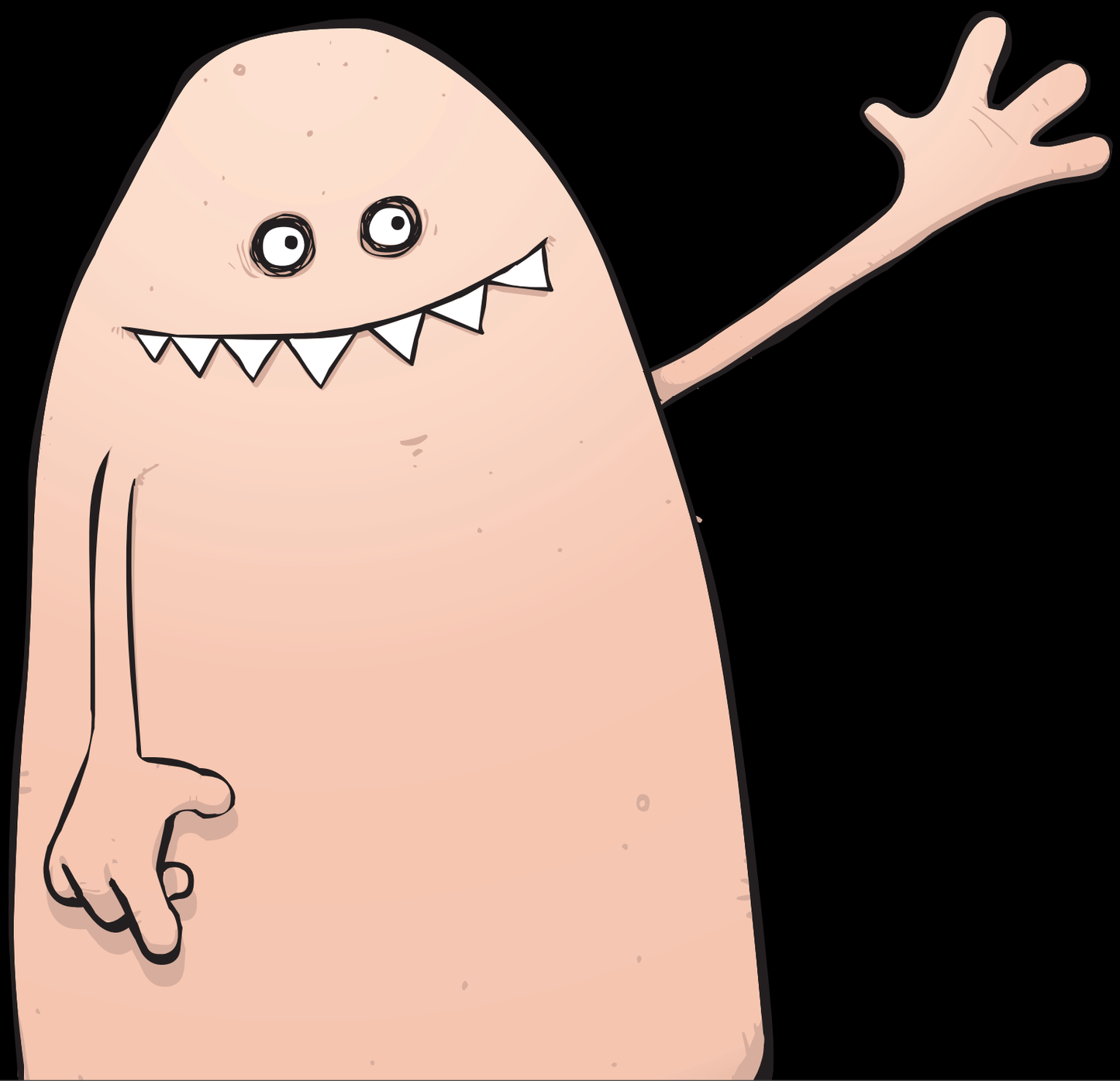
An Infrastructure in Transition

Joe Stump, Lead Architect, Digg

Introductions



✓ 35,000,000 uniques
✓ 3,500,000 users
✓ 15,000 requests / sec
✓ Hundreds of servers

“Web 2.0 sucks (for scaling).”

Joe Stump

What’s Scaling?

What’s Scaling?

Specialization

What’s Scaling?

