



Beat *the* Devil

towards a Drupal performance benchmark

<http://drupal.org/user/770300>
@rodricels

<http://drupal.org/user/8859>
@perusio

<http://drupal.org/user/39078>
@NITEMAN_es

**We'll kill a lot of sacred
cows**



**the murder weapon
will be**

Occam's razor



**which amounts to the
principle of least effort**

laziness is good

hopefully in the end you'll
be **puzzled**

good things happen to
puzzled people who
obsess

simplest
laziest
easiest

performance is for
everyone



do you **have**
performance **issues?**

A close-up photograph of a complex mechanical system featuring several interlocking metal gears. The gears are made of polished metal, likely steel, and are arranged in a series of concentric or overlapping paths. Some of the gear housing or support structures are painted in bright red and blue, creating a high-contrast visual. The lighting is dramatic, with strong highlights on the metallic surfaces and deep shadows in the gaps between the teeth of the gears.

system thinking

mathematics is
for everyone

request per second
vs
seconds per **request**

MTTR

**Mean Time To
Recovery**

vs

MTBF

**Mean Time Between
Failure**

TELEQUOTE TICKER SERVICE

THE BUNKER-RAMO CORPORATION

| | | | |
|-----------------------|----------------------|------------------|---------------------|
| ACK 109 1/2 66 1/2 | ACY 62 1/2 47 1/2 | AFI 56 1/2 48 | AL 22 1/2 20 1/2 |
| 0 1 6 | 5 6 7 | 5 2 1 | 2 4 6 |
| 0 2 2 | 5 6 7 | 5 2 / | 2 4 7 |
| 0 3 4 | 5 7 4 | 5 2 4 | 2 4 7 |
| 0 2 2 | 5 6 5 | 5 2 / | 2 4 5 |
| 0 3 / | 5 7 3 | | 6 |
| CO 67 1/2 51 1/2 | CRI 35 1/2 22 1/2 | | |
| 6 5 6 | 2 3 5 | | |
| 6 6 1 | 2 3 5 | 4 0 | 4 |
| 6 6 1 | 2 4 / | | 5 3 |
| 6 | 4 | | |

| | |
|-----|------------|
| NI | 61 3/8 |
| TXU | 65 60 1/8 |
| CNG | 60 7/8 |
| WK | 39 3/4 |
| GPU | 25 31 1/2 |
| HC | 20 3/8 |
| CQ | 5 5/5 |
| CRI | 72 1/4 |
| TG | 35 102 1/4 |
| JLP | 240 96 1/4 |

| | |
|-------|-------------|
| MGP | 19 1/4 |
| HYD | 12 1/4 |
| UEX.U | 24 1/4..... |
| NVD | 100 1/2 |
| GKM | 15 3/4 |
| INX | 13 |
| SOD | 159 1/2 |
| FLY | 25 41 |
| HAV | 25 8 1/4 |
| NUM | 40 3/4..... |
| .. | |

| | |
|---------------------|----|
| AS 65 1/2 51 1/2 | AT |
| 6 3 3 | 2 |
| 6 3 3 | 2 |
| 6 3 3 | 2 |
| 6 2 3 | 2 |
| 6 3 2 | 2 |

| | |
|------------------|----|
| EAL 26 1/2 19 | ED |
| 2 5 4 | 8 |
| 2 5 6 | 8 |
| 2 6 3 | 8 |
| 2 5 5 | 8 |
| 2 6 3 | 8 |

| | |
|------------------|---|
| NND 53 43 1/2 | N |
| 4 5 2 | |
| 4 5 4 | |
| 4 6 2 | |
| 4 5 4 | |
| 4 6 | |

| | |
|-------|--|
| AVT | |
| 1 3 5 | |
| GRC | |
| 2 4 | |
| PY | |
| 4 1 | |

do you monitor **live**
system
performance?

live **monitoring** tools



Munin



Cacti

drupal accesslog table
has tons of useful data

beware the logs
logstash / graylog
aggregate them



economics
costs per
request/volume/peek

operational costs
Return On Investment

we're talking about **\$speed**

\$speed = cost_effective_performance



slowness
downtime
operations
costs you money

it's **not** the
tools you have

it's the **use** you
make of them

have you **benchmarked**
your Drupal?

complex is easy and **fragile**

simple is hard and **resilient**

Keep It Simple Stupid

Know your stack
Know your targets
Monitor your performance

find where the **problems** are
don't **fix** things that aren't broken
the worse the first
one change a time
&
keep a **log** of your actions

dissecting

frontend vs backend

backend

static vs dynamic

dynamic

processing vs data gather

beyond development & **operations**

good developers are those that take
all aspects in consideration

test systems

keep everything as close as
possible to **production**

test systems

**take advantage of
configuration management
to reproduce your live
infrastructure**

test systems

some software has its
performance directly
tied to the number of **cores**
and/or
the amount of **RAM**

