Tag and Release

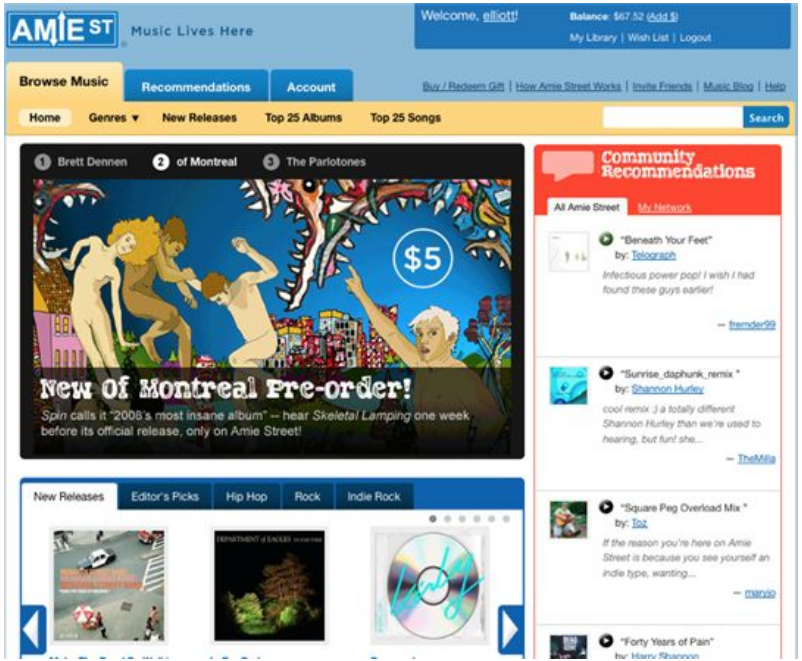
Monitoring Increasingly Distributed Applications

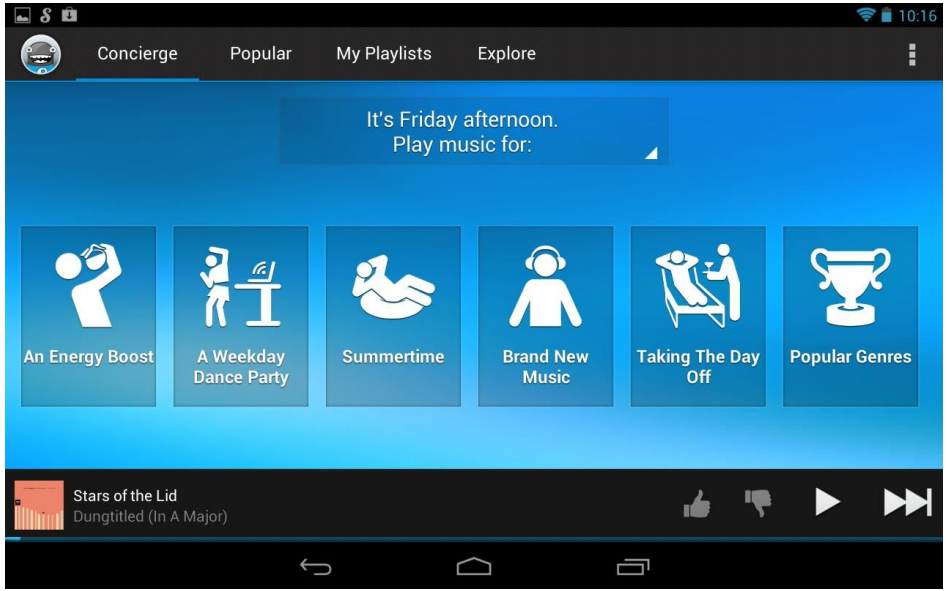
dkuebric / dan@appneta.com

AppNeta

Outline

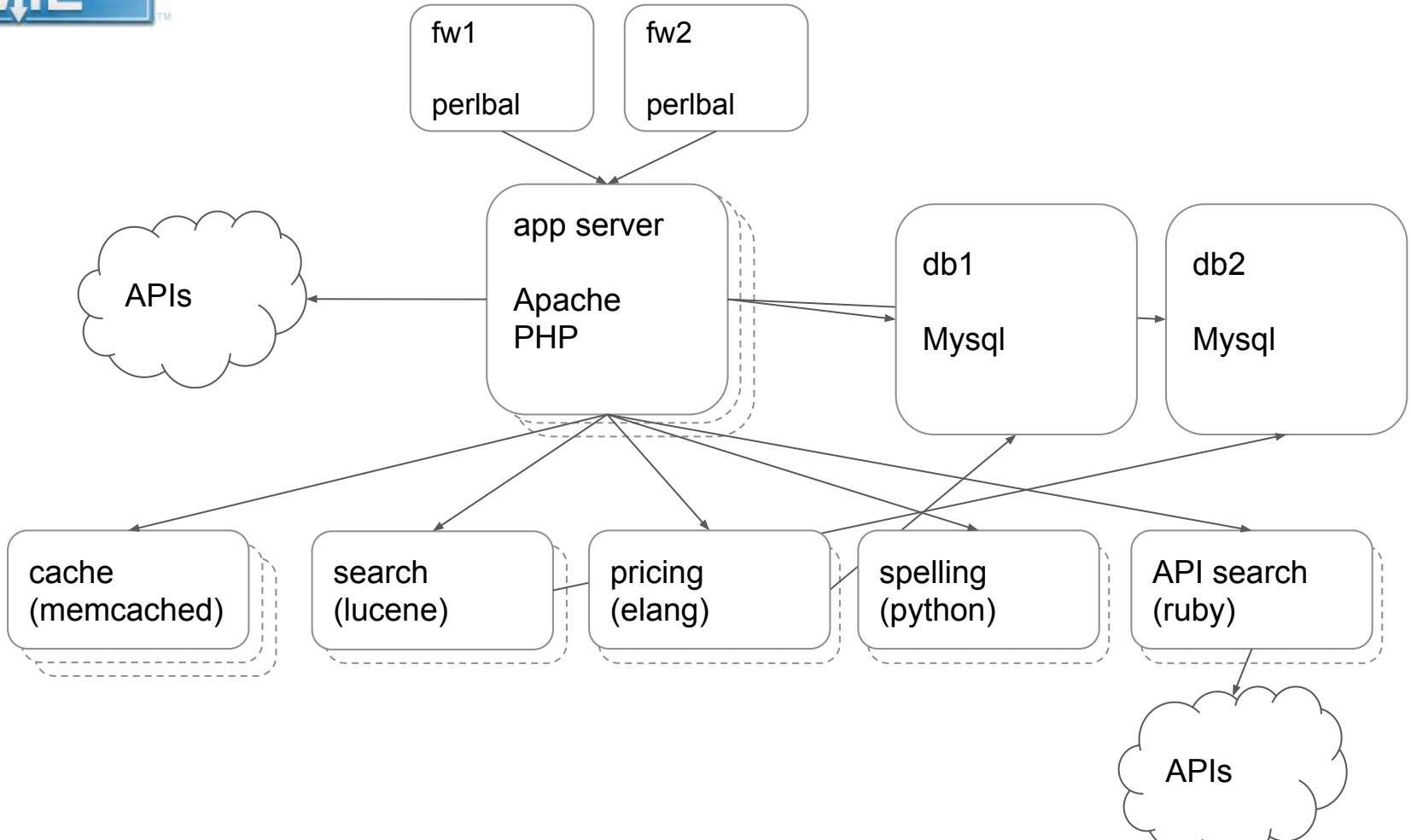
- What is distributed tracing?
- Who’s doing it, and how?
- Challenges, and future directions?


AMIE ST
Music Lives Here
Welcome, elliott!
Balance: $67.52 (Add $)
My Library | Wish List | Logout
Browse Music
Recommendations
Account
Buy / Redeem Gift | How Amie Street Works | Invite Friends | Music Blog | Help
Home
Genres
New Releases
Top 25 Albums
Top 25 Songs
Search
1 Brett Dennen
2 of Montreal
3 The Parlotones
$5
New Of Montreal Pre-order!
Spin calls it "2008's most insane album" -- hear Skeletal Lamping one week before its official release, only on Amie Street!
Community Recommendations
All Amie Street
My Network
"Beneath Your Feet"
by: Telograph
Infectious power pop! I wish I had found these guys earlier!
-- fremder99
"Sunrise_daphunk_remix "
by: Shannon Hurley
cool remix :) a totally different Shannon Hurley than we're used to hearing, but fun! she...
-- TheMilla
"Square Peg Overload Mix "
by: Toz
If the reason you're here on Amie Street is because you see yourself an indie type, wanting...
-- manyo
"Forty Years of Pain"
New Releases
Editor's Picks
Hip Hop
Rock
Indie Rock


10:16
Concierge
Popular
My Playlists
Explore
It's Friday afternoon.
Play music for:
An Energy Boost
A Weekday Dance Party
Summertime
Brand New Music
Taking The Day Off
Popular Genres
Stars of the Lid
Dungtitled (In A Major)

