Behind the scenes - time series data

David Mytton

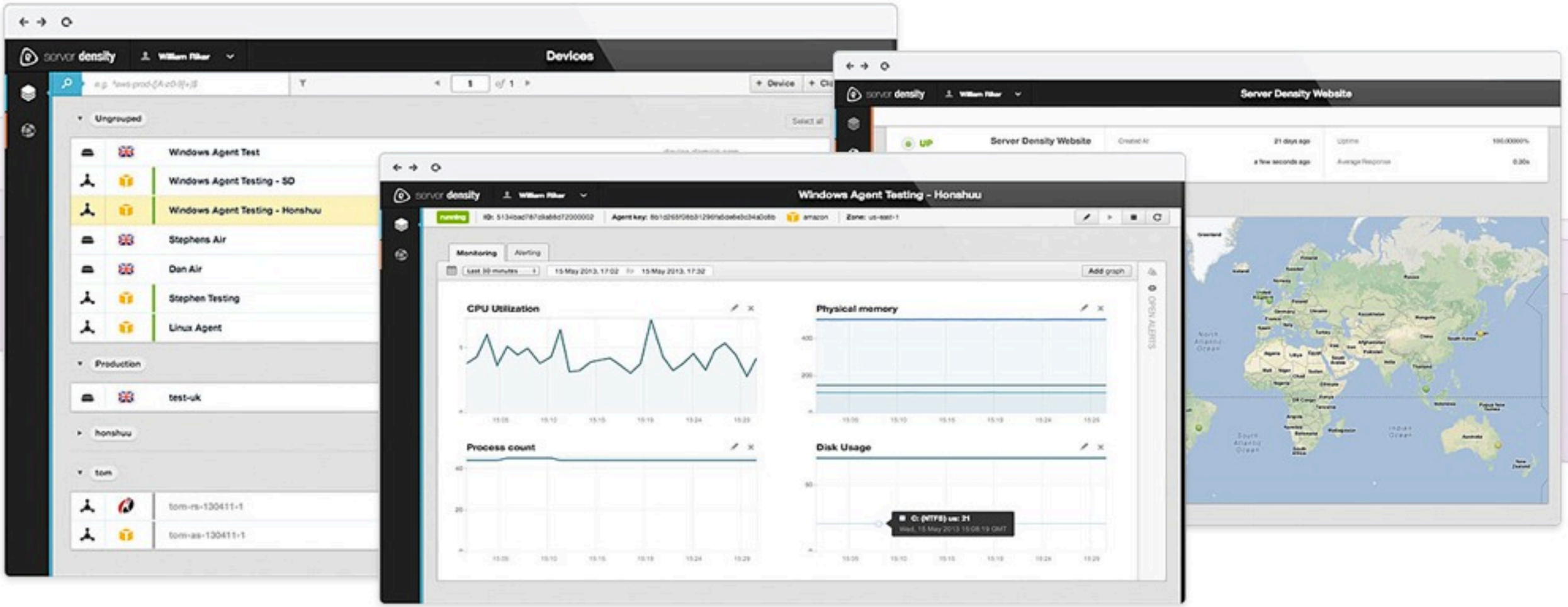
Woop Japan!

