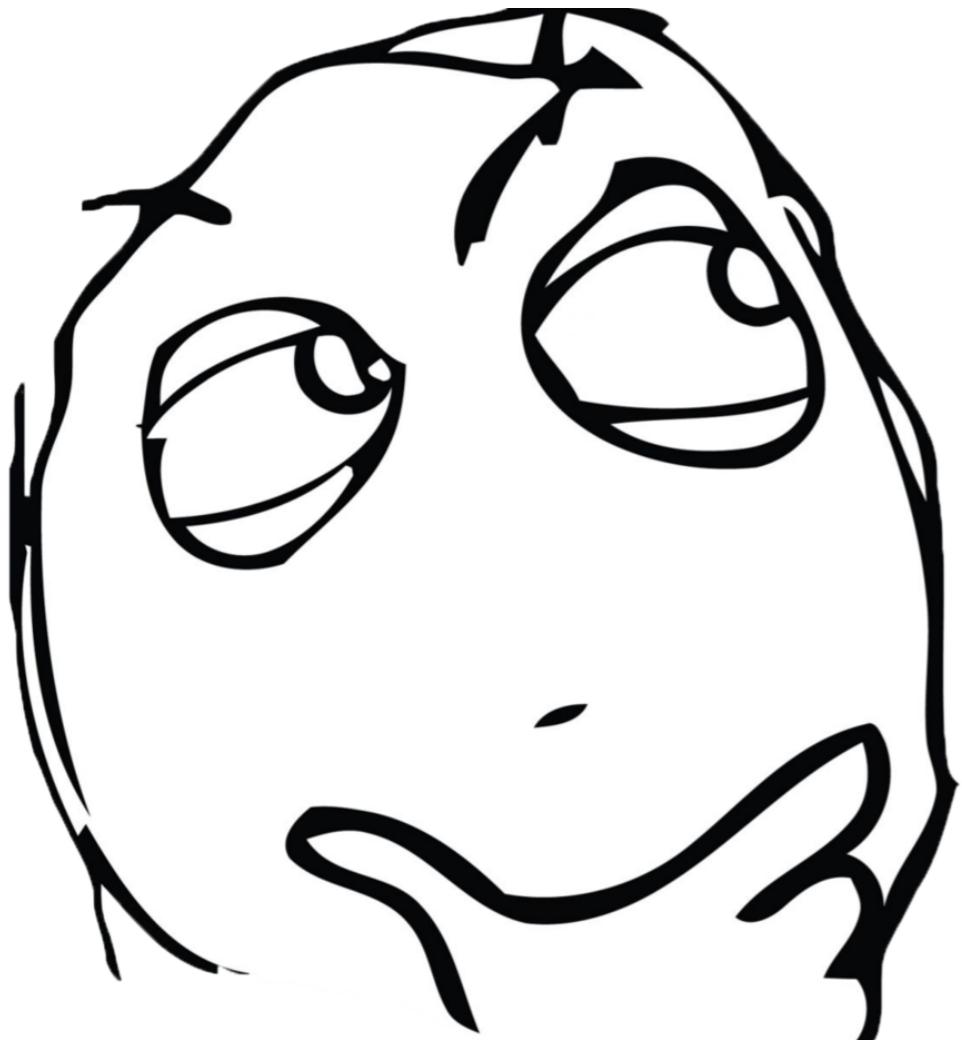


PERFORMANCE OPTIMISATION

HOW DO I GO ABOUT IT?