Thrift Shop

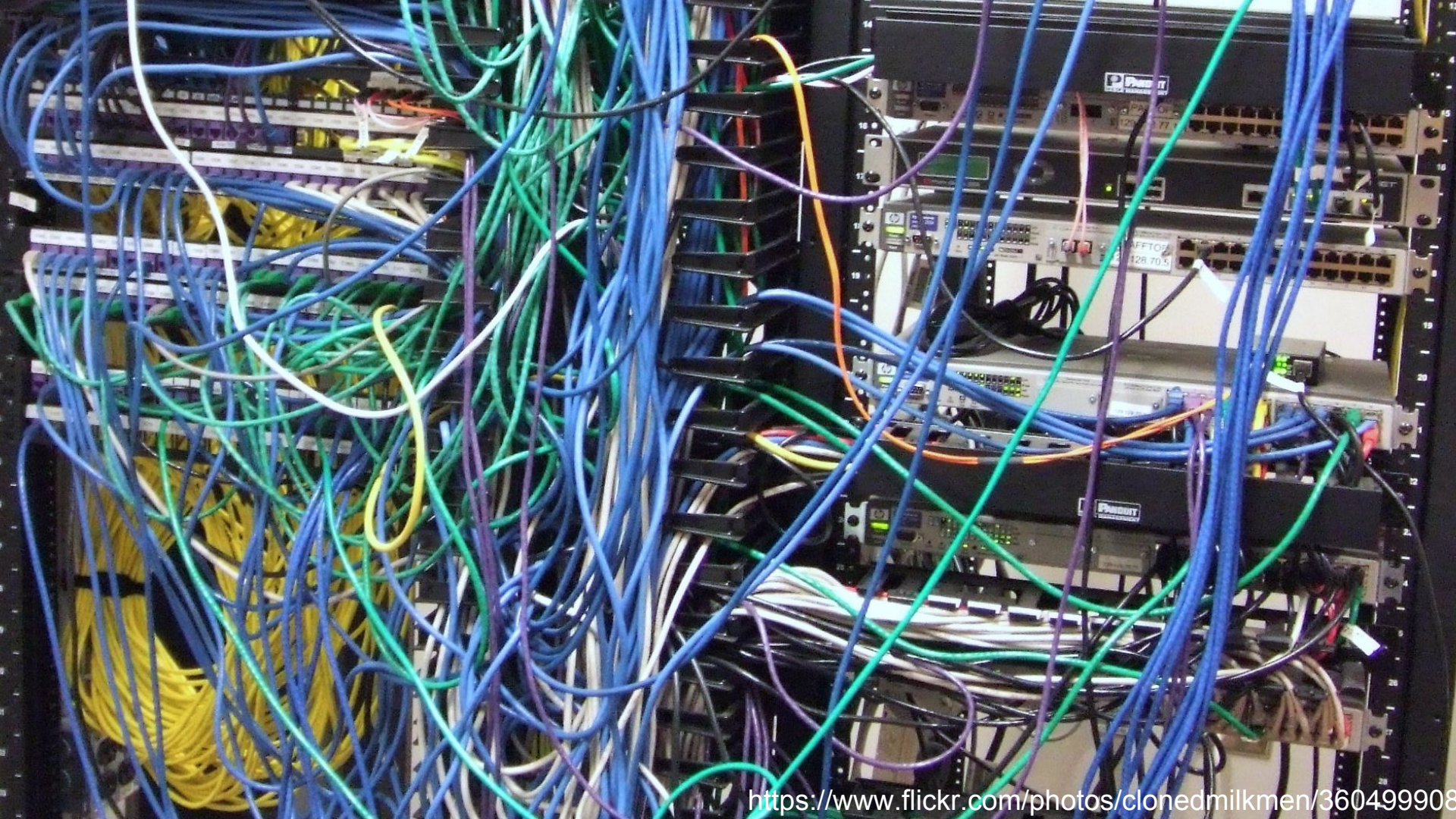
- Frontend web app: PHP
- Text search: lucene-based, via thrift
- Pricing service: erlang, via thrift
- Spelling corrector: python bindings around xapian, via thrift
- Content provider search: ruby, via thrift
- ...


fw1
perlbal
fw2
perlbal
app server
Apache
PHP
APIs
db1
Mysql
db2
Mysql
cache
(memcached)
search
(lucene)
pricing
(elang)
spelling
(python)
API search
(ruby)
APIs

Q: Why do you remember this so well?

Q: Why do you remember this so well?

A: ops


https://www.flickr.com/photos/clonedmilkmen/360499908

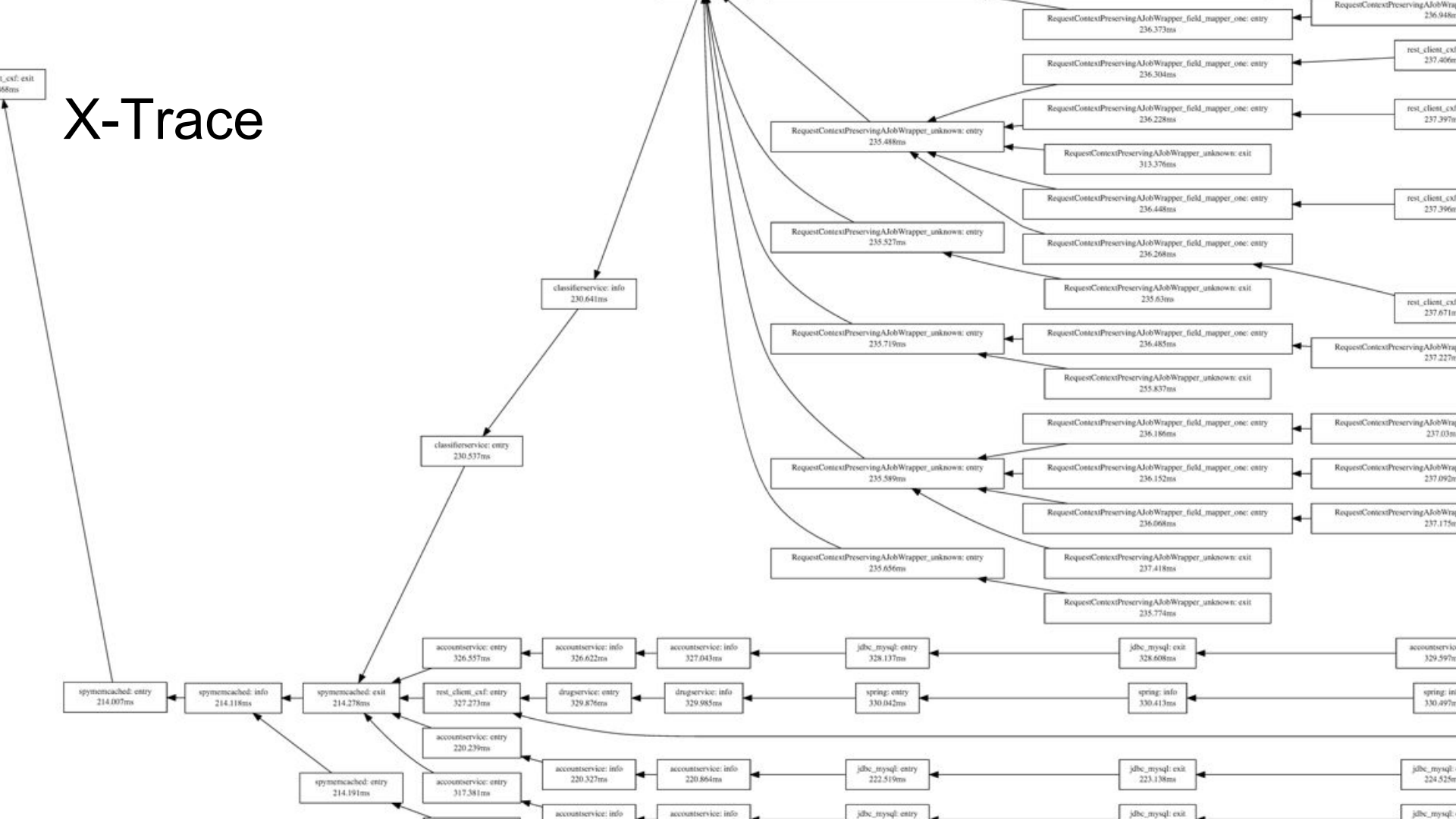
Things we had

- Ganglia
- Nagios
- Thrift
  - Per-service status page
  - Service status page
- Logs

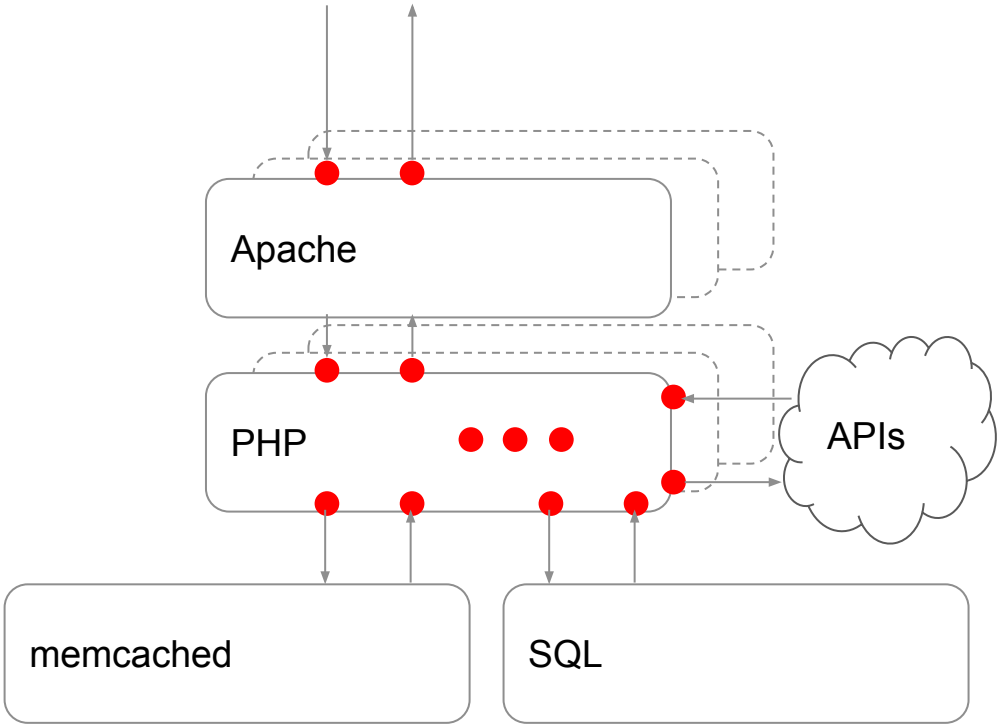
Sample performance / debug workflow

1. Are any services outright down?
2. Hit refresh N times -- how many times were problematic?
3. Systematically tail the logs of every service on every machine
4. Check database processlist
5. SSH in and poke around
6. Deploy new release with debug logging
7. Google