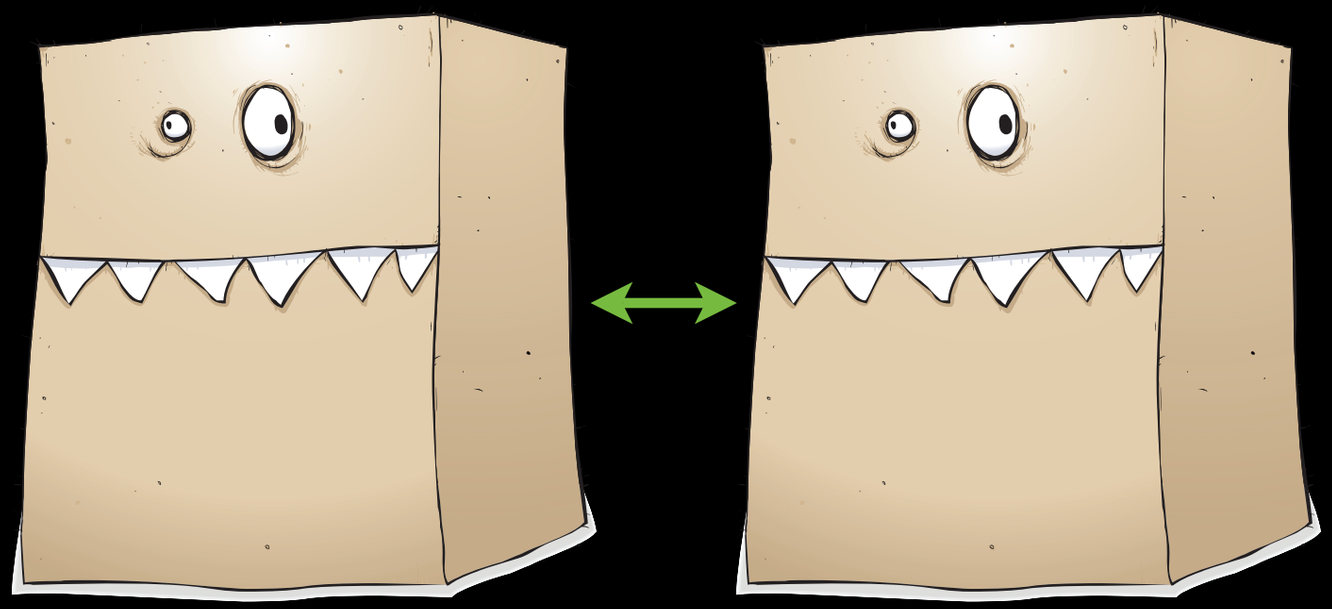
Severe Hair Loss

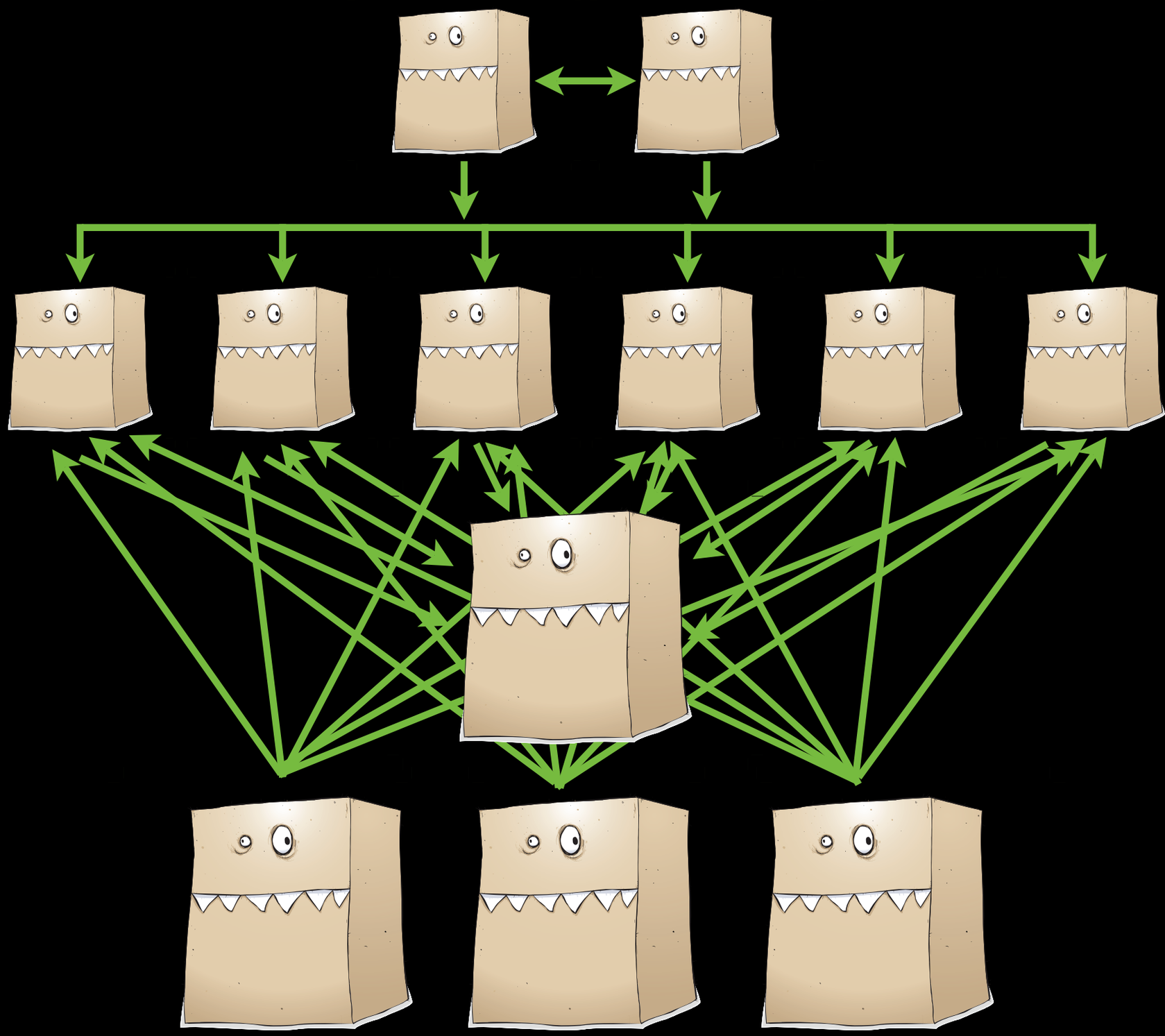