server **density**

cloud **management**

server **monitoring**

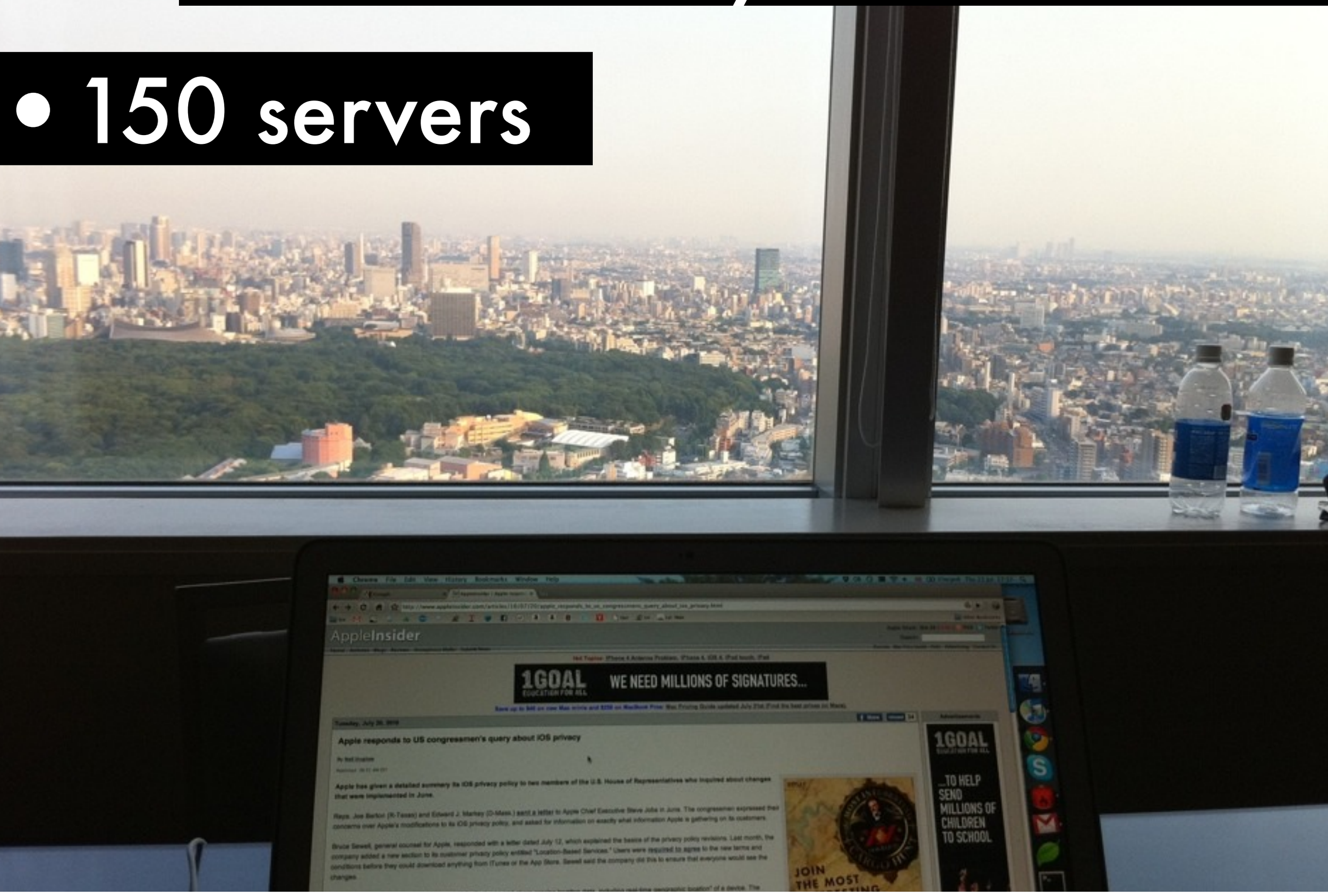
website **monitoring**



Server Density Infrastructure

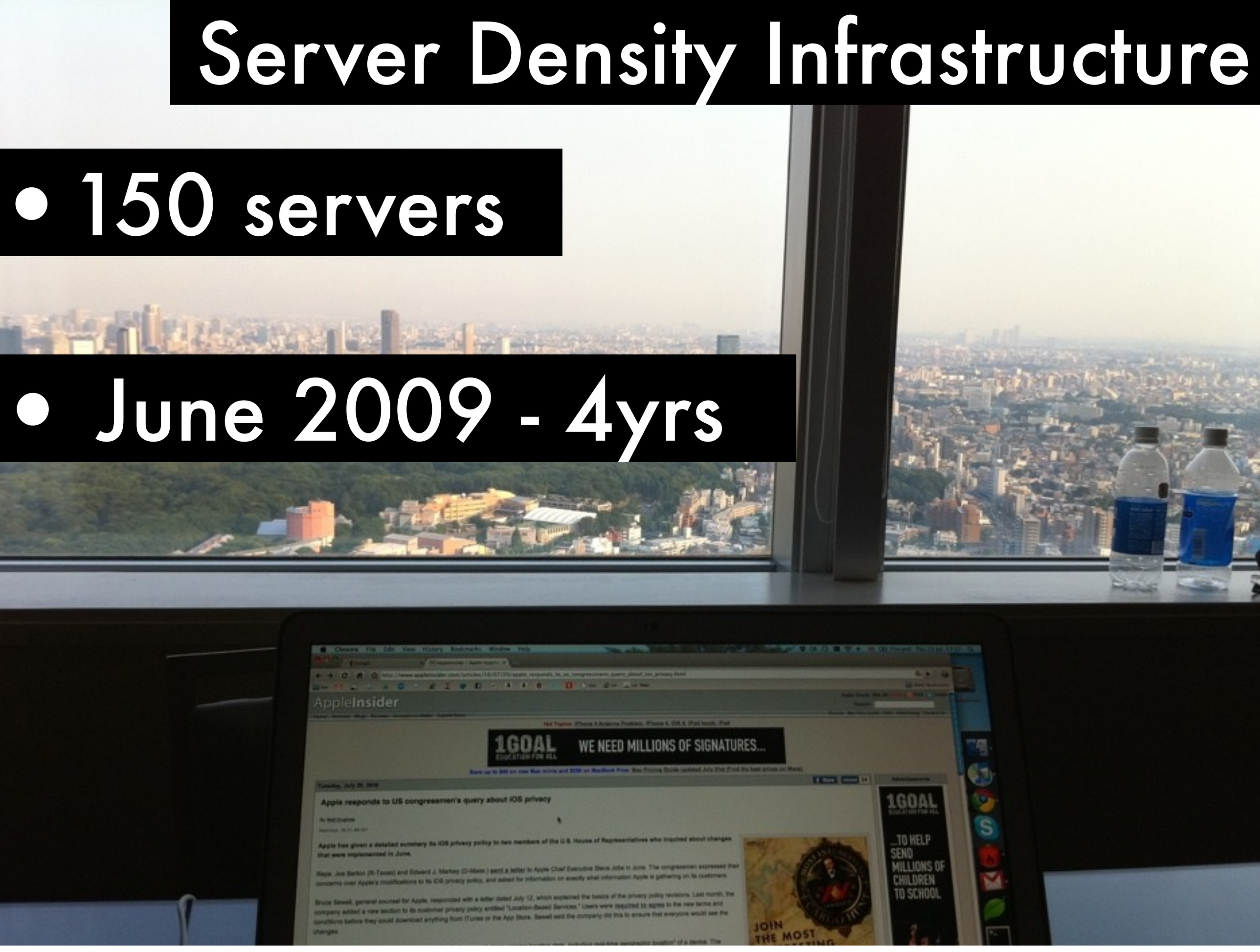
Server Density Infrastructure

- 150 servers



Server Density Infrastructure

- 150 servers
- June 2009 - 4yrs



Server Density Infrastructure

- 150 servers
- June 2009 - 4yrs
- MySQL -> MongoDB

Server Density Infrastructure

- 150 servers
- June 2009 - 4yrs
- MySQL -> MongoDB
- 25TB data per month

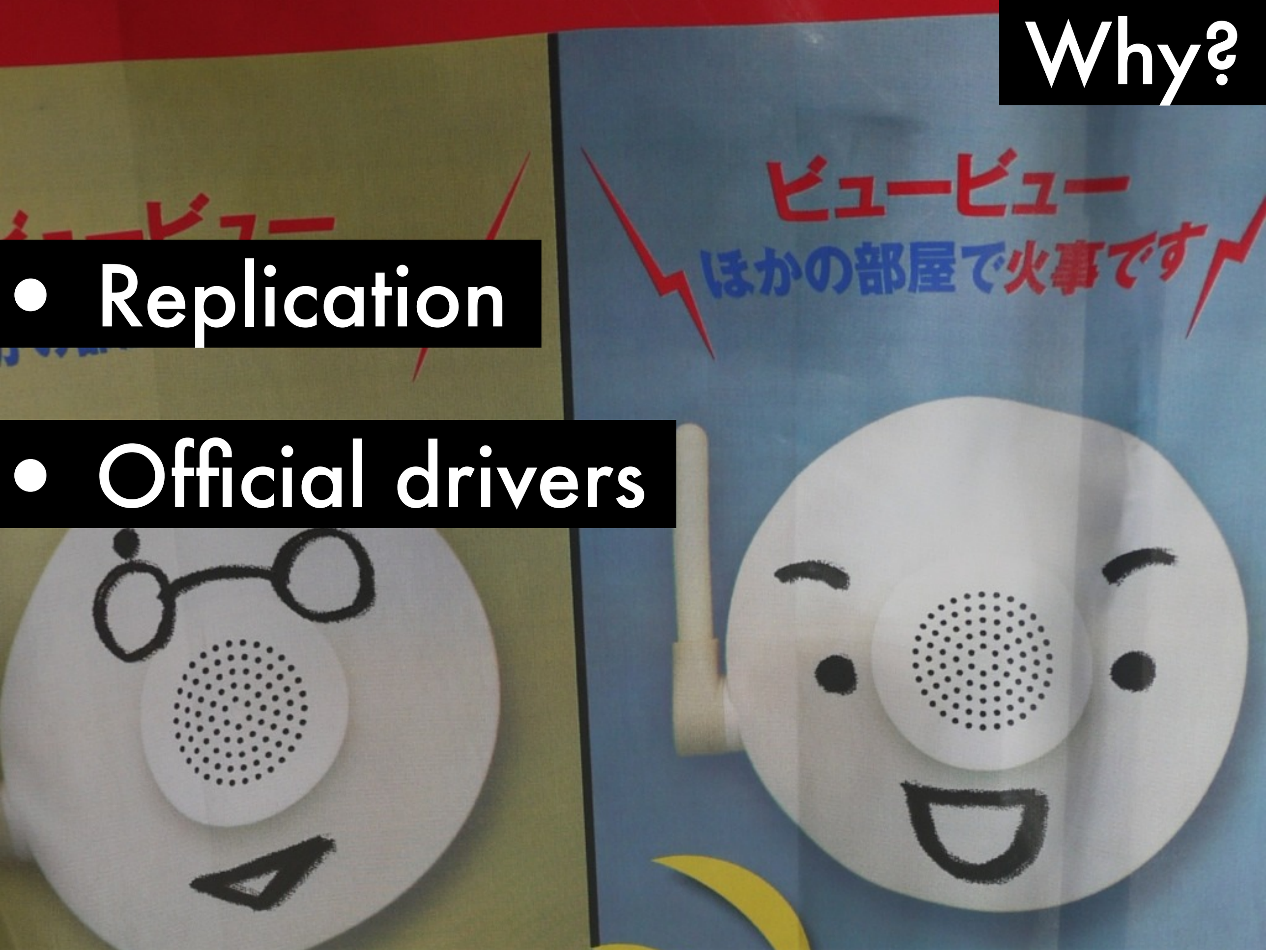
Why?

Why?

- Replication

Why?

- Replication
- Official drivers



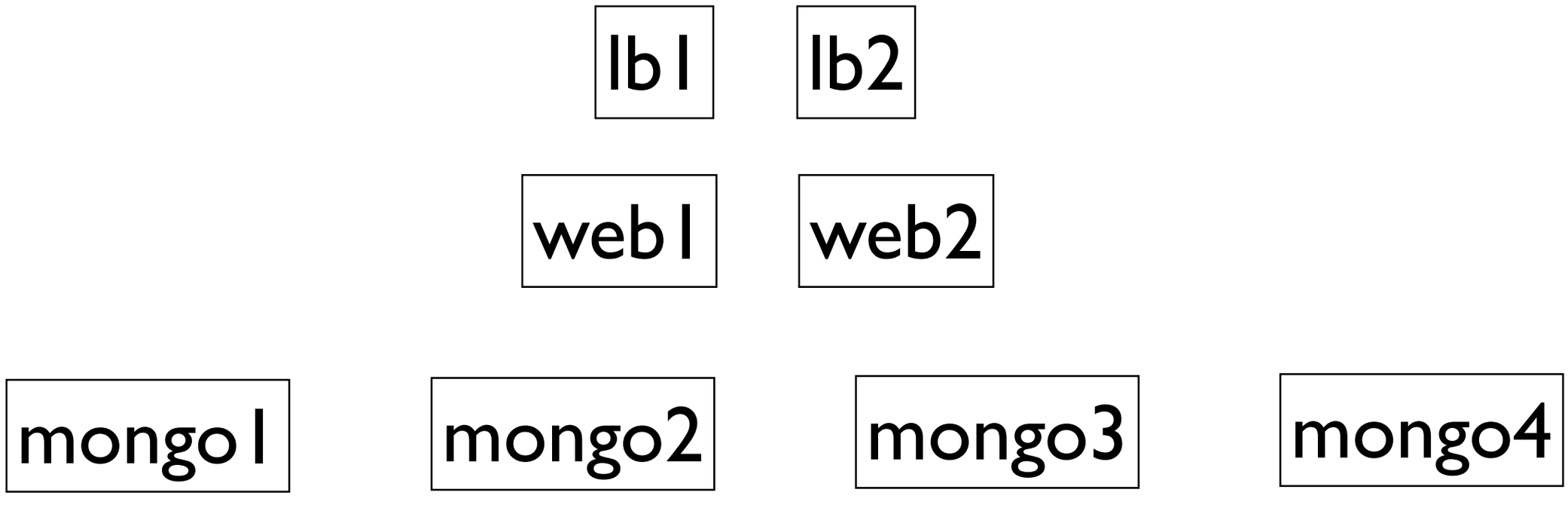
Why?

- Replication
- Official drivers
- Easy deployment

Why?