X-Trace


RequestContextPreservingAJobWrapper_field_mapper_one: entry
236.373ms
RequestContextPreservingAJobWrapper_field_mapper_one: entry
236.304ms
RequestContextPreservingAJobWrapper_field_mapper_one: entry
236.228ms
RequestContextPreservingAJobWrapper_unknown: entry
235.488ms
RequestContextPreservingAJobWrapper_unknown: exit
313.376ms
RequestContextPreservingAJobWrapper_field_mapper_one: entry
236.448ms
RequestContextPreservingAJobWrapper_unknown: entry
235.527ms
RequestContextPreservingAJobWrapper_field_mapper_one: entry
236.268ms
RequestContextPreservingAJobWrapper_unknown: exit
235.63ms
classifierservice: info
230.641ms
RequestContextPreservingAJobWrapper_unknown: entry
235.719ms
RequestContextPreservingAJobWrapper_field_mapper_one: entry
236.485ms
RequestContextPreservingAJobWrapper_unknown: exit
255.837ms
classifierservice: entry
230.537ms
RequestContextPreservingAJobWrapper_field_mapper_one: entry
236.186ms
RequestContextPreservingAJobWrapper_unknown: entry
235.589ms
RequestContextPreservingAJobWrapper_field_mapper_one: entry
236.152ms
RequestContextPreservingAJobWrapper_field_mapper_one: entry
236.068ms
RequestContextPreservingAJobWrapper_unknown: entry
235.656ms
RequestContextPreservingAJobWrapper_unknown: exit
237.418ms
RequestContextPreservingAJobWrapper_unknown: exit
235.774ms
accountservice: entry
326.557ms
accountservice: info
326.622ms
accountservice: info
327.043ms
jdbc_mysql: entry
328.137ms
jdbc_mysql: exit
328.608ms
spymemcached: entry
214.007ms
spymemcached: info
214.118ms
spymemcached: exit
214.278ms
rest_client_cxf: entry
327.273ms
drugservice: entry
329.876ms
drugservice: info
329.985ms
spring: entry
330.042ms
spring: info
330.413ms
accountservice: entry
220.239ms
accountservice: info
220.327ms
accountservice: info
220.864ms
jdbc_mysql: entry
222.519ms
jdbc_mysql: exit
223.138ms
spymemcached: entry
214.191ms
accountservice: entry
317.381ms

Example: Drupal request handling



Drupal TraceView project

TraceView

View | Version control | Revisions | Automated Testing

Posted by pdrake on *September 3, 2013 at 12:04pm*

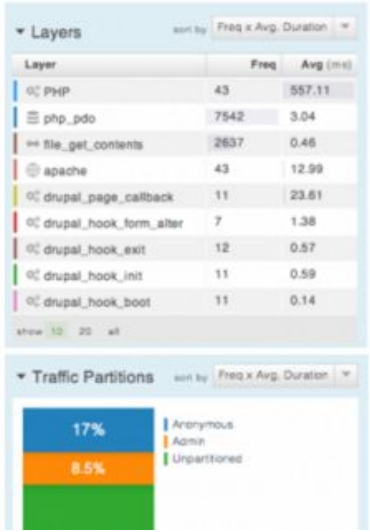
AppNeta TraceView (formerly Tracelytics) is a SAAS application performance management tool which provides full-stack application tracing across any number of hosts. This module is a replacement for the original tracelytics module.

The tracelytics module exposes Drupal-specific performance details to the TraceView agent:

- hook-level data for core and some contrib modules
- request partitioning by role or function
- per-menu-item filtering
- real user monitoring (RUM) JavaScript injection

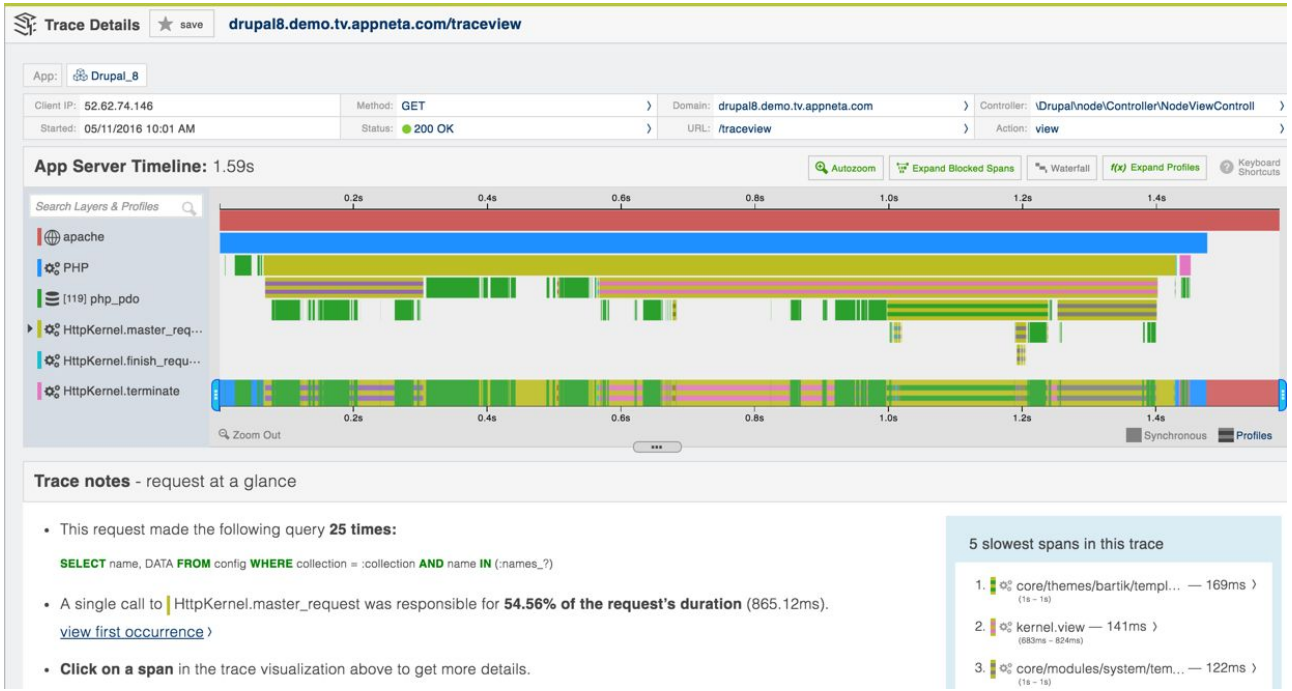
For more information, watch the video introduction to TraceView & Drupal.

This module is not maintained by AppNeta but does include properly attributed contributions from AppNeta employees.



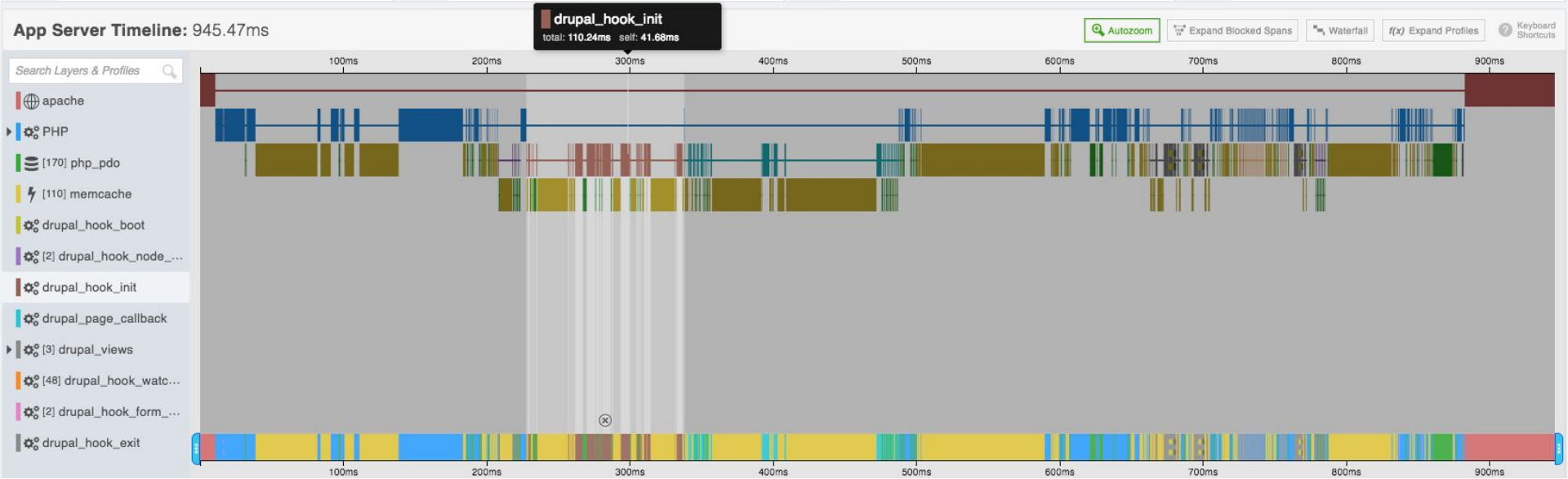
D6/7: https://www.drupal.org/project/traceview
D8: https://www.drupal.org/node/2113637

Drupal 8 request handling



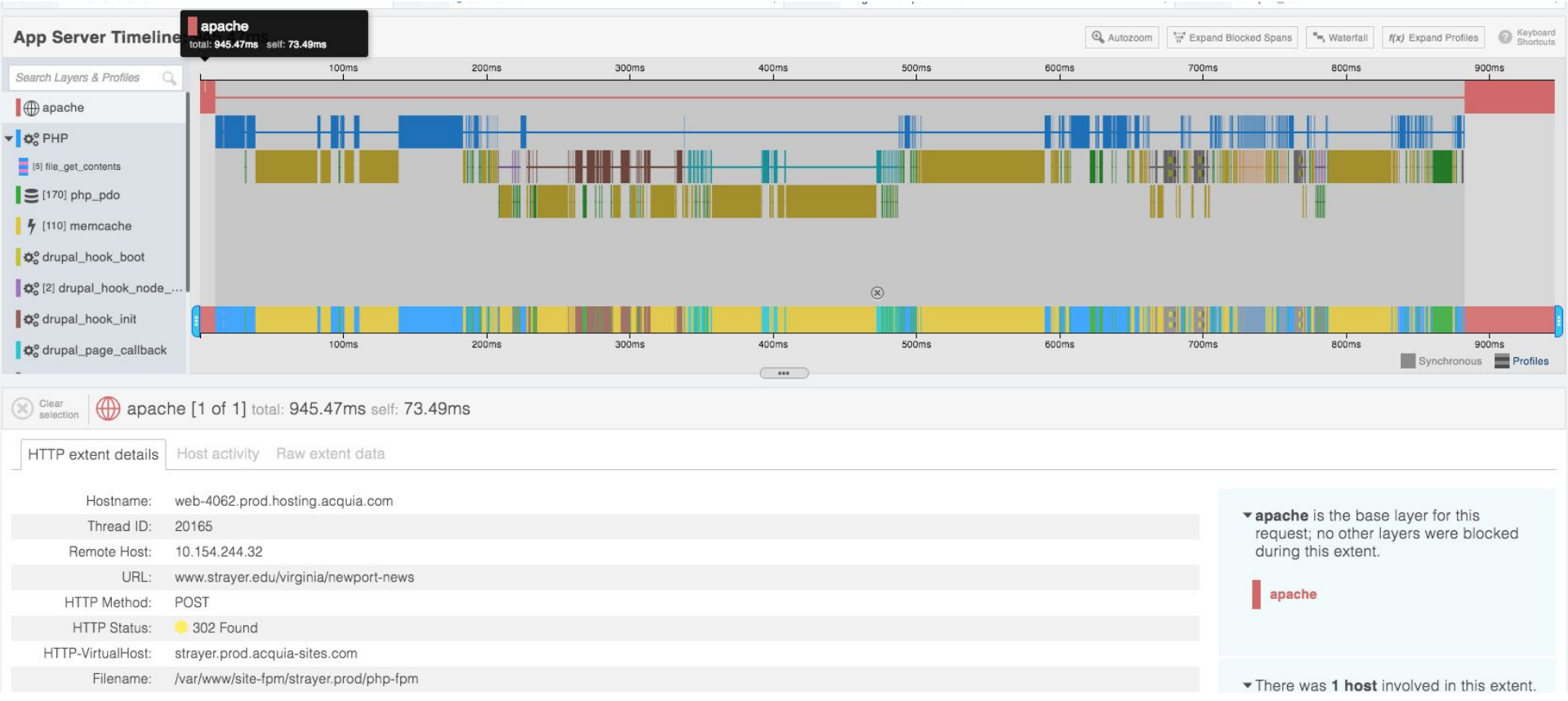
https://helloapp.tv.appneta.com/traces/view/FECA51A4134E765EBB04717C1D07F64352DE49E0

