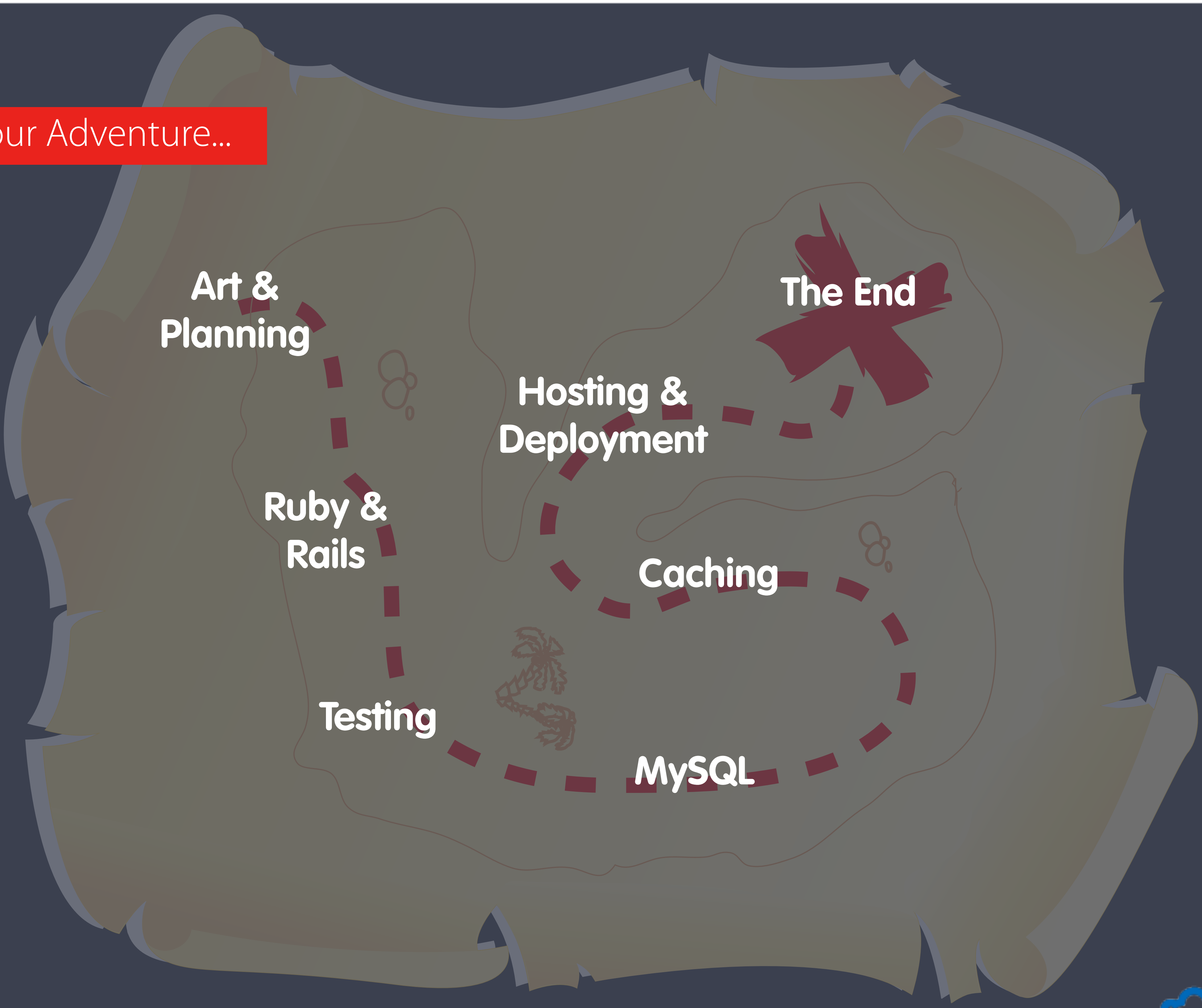
Scalability + Performance

a choose your own adventure game

hosted by James Cox

smokeclouds

Begin Your Adventure...



smokeclouds

OH NOES!

smokeclouds

Scalability is an artform

“EVERYTHING takes time. No amount of super fast assembler based multiplexed evented code will get around that”

“I'm not saying don't try to make it fast. What I'm saying is first thing programmers do is they run off with faulty statistics to "tune" their system, completely ignoring the fact that many times a simple redesign (or complex improvement) can just eliminate the problem entirely. ”