- Replication
- Official drivers
- Easy deployment
- Fast out of the box

Architecture



Architecture

Global IP

lb1 lb2

web1 web2

mongo1a

mongo2a

mongo3a

mongo4a

lb3 lb4

web3 web4

mongo1b

mongo2b

mongo3b

mongo4b

Performance

- Fast network

Picture is unrelated! Mmm, ice cream.

Performance

- Fast network

EC2 10 Gigabit Ethernet

- Cluster Compute
- High Memory Cluster
- Cluster GPU
- High I/O
- High Storage

Single Public and Private Ports (2 ports total)	
10 Mbps Ports	$0.00
100 Mbps Ports	$0.00
1 Gbps Ports	$20.00
10Gbps Ports*	$100.00

* Where available/Must meet hardware requirements

Dual Public and Private Ports (4 ports total)	
10 Mbps Dual Ports (20 Mbps max. throughput)*	$10.00
100 Mbps Dual Ports (200 Mbps max. throughput)*	$20.00
1 Gbps Dual Ports (2 Gbps max. throughput)*	$40.00

* Where available/Must meet hardware requirements

- Network cards
- VLAN separation

Performance

- Fast network

Workload: Read/Write?

What is being stored?

Result set size

- Read / write: adds to replication oplog
- Images? Web pages? Tiny documents?
- What is being returned? Optimised to return certain fields?

Performance

- Fast network

Inter-DC LAN

```
15/05/13 07:42:23 david@mtx2-md1a ~: ping mtx2-md2a
PING mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87) 56(84) bytes of data.
64 bytes from mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87): icmp_req=1 ttl=64 time=0.472 ms
64 bytes from mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87): icmp_req=2 ttl=64 time=0.570 ms
64 bytes from mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87): icmp_req=3 ttl=64 time=0.425 ms
64 bytes from mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87): icmp_req=4 ttl=64 time=0.398 ms
64 bytes from mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87): icmp_req=5 ttl=64 time=0.623 ms
64 bytes from mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87): icmp_req=6 ttl=64 time=0.444 ms
64 bytes from mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87): icmp_req=7 ttl=64 time=0.297 ms
^C
--- mtx2-md2a.wdc.sl.serverdensity.net ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 5997ms
rtt min/avg/max/mdev = 0.297/0.461/0.623/0.101 ms
```

- Latency

Performance

- Fast network

Inter-DC LAN

```
15/05/13 07:42:23 david@mtx2-md1a ~: ping mtx2-md2a
PING mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87) 56(84) bytes of data.
64 bytes from mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87): icmp_req=1 ttl=64 time=0.472 ms
64 bytes from mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87): icmp_req=2 ttl=64 time=0.570 ms
64 bytes from mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87): icmp_req=3 ttl=64 time=0.425 ms
64 bytes from mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87): icmp_req=4 ttl=64 time=0.398 ms
64 bytes from mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87): icmp_req=5 ttl=64 time=0.623 ms
64 bytes from mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87): icmp_req=6 ttl=64 time=0.444 ms
64 bytes from mtx2-md2a.wdc.sl.serverdensity.net (10.56.164.87): icmp_req=7 ttl=64 time=0.297 ms
^C
--- mtx2-md2a.wdc.sl.serverdensity.net ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 5997ms
rtt min/avg/max/mdev = 0.297/0.461/0.623/0.101 ms
```

```
15/05/13 08:13:27 david@mtx2-md1a ~: ping mtx2-md1b
PING mtx2-md1b.sjc.sl.serverdensity.net (10.52.8.160) 56(84) bytes of data.
64 bytes from mtx2-md1b.sjc.sl.serverdensity.net (10.52.8.160): icmp_req=1 ttl=54 time=71.7 ms
64 bytes from mtx2-md1b.sjc.sl.serverdensity.net (10.52.8.160): icmp_req=2 ttl=54 time=72.1 ms
64 bytes from mtx2-md1b.sjc.sl.serverdensity.net (10.52.8.160): icmp_req=3 ttl=54 time=71.9 ms
64 bytes from mtx2-md1b.sjc.sl.serverdensity.net (10.52.8.160): icmp_req=4 ttl=54 time=72.1 ms
64 bytes from mtx2-md1b.sjc.sl.serverdensity.net (10.52.8.160): icmp_req=5 ttl=54 time=71.9 ms
64 bytes from mtx2-md1b.sjc.sl.serverdensity.net (10.52.8.160): icmp_req=6 ttl=54 time=72.0 ms
64 bytes from mtx2-md1b.sjc.sl.serverdensity.net (10.52.8.160): icmp_req=7 ttl=54 time=71.9 ms
^C
--- mtx2-md1b.sjc.sl.serverdensity.net ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6007ms
rtt min/avg/max/mdev = 71.780/72.000/72.158/0.235 ms
```

Cross USA Washington, DC - San Jose, CA

Performance

- Fast network

Location	Ping RTT Latency
Within USA	40-80ms
Trans-Atlantic	100ms
Trans-Pacific	150ms
Europe - Japan	300ms

Ping – low overhead
Important for replication

Failover

- Replication

Failover

- Replication
- Master/slave

- One master accepts all writes
- Many slaves staying up to date with master
- Can read from slaves

Failover

- Replication
- Master/slave
- Min 3 nodes

Minimum of 3 nodes to form a majority in case one goes down.
All store data.
Odd number otherwise != majority
Arbiter

Failover

- Replication
- Master/slave
- Min 3 nodes
- Automatic failover

Drivers handle automatic failover. First query after a failure will fail which will trigger a reconnect. Need to handle retries

Performance

- Replication lag

Location	Ping RTT Latency
Within USA	40-80ms
Trans-Atlantic	100ms
Trans-Pacific	150ms
Europe - Japan	300ms