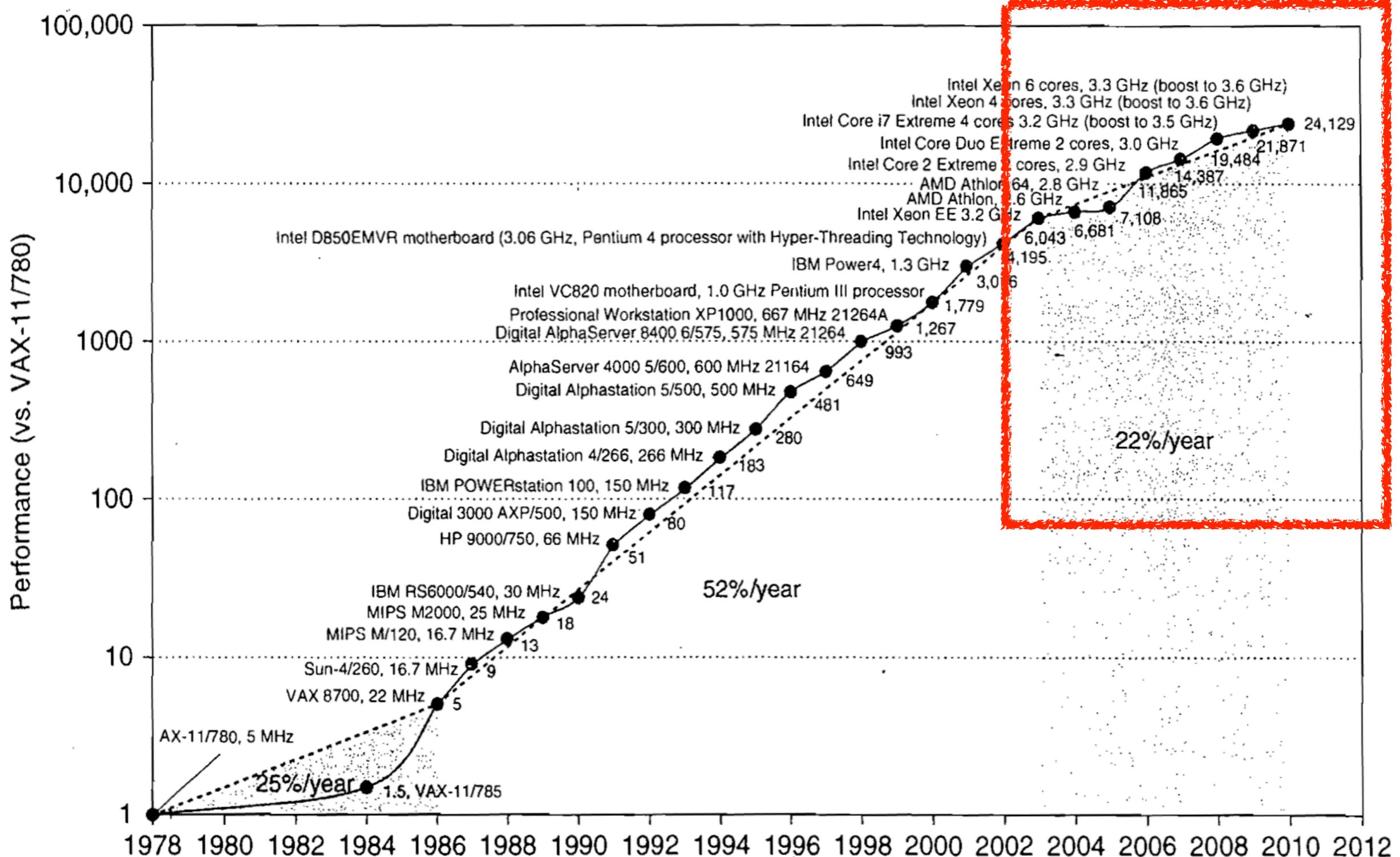
KAT ZIEŃ | @KASIAZIEN

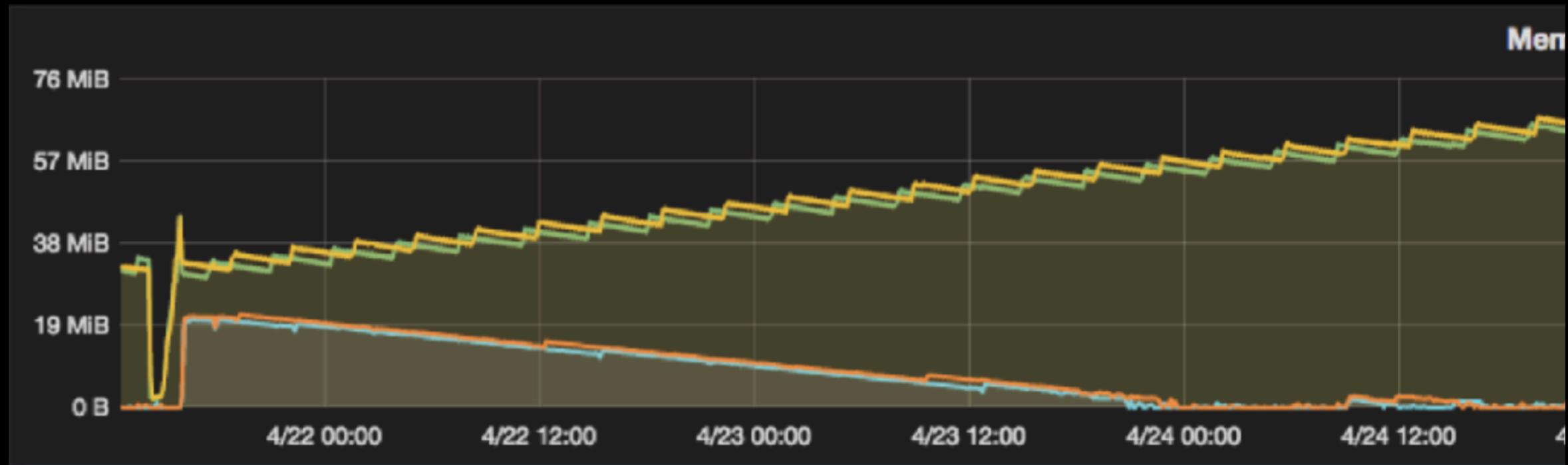
LONGHORN PHP

3 MAY 2019