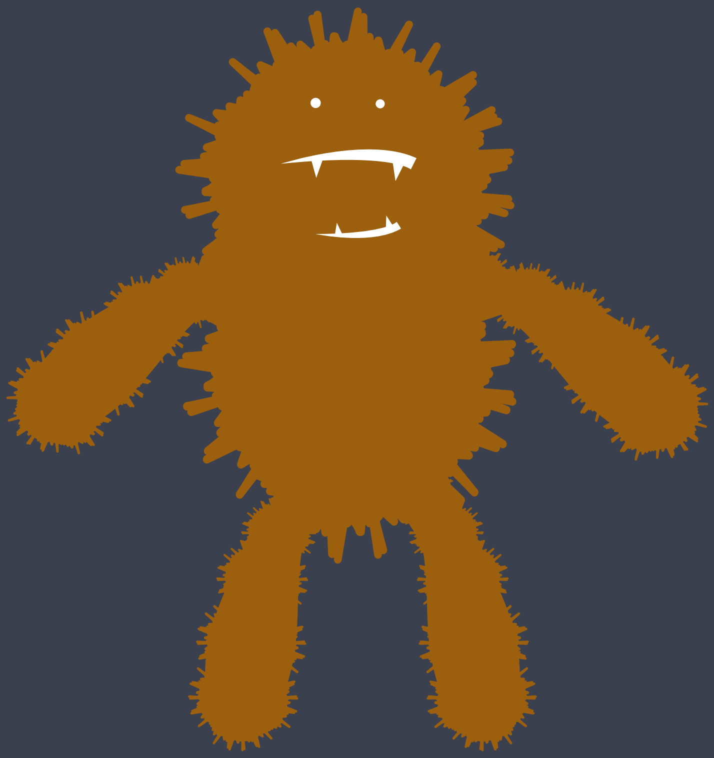
“If you can’t make the computer fast, trick the people to think it’s fast”

smokeclouds

Scalability is an artform

There are no easy answers.

... except maybe this:



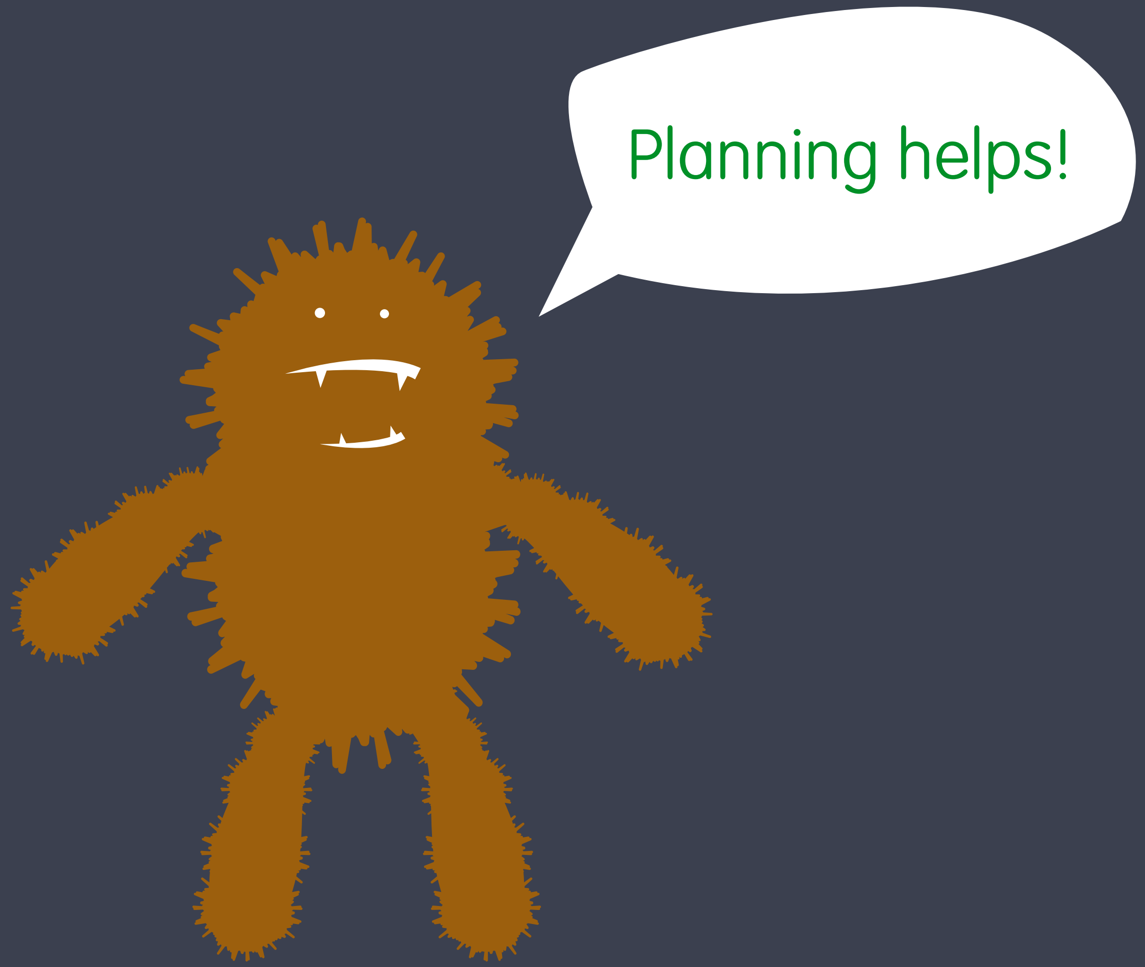
measure, refactor.

