Speed up your data caches with Heisencache

Frédéric G. Marand

#drupaldevdays

wunderkraut • Smile • adyax • Kaliop • MD SYSTEMS • actency • TRAINED PEOPLE • Amazee Labs

CONTEXT

MEASURING

IMPLEMENTATION

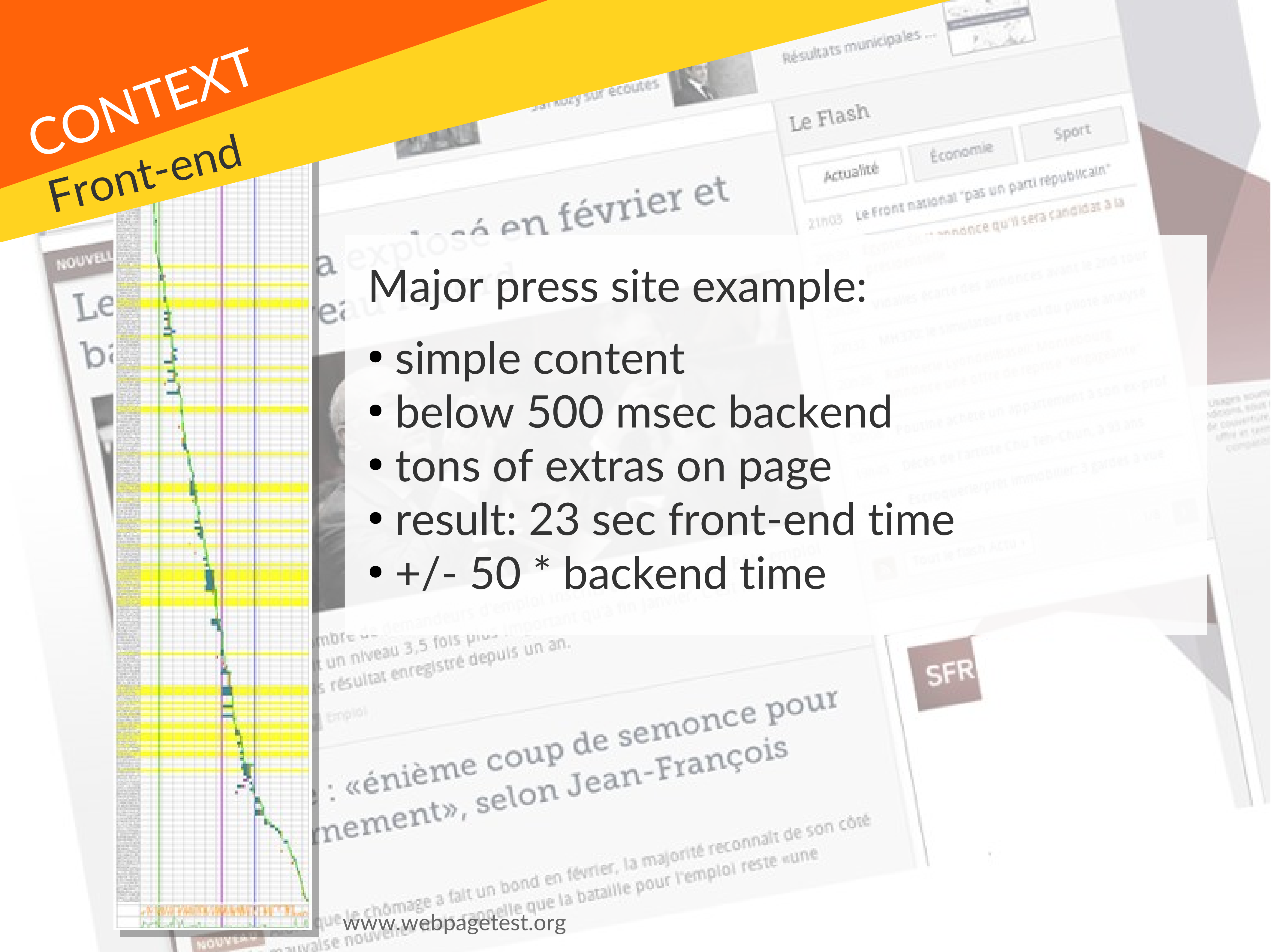
RESULTS

Perception

- Front-end dominates
- Downloads start after the page is served
- Not all sites are equal when it comes to the back/front performance ratio

