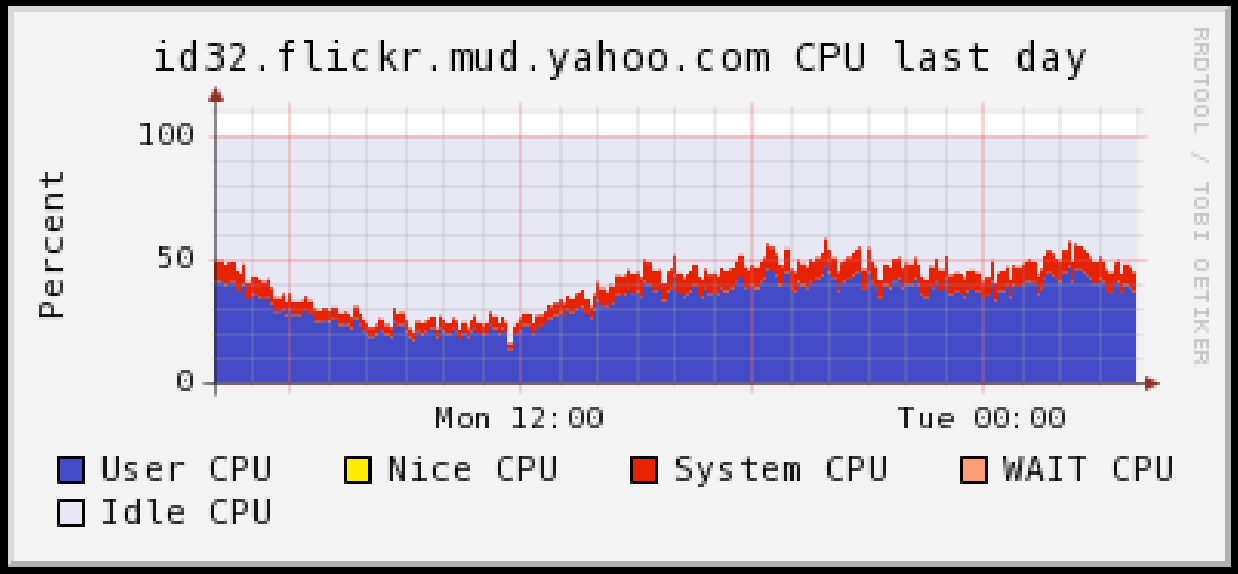
user registrations per day

etc etc etc

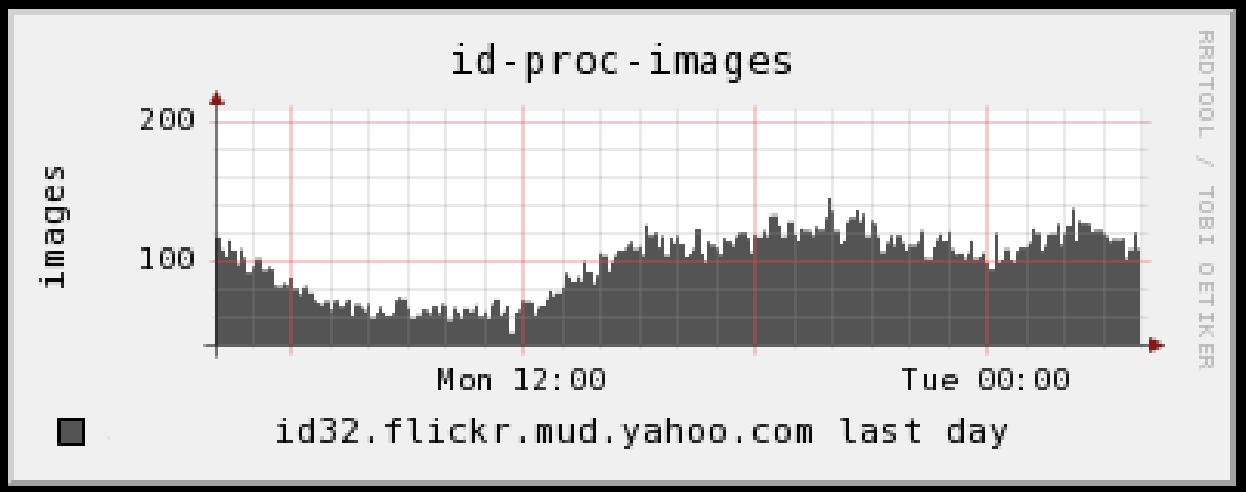
Tie application metrics to system metrics



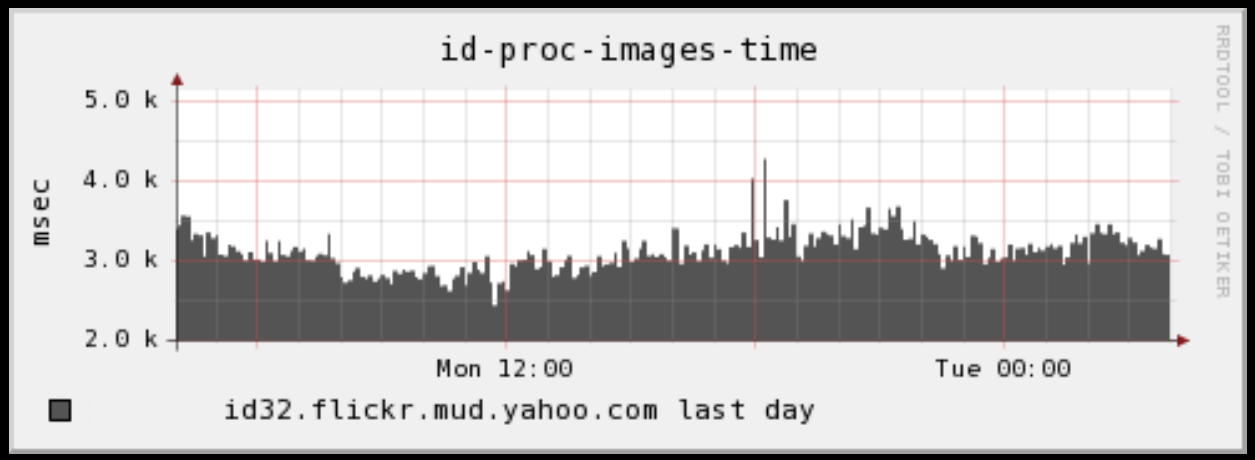
Pretty!! But what does this mean?