- Replication lag

Replication Lag

1. Reads: eventual consistency

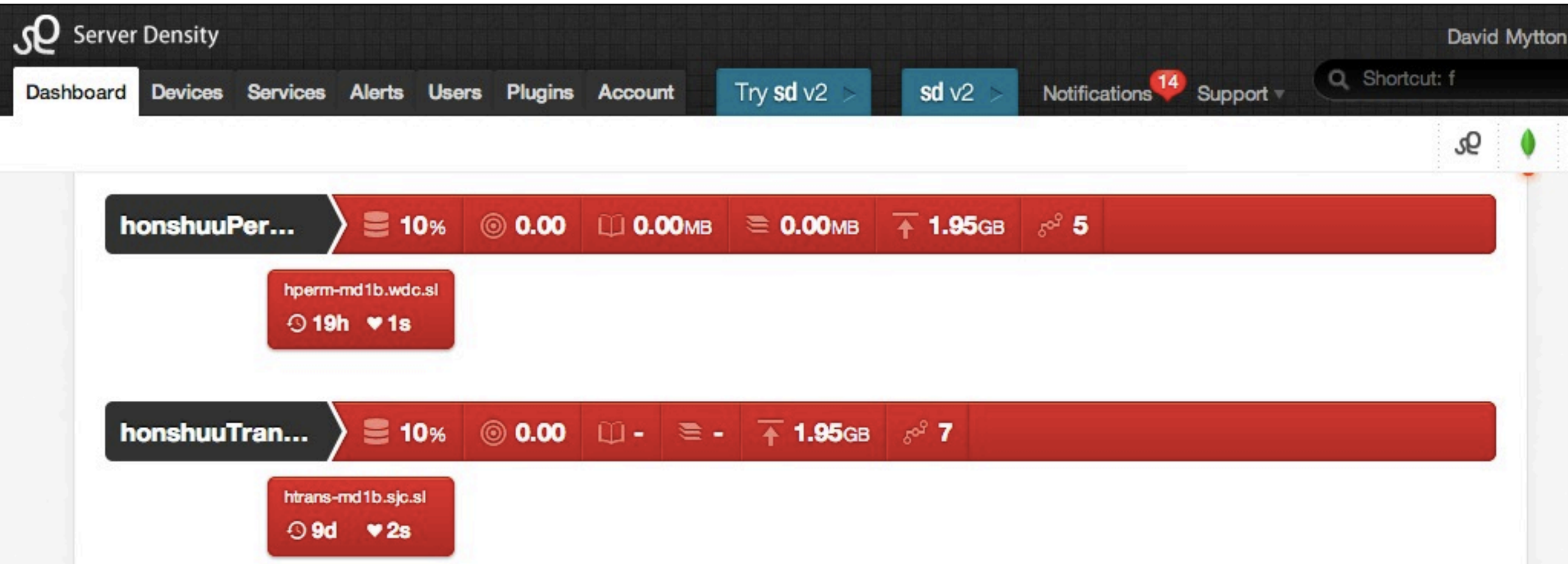
Replication Lag

1. Reads: eventual consistency

2. Failover: slave behind

Slave behind

Failover: out of date master



Old data
Rollback

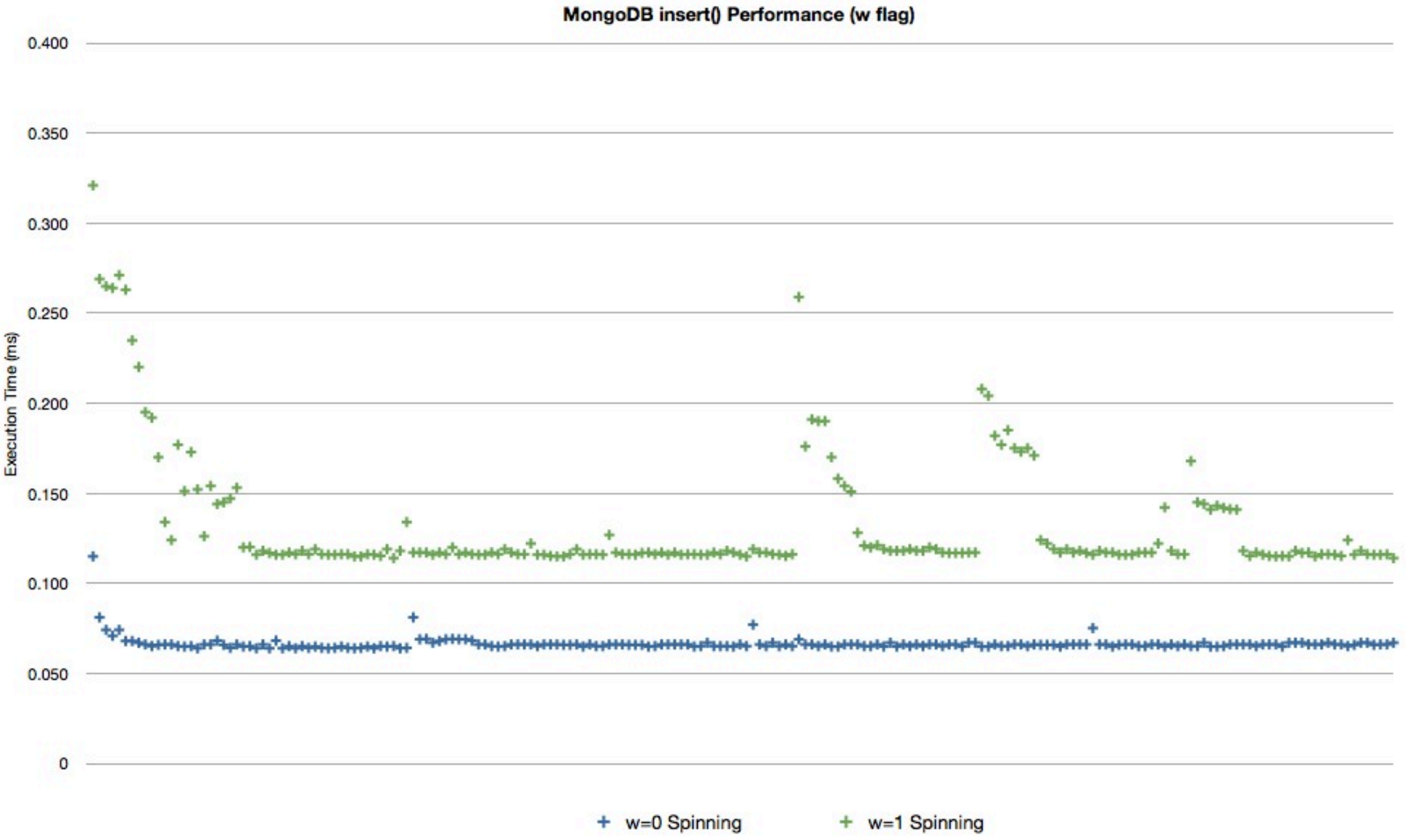
MongoDB WriteConcern

- Safe by default

```
>>> from pymongo import MongoClient
>>> connection = MongoClient(w=int/str)
```

Value	Meaning
0	Unsafe
1	Primary
2	Primary + x1 secondary
3	Primary + x2 secondaries

wtimeout - wait for write before raising an exception


MongoDB insert() Performance (w flag)
Execution Time (ms)
0.400
0.350
0.300
0.250
0.200
0.150
0.100
0.050
0
w=0 Spinning
w=1 Spinning

Performance

- Fast network
- More RAM

Picture is unrelated! Mmm, ice cream.

Amazon EC2 Instance Types

Standard On-Demand Instances	Linux/UNIX Usage
~~Small (Default)~~	~~$0.085 per hour~~
Large	$0.34 per hour
Extra Large	$0.68 per hour
Micro On-Demand Instances	
Micro	$0.02 per hour
High-Memory On-Demand Instances	
Extra Large	$0.50 per hour
Double Extra Large	$1.00 per hour
Quadruple Extra Large	$2.00 per hour
High-CPU On-Demand Instances	
~~Medium~~	~~$0.17 per hour~~
Extra Large	$0.68 per hour
Cluster Compute Instances	
Quadruple Extra Large	$1.60 per hour
Cluster GPU Instances	
Quadruple Extra Large	$2.10 per hour

