

# Scalability

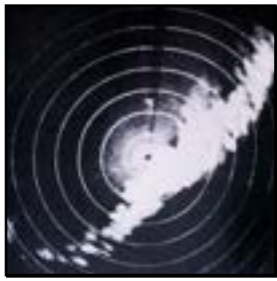


# Boot Camp

SXSW 2008



Jakob Heuser  
Alan Kasindorf  
Blaine Cook  
Sandy Jen  
Kerry Miller



# Briefing

## Panelists

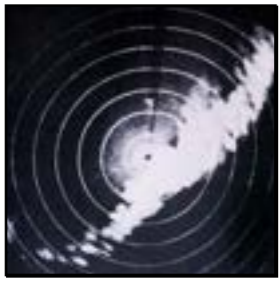
Kerry Miller - BusinessWeek

Alan Kasindorf (aka dormando) - Six Apart

Jakob Heuser - Gaia Online

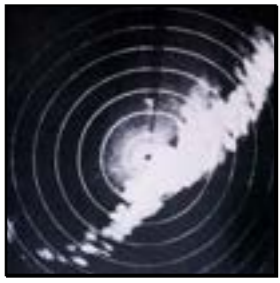
Blaine Cook - Twitter

Sandy Jen - Meebo



# Why Scale

- Good problem
- Doesn't have to cost
- It's an "everybody" thing

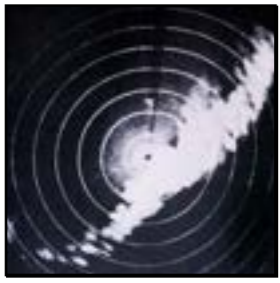


# The Regimen

- Problem
- Concepts
- Abstract Solutions
- Google Time
- Conversation

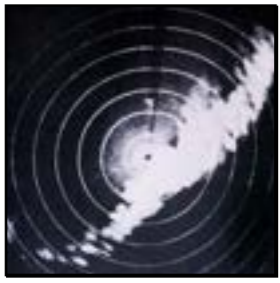
# Monitoring

avoid working in the dark



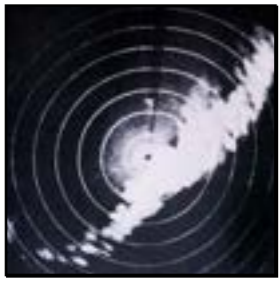
# Get it on the radar

- Understand the “pain points”
- Live and die by monitoring
- Monitor **EVERYTHING**



# “Everything”?

- Disk I/O
- Memory
- Bandwidth
- Page Load Times
- The list goes on...



# Google Time

- Ganglia
- Hyperic
- sar and sysstat - simple, you already have it
- Know your tool, whatever it is