It means that with about 60% total CPU...



It means we can process ~120 images per minute



...and we can process them in ~3.5 seconds (on average)

Benchmarking

Great for comparing hardware platforms and configurations

BUT

not ***exactly*** like a bike messenger

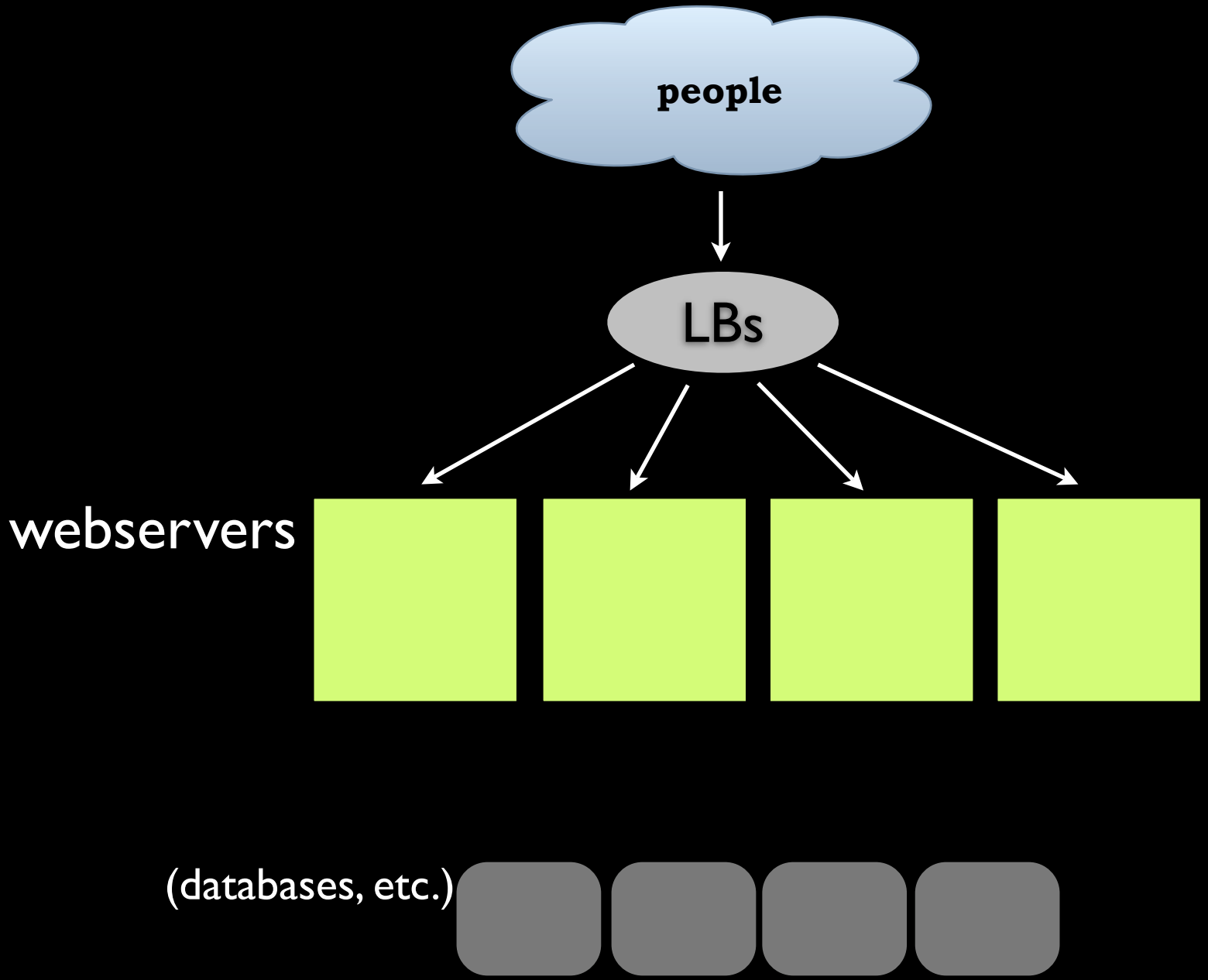
Finding your ceilings

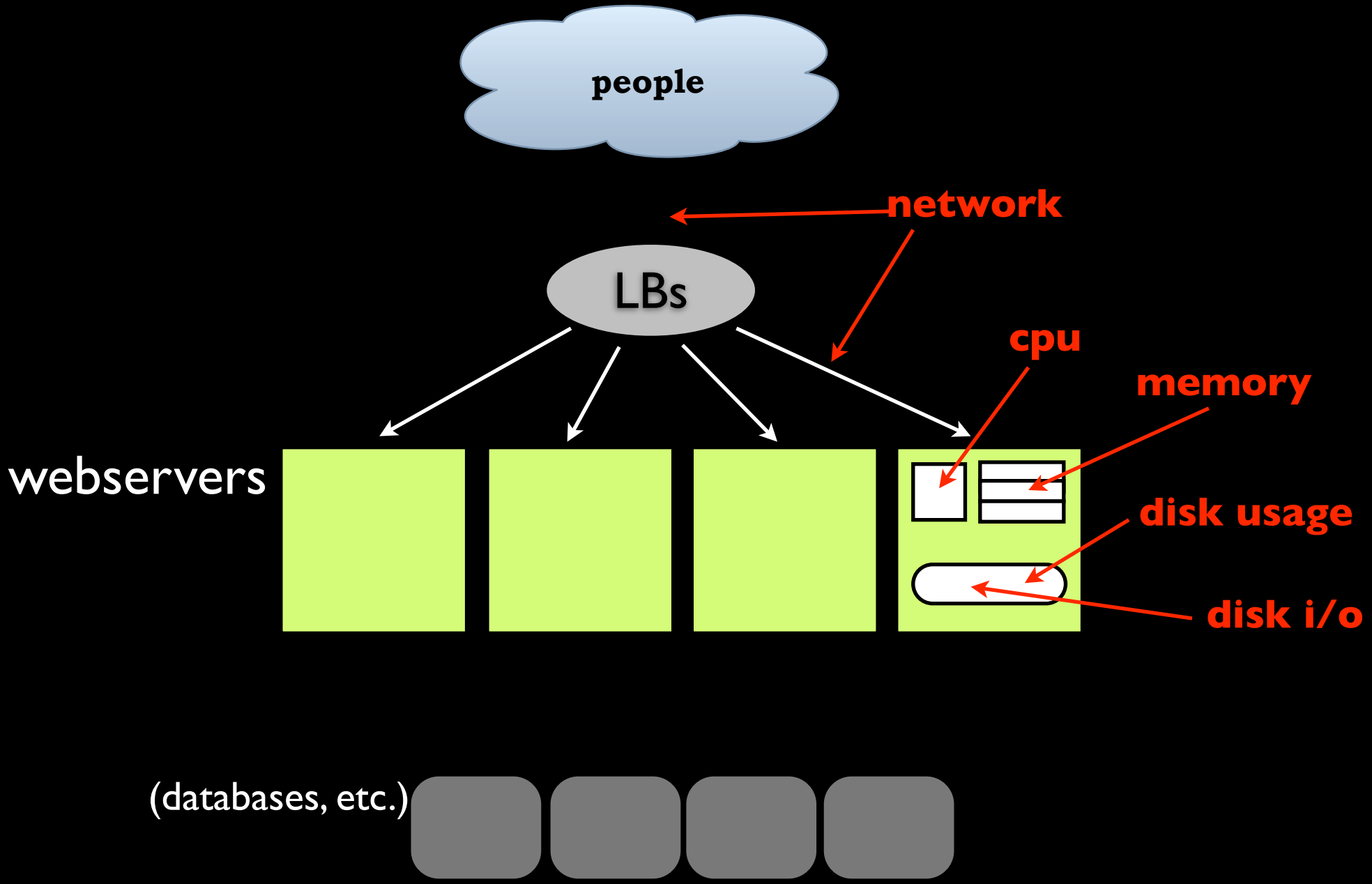
- Use real data, from production servers (if at all possible)
- No, really

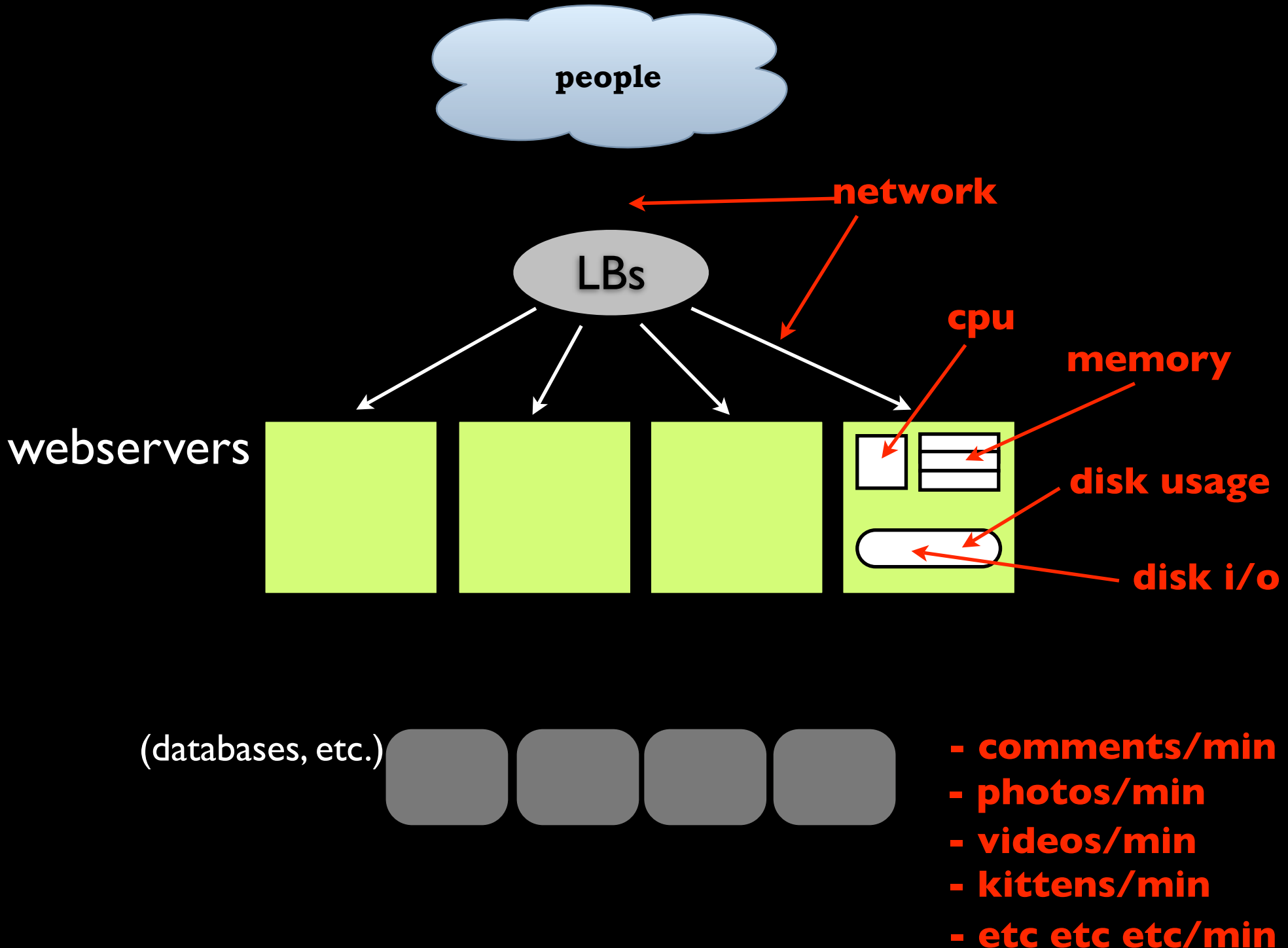
How much traffic can each webserver take before it dies?