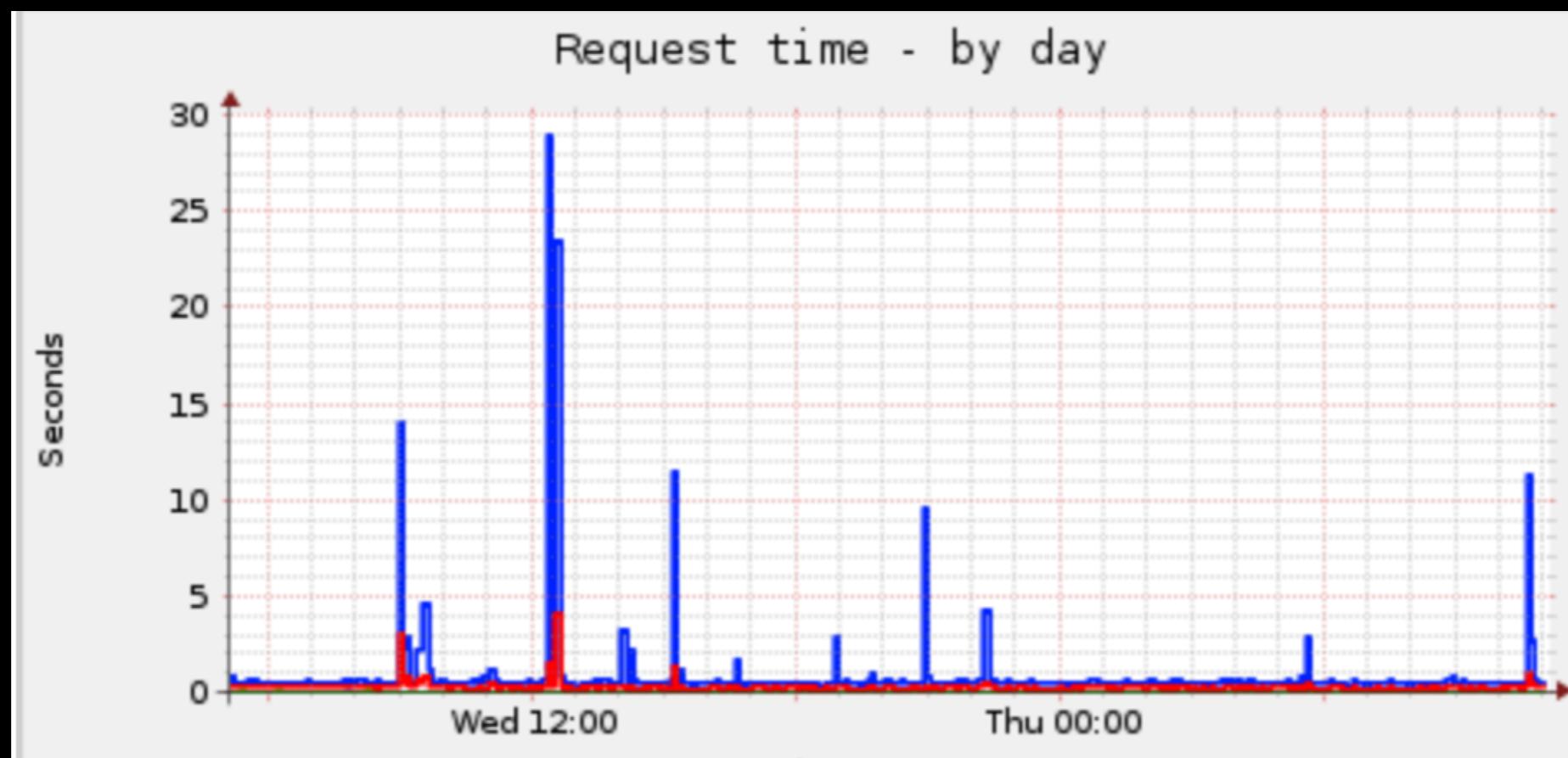
W H Y

- 👍 fast is better than slow (duh)
- 👍 memory efficient is good
- 💰 saving money is good
- 😬 running out of memory in production is bad
- 😱 suddenly running out of memory in production is BAD
- 🍿 language and hardware only take you so far