What’s Performance?

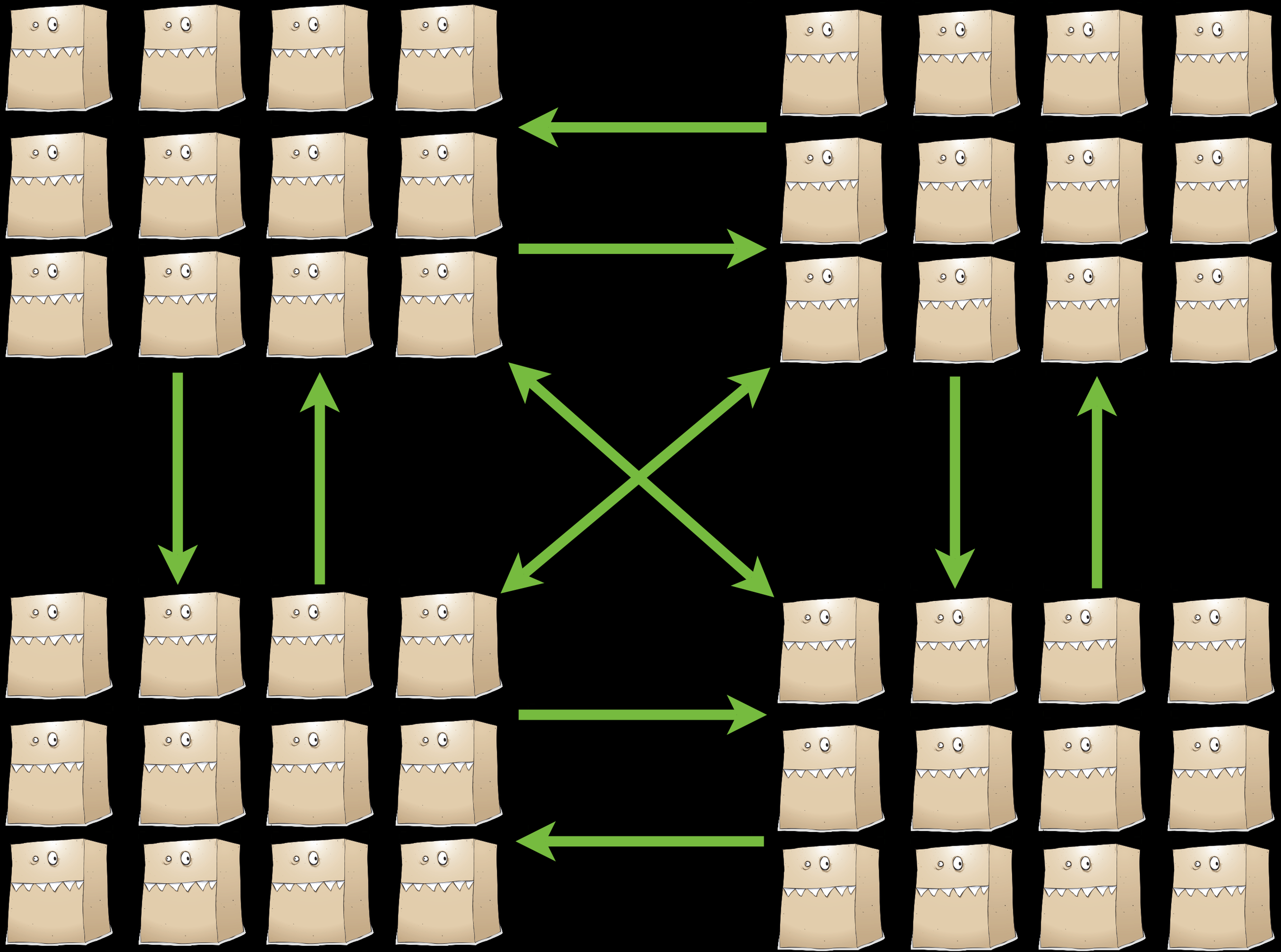
What’s Performance?