measure, refactor.

measure, refactor.

smokeclouds

Scalability is an artform



back to the map

smokeclouds

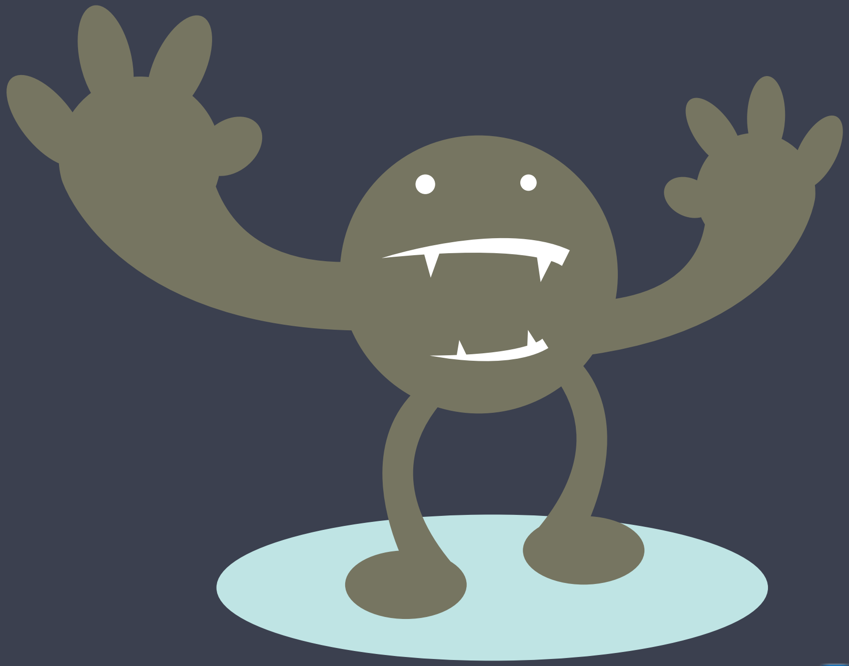
MySQL: Built To Perform

- mysql> \s

- Threads: 3, Questions: 10,171,505, Slow queries: 334, Opens: 224, Flush tables: 1, Open tables: 106