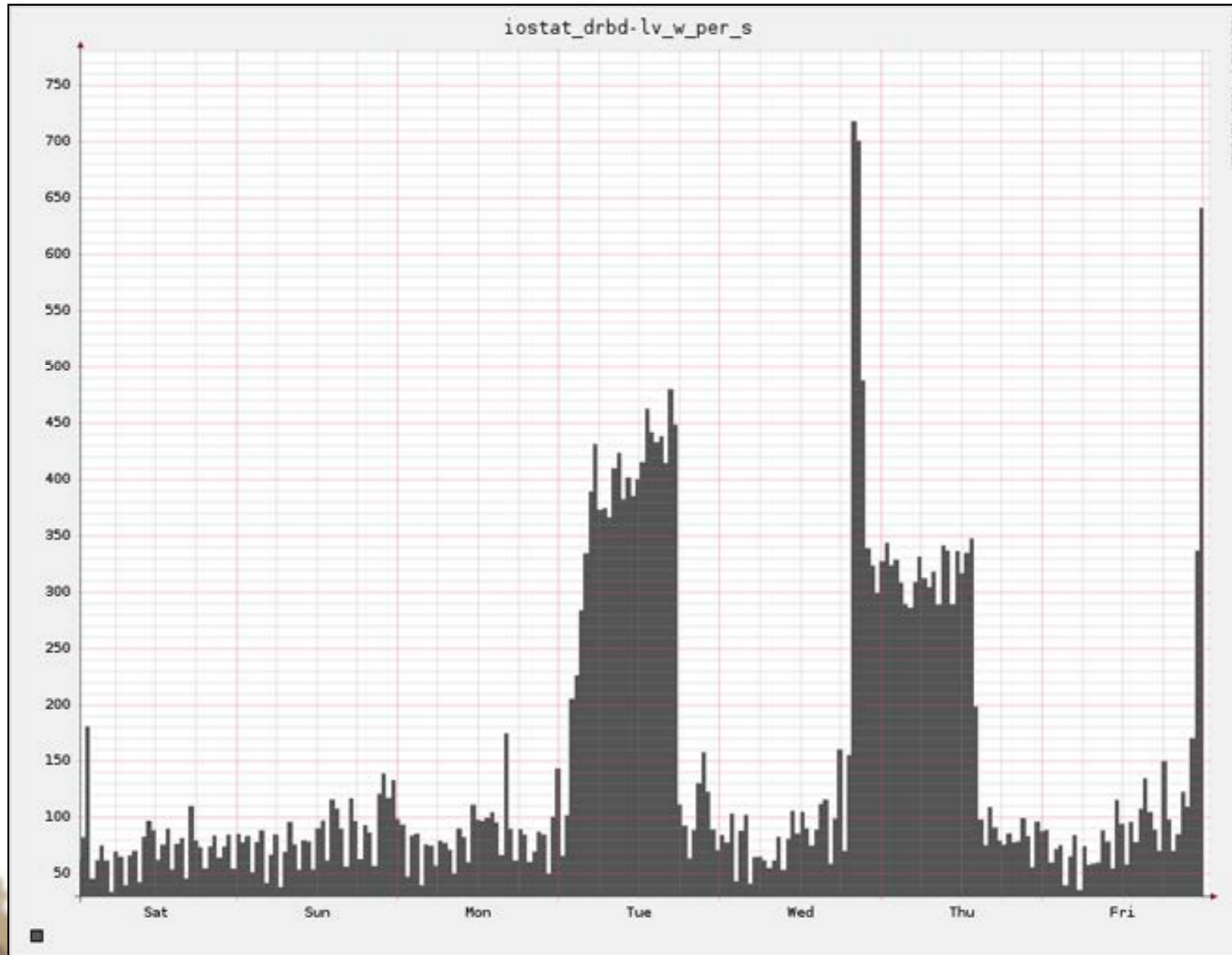
# File System Solutions

users like making tons of lolcats  
and storing them on your website



# Never gonna be a hero

## Disk IO Graph





# Use what you have

- Don't waste capacity
- Use someone else's space
- Avoid a single "authority" on a file



# Google Time

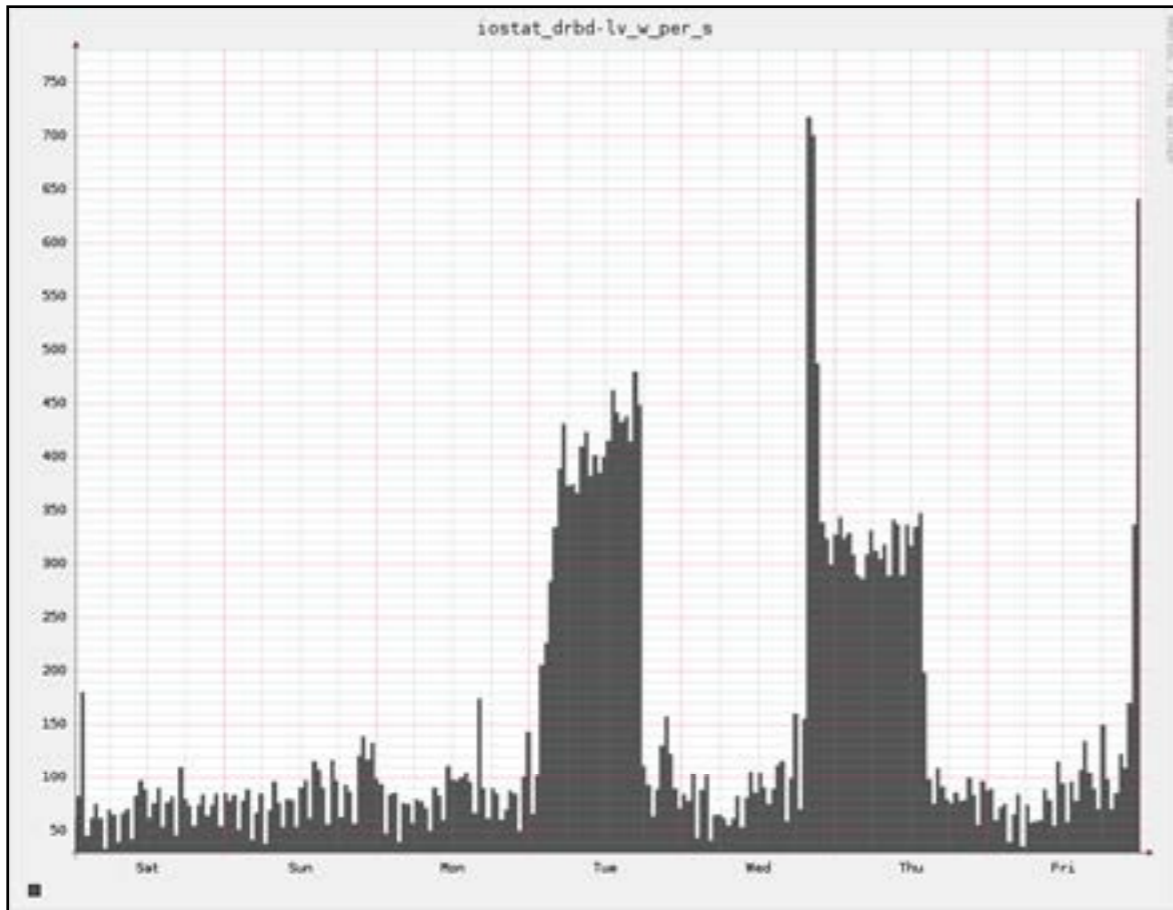
- DRBD + OCFS
- Amazon S3
- MogileFS (Danga Software)
- lustre