sacred cow of **frontend**



CDN (expiration logic)
cloud servers

frontend vs backend

number of req. / parallelization

blocking events

data side: **DNS** resolution

data side: **download** time

data side: **size** / **weight**

order matters



frontend vs backend tools

Firebug

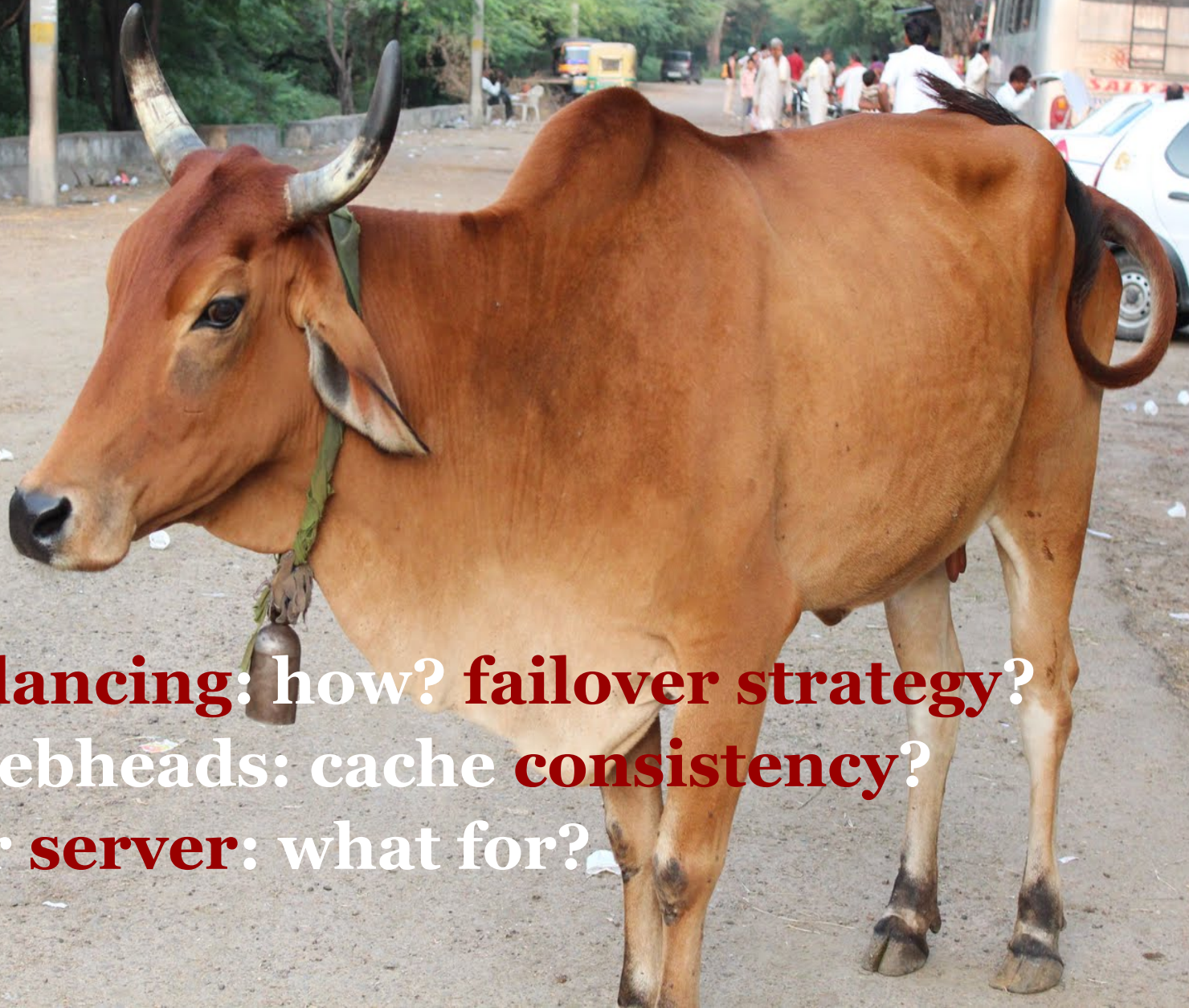
Chrome tools

Google Speed Tracer

Yslow

Web services [*] [*] [*] [*]

sacred cow of **backend**



load **balancing**: how? **failover** strategy?
more webheads: cache **consistency**?
another **server**: what for?