CONTEXT

Front-end



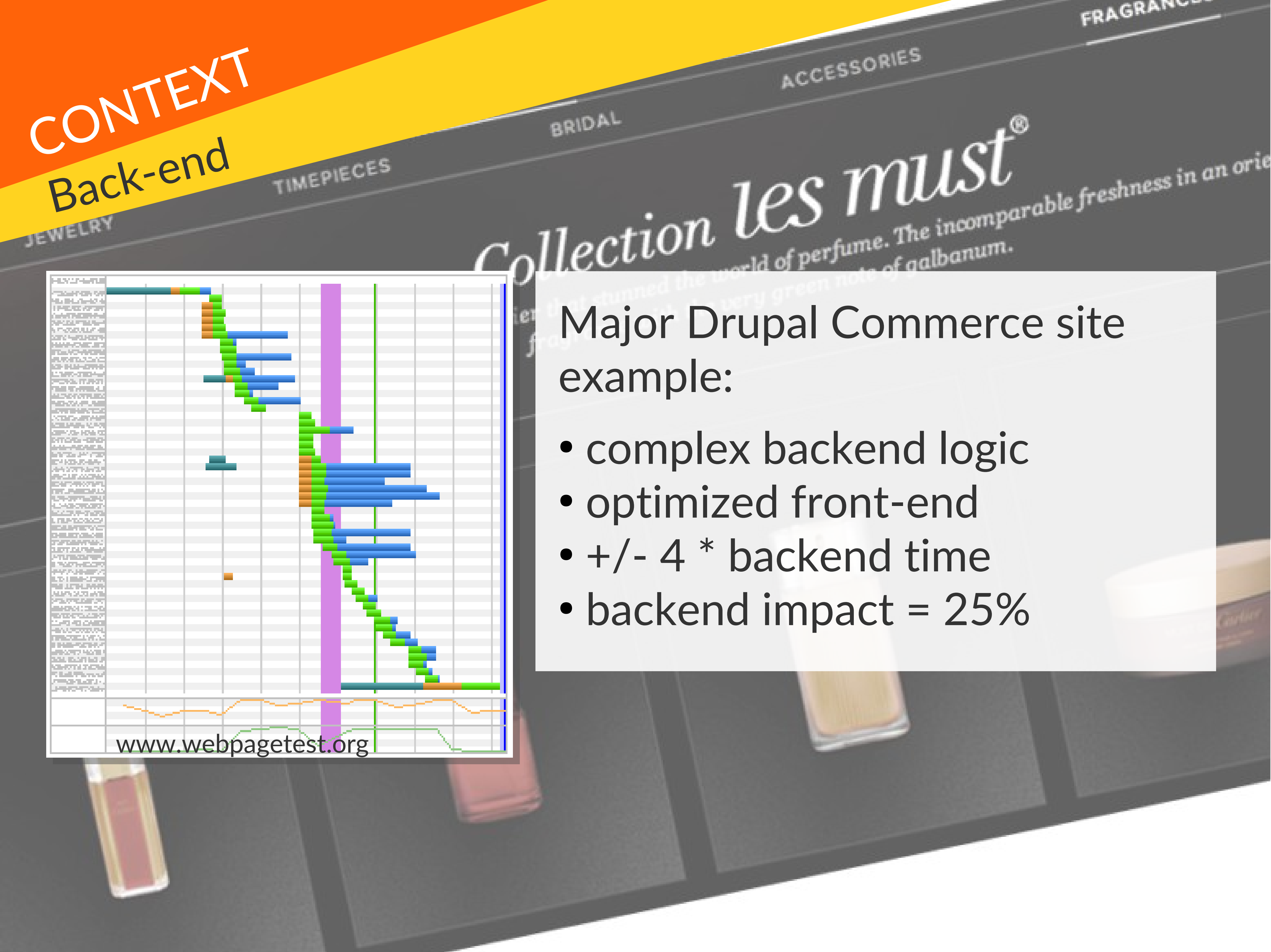
Major press site example:

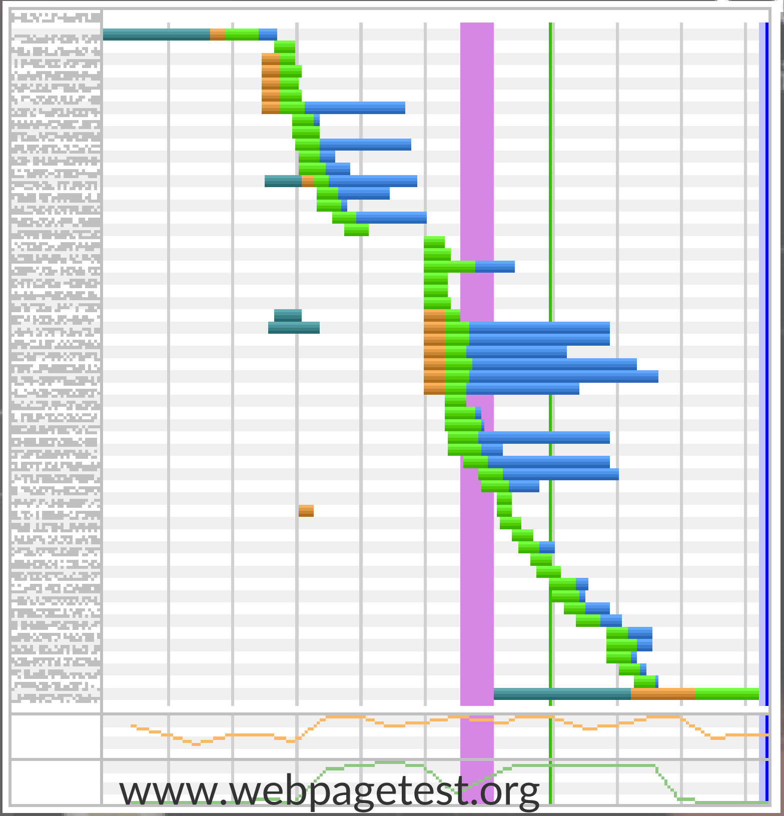
- simple content
- below 500 msec backend
- tons of extras on page
- result: 23 sec front-end time
- +/- 50 * backend time

www.webpagetest.org

CONTEXT

Back-end





Major Drupal Commerce site example:

- complex backend logic
- optimized front-end
- +/- 4 * backend time
- backend impact = 25%