# The Database Layer

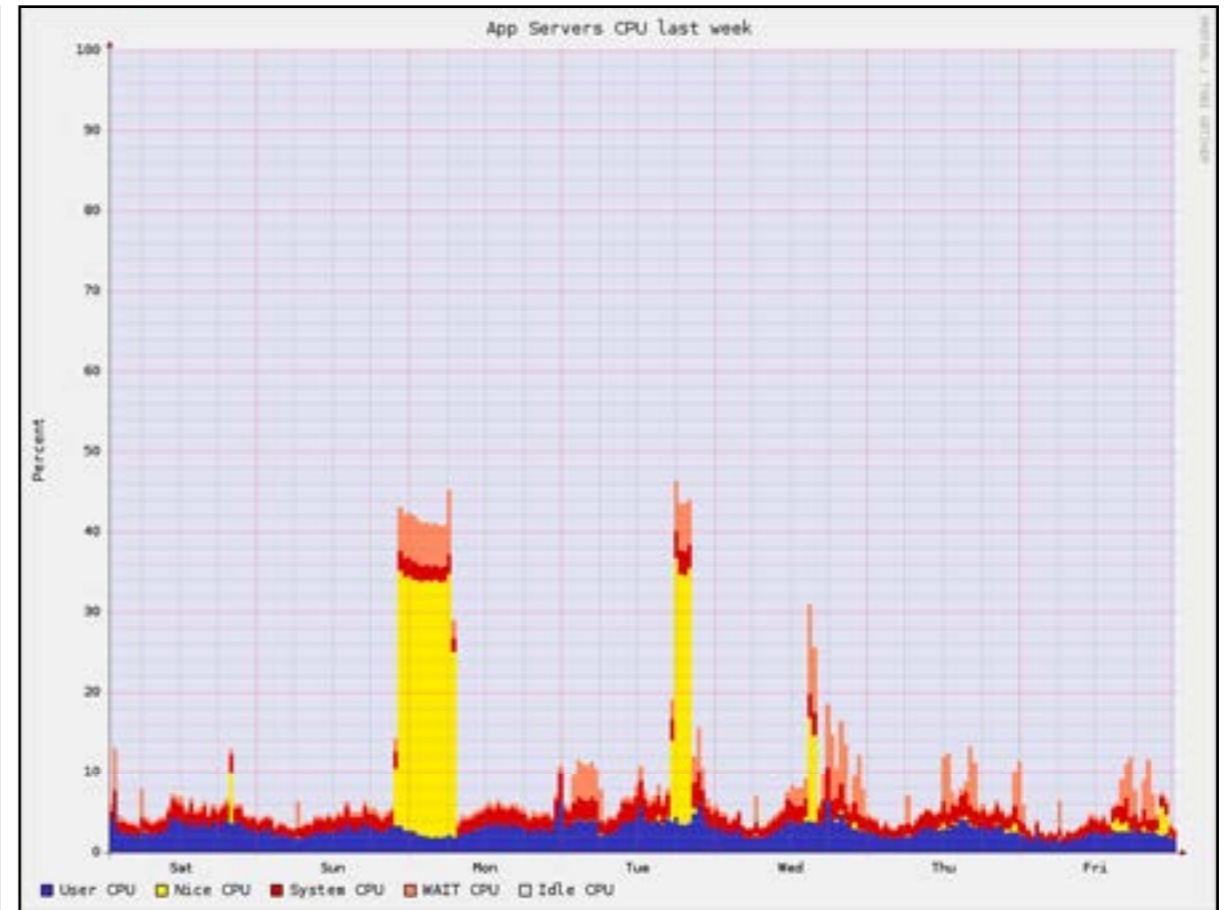
your most common, but  
hardest to solve  
bottleneck