32-bit = Don't Use

Typical MongoD

ConfigD / Arbiter

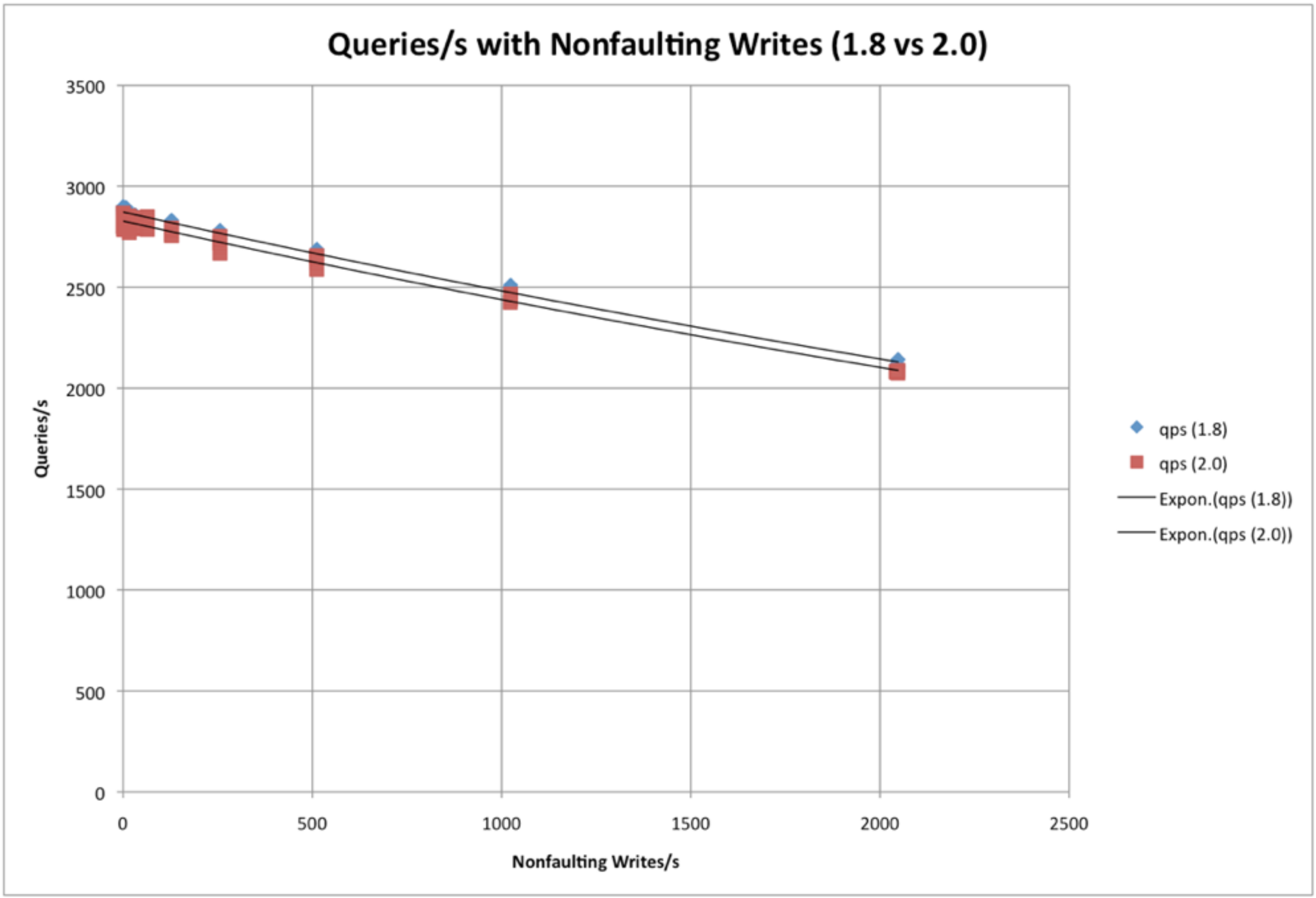
Big MongoD

32-bit = Don't Use

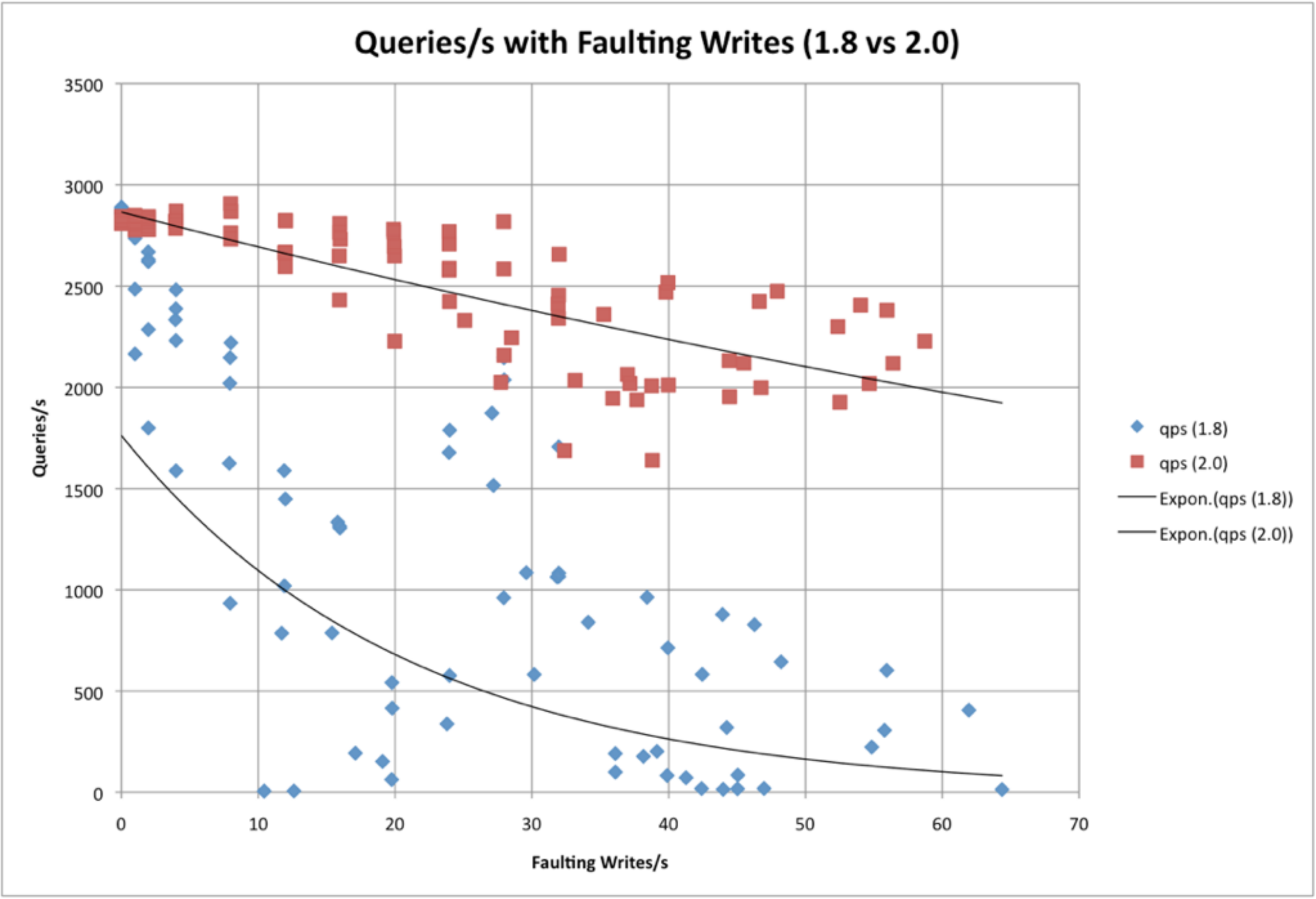
High CPU not necessary

http://www.slideshare.net/jrosoff/mongodb-on-ec2-and-ebs

No 32 bit
No High CPU
RAM RAM RAM.



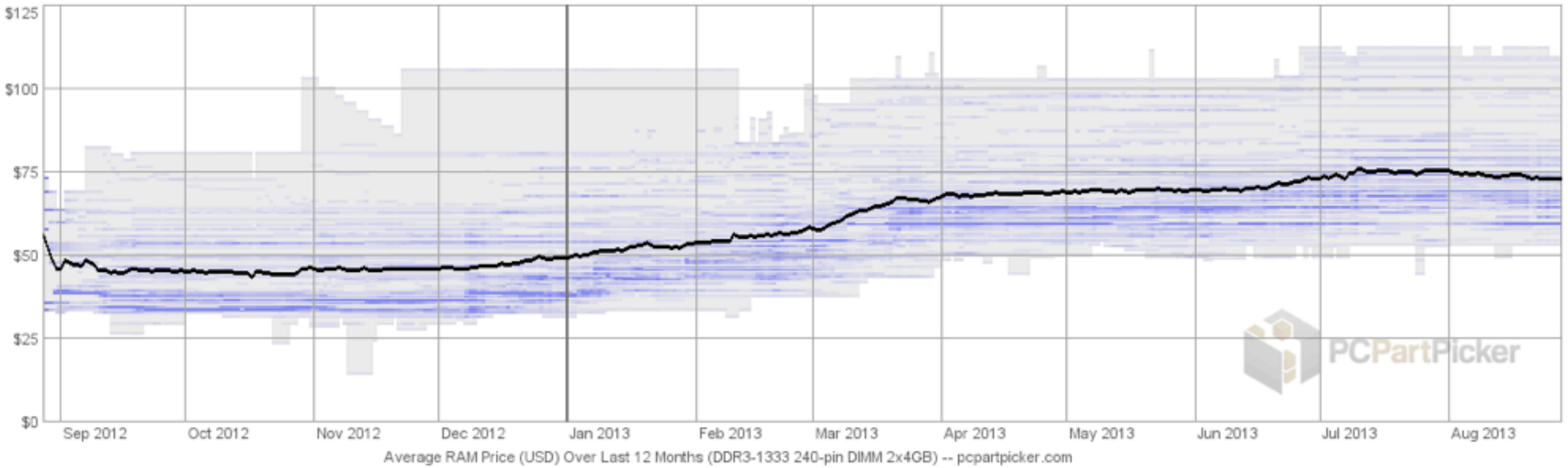
http://blog.pythonisito.com/2011/12/mongodbs-write-lock.html



http://blog.pythonisito.com/2011/12/mongodbs-write-lock.html

Performance

More RAM = expensive



x2 4GB RAM 12 month Prices


RAM
SSDs
Spinning disk
Cost
Speed

Performance

Softlayer disk pricing

Unit	Monthly
32GB SSD	$25.00
50GB SSD	$30.00
64GB SSD	$40.00
100GB SSD	$100.00
200GB SSD	$125.00
400GB SSD	$200.00
800GB SSD	$300.00

Unit	Monthly
250GB - SATA II Hard Drive	$20.00
500GB - SATA II Hard Drive	$30.00
750GB - SATA II Hard Drive	$40.00
1.00TB - SATA II Hard Drive	$50.00
2.00TB - SATA II Hard Drive	$60.00
3.00TB - SATA III Hard Drive	$80.00
4.00TB - SATA III Hard Drive	$100.00

Unit	Monthly
73GB SA-SCSI 10K Hard Drive	$30.00
73GB SA-SCSI 15K Hard Drive	$50.00
147GB SA-SCSI 10K Hard Drive	$50.00
147GB SA-SCSI 15K Hard Drive	$75.00
300GB SA-SCSI 10K Hard Drive	$75.00
300GB SA-SCSI 15K Hard Drive	$100.00
450GB SA-SCSI 15K Hard Drive	$125.00
600GB SA-SCSI 15K Hard Drive	$150.00

Performance

EC2 disk/RAM pricing

Standard On-Demand Instances	
Small (Default)	$0.060 per Hour
Medium	$0.120 per Hour
Large	$0.240 per Hour
Extra Large	$0.480 per Hour

$43/m

High-Memory On-Demand Instances	
Extra Large	$0.410 per Hour
Double Extra Large	$0.820 per Hour
Quadruple Extra Large	$1.640 per Hour