Battle plan

- Level 1: page caching
  - Opcode cache
  - Varnish, CDN
- Level 2: storage tuning
  - MySQL slow queries
  - MongoDB, Redis, …
- Level 3: cache tuning
  - On we go…

CONTEXT

LESS DB QUERIES MEANS **DATA CACHING**

Drupal *needs* caching

Cache is a **good** thing

«Cache is king»

Steve Souders, author of *High-Performance web sites*
http://fr.slideshare.net/souders/cache-is-king

Too much of a good thing is WONDERFUL

Photo by: Alan Light

CONTEXT

(Not *that* kind of good thing)

CONTEXT

CACHING *CAN* BE BAD

- Memcached speed +/- like good MySQL
- Can be worse: clustering network issues
- A miss costs more than uncached work

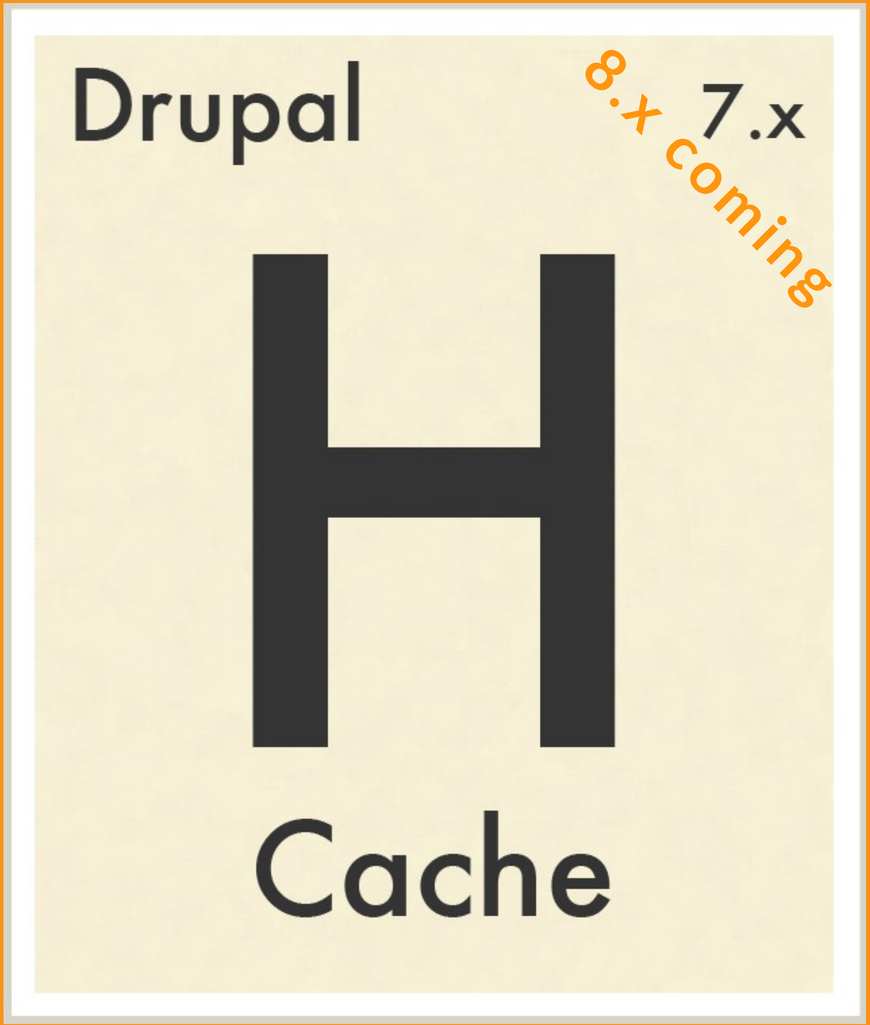
CONTEXT

HOW DO YOU KNOW WHEN YOU'VE HAD ENOUGH?

MEASURE!

QUANTUM COPS
-The Uncertainty Principality-
Do you have any idea how fast you were going back there?
No.
Crap. Me neither.
SaintGasoline.com

Introducing HEISENCACHE



CONTEXT

MEASURING

IMPLEMENTATION

RESULTS

MEASURING

WHY *HEISEN*CACHE?