Who cares?

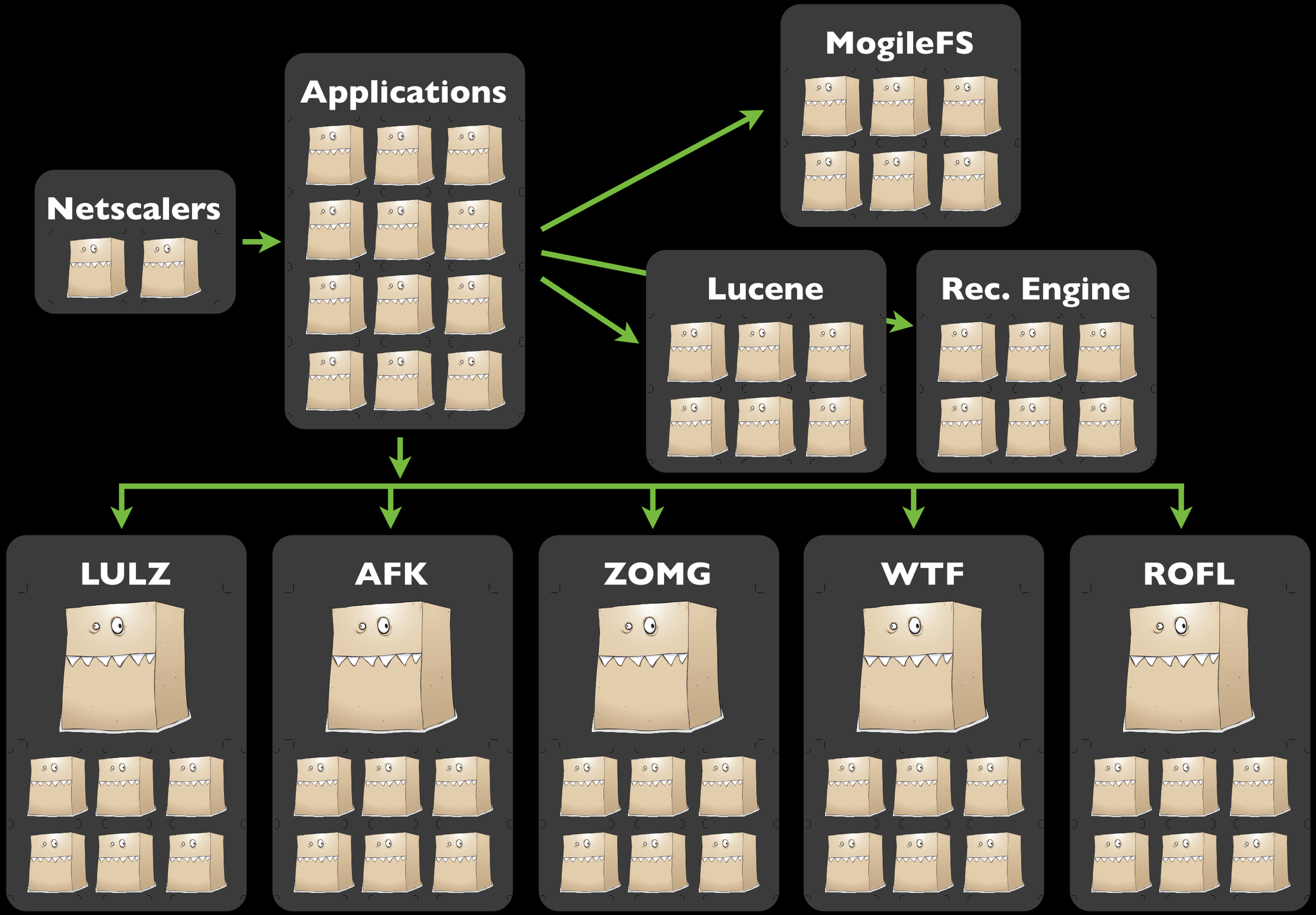
4 Stages of Scaling







As it stands ...


Netscalers
Applications
MogileFS
Lucene
Rec. Engine
LULZ
AFK
ZOMG
WTF
ROFL

Building Blocks

- MogileFS
  - 9 nodes
  - 2.8TB of files
- Gearman
  - Each application server
  - 400,000 jobs / day
- Memcached
  - 25 nodes
  - 2GB / node

Moving forward ...