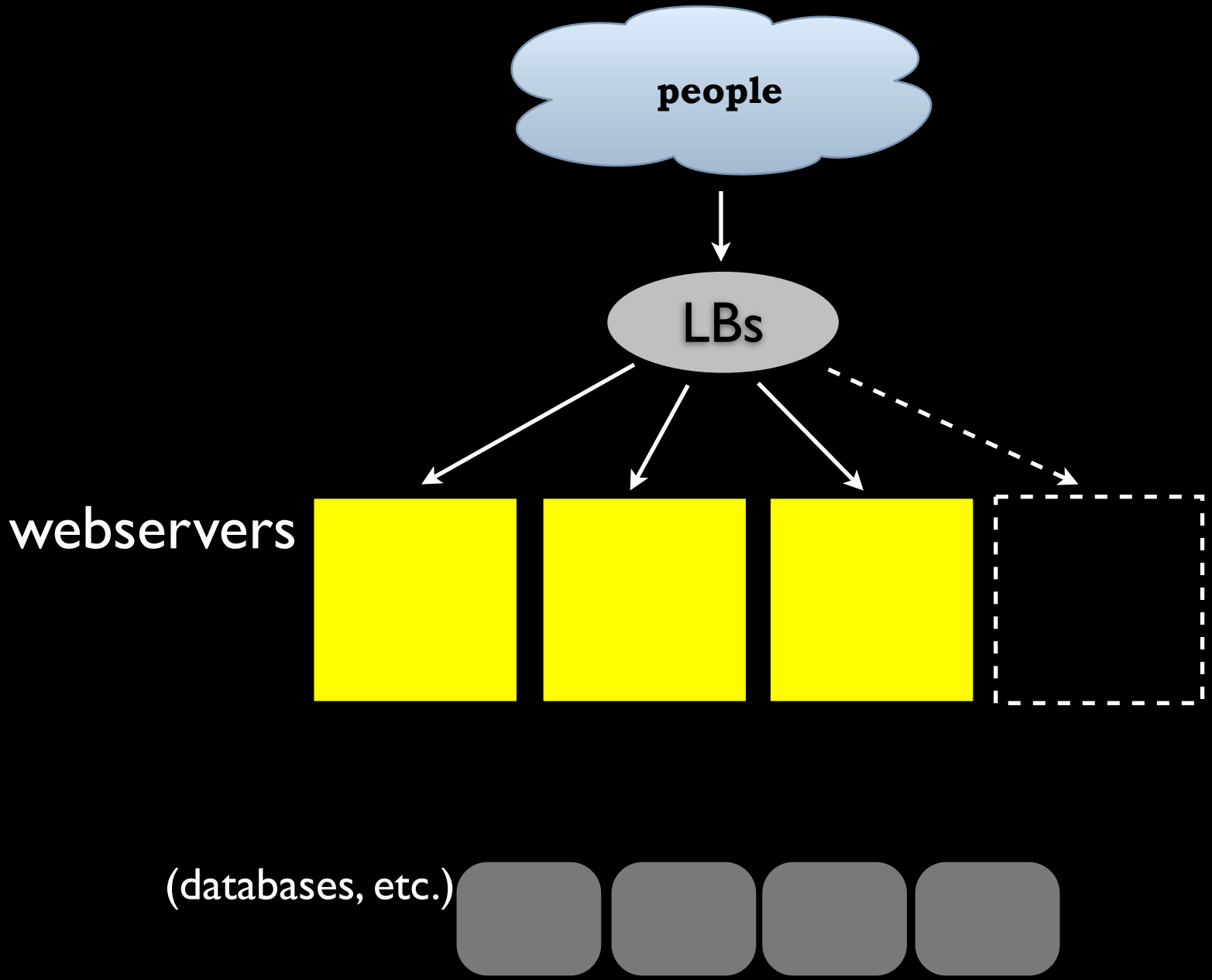
How many webservers can fail before we're screwed?

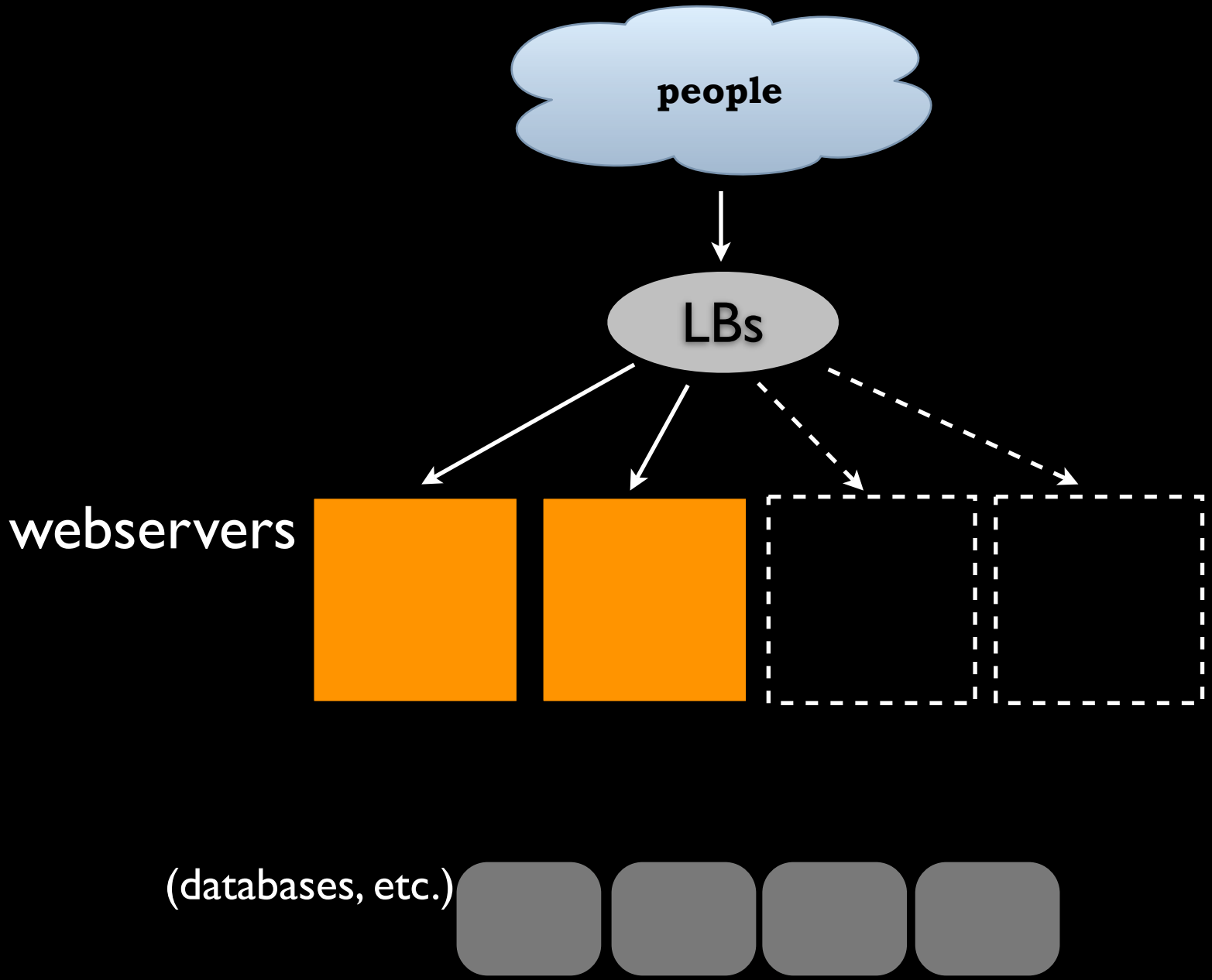
When should I add more webservers?