$295/m

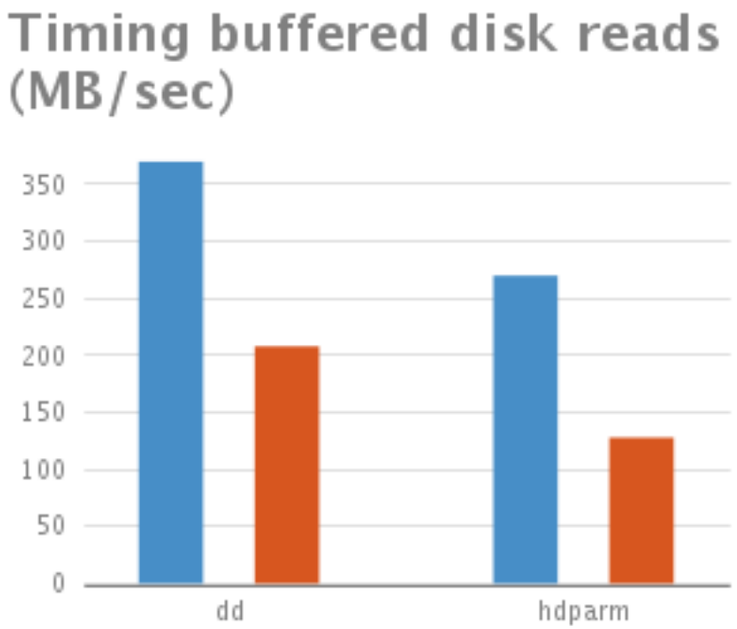
High-Memory Cluster On-Demand Instances	
Eight Extra Large	$3.500 per Hour
Cluster GPU Instances	
Quadruple Extra Large	$2.100 per Hour
High-I/O On-Demand Instances	
Quadruple Extra Large	$3.100 per Hour

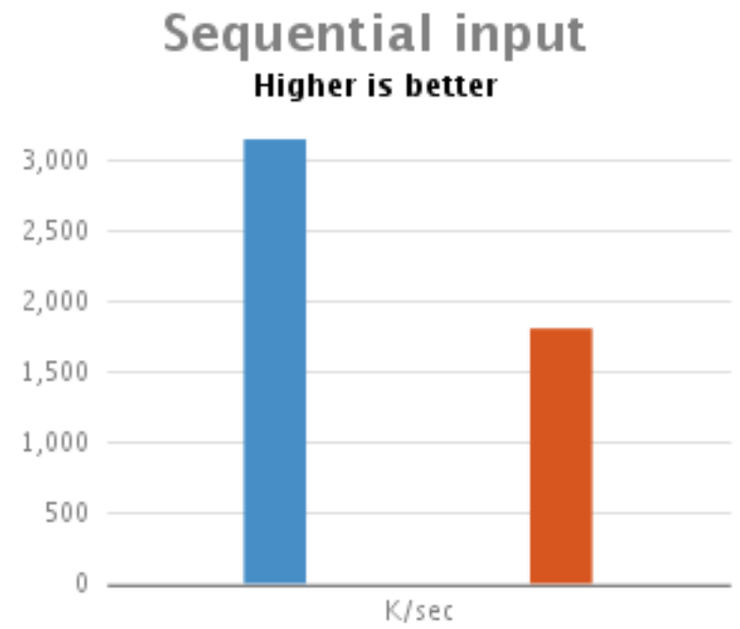
$2520/m

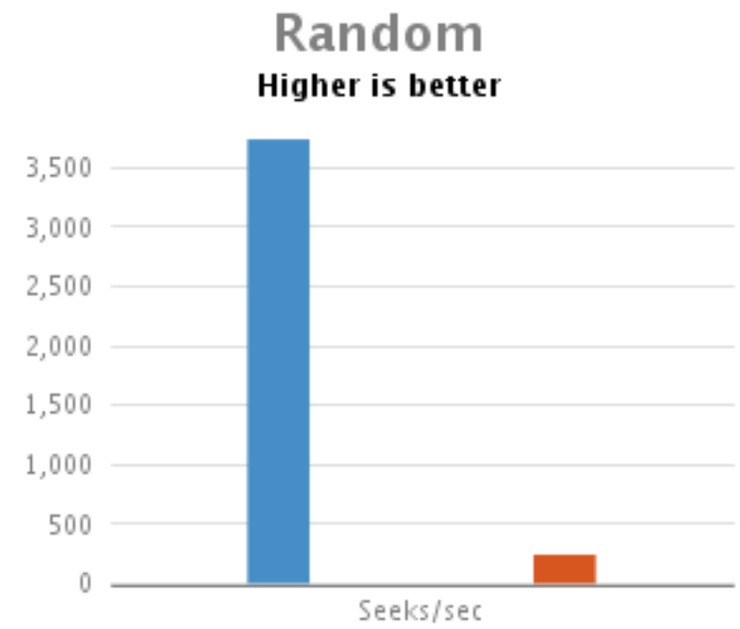
$2232/m

Performance

SSD vs Spinning



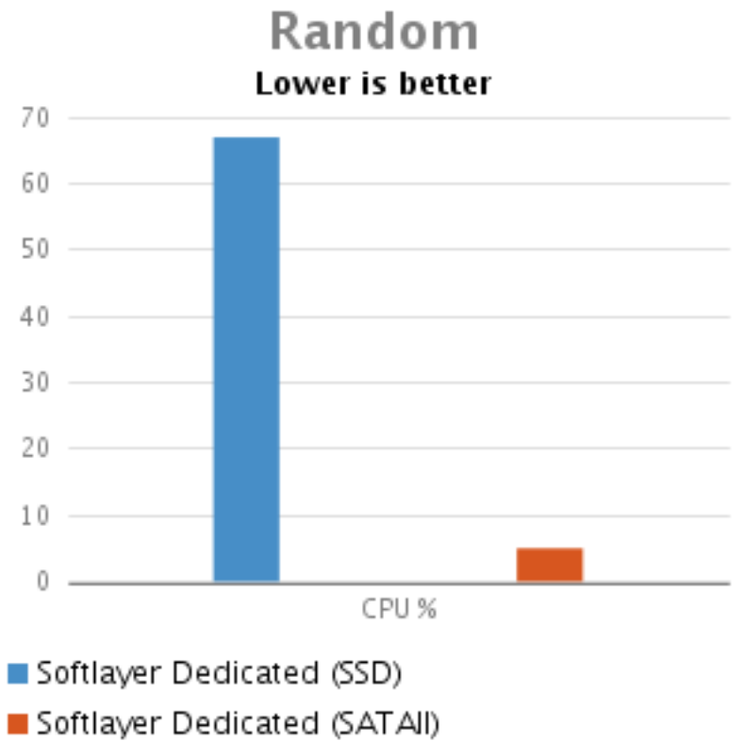


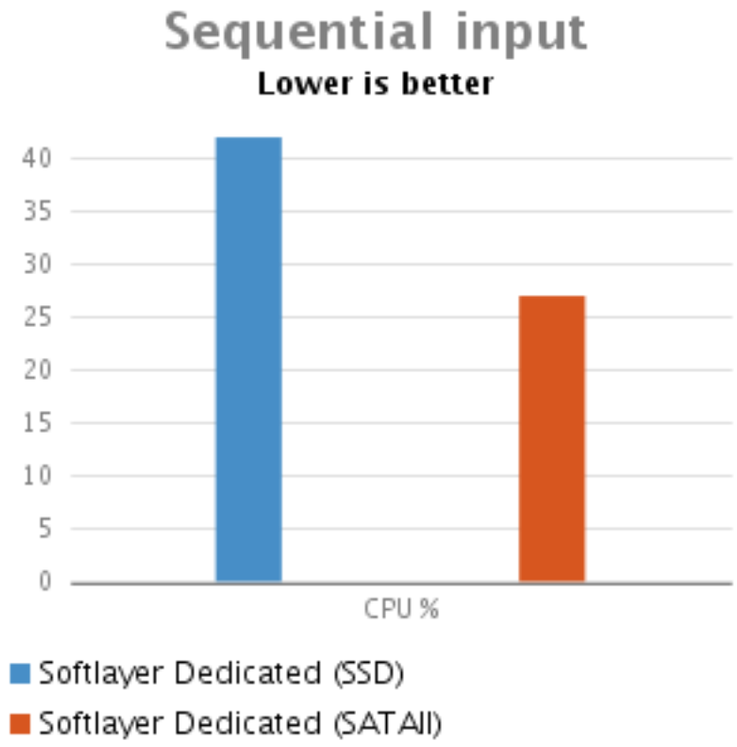


SSDs are better at buffered disk reads, sequential input and random i/o.

Performance

SSD vs Spinning





However, CPU usage for SSDs is higher. This may be a driver issue so worth testing your own hardware. Tests done using Bonnie.

Cloud?

Cloud?

- Elastic workloads

Cloud?

- Elastic workloads
- Demand spikes

Cloud?

- Elastic workloads
- Demand spikes
- Unknown requirements

Dedicated?

Dedicated?

- Hardware replacement

Dedicated?

- Hardware replacement
- Managed/support

Dedicated?

- Hardware replacement
- Managed/support
- Networking

Colo?

Colo?

- Hardware spec/value

Colo?

- Hardware spec/value
- Total cost

Colo?

- Hardware spec/value
- Total cost
- Internal skills?

Colo?

- Hardware spec/value
- Total cost
- Internal skills?
- More fun?!