PHP Fatal error: Allowed memory size of 134217728 bytes exhausted



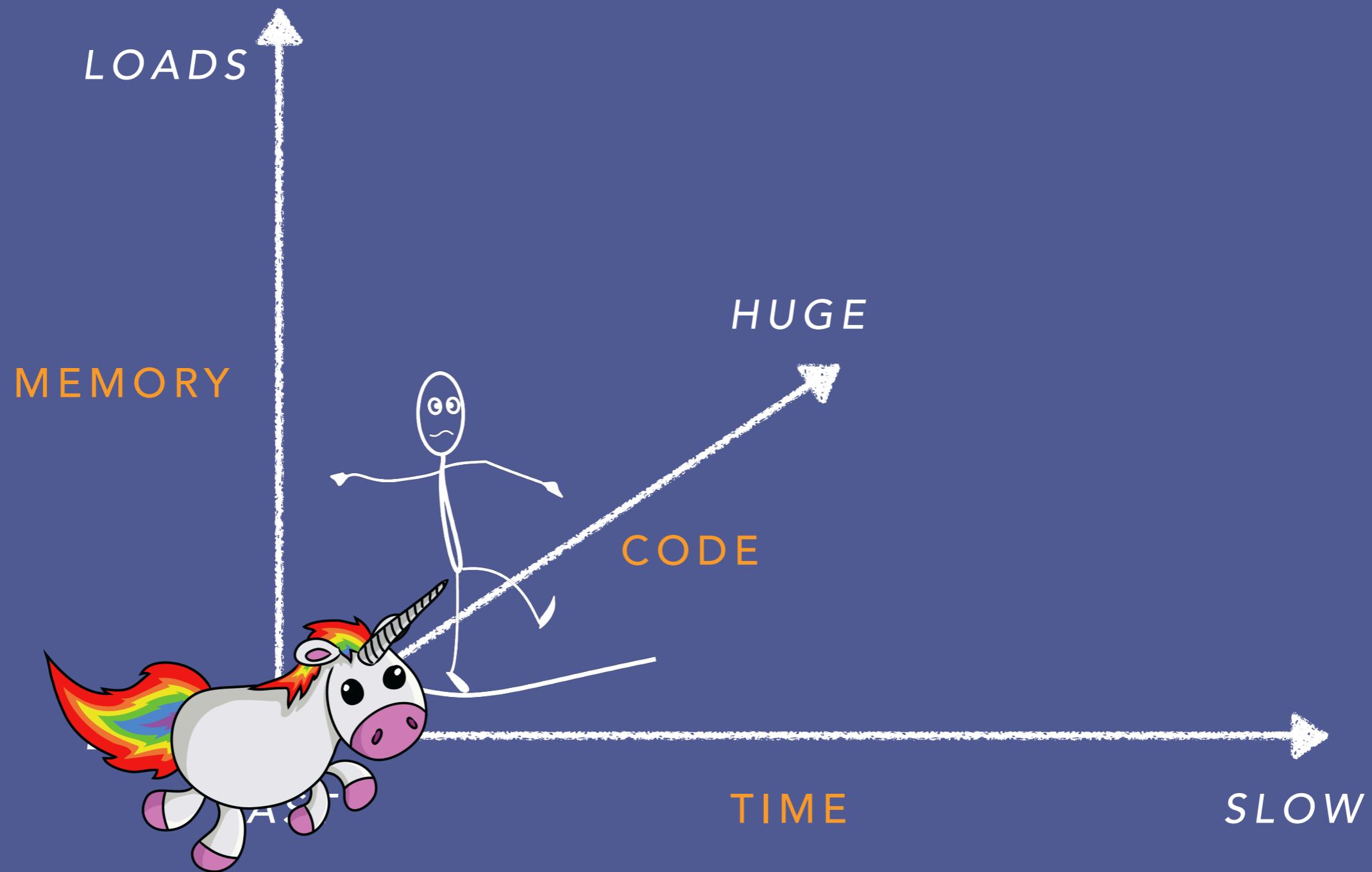
W H E N

 development load testing monitoring production

W H A T

 execution time / no of operations memory usage / allocation design (concurrency) server (eg. opcache settings) hardware

GOAL





memory leaks



timeouts



if concurrency: race conditions, deadlocks / livelocks



redundancy / inefficiency



dead code / unused libraries

<sane>

- don't evaluate in your loop condition, do it once before the loop

```
for ($i = 0, $count = count($array); $i < $count; $i++)
```

- watch out for DB queries inside loops

</sane>

- use empty(\$array) instead of count(\$array) === 0
- use \$a === \$b instead of strcmp(\$a, \$b) === 0
- use single quotes instead of double quotes unless you're concatenating variables
- foreach vs array_walk()
- use regex
- don't use regex
- ...