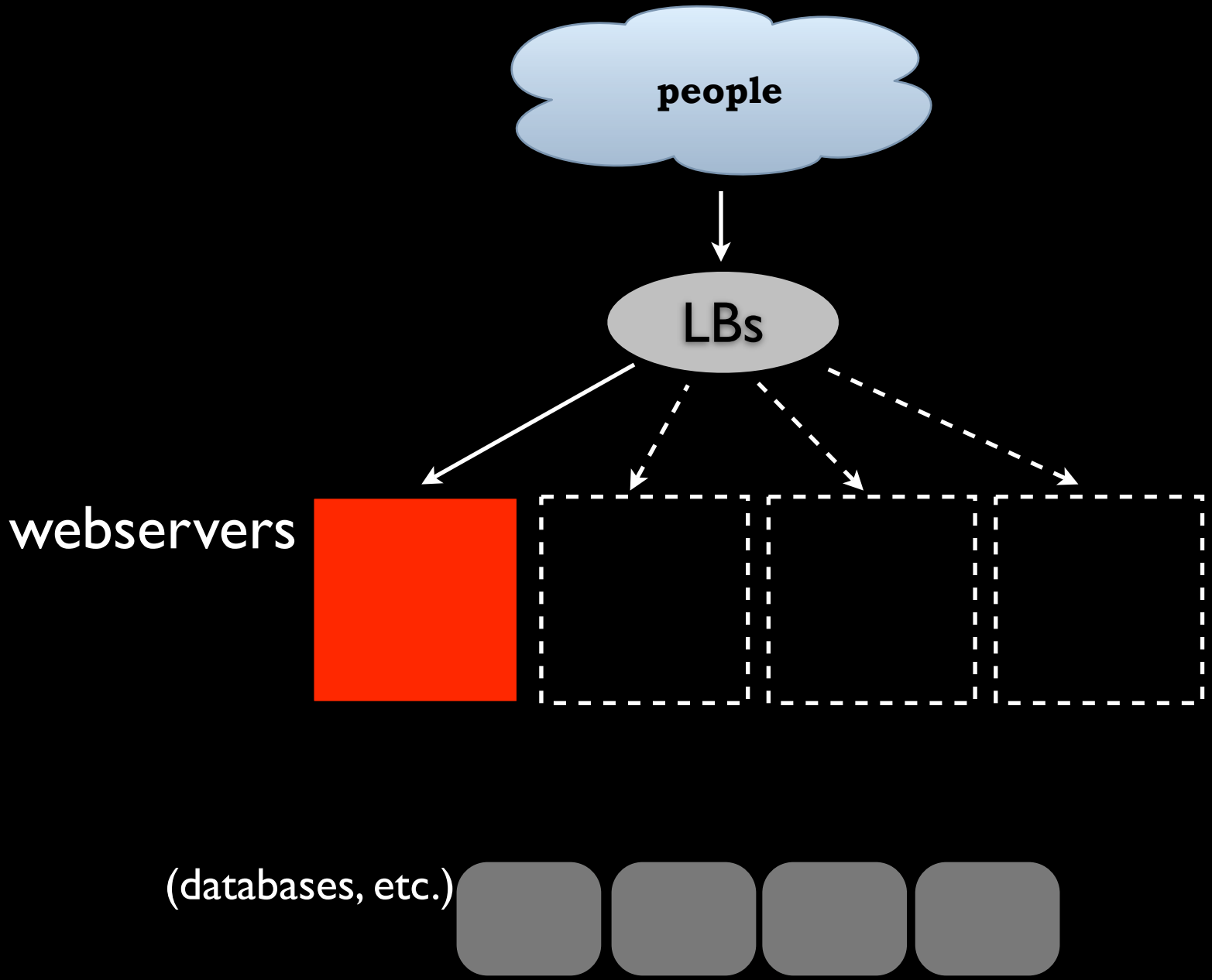

people
LBs
webservers
(databases, etc.)