# Enemy diversion



Disk IO



CPU Usage



# The real problem

## Show Processlist Output

```
mysql> show processlist;
```

Id	User	Host	db	Command	Time	State	Info
43	root	localhost	NULL	Query	0	NULL	show processlist
48	root	localhost	example	Query	200	Sending data	SELECT SQL_CALC_FOUND_ROWS * FROM `users` WHERE email like "%bb%" LIMIT 1
49	root	localhost	example	Query	183	Sending data	SELECT SQL_CALC_FOUND_ROWS * FROM `users` WHERE email like "%bb%" LIMIT 1
53	root	localhost	example	Query	147	Sending data	SELECT SQL_CALC_FOUND_ROWS * FROM `users` WHERE email like "%bb%" LIMIT 1
48	root	localhost	example	Query	120	Sending data	SELECT SQL_CALC_FOUND_ROWS * FROM `users` WHERE email like "%bb%" LIMIT 1
48	root	localhost	example	Query	90	Sending data	SELECT SQL_CALC_FOUND_ROWS * FROM `users` WHERE email like "%bb%" LIMIT 1
48	root	localhost	example	Query	47	Sending data	SELECT SQL_CALC_FOUND_ROWS * FROM `users` WHERE email like "%bb%" LIMIT 1
48	root	localhost	example	Query	27	Sending data	SELECT SQL_CALC_FOUND_ROWS * FROM `users` WHERE email like "%bb%" LIMIT 1



# Another war zone

- Make “:) SQL” not “:( SQL”
- Horizontal Partitioning
- Caching Layer





# Google Time

- Memcache
- HiveDB
- CouchDB / Hypertable
- MySQL Consultant

# Parallel Processing

you don't have to do it all right now



# Under siege

```
mysql> select * from timelog where lapse > 5 limit 5
+-----+-----+-----+
| uri      | start          | lapse |
+-----+-----+-----+
| /example | 1204954951.8172 | 39.901 |
| /example | 1204954971.4523 | 35.712 |
| /example | 1204956191.2133 | 50.912 |
| /example | 1204959183.6785 | 20.743 |
| /example | 1204981923.8372 | 70.621 |
+-----+-----+-----+
```

Slow Query Log



# Smarter, not stronger

- Consistent for current user, not everyone
- Design code for parallel steps
- cronjobs



# Google Time

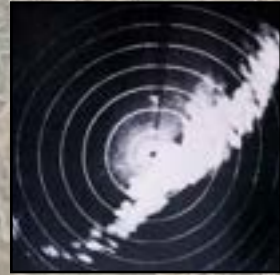
- Starling
- Gearman
- TheSchwartz

# Regroup

# Regroup

- All these technologies are built to be asynchronous
- An amazing amount of your app can be asynchronous too

<http://www.slideshare.net/Jakobo>



# Scalability



# Boot Camp

SXSW 2008



Jakob Heuser  
Alan Kasindorf  
Blaine Cook  
Sandy Jen  
Kerry Miller



# Resources

## **Monitoring**

<http://ganglia.sourceforge.net>

<http://www.hyperic.com>

<http://www.nagios.org>

<http://pagesperso-orange.fr/sebastien.godard/>

<http://www.cacti.net>

## **CDN**

[http://en.wikipedia.org/wiki/Content\\_Delivery\\_Network](http://en.wikipedia.org/wiki/Content_Delivery_Network)

[http://en.wikipedia.org/wiki/Akamai\\_Technologies](http://en.wikipedia.org/wiki/Akamai_Technologies)



# Resources

## **File Systems**

<http://www.danga.com/mogilefs/>

<http://www.lustre.org>

<http://www.drbd.org>

<http://oss.oracle.com/projects/ocfs/>

## **Database**

<http://www.planetmysql.com>

<http://datacharmer.blogspot.com/>

<http://www.danga.com/memcached/>

## **Job and Queue Systems**

<http://www.danga.com/gearman/>

<http://rubyforge.org/projects/starling/>



# Image Credit

[http://www.history.noaa.gov/stories\\_tales/radar.html](http://www.history.noaa.gov/stories_tales/radar.html) (radar)

<http://www.thatpoliticalblog.com/serendipity/plugin/tag/TPB+Information> (bar chart)

<http://www.digitalearth.com.au/category/general/> (disk on fire)

<http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=160210671355> (dump truck)

<http://420.thrashbarg.net/> (marching penguins)