DELL
DELL
DELL

Colo experiment

- Build master (buildbot): VM x2 CPU 2.0Ghz, 2GB RAM
  - $89/m
- Build slave (buildbot): VM x1 CPU 2.0Ghz, 1GB RAM
  - $40/m
- Staging load balancer: VM x1 CPU 2.0Ghz, 1GB RAM
  - $40/m
- Staging server 1: VM x2 CPU 2.0Ghz, 8GB RAM
  - $165/m
- Staging server 2: VM x1 CPU 2.0Ghz, 2GB RAM
  - $50/m
- Puppet master: VM x2 CPU 2.0Ghz, 2GB RAM
  - $89/m

Total: $473/m

Colo experiment

- Dell 1U R415
- x2 8C AMD 2.8Ghz
- 32GB RAM

Colo experiment

- Dell 1U R415
- x2 8C AMD 2.8Ghz
- 32GB RAM
- Dual PSU, NIC

Colo experiment

- Dell 1U R415
- x2 8C AMD 2.8Ghz
- 32GB RAM
- Dual PSU, NIC
- x4 1TB SATA hot swappable

Colo: Networking

- 10-50Mbps: £20-25/Mbps/m
- 51-100Mbps: £15/Mbps/m
- 100+Mbps: £13/Mbps/m

Colo: Metro

- 100Mbps: £300/m
- 1000Mbps: £750/m

Colo: Power

- £300-350/kWh/m
- 4.5A = £520/m
- 9A = £900/m

How we charge for our services

TelecityGroup
where content meets connectivity

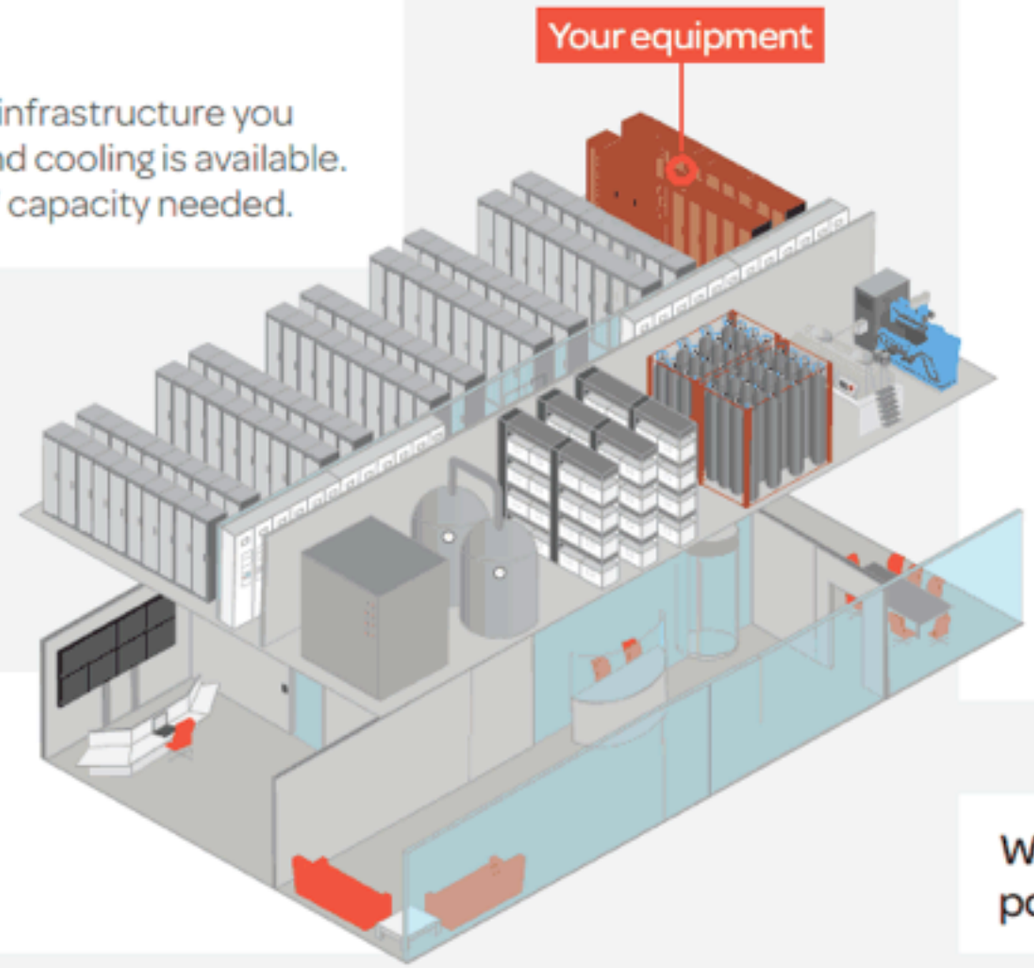
At TelecityGroup we build open, transparent and lasting partnerships with all our customers. Our flexible approach means we can deliver solutions unique to every business, including how customers pay for data centre costs.

Set-up costs (one-off cost)

Providing the racks, connectivity and power feed, ready for installation of your equipment.

Capacity reservation fee (annually recurring cost)

This ensures the core data centre infrastructure you may need including, power, UPS and cooling is available. This is calculated based on the kW capacity needed.



At TelecityGroup we always have enough infrastructure to support every contracted customer requirement.

Power consumption (monthly cost)

There are three ways you can pay for the power needed to run and cool your equipment:

1. Metered power
With our metered option you only pay for what you use at a fixed rate per kW hour. Ideal for customers who prefer to only pay for what they use.

2. Partially inclusive power
Includes a fixed element and variable element charged at a fixed rate per kW hour of power used, ideal for customers who prefer to have a predominantly fixed monthly cost. Usage will be metered and any additional cost will be invoiced.

3. Fully inclusive power
A flat invoice for all of the power you have reserved to run your equipment, charged at a fixed rate per kW hour. Ideal for customers who prefer not to receive variable invoices.

With TelecityGroup you can choose how much power you need.

Backups

What is the use case?

- What kind of disaster?
- Store backups offsite

How log do you keep the backups for?
How far do they go back?
How recent are they?

Latency issue - further away geographically, slower the transfer time
Partition backups to get critical data restored first

```
david@asriel ~: scp david@stelmaria:~/local/local.11 .
local.11                      100% 2047MB    6.8MB/s   05:01
```

Restore time

- Needed to resync a database server across the US
- Take too long; oplog not large enough
- Fast internal network but slow internet