MEASURING

$$\Delta p \Delta x \geq \frac{\hbar}{2}$$

- Reduce uncertainty
- Minimize observer impact
- Don't push the analogy too far

MEASURING

THE COST OF MEASUREMENT

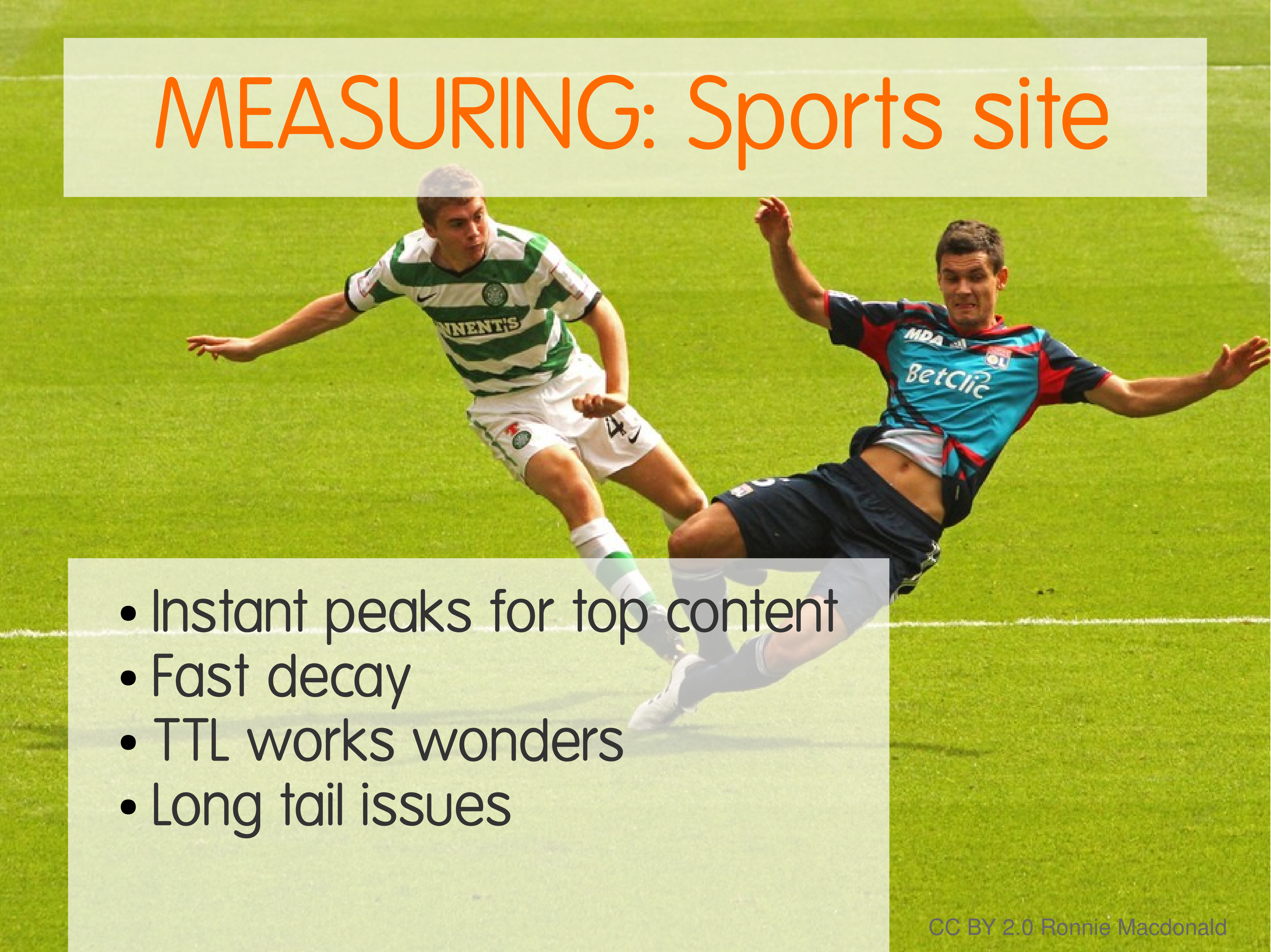
- Observer code runs within Drupal
- Needs to be invoked → more CPU
- Needs to store data → more I/O

MEASURING

CACHING BEHAVIOUR

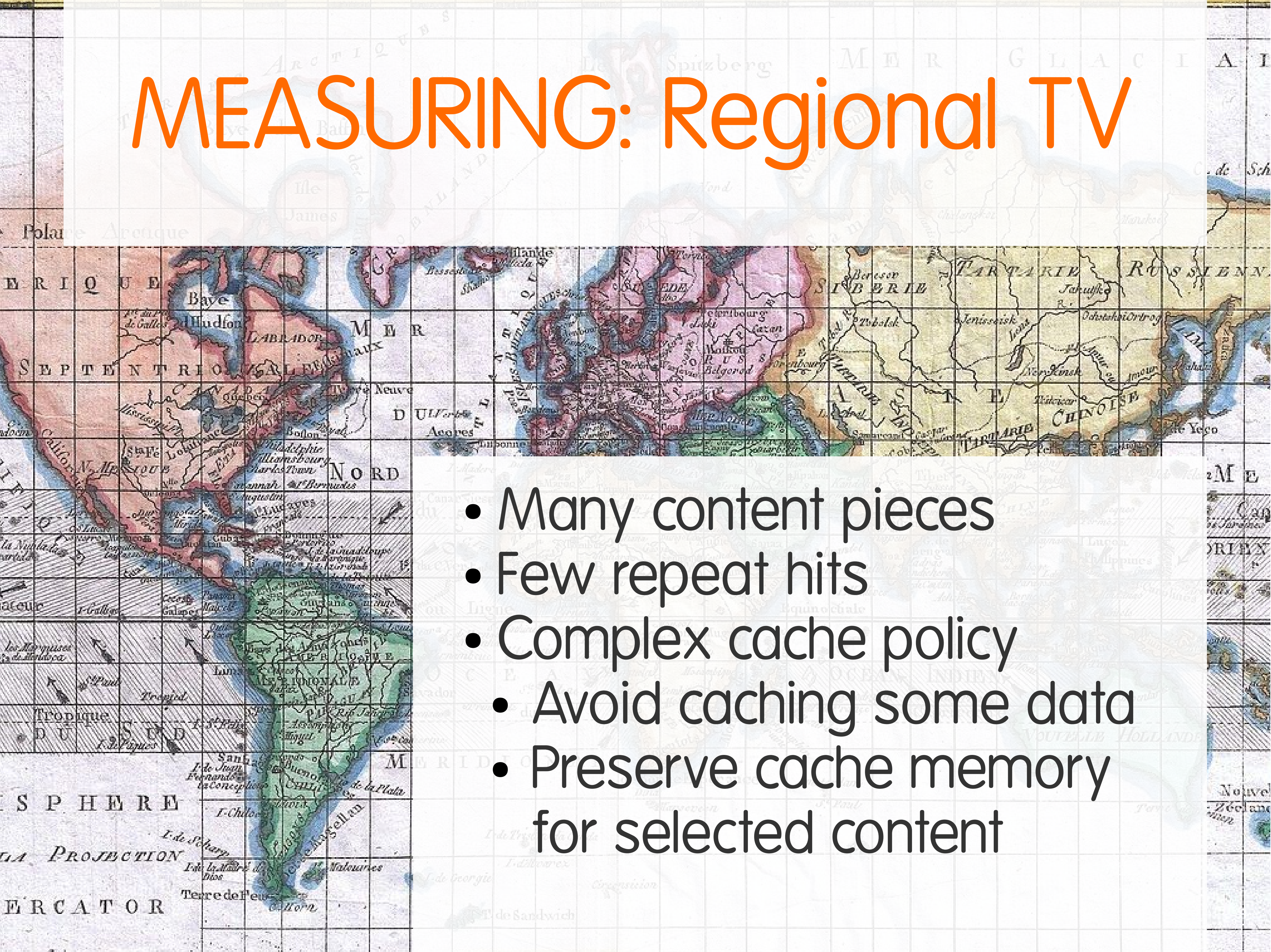
vs.