Netscalers

Messaging

Netscalers

Applications

Applications

Services

Services

MogileFS

IDDB

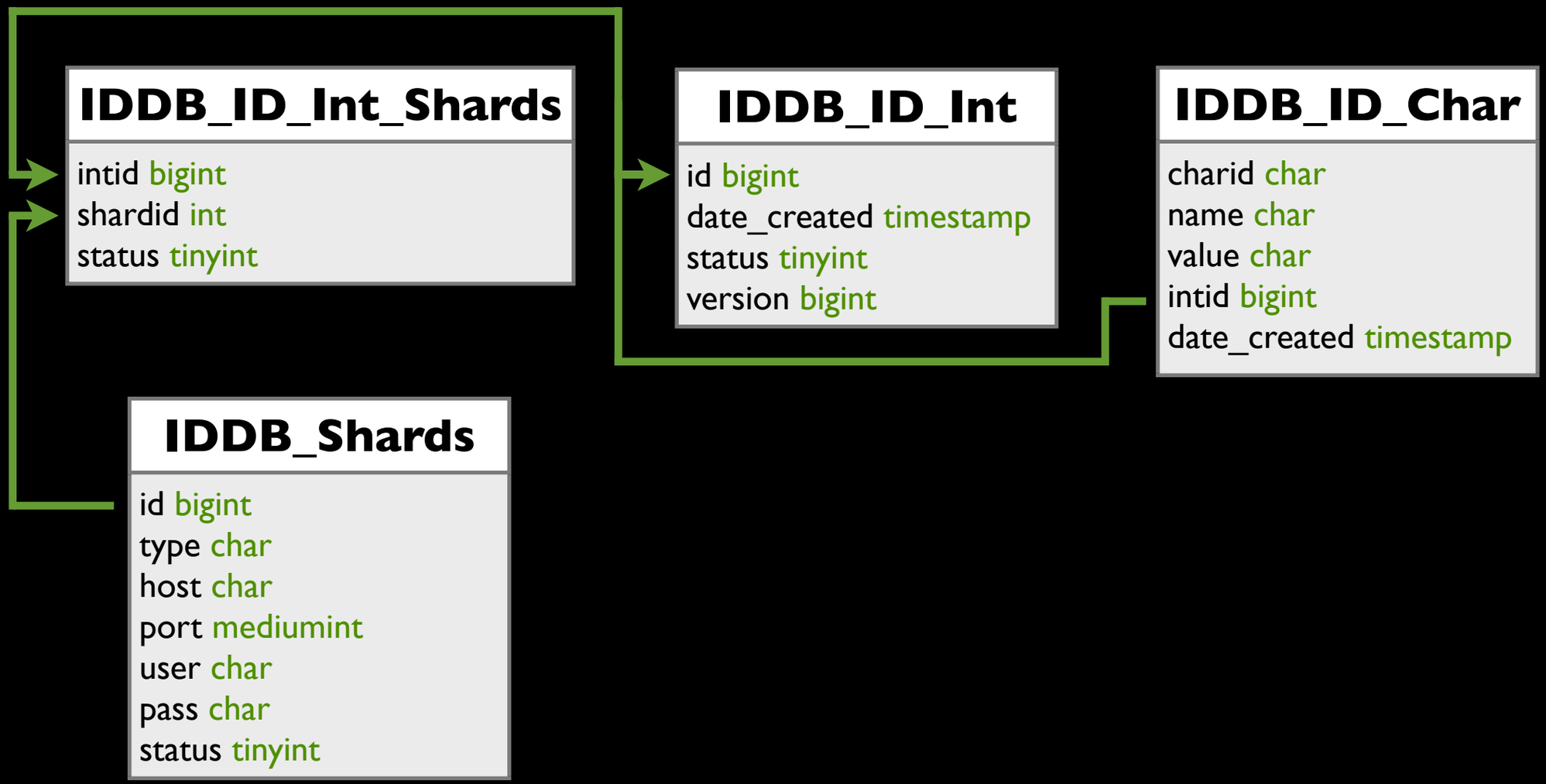
Lucene

Rec. Engine

IDDB

IDDB

✓ Elastic horizontal partitions
✓ Heterogenous partition types
✓ Muti-homed
✓ ID’s live in multiple places
✓ Partitioned result sets


IDDB_ID_Int_Shards
intid bigint
shardid int
status tinyint
IDDB_ID_Int
id bigint
date_created timestamp
status tinyint
version bigint
IDDB_ID_Char
charid char
name char
value char
intid bigint
date_created timestamp
IDDB_Shards
id bigint
type char
host char
port mediumint
user char
pass char
status tinyint

MemcacheDB

✓ Memcached + BDB
✓ 28,000+ writes a second
✓ Persistent key/value storage
✓ Works with Memcached clients

War stories ...