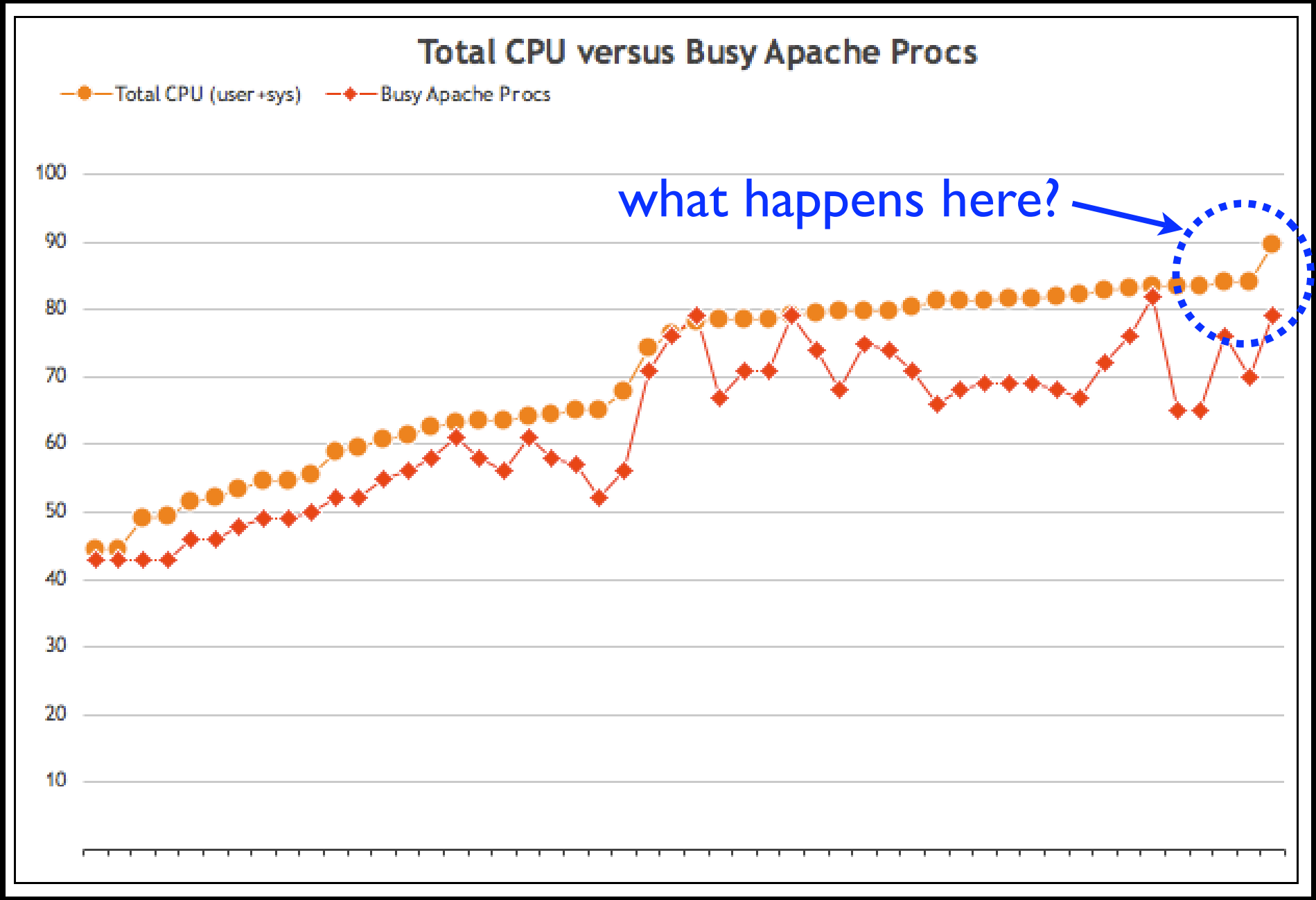

people
network
LBs
cpu
memory
webservers
disk usage
disk i/o
(databases, etc.)


people
network
LBs
cpu
memory
webservers
disk usage
disk i/o
(databases, etc.)
- comments/min
- photos/min
- videos/min
- kittens/min
- etc etc etc/min


people
LBs
webservers
(databases, etc.)