LOAD PROFILE



MEASURING: Sports site

- Instant peaks for top content
- Fast decay
- TTL works wonders
- Long tail issues

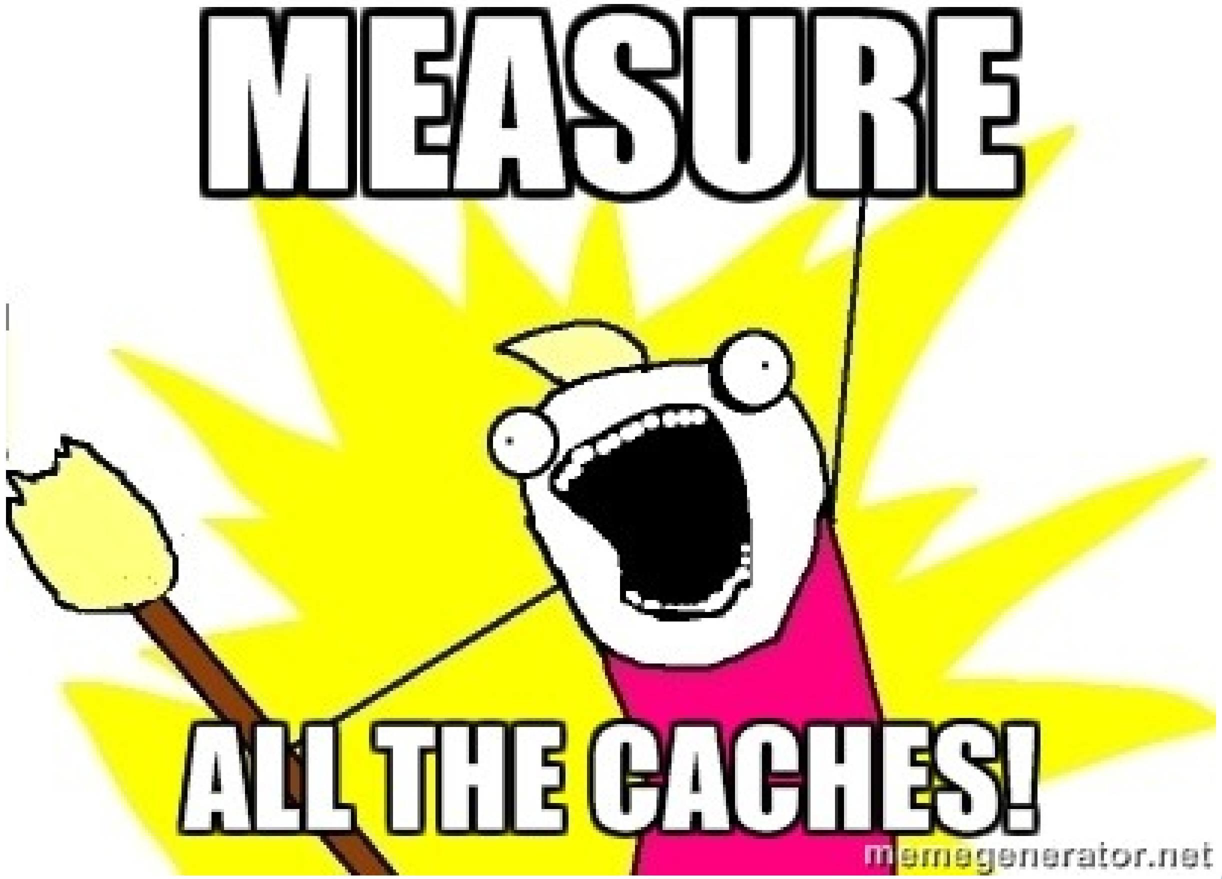


MEASURING: Regional TV

- Many content pieces
- Few repeat hits
- Complex cache policy
  - Avoid caching some data
  - Preserve cache memory for selected content