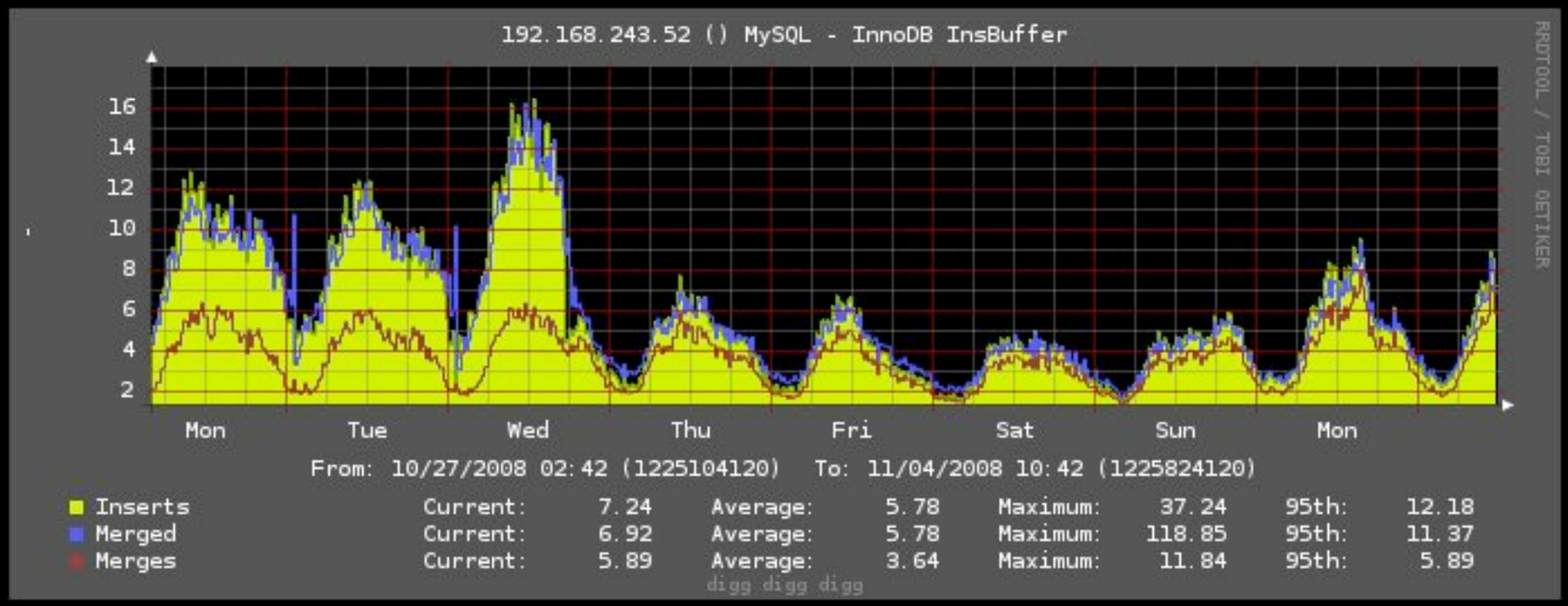
Digg Images

- ✓ 15,000 - 17,000 submissions per day
- ✓ Crawl for images, video embeds, source, other meta data
- ✓ Ran in parallel via Gearman

Green Badges

- ✓ 230,000+ Diggs per day
- ✓ Most active Diggers are also most followed
- ✓ 3,000 writes per second
- ✓ Ran in background via Gearman
- ✓ Eventually consistent

user_ip_views



Digg Comments

- ✓ Switched to explicit caching