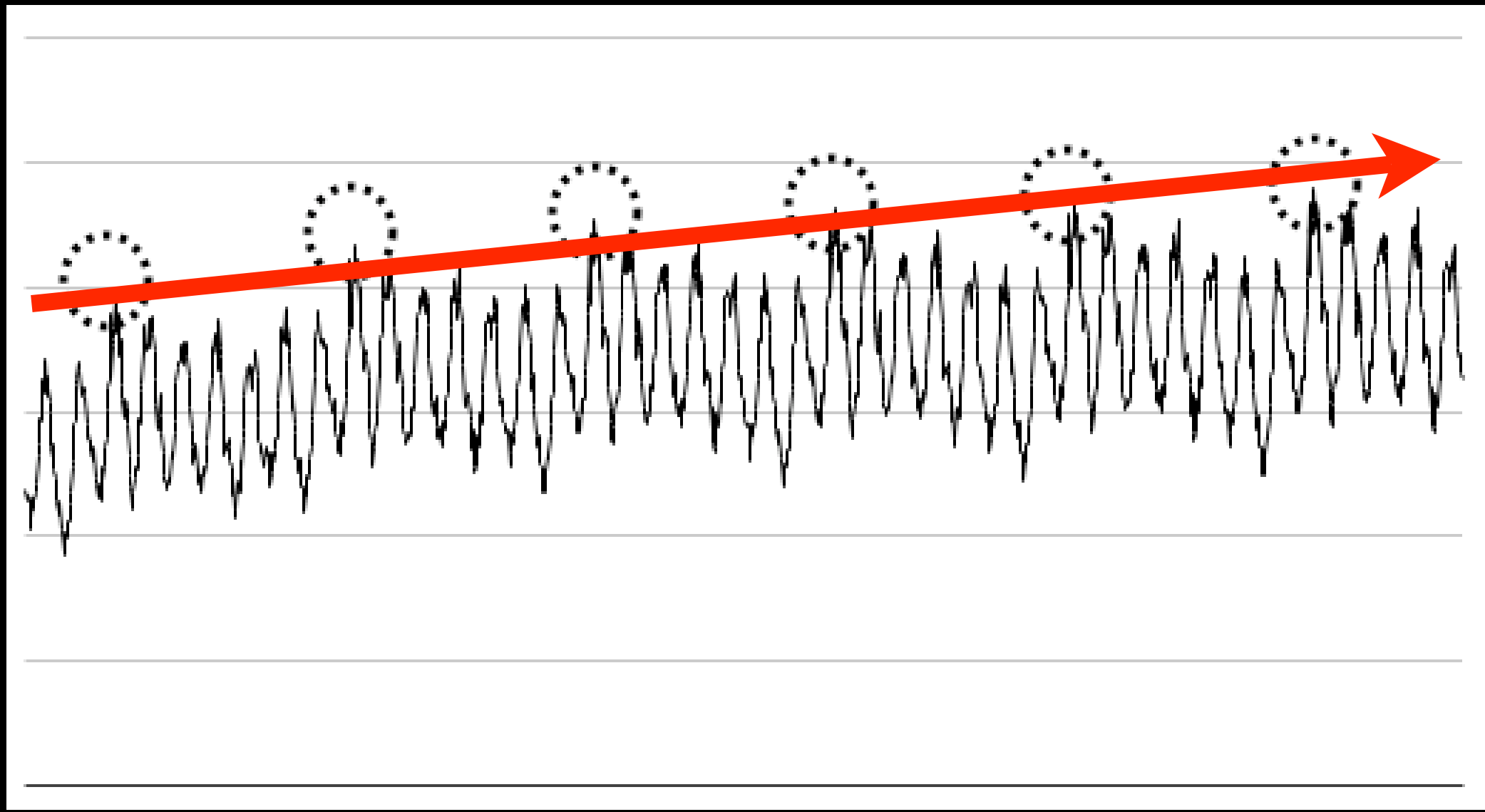

people
LBs
webservers
(databases, etc.)


people
LBs
webservers
(databases, etc.)



Ceiling = upper limit of "work" (and resources)

Trends of peaks



Time

Benchmarking

Might be your only option if you have a single server.

some good benchmarking tools:

Siege
http://www.joedog.org/JoeDog/Siege

httperf/autobench
http://www.hpl.hp.com/research/linux/httperf/
http://www.xenoclast.org/autobench

sysbench
http://sysbench.sf.net

Economics

Time makes everything cheaper

(the Moore's Law thing)

BUT

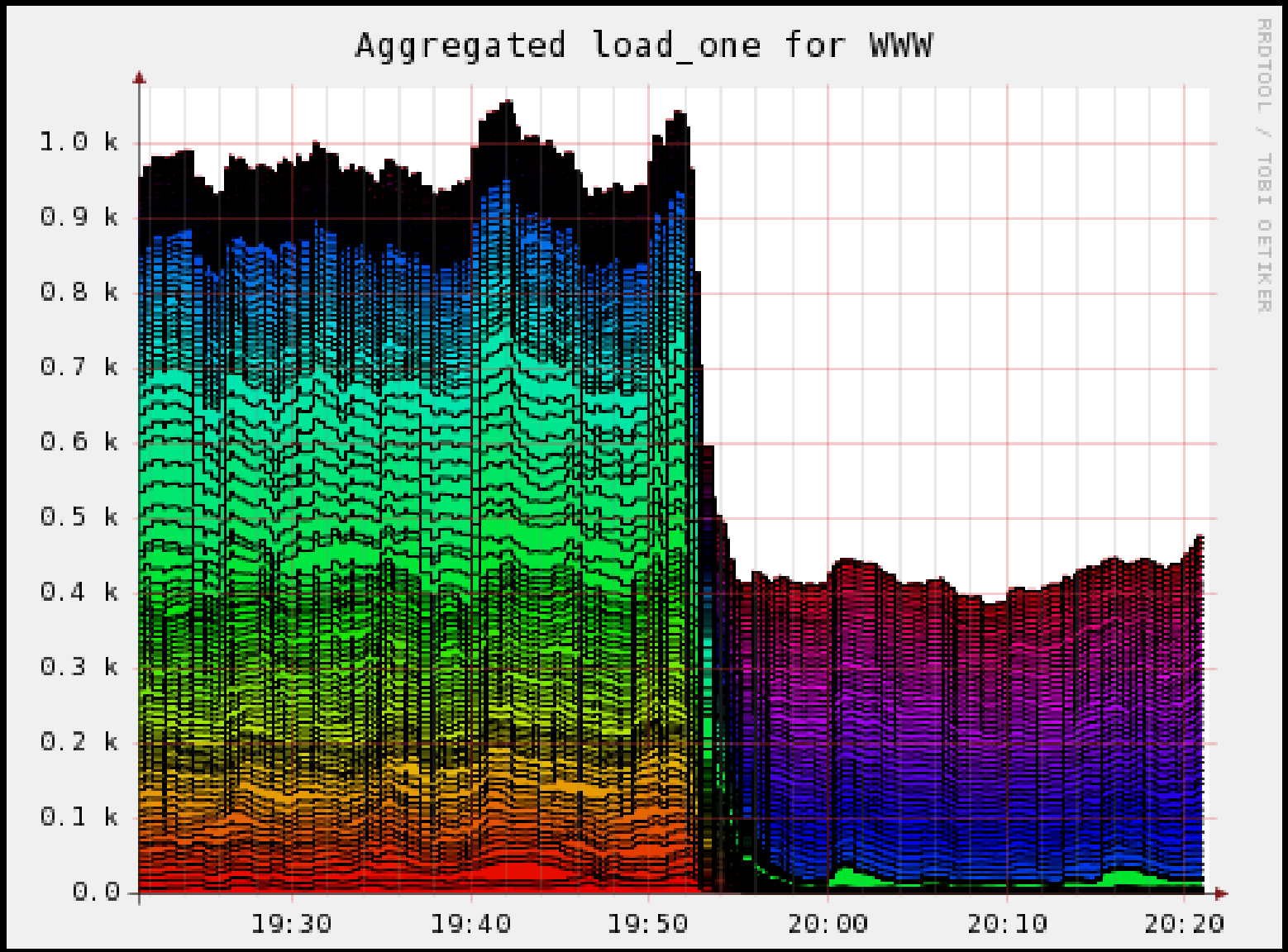
you don't have a lot of time to
wait around, do you?