backend: static vs dynamic

anonymous vs registered

cacheable vs non cacheable

HTTP benchmarking tools

ab

Jmeter

Httpperf

wrk

Gatling

TCP copy

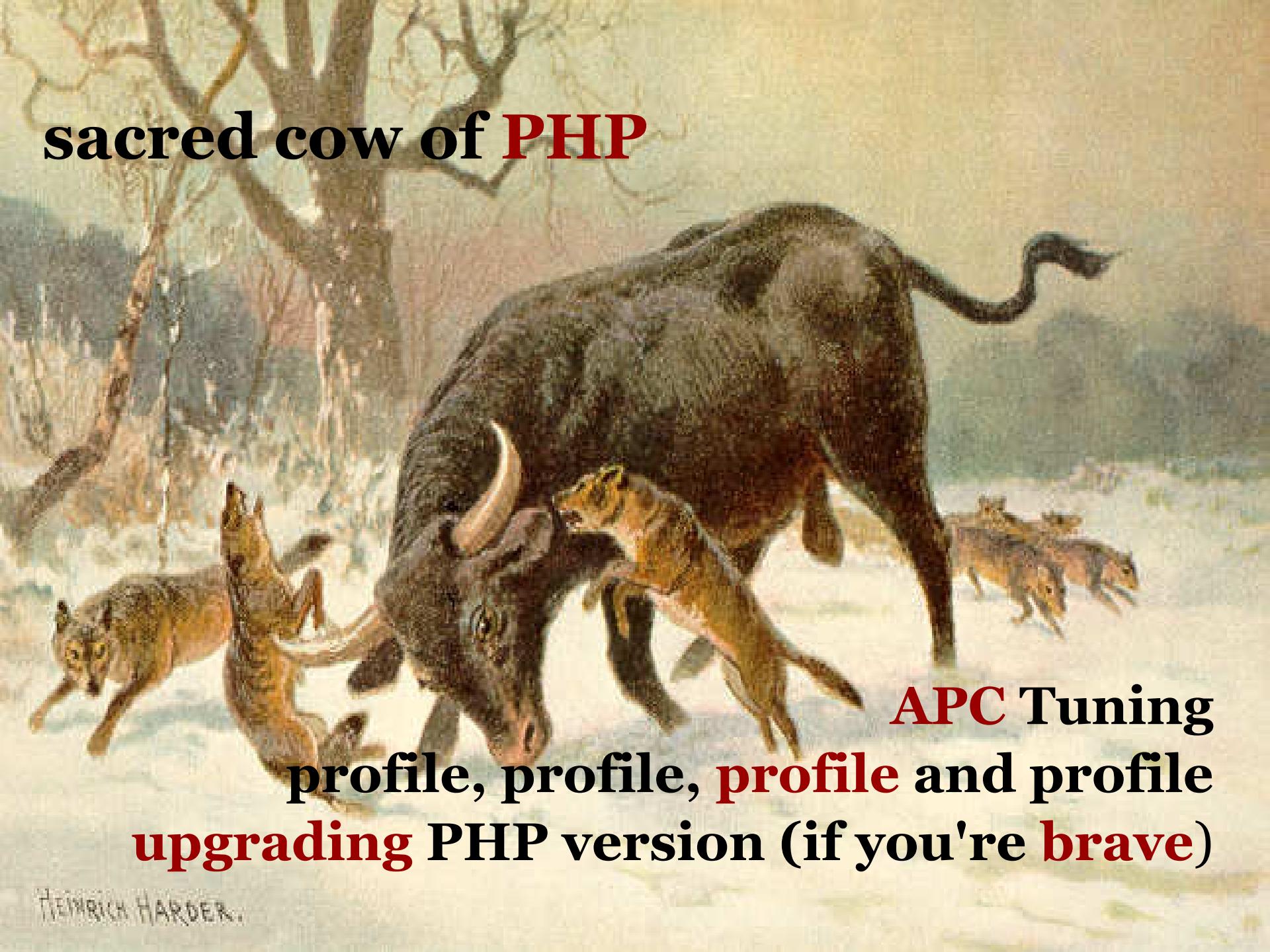
sacred cow of PHP

HipHop (hacked libs)

PHP Extensions (C / PECL)



sacred cow of **PHP**



APC Tuning
profile, profile, **profile** and profile
upgrading PHP version (if you're **brave**)

HEINRICH HARDER.

dynamic processing vs data gather

locks
difficult to dissect
devel

benchmarking / profiling **PHP**

Xdebug

Webgrind

Xhprof

**don't even try without an
opcode cache
& remember the hard disk**



sacred cow of **databases**

NoSQL
denormalization
sharding

A close-up, high-contrast photograph of a man's face, looking directly at the camera with a serious, intense expression. His eyes are slightly squinted, and his skin appears tanned. The background is dark and out of focus.

sacred cow of **databases**

NoSQL

=> helps with **writes**

denormalization

=> helps with **HUGE** DBs

sharding

=> helps with **Big** DBs

MySQL Tune & Benchmarking tools

`https://tools.percona.com/`

Percona toolkit  (formerly `maat-kit`)

tcpdump + Percona toolkit

Logs are vital!

Every engine has strong & weakness

Better on bare metal!

root hardware causes

network & DNS play a role

static content depends mainly on **I/O**

dynamic content (**php**) depends on **CPU**

database server mainly depends on **RAM**

extra balls:

Drupal known issues (DB): Watchdog, sessions, accesslog... history...

Please do stress, load & stability tests regularly in your live system

**This is *our way*, what is
yours?**

(no cows were harmed in the making of this
presentation)

¿Questions?

Pablo Picasso [speaking of computers]:
"But they are useless. They can only give you
answers."

DOES NOT SIMPLY



BENCHMARK DRUPAL