MEASURING

OBSERVE CACHE IN



MEASURING in production

- Precision vs. velocity?
- Performance module
- Cannot write to the DB in real-time
- Cached pages
- Observing Drupal bootstrap

CONTEXT

MEASURING

IMPLEMENTATION

RESULTS

IMPLEMENTATION: early hits

DRUPAL 7 BOOT SEQUENCE

- Page cycle: index.php → drupal_bootstrap
  - Phases: Configuration, Page Cache, DB, Variables, Session, Page Header, Language, Full
  - Cache handlers declaration: settings.php
  - Exotic early hits : sites.php, settings.php, drupal_settings_initialize()
  - Common early hits : _drupal_bootstrap_page_cache
- Need to work before DB and module system

IMPLEMENTATION: early hits

SOLUTION

D7 : Use a standalone event system

D8 : Use the SF2 EventDispatcher

IMPLEMENTATION: operation

IDEAS

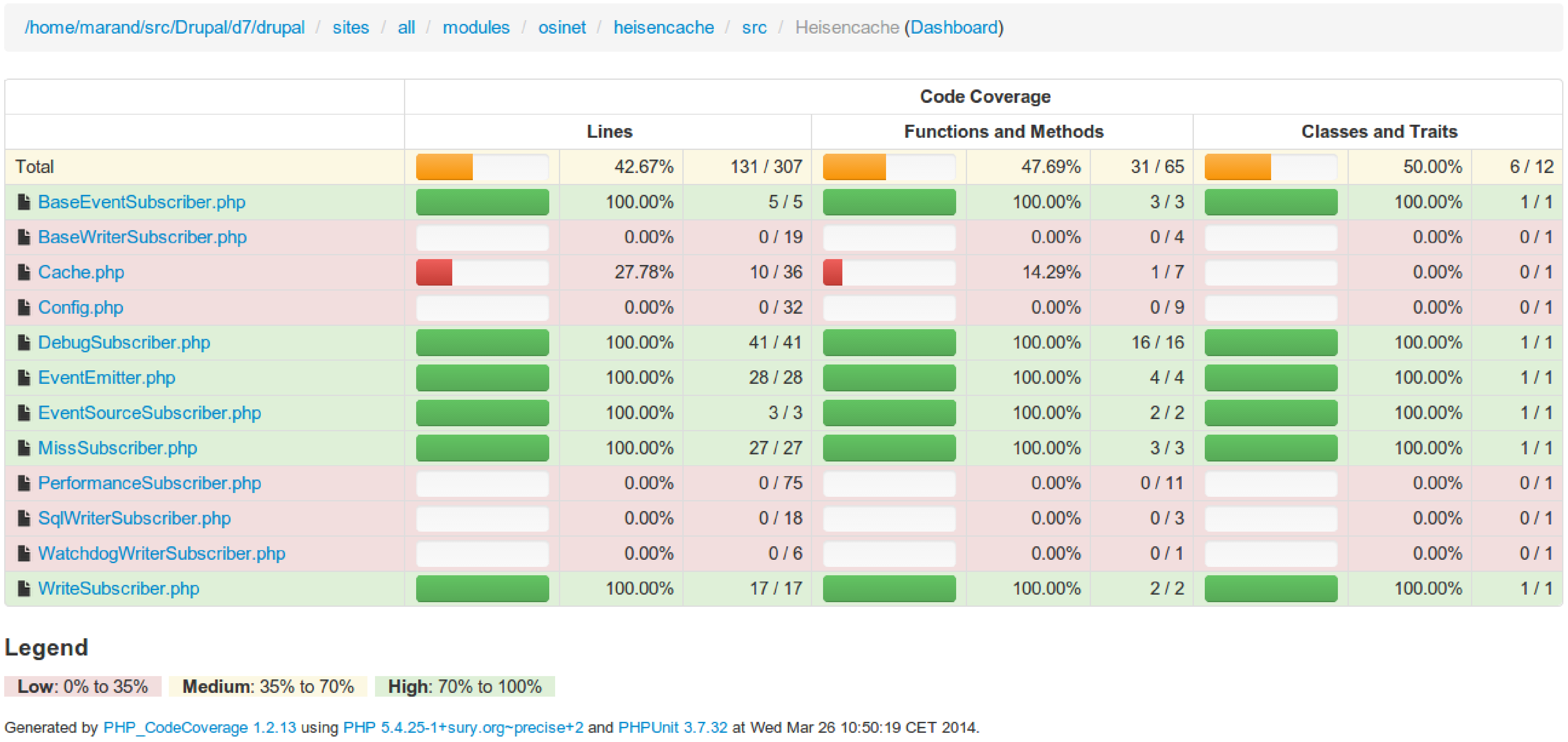
- Doctrine, NodeJS and Symfony event systems
- Use dependency injection, but no DIC (on D7)
- Use the standard DIC and event system (on D8)

IMPLEMENTATION: operation

BONUS

- D7 : Composer for deployment
  - Before hooks, so no composer_manager
- D8 : normal service, easy to build
- Easy unit testing → decent code coverage

IMPLEMENTATION: coverage



/home/marand/src/Drupal/d7/drupal / sites / all / modules / osinet / heisencache / src / Heisencache (Dashboard)