- ✓ Intelligently grouped objects in cache
- ✓ Sorting, limiting, etc. done in the application layer
- ✓ 200% to 300% gains in performance

Data Migration

- ✓ Vertical partitioning
- ✓ Migrate in background processes
- ✓ Use the bots
- ✓ Keep track of migration
- ✓ Retry failed migrations automatically

Things to ponder ...

CAP Theorem

Have I ran the numbers?

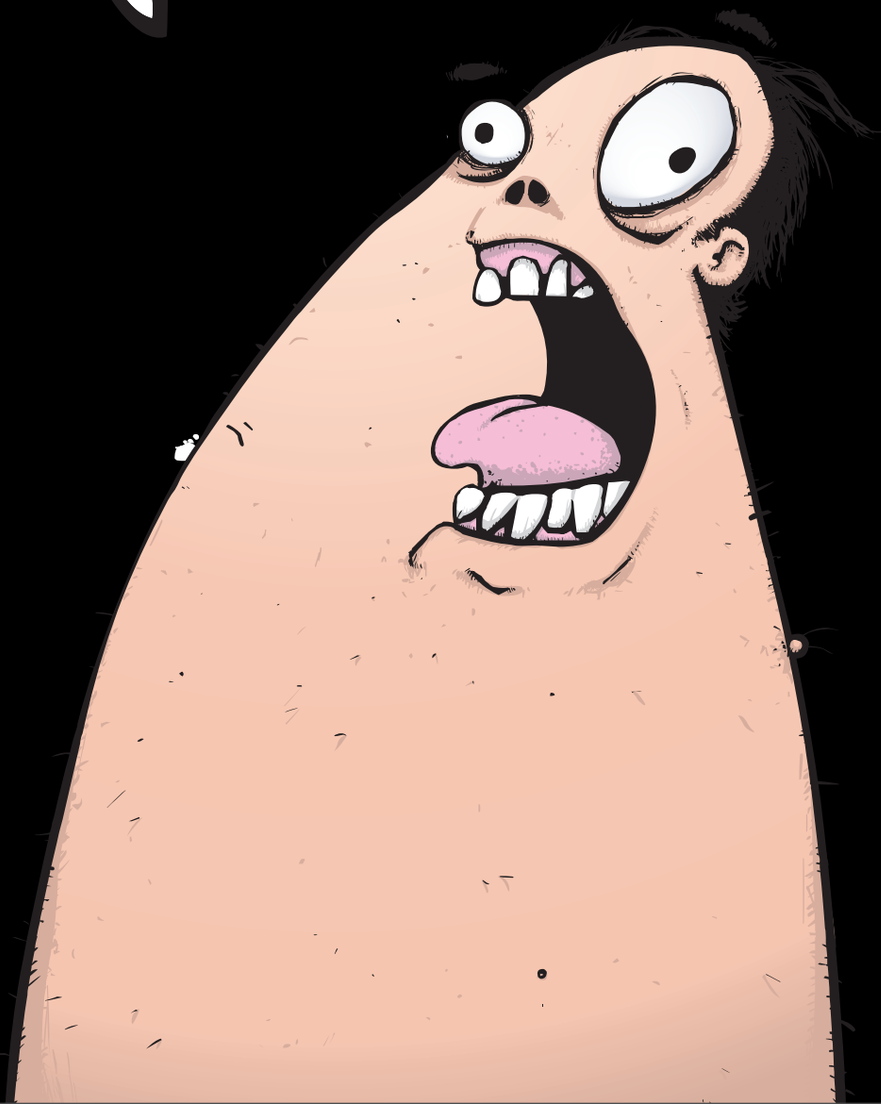
Is MySQL the best solution?