Example Drupal 7 request

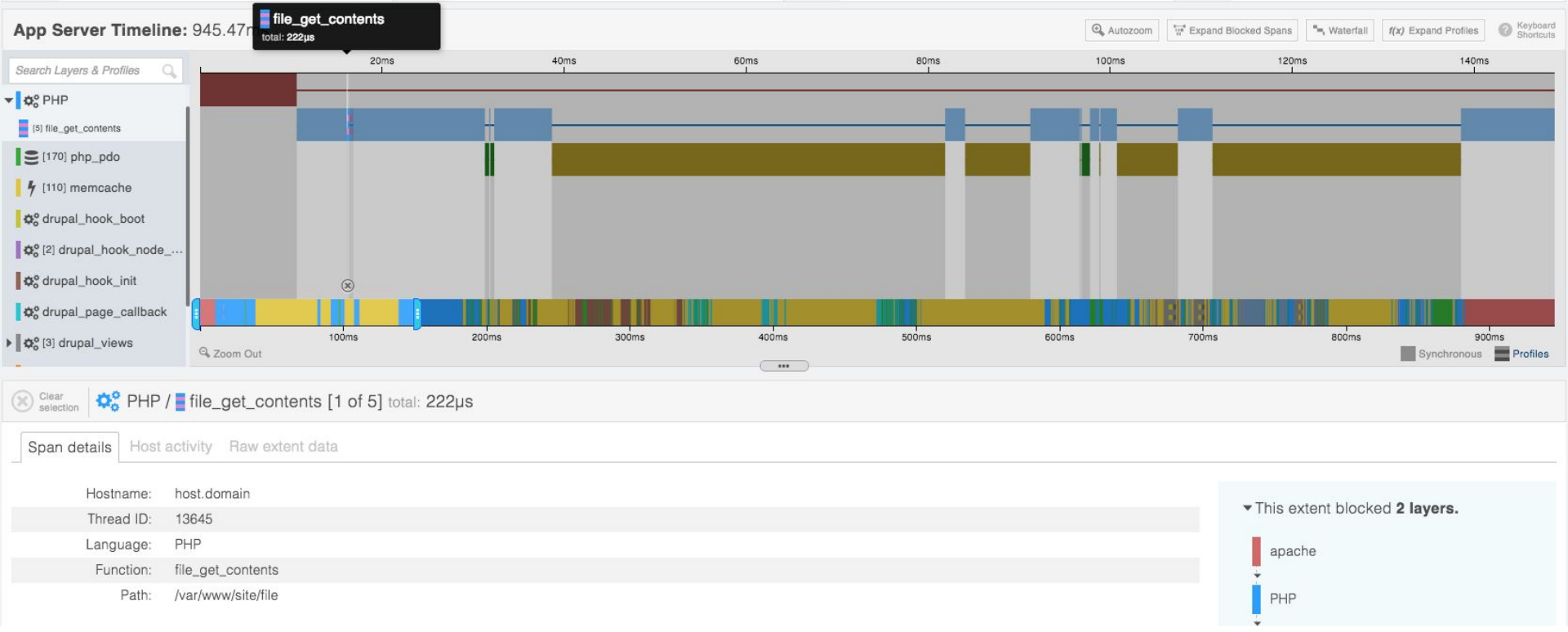


Example Drupal 7 request

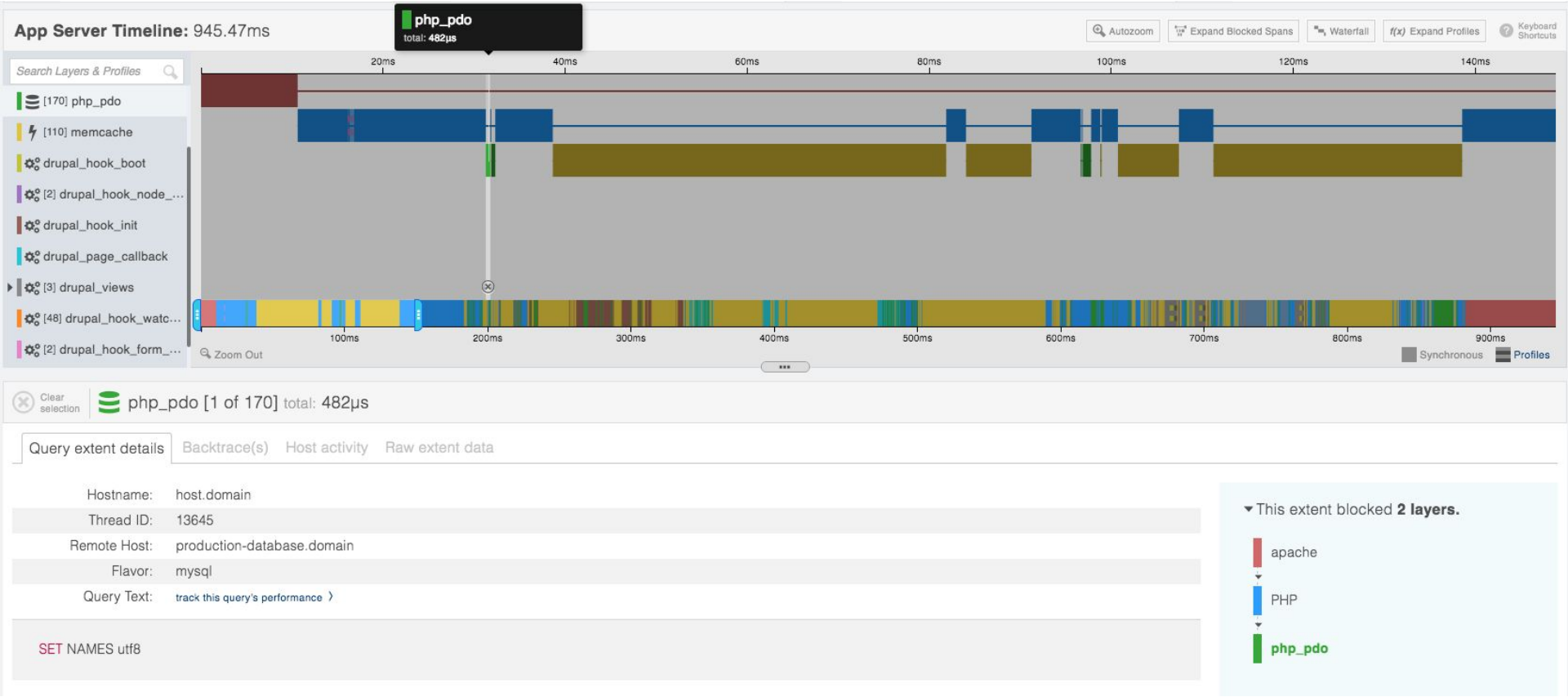
Example Drupal 7 request



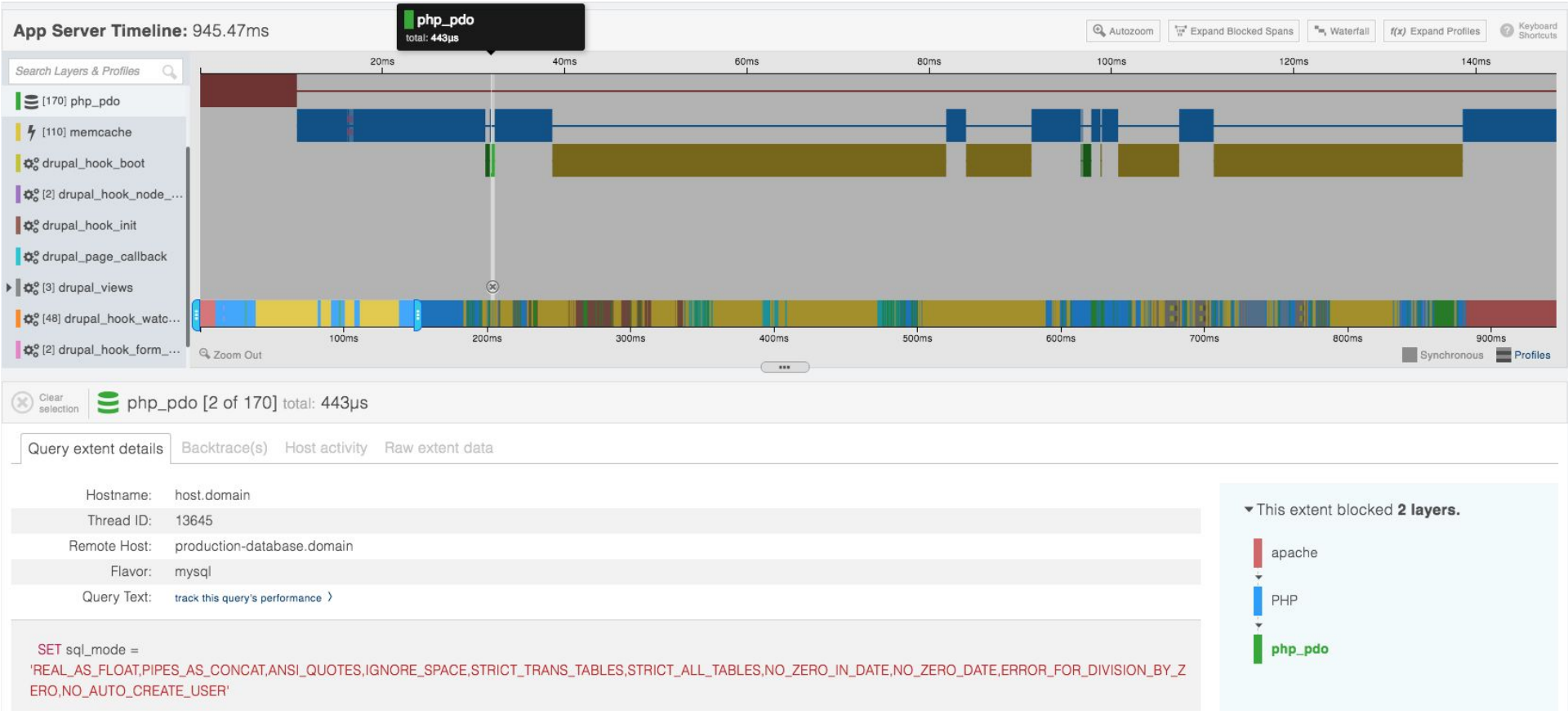
Example Drupal 7 request



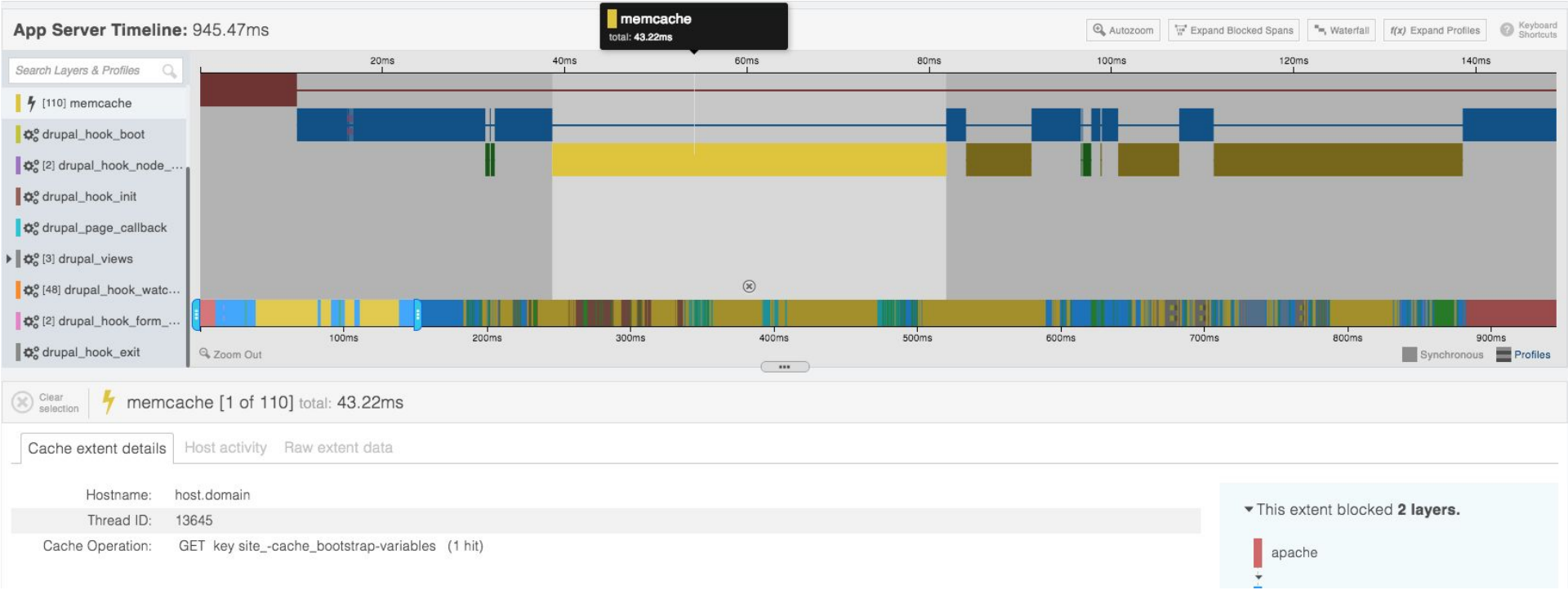
Example Drupal 7 request



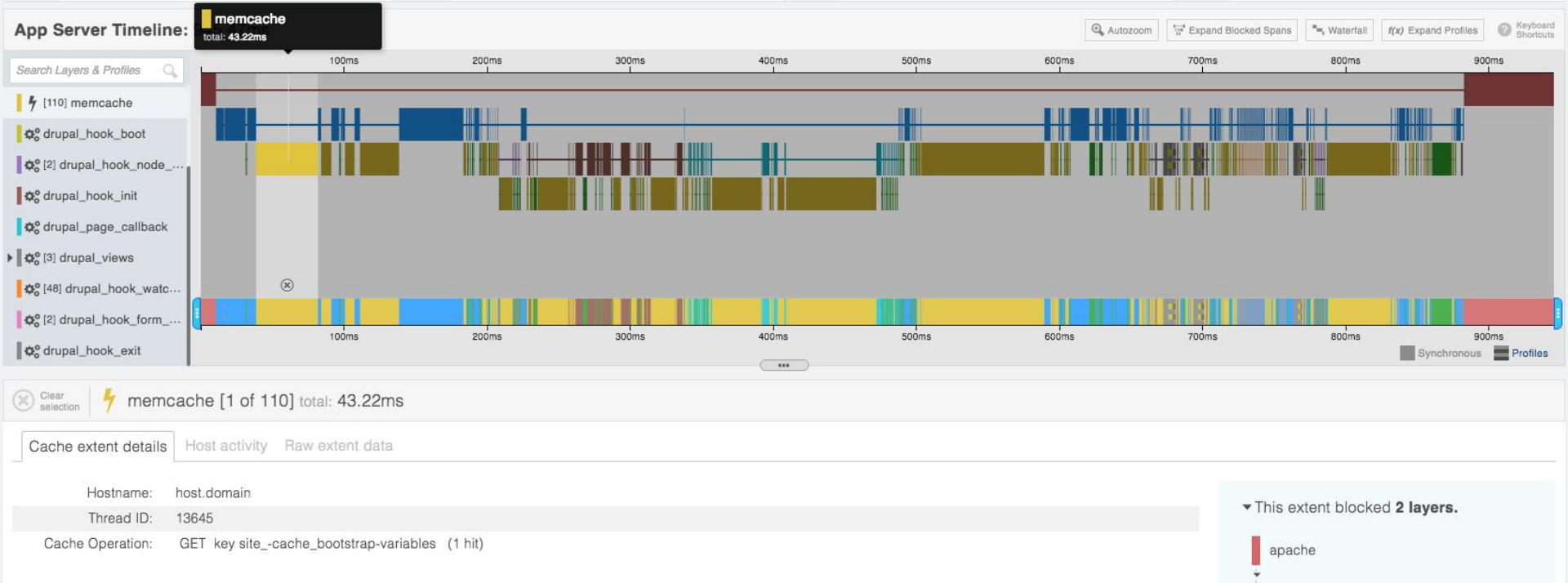
Example Drupal 7 request



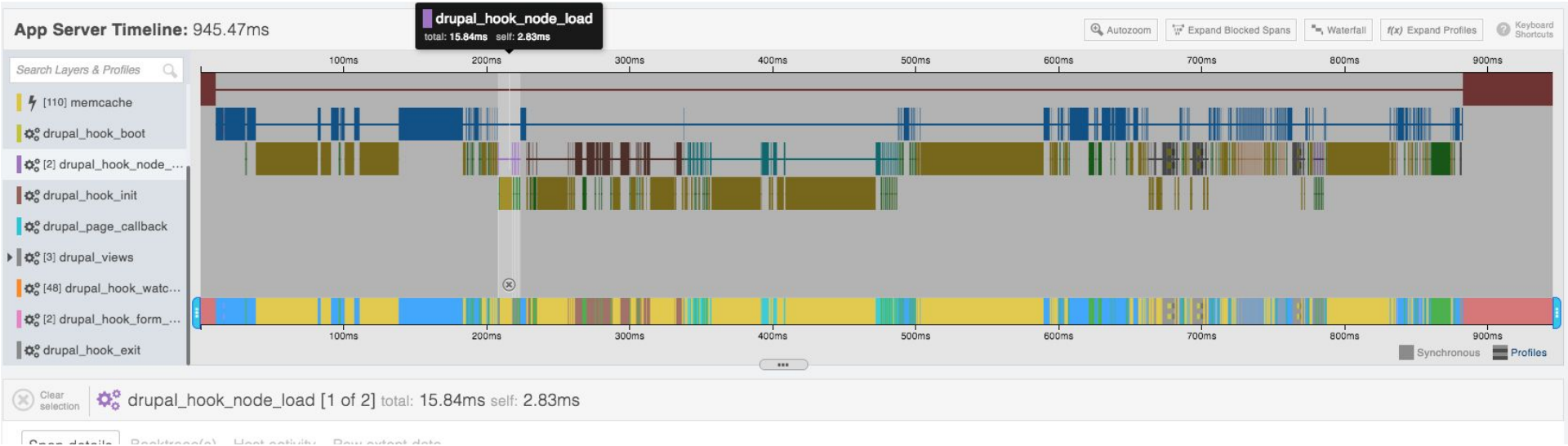
Example Drupal 7 request



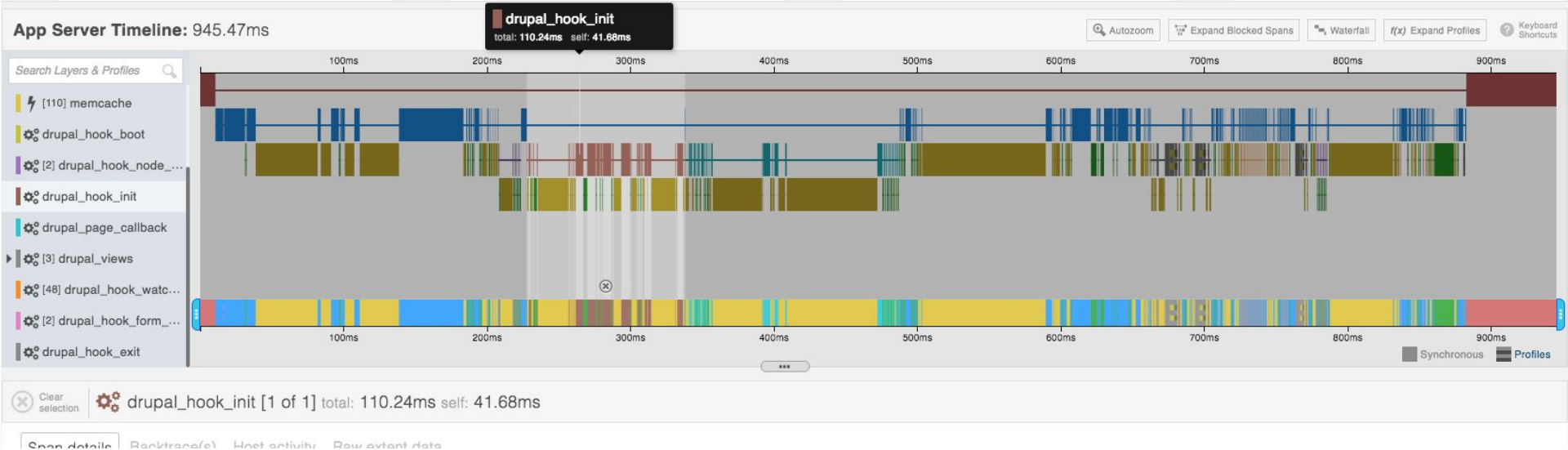
Example Drupal 7 request



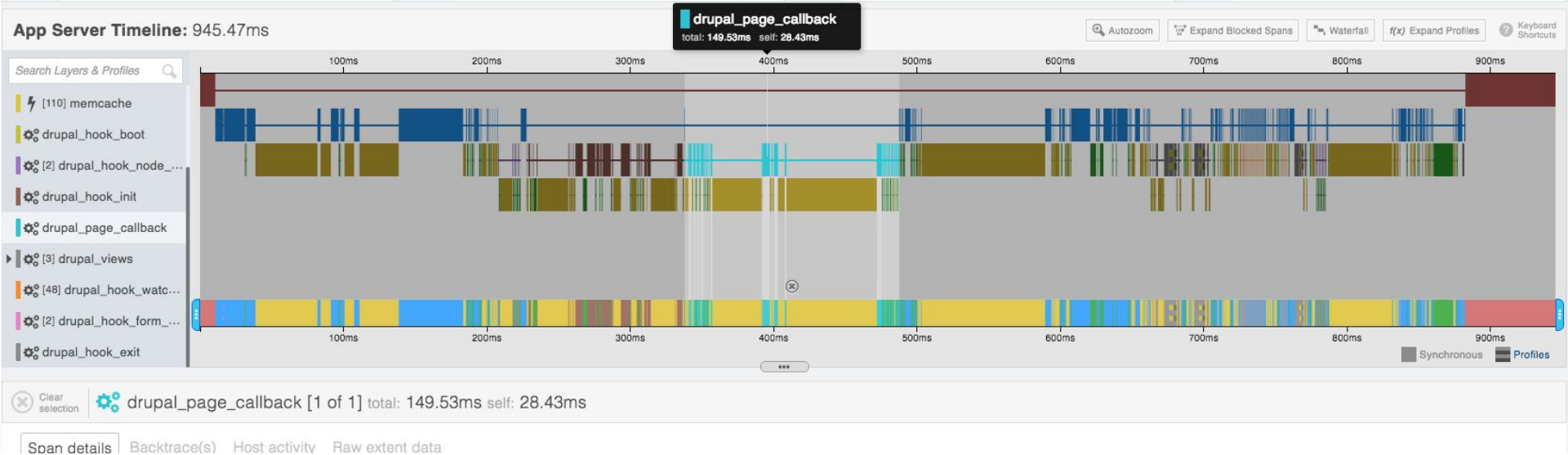
Example Drupal 7 request



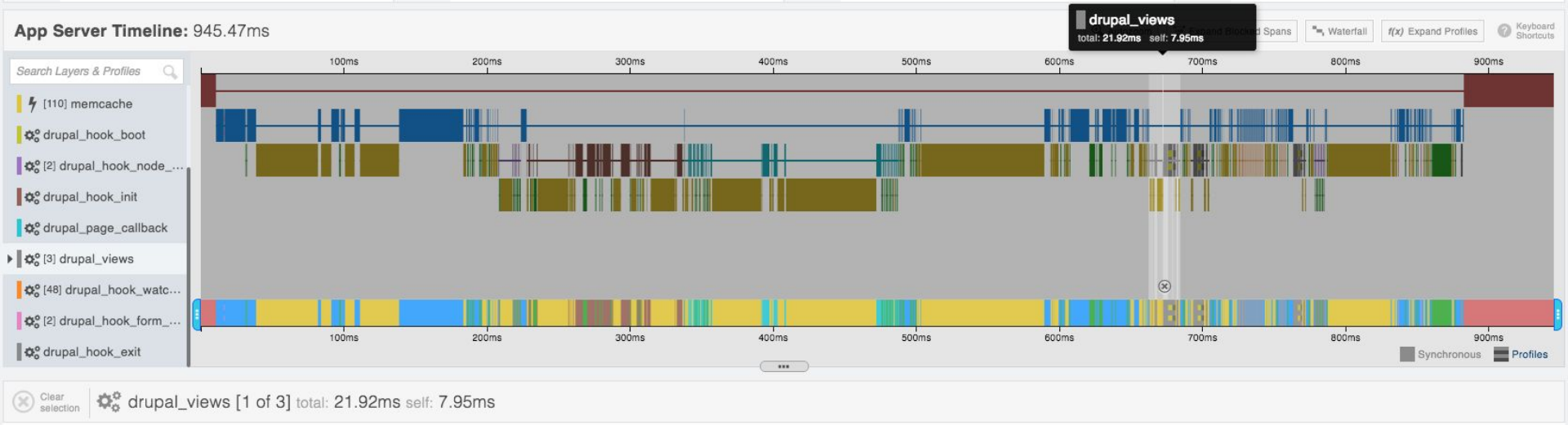
Example Drupal 7 request



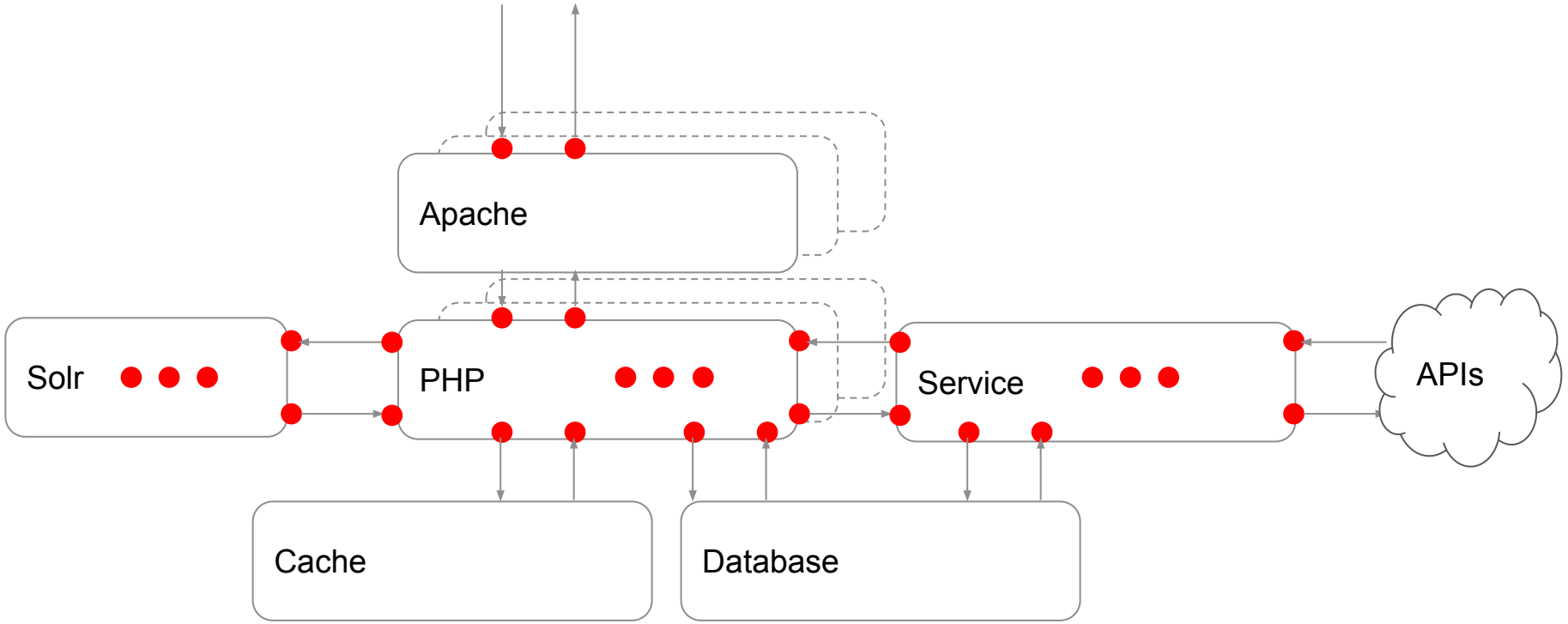
Example Drupal 7 request