- Queries per second (average): 277.1

- Uptime: 10 hours 11 min 47 sec

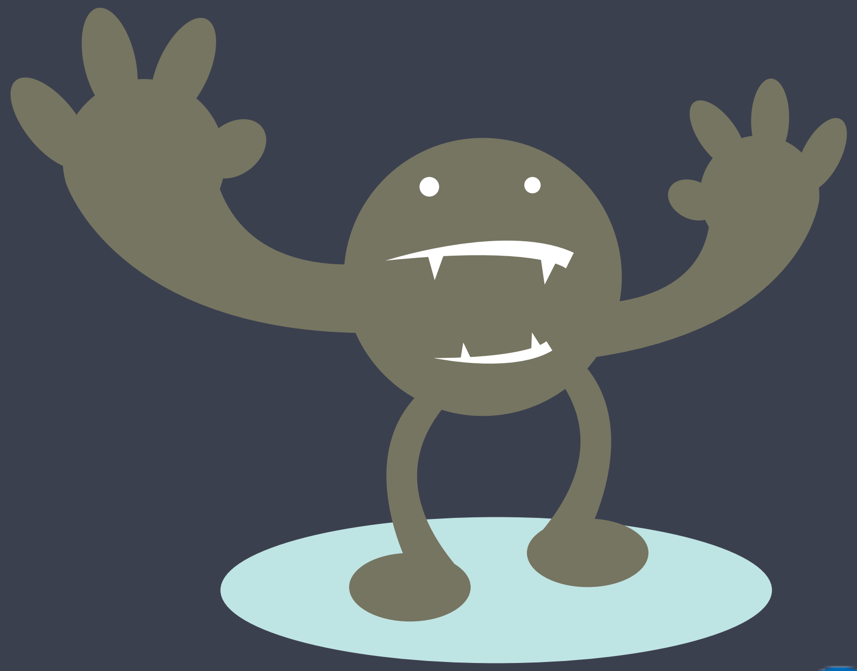


smokeclouds

7

MySQL: Built To Perform

- Ditch the query cache. It doesn‘t help you like you think.
- Become best friends with the various table engines - InnoDB, MyISAM, NBD, etc.
- Watch the process list. Hawk like.
- Know which queries hurt..



smokeclouds 8

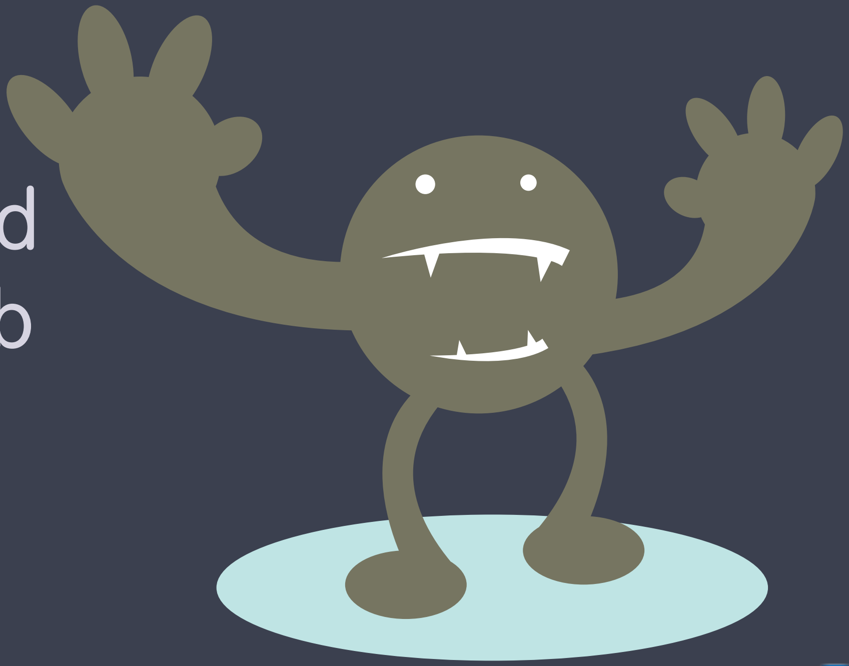
MySQL: Built To Perform

Some key variables to tune:

max_allowed_packet & key_buffer_size
(keeps your index quicksort from chunking)

Care about **wait_timeout**, **net_write_timeout**, **net_read_timeout** and **max_connections** if you use networked db servers

Consider ditching ActiveRecord



back to the map

smokeclouds

Caching: how to fix nearly everything

This really defines 80/20.
There are few scenarios which are truly uncacheable.

Memcached is the defacto tool, and cache_fu is the best way in.

memcache is your friend...

... learn to love it like it loves you. :)

except on jruby

smokeclouds 10

Caching: how to fix nearly everything

be wary of over-reliance on memcache.

Facebook runs over 200gb of memcache - twitter consumes about 40gb.

restarting your cache is slow....

so find ways to mitigate a cold cache.

smokeclouds

Caching: how to fix nearly everything

File I/O lets you down almost all the time.

NFS is not as bad as it used to be.

Consider offloading to s3 - it's not as expensive as you think.