Can I do this later?

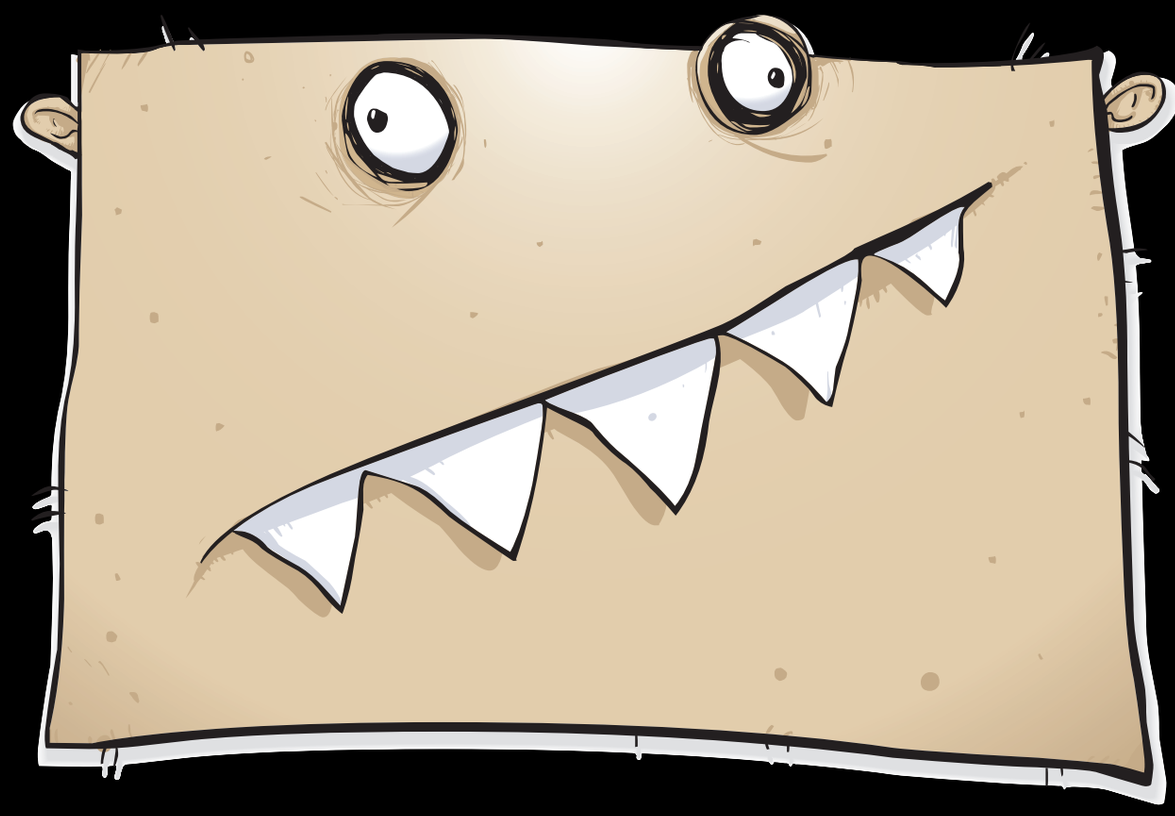
How can I partition this data?

How should I cache this data?

Questions?!



Contact/Flame Me



Joe Stump
joe@digg.com
http://joestump.net