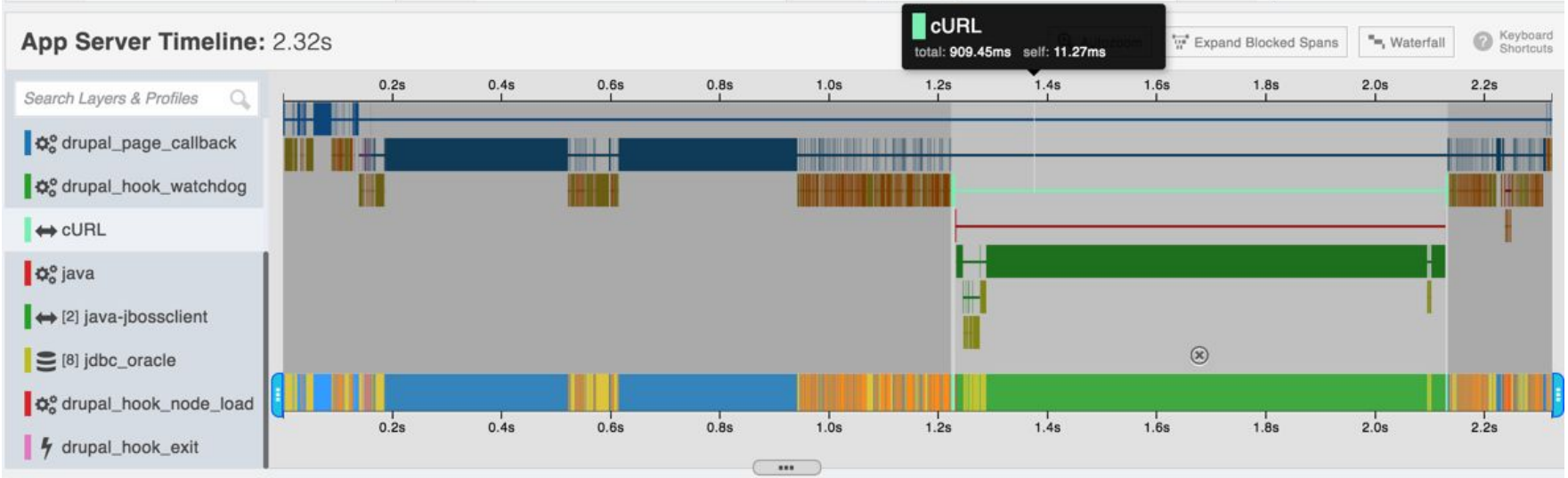


Example Drupal 7 request



Example Drupal request: more distributed





Clear selection

cURL [1 of 1] total: 909.45ms self: 11.27ms

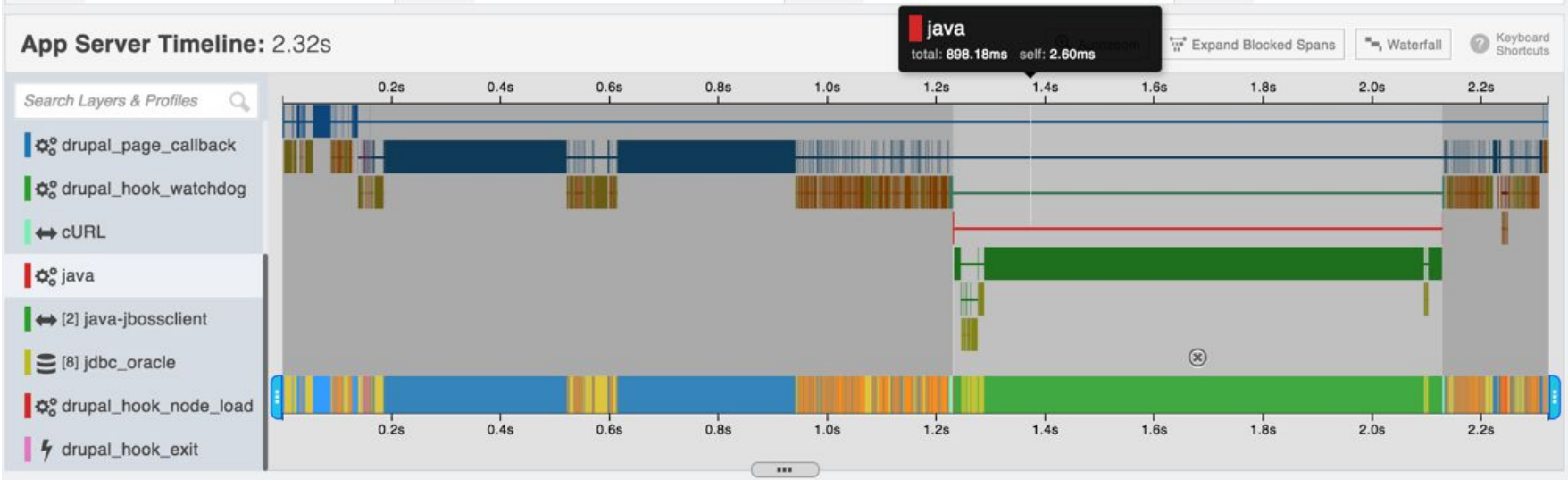
RPC extent details | Backtrace(s) | Host activity | Raw extent data

Hostname:	appserver.domain
Thread ID:	15012
Remote Host:	javaservice.domain
RPC:	http://javaservice.domain/endpoint/get?user=asdfjk

▼ This extent blocked **2 layers.**

PHP

drupal_page_callback



Clear selection

java [1 of 1] total: 898.18ms self: 2.60ms

HTTP extent details | Host activity | Raw extent data

Hostname:	javaservice.domain
Thread ID:	20773818
Remote Host:	10.4.35.65
URL:	http://javaservice.domain/endpoint/get?user=asdfjk
HTTP Method:	GET