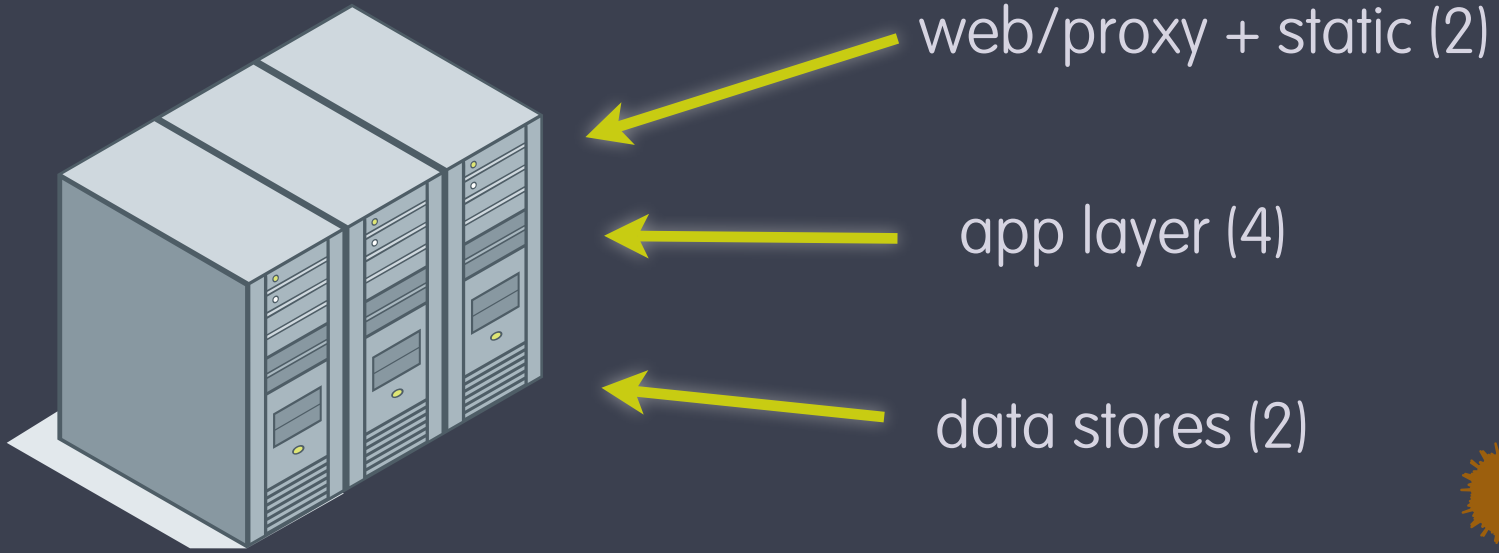
or, if you have money, buy a blade center, SAN or NAS.

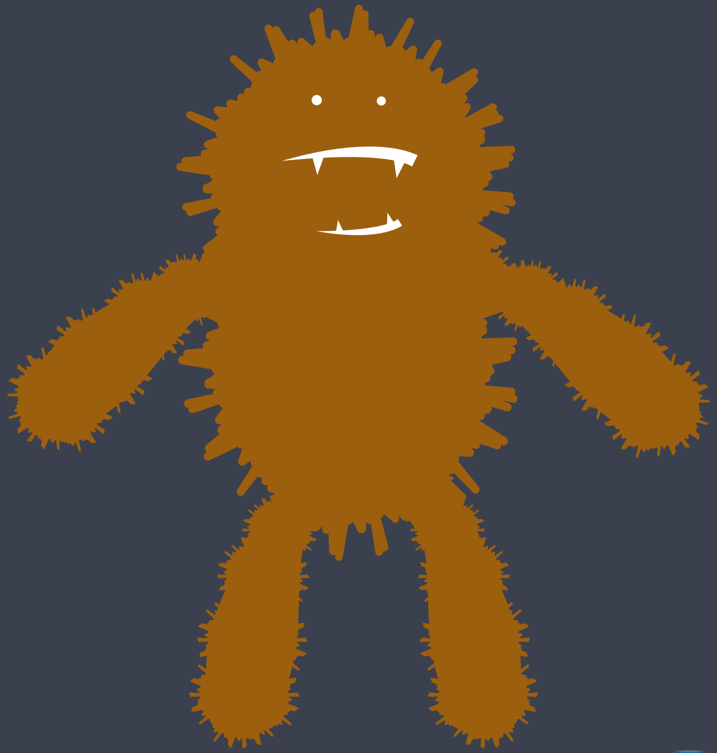
back to the map

smokeclouds

Deployment: get it out there for real

Going alone? That's ok. Here's a rough guide...