	Code Coverage								
	Lines			Functions and Methods			Classes and Traits		
Total		42.67%	131 / 307		47.69%	31 / 65		50.00%	6 / 12
BaseEventSubscriber.php		100.00%	5 / 5		100.00%	3 / 3		100.00%	1 / 1
BaseWriterSubscriber.php		0.00%	0 / 19		0.00%	0 / 4		0.00%	0 / 1
Cache.php		27.78%	10 / 36		14.29%	1 / 7		0.00%	0 / 1
Config.php		0.00%	0 / 32		0.00%	0 / 9		0.00%	0 / 1
DebugSubscriber.php		100.00%	41 / 41		100.00%	16 / 16		100.00%	1 / 1
EventEmitter.php		100.00%	28 / 28		100.00%	4 / 4		100.00%	1 / 1
EventSourceSubscriber.php		100.00%	3 / 3		100.00%	2 / 2		100.00%	1 / 1
MissSubscriber.php		100.00%	27 / 27		100.00%	3 / 3		100.00%	1 / 1
PerformanceSubscriber.php		0.00%	0 / 75		0.00%	0 / 11		0.00%	0 / 1
SqlWriterSubscriber.php		0.00%	0 / 18		0.00%	0 / 3		0.00%	0 / 1
WatchdogWriterSubscriber.php		0.00%	0 / 6		0.00%	0 / 1		0.00%	0 / 1
WriteSubscriber.php		100.00%	17 / 17		100.00%	2 / 2		100.00%	1 / 1

Legend

Low: 0% to 35% **Medium**: 35% to 70% **High**: 70% to 100%

Generated by PHP_CodeCoverage 1.2.13 using PHP 5.4.25-1+sury.org~precise+2 and PHPUnit 3.7.32 at Wed Mar 26 10:50:19 CET 2014.

IMPLEMENTATION
cache (factory, backend) driver

DECORATOR PATTERN

- Original Inspiration: authcache
- Read existing cache configuration
- D7 : wrap settings, claim to be the sole cache provider
- D8 : decorate CacheFactory, CacheBackends
- Handle requests per the original configuration, but enhance the service

IMPLEMENTATION: events

EMIT EVENTS AROUND ALL OPERATIONS

- Configuration
- Initialization
- Cache operations
- ...and page termination

IMPLEMENTATION: late hits

CATCHING LATE HITS 1/2

- D7 hook_exit ? Lots of code after that

  @see drupal_page_footer()
- D7 Catching page caching ?
  - not triggered on AJAX callbacks @see ajax_deliver()
  - poormancron can run lots of code after that

IMPLEMENTATION: late hits

CATCHING LATE HITS 2 / 2

- D7/D8 Catching the session commit?
  - Only if a session was started
  - Dirty interactions
  - Session regeneration
- D8 : kernel.terminate, kernel.response ?
- => Shutdown function stack

IMPLEMENTATION: storing data

NEEDS

- minimize I/O load
  → aim for #writes <= 1
- «Fingers crossed»-inspired strategy
  - Keep data in memory during the page lifecycle
  - Write it at end of page

IMPLEMENTATION: storing data

CHALLENGES

- Writing after the last possible cache operation
- Write while classes are still available
- D7 : Passing information within an event-oriented procedural code base

IMPLEMENTATION: D7 events

EventEmitter (à la Node)

- narrowcast events created by sources
- subscribers add/remove events on the fly
  - can further tighten narrowcasting

IMPLEMENTATION: D7 events

EventSubscriberInterface

- Doctrine/Symfony
- + (add | remove)Event()

EventSourceInterface

- Define events a source can emit
- Base source of events: Cache API

IMPLEMENTATION: events

D7/D8 SYNTHETIC EVENTS

- API Limitations: post-operation cache events do not get the operation settings
- Enable immediate event reconciliation
- Event susbcribers can also be sources

IMPLEMENTATION: events

D7/D8 SYNTHETIC EVENT EXAMPLES

- MissSubscriber: *miss* info for Cache::get()
- PerformanceSubscriber: *timing* info for all ops
- WriteSubscriber: single event for *write* and *delete* ops