Vertical scaling

Horizontal architectures

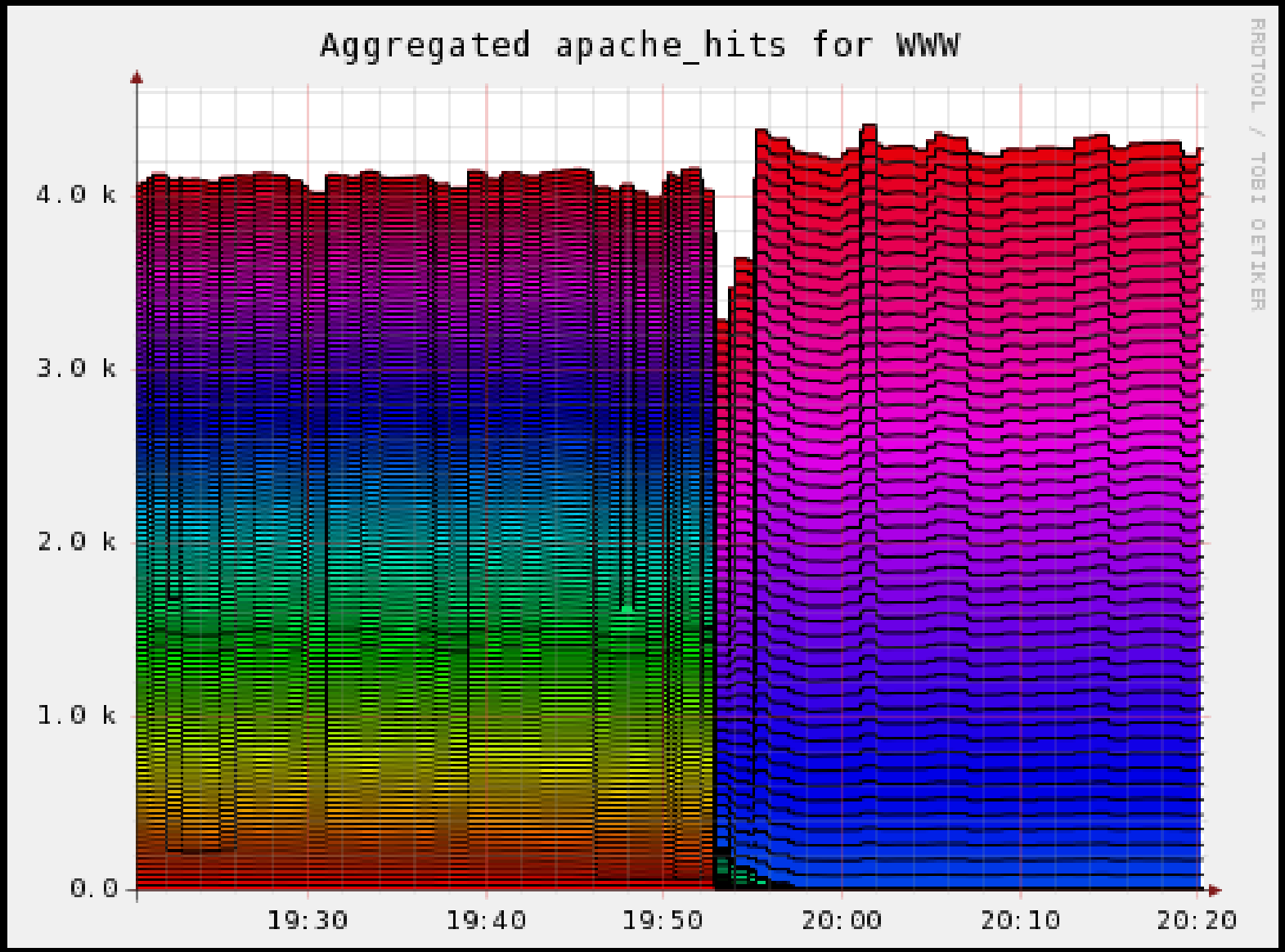

Diagonal scaling

Diagonal scaling



Replacing 67 dual-core webservers with 18 dual quads

Diagonal scaling



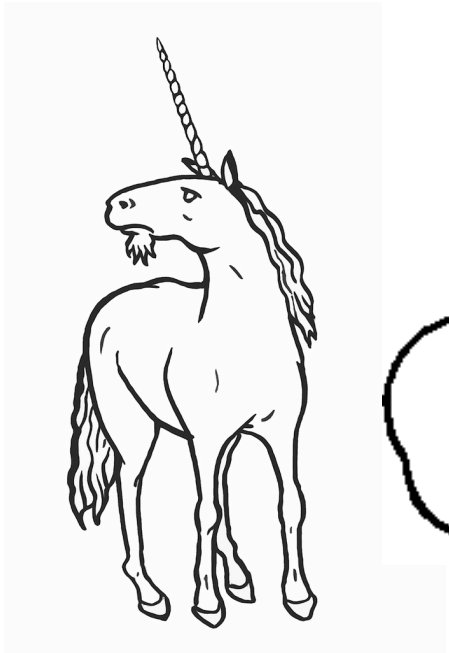
more traffic from less machines

Diagonal Scaling

servers	CPUs per server	RAM per server	drives per server	total power (W) @60% peak
67	2	4GB	1x80GB	8763.6
18	8	4GB	1x146GB	2332.8

~70% less power
49U less rack space

Utility Computing



Disclosure: We don't use clouds at Flickr.

(but we know folks who do)

Help with deployment timelines

Help with procurement timelines

BUT

Still have to pay attention

Many people use the same forecasting methods

Use Common Sense(tm)

Pay attention to the right metrics

Don’t pretend to know the exact future

Measure constantly, adapt constantly

Complex simulation and modeling is rarely worth it

Don’t expect tuning and tweaking will ever win you any excess capacity

Some more stats

Serving 32,000 photos per second at peak

Consuming 6-8TB per day

Consumed >34TB per day during Y!Photos migration

~3M uploads per day, 60 per second at peak

June 23-24, 2008

20% off discount: “vel08js”

We Are Hiring!

(DBA, engineers)

http://flickr.com/photos/85013738@N00/542591121/
http://flickr.com/photos/formatc1/2301500208/
http://flickr.com/photos/mikefats/11546240/
http://flickr.com/photos/kanaka/491064256/
http://flickr.com/photos/randysonofrobert/1035003071/
http://flickr.com/photos/halcyonsnow/446166047/
http://flickr.com/photos/wwworks/2313927146/
http://flickr.com/photos/sunxez/1392677065/
http://flickr.com/photos/spacesuitcatalyst/536389937/
http://flickr.com/photos/theklan/1276710183/