HTTP Status:	200 OK

▼ This extent blocked **3 layers.**

- PHP
- drupal_page_callback
- cURL
- **java**

Great minds...

- Distributed tracing based on ID propagation
  - Google Dapper (200x? Published paper 2010)
  - Twitter Zipkin (Open-sourced 2012, 3rd party PHP support)
  - Etsy Cross Stitch (2014ish)
  - OpenTracing (2016ish)
- Commercial APM -- semi-distributed tracing
  - New Relic
  - AppDynamics

Challenges: Instrumentation Points

```
function interesting_method(...) {
    log_entry(...);
    _do_stuff();
    log_exit(...);
}
```

Challenges: Trace ID Propagation

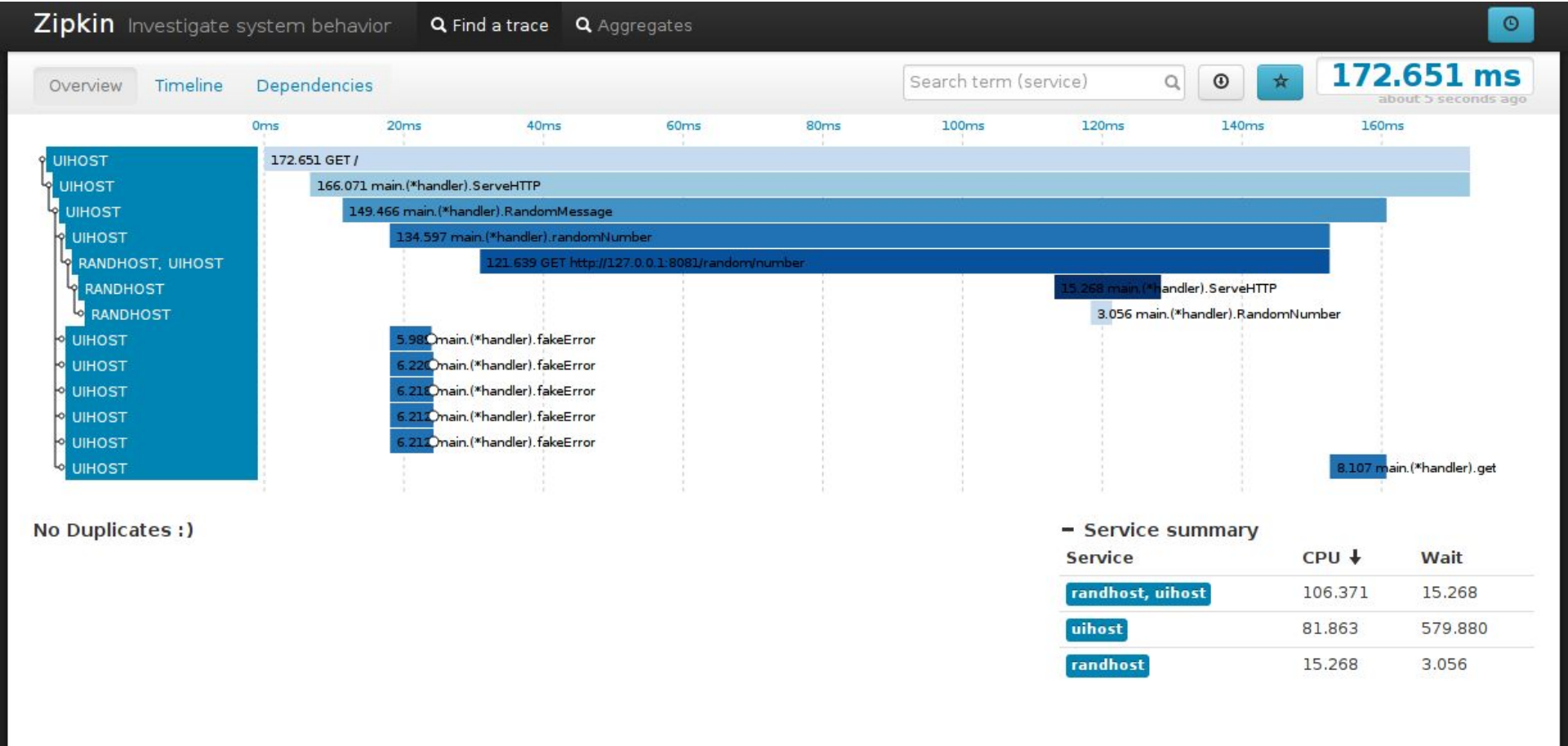
```
function interesting_method(trace_id,...) {
    log_entry(trace_id, ...);
    _do_stuff(?);
    log_exit(trace_id, ...);
}
```

Optional in PHP! Could use glo
due to single-request handling
model.

Challenges: Trace ID Propagation

```
function http_rpc_call(...) {
    log_entry(...);
    $opt = array(modified_headers);
    drupal_http_request($url, $opt);
    log_exit(...);
}
```

Challenges: Extracting Value



Rich data set

- Distributed tracing “only”
  - Follow request flow through application
  - Understand end-to-end latency
  - Associate backend load with frontend requests
  - Provide errors with distributed context
- While you’re in there
  - Latency of queries, RPC calls, in each tier
  - Slow code
  - Cache hit/miss ratio
  - Errors and exceptions
  - Custom tagging/categorization of data
  - ...

How does it actually work?