web: nginx, thin, rack
app: mongrel, glassfish, swiftiply
db: MySQL, SDB, PostgreSQL

smokeclouds

Deployment: get it out there for real

Engine Yard™

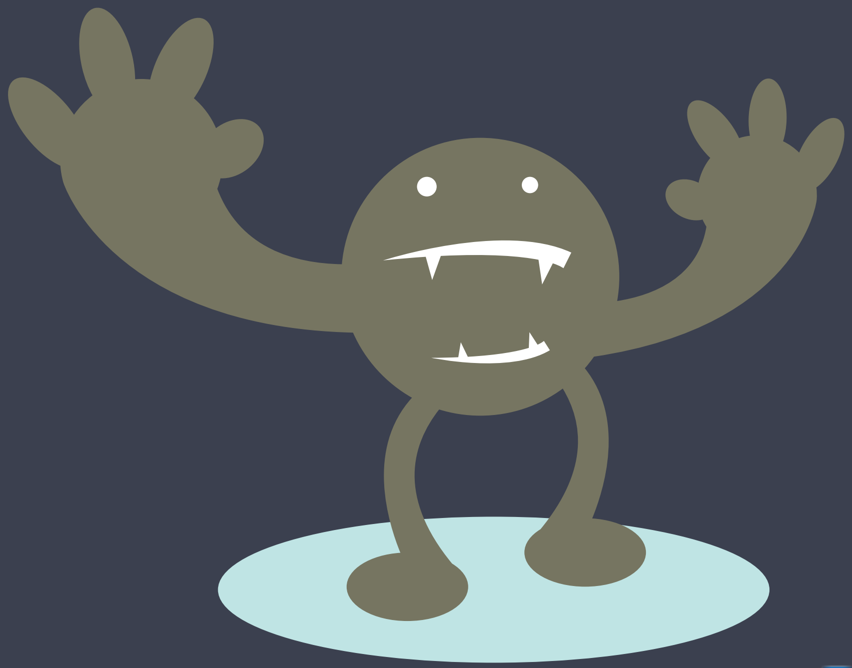
back to the map

smokeclouds

Rails: can be monkey patched

Don’t be afraid to improve it

But follow edge if you do.



smokeclouds 15

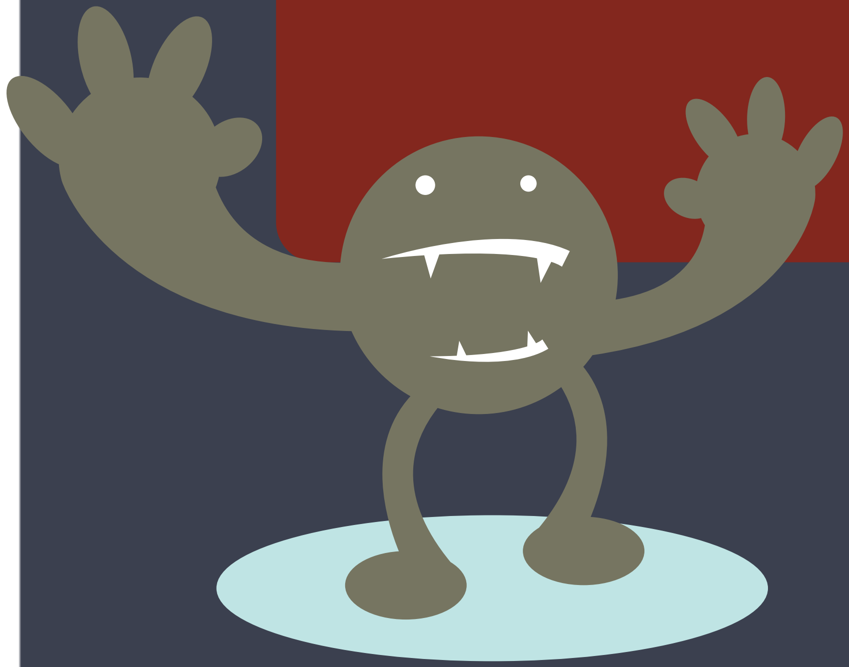
Rails: can be monkey patched

ActiveRecord:

Take only what you need. :select, :limit, :offset

Eagerly associate with :include => :association

@var ||= Model.find(...)



smokeclouds 16

Rails: can be monkey patched

Views & Controllers:

Use HTML helpers sparingly (url_for, etc)

Consider template optimization

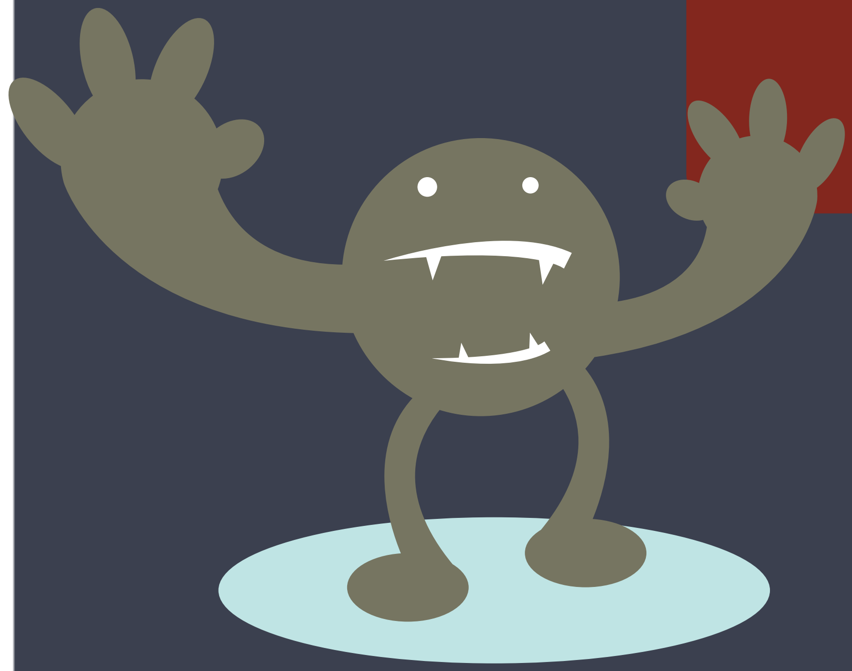
Know how your rendering engine works.

smokeclouds

Ruby: can be monkey patched too

Meta programming monkey patching

```
Class String
    def new_behavior
        ...
    end
end
```



Don't forget constants and class variables

back to the map

smokeclouds

Testing: verify your performance

Be sure that your performance is what you think it is.

Measure: response times, concurrent reqs & server load.

Look for a tight standard deviation in requests.

Keep asking users: "What's Slow?"

tools: httperf, ab, flood, your brain.

back to the map

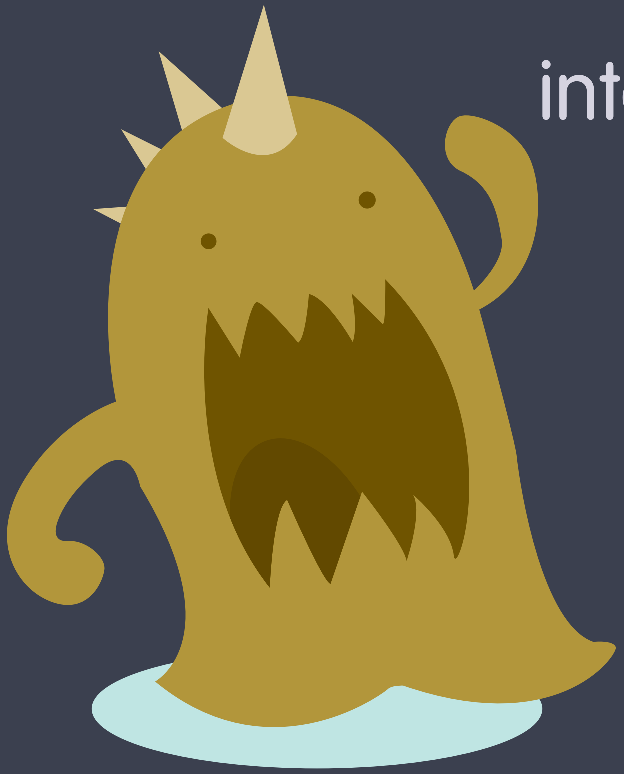
smokeclouds

The end: thanks for not sleeping

need more? email james@smokeclouds.com

will code for trips to cool places. :)

interactive deck at http://media.imaj.es/scale/



colophon:
font: vag rounded, metabook
art: iStockPhoto, ryan putnam + spotblind

back to the map

smokeclouds