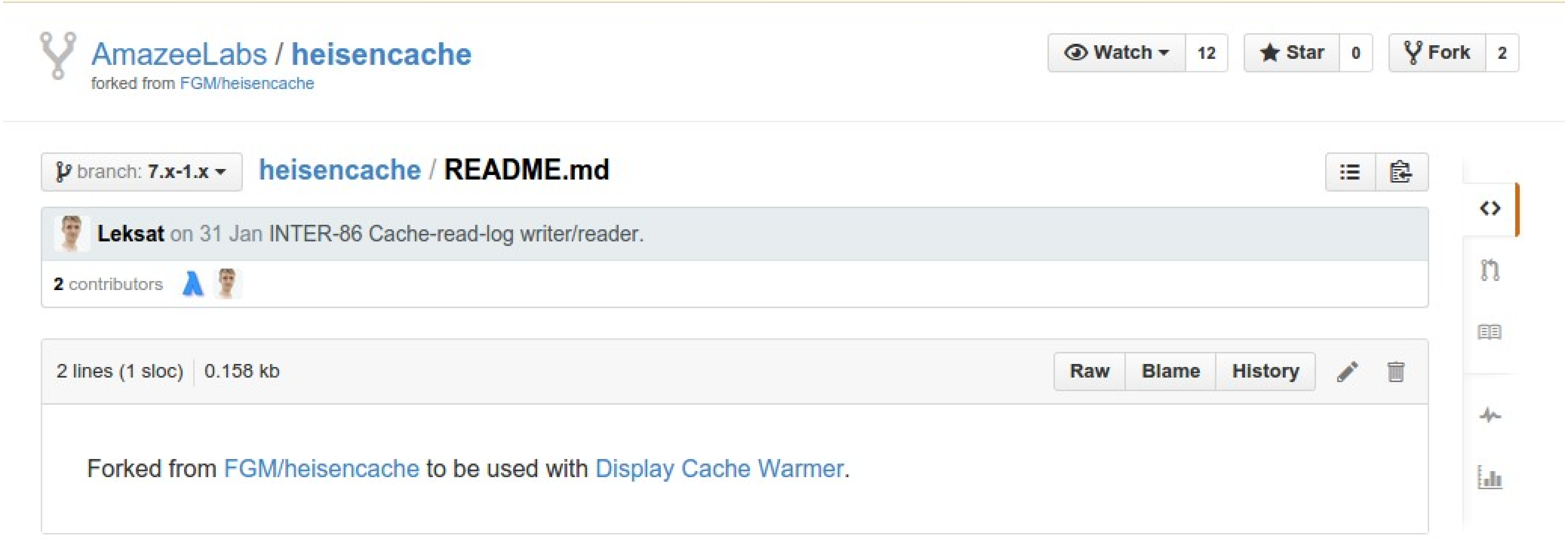
IMPLEMENTATION: setup

`settings.heisencache.inc`
D7 CONFIGURATION

- Retrieve the EventEmitter from Config instance
- Create EventSubscriber instances as chosen
- Register them on chosen events
- DebugSubscriber listen to **all** events
  - not in production !
- Don't forget to include a WriterSubscriber
- D8 : customize development.services.yml

IMPLEMENTATION: extending

HEISENCACHE IS A CODER TOOL


AmazeeLabs / heisencache
forked from FGM/heisencache

heisencache / README.md

Leksat on 31 Jan INTER-86 Cache-read-log writer/reader.

2 contributors

2 lines (1 sloc) | 0.158 kb

Forked from FGM/heisencache to be used with Display Cache Warmer.

IMPLEMENTATION: extending

EASY TO EXTEND :

- Write additional EventSubscriber classes
- Add them to your configuration

ALREADY EXTENDED :

- WriterSubscriber (France Télévisions)
- CacheReadLogWriterSubscriber (AmazeeLabs)
- Alternate loader (AmazeeLabs)

IMPLEMENTATION: extending

MOST TYPICAL

- Create new subscriber for custom conditions
- Create new writer classes
- Target alternate stores for speed and ease
  - MongoDB
  - K/V or data structure store (Redis)
  - Message queue (Beanstalkd, RabbitMQ, ZeroMQ, etc)

IMPLEMENTATION: UI

- ROLL YOUR OWN!
- Use `WatchdogSubscriber` data
  - Use `admin/reports/dblog` or rework it
- Views integration
  - Use the `SqlWriterSubscriber` data
  - Two default views provided (Sql, Watchdog)
- Symfony WebProfiler toolbar
  - Replace/complement existing cache report

OSInet
formation

IMPLEMENTATION: UI

- Data collecting: raw data, big volume
- Three-step data processing
  - Collect → Heisencache
  - Cook → Process data based on your needs
  - Consumer → Visualize processed data
- Sweet spot
  - Use a queue (Beanstalkd, etc) instead of cron
  - Time-series database for longitudinal analysis : RRD, InfluxDB, OpenTSDB ...

IMPLEMENTATION: UI

Comparing with network analysis

- Heisencache is like
  - libpcap / tcpdump / iptrace / snoop ...
- Someone has to design a Wireshark on top of it

CONTEXT

MEASURING

IMPLEMENTATION

RESULTS

RESULTS: instant data

TYPICAL USEFUL INSTANT RESULTS

- Repeated misses
  - Usual suspects: default Memcached and big writes (prod)
  - Rewriting a variable on most pages (dev)
- Many calls to same key
  - Usual suspect: missing or broken static cache
- Many calls to related keys
  - Usual suspect: code loop instead of cache multiple

RESULTS: longitudinal analysis

TYPICAL USEFUL TIME-SERIES RESULTS

- Size of known-to-be-growing keys. Usual suspects:
  - Translation cache → Someone left a `t($foo)` somewhere
  - Context cache → Contexts are piled instead of refactored
  - Views plugins → Hard Views problem. Partial fix only.
- Miss rate shooting up from baseline on a bin
  - Call for instant analysis on that bin: likely a code regression
- Response time shooting up on a normally stable key
  - Network/server problem, bin saturation
  - Call for instant analysis on the cache instance

OSInet formation

RESULTS : beyond debugging

- Storing cache hit patterns for cache warming
  - Pre-warm cache on keys selected with a Heisencache WriterSubscriber
- Exemple with the AmazeeLabs fork :

https://github.com/AmazeeLabs/heisencache

Drupal 8?

WHEN?

- When bigger sites start to deploy D8
- ...and need them to go faster :-)
- First steps already available :
- https://github.com/FGM/heisencache

Drupal, *faster*

http://www.osinet.fr/

OSInet
formation