- PHP extension
  - Hook into core methods
- TraceView Module
  - Hook into key events -- take timing and attributes
- Drupal 8 module, for example:
  - Event Dispatcher -- log timing of different kernel actions, etc
  - Event Subscriber -- figure out if user is anon/authenticated/admin
  - Service Provider -- alter base template class
    - Wrapper for Twig -- get timing and info on templates

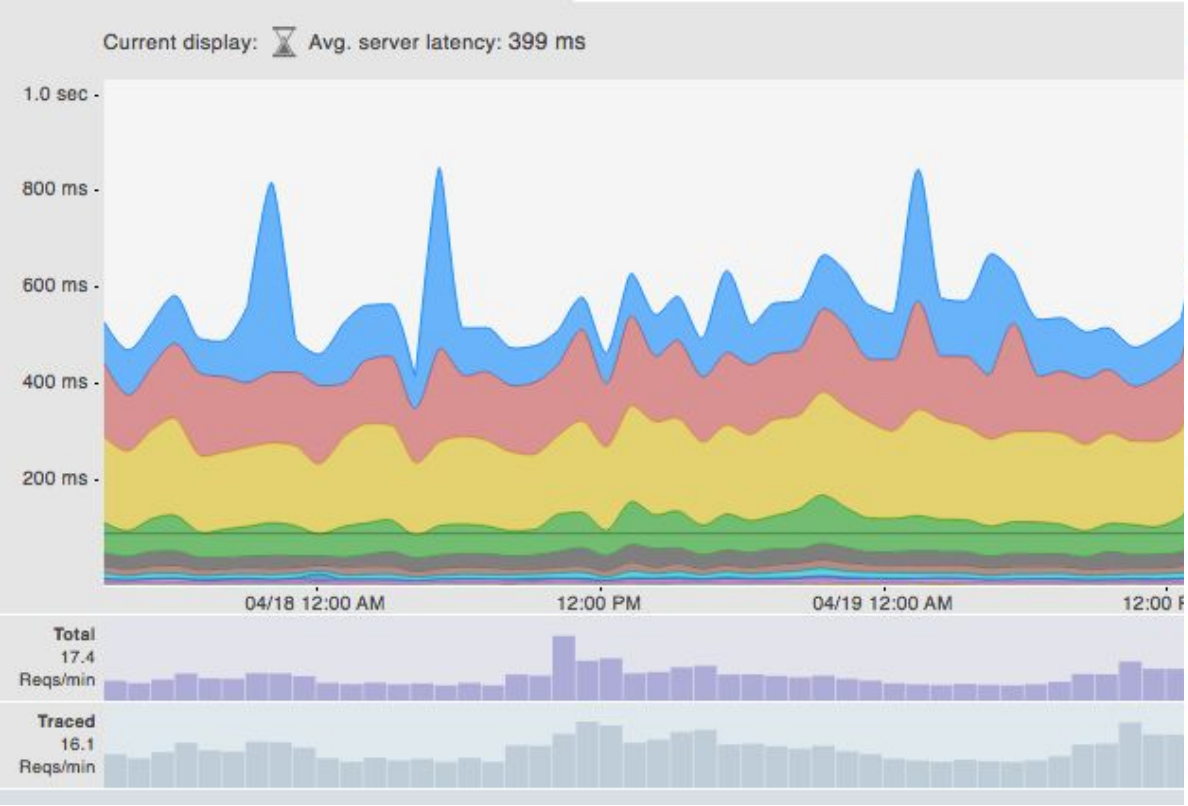
How does it actually work?

```
class TraceViewContainerAwareEventDispatcher extends ContainerAwareEventDispatcher
{
    public function dispatch($eventName, Event $event = null)
    {
        // On an untraced request, bail out early.
        if (!oboe_is_tracing()) {
            return parent::dispatch($eventName, $event);
        }
            …
        // Figure out what event we're dispatching
        if ($is_request) {
                oboe_log(($event->getRequestType() === HttpKernelInterface::MASTER_REQUEST) ? 'HttpKernel.
master_request' : 'HttpKernel.sub_request', "entry", array('Event' => get_class($event)), TRUE);
                oboe_log(NULL,"profile_entry", array('Event' => get_class($event), 'ProfileName' => $eventName),
TRUE);
        } elseif ($is_finish_request) {
         ...

        // Try to dispatch the event as normal.
        try {
            $ret = parent::dispatch($eventName, $event);
        // Catch any exceptions that occur during dispatch.
        } catch (\Exception $e) {
          ...
        }

        // And mark the end timing as well
```

Aggregate performance

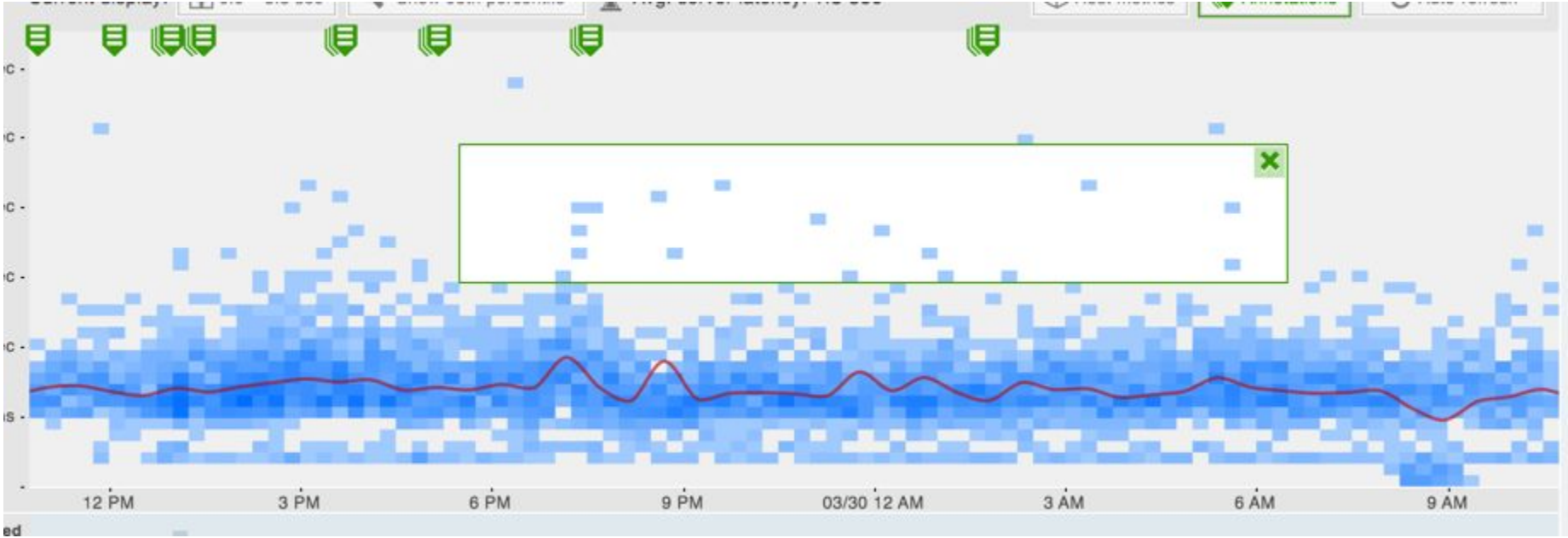


Layers by Frequency

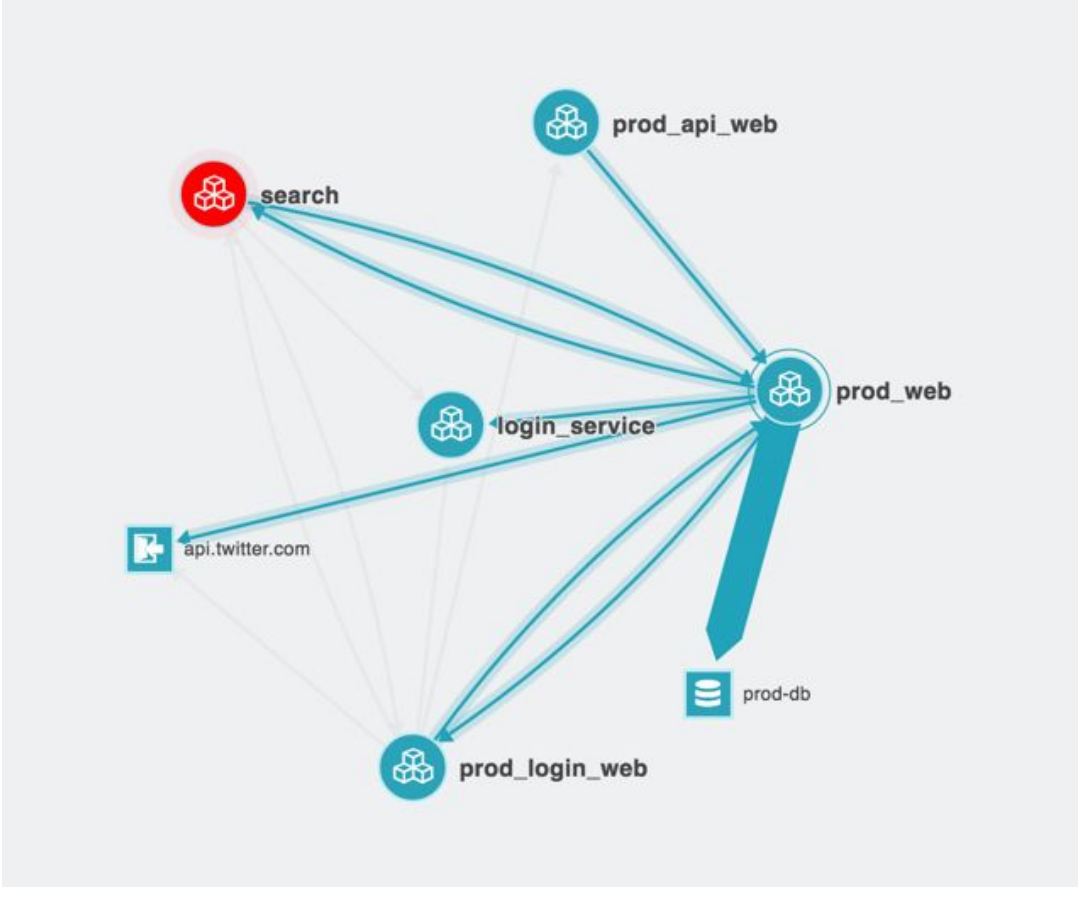
Layer	# Calls	Avg (ms)
php_pdo	4044112	0.42
memcache	1178948	2.48
drupal_hook_watchdog	66048	0.14
drupal_hook_node_load	59139	1.95
drupal_hook_form_alter	28098	1.00
apache	25839	91.10
PHP	23853	84.28
drupal_hook_boot	19594	0.09
drupal_hook_exit	19142	0.86
drupal_hook_init	15765	14.78
drupal_page_callback	11603	15.54
drupal_views	9420	51.10
cURL	525	59.95
drupal_http_request	438	98.12
drupal_panels	73	39.13
php_mysqli	72	0.67
php_http	8	317.79

Show 10 20 all

Outliers, trends



Topology mapping



Thanks!

twitter.com/dkuebric
appneta.com

dkuebric / dan@appneta.com

AppNeta