1d, 1h, 58m

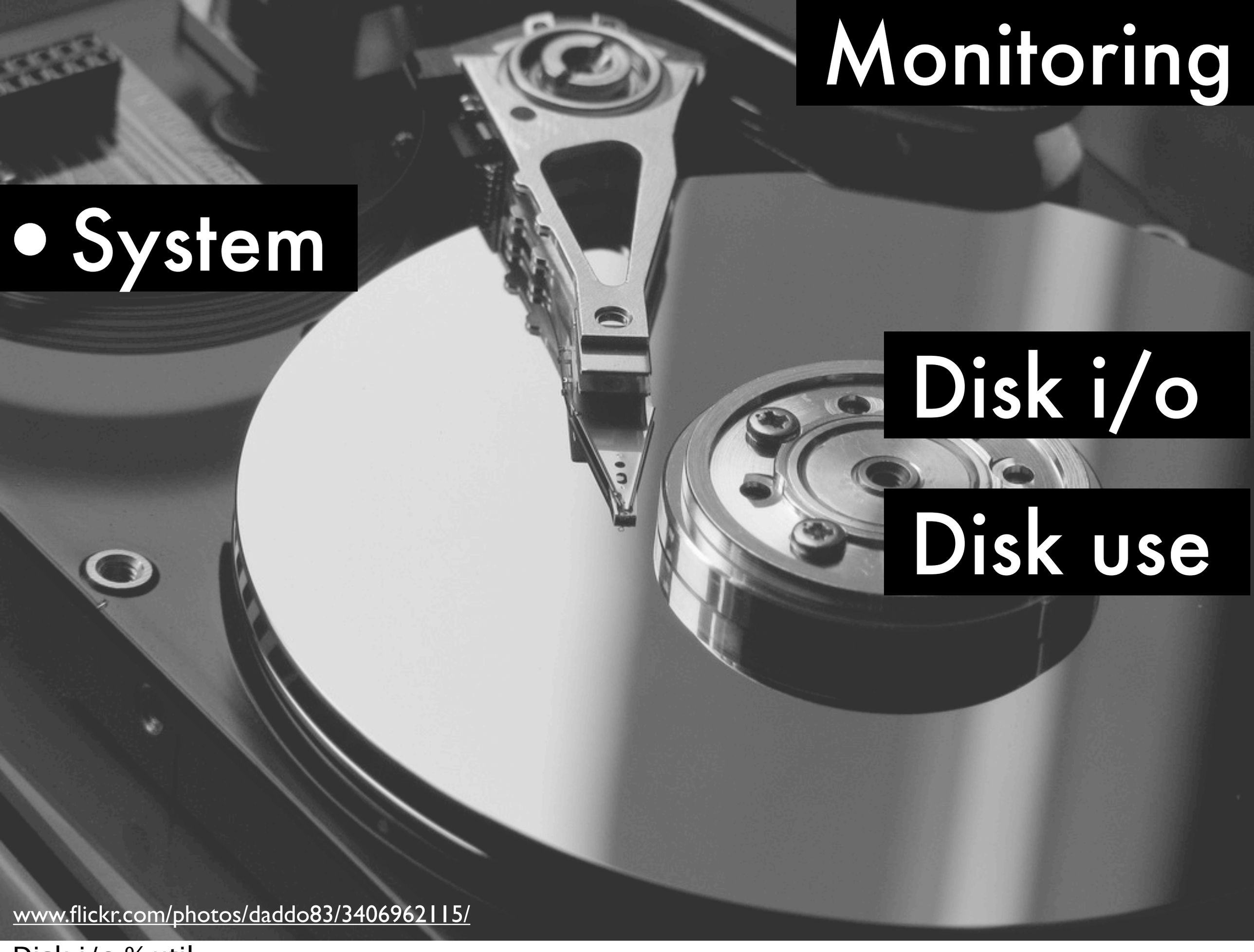
11.22MB/s

Monitoring

- System

Disk i/o

Disk use



www.flickr.com/photos/daddo83/3406962115/

Disk i/o % util
Disk space usage

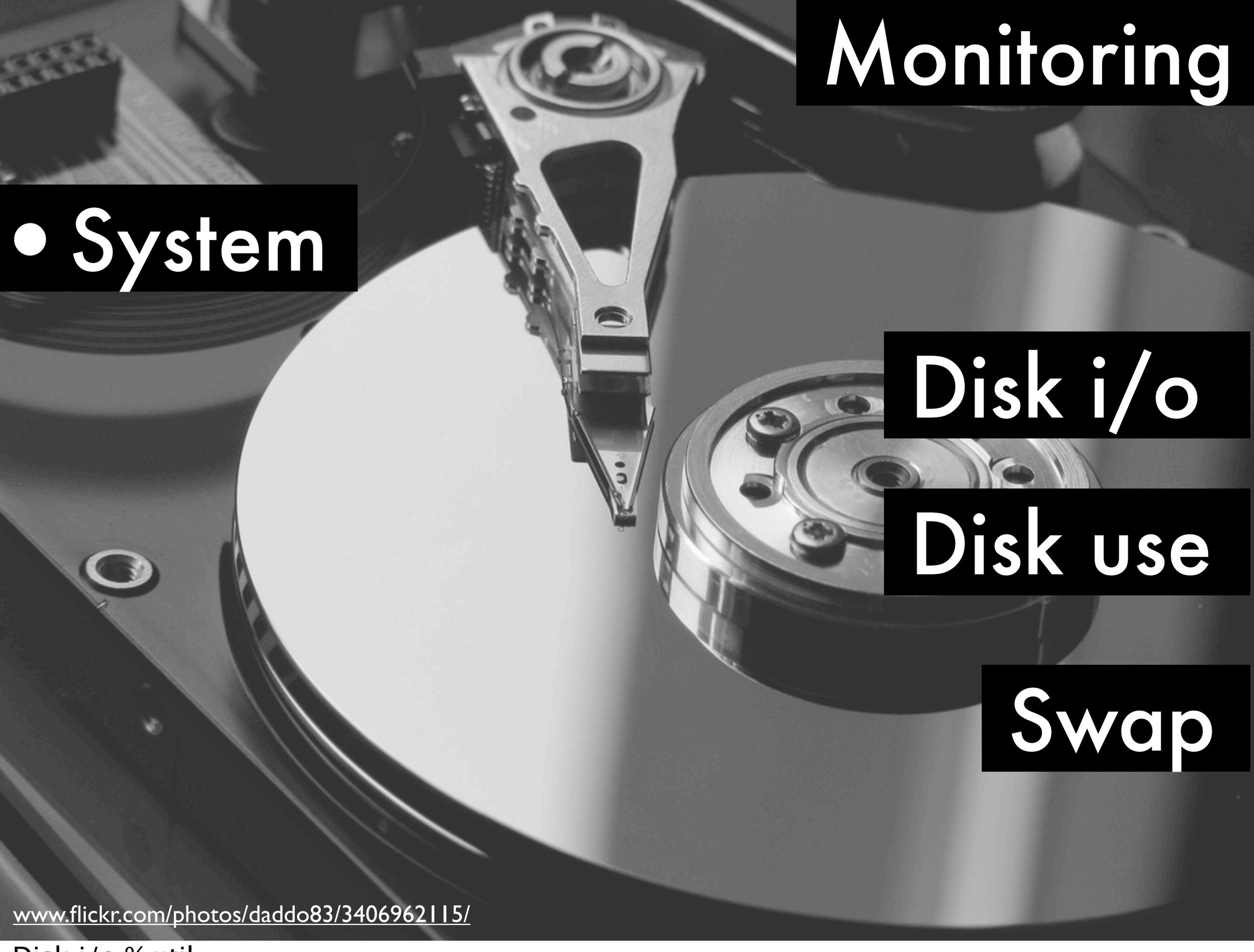
Monitoring

- System

Disk i/o

Disk use

Swap



www.flickr.com/photos/daddo83/3406962115/

Disk i/o % util
Disk space usage


空
雷門角 栄盛堂雷おこし本舗直営
雷5656会館
ゴロゴロ
Monitoring
• Replication
Slave lag
State
www.flickr.com/photos/daddo83/3406962115/

Monitoring tools

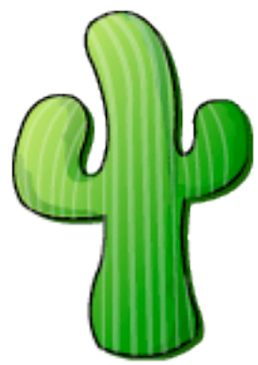
Run yourself

Nagios®

MUNIN

ZABBIX

Ganglia



So Server Density is the tool my company produces but if you don’t like it, want to run your own tools locally or just want to try some others, then that’s fine.

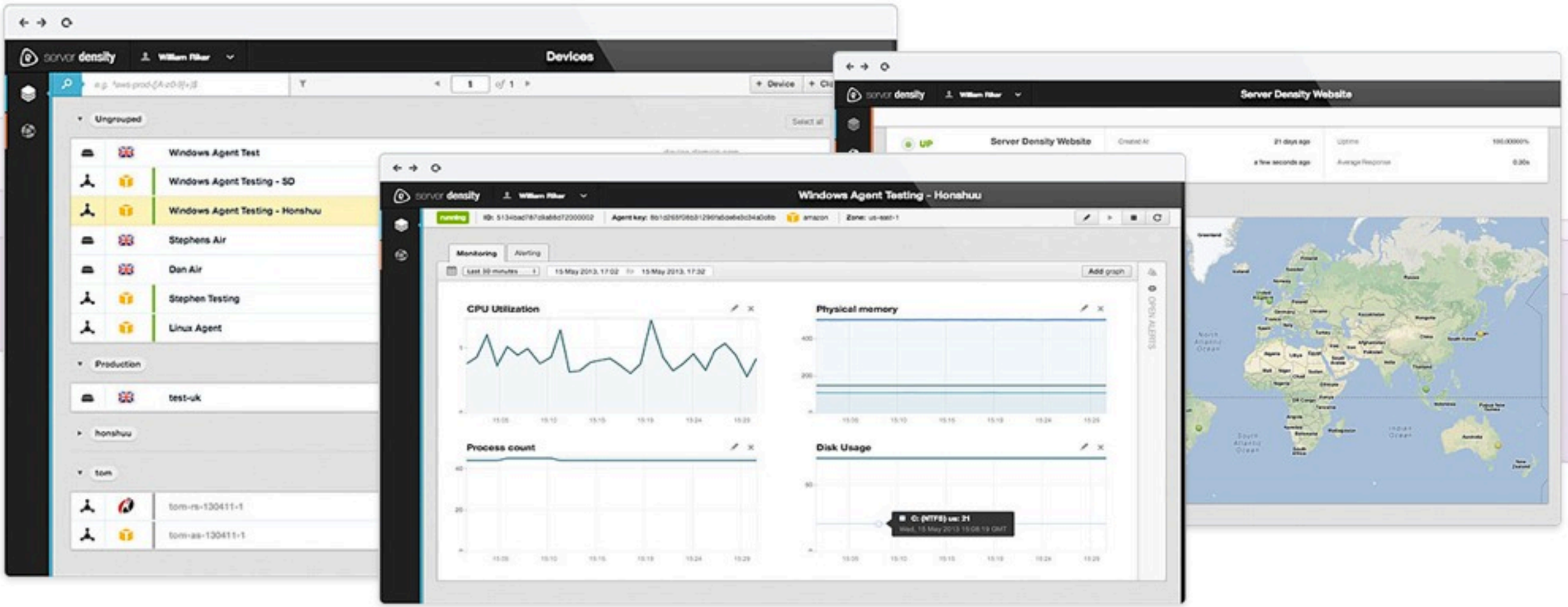
Monitoring tools

server density

cloud management

server monitoring

website monitoring



www.serverdensity.com


David Mytton
@davidmytton
david@serverdensity.com
blog.serverdensity.com
Woop Japan!