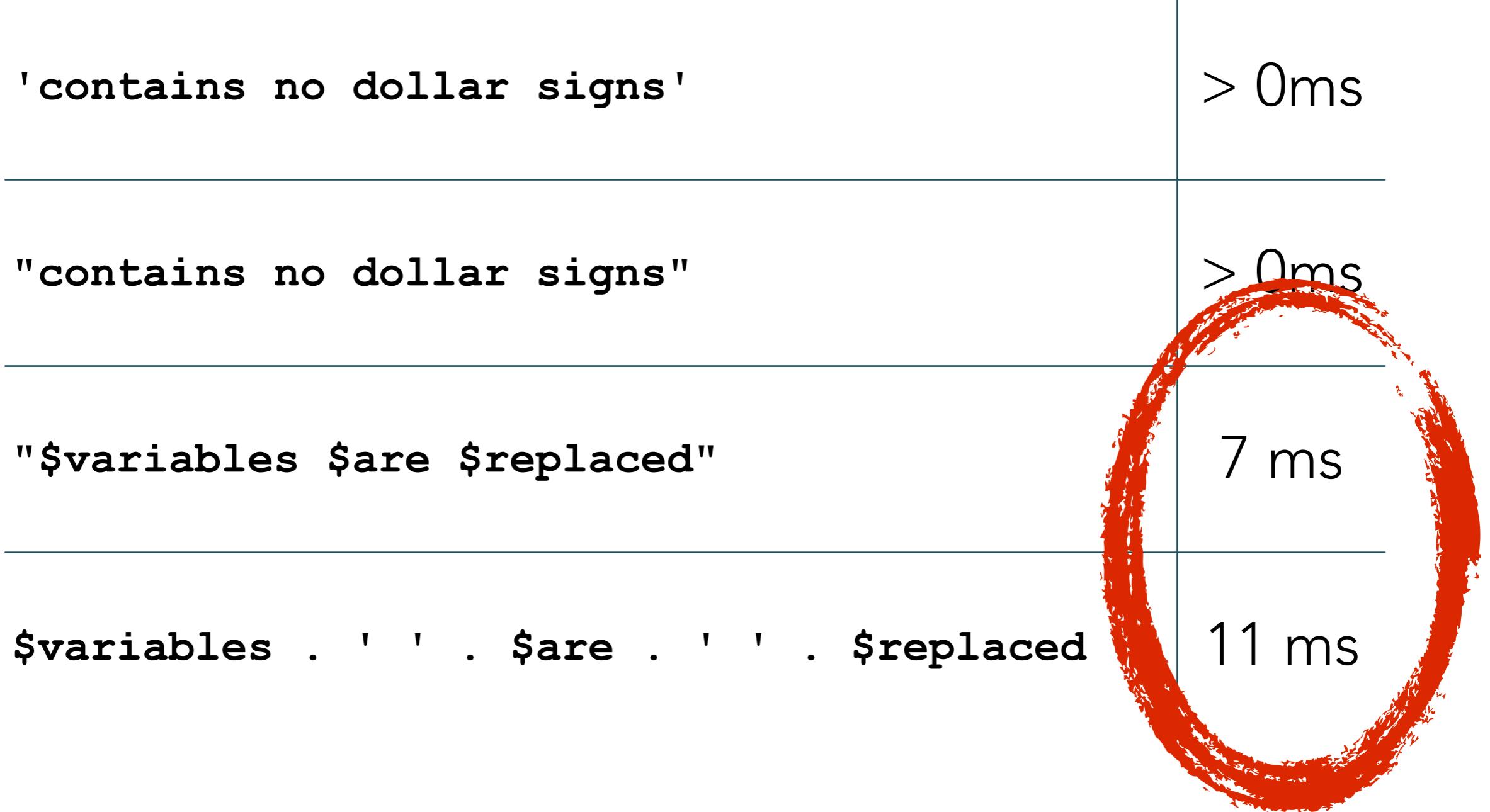
 memory leaks timeouts if concurrency: race conditions, deadlocks / livelocks redundancy / inefficiency dead code / unused libraries

MICRO OPTIMISATIONS

99% of the time, ignore them every time



THE SINGLE VS DOUBLE QUOTE MYTH



SOURCE: [HTTP://MAETTIG.COM/CODE/PHP/PHP-PERFORMANCE-BENCHMARKS.PHP](http://maettig.com/code/php/php-performance-benchmarks.php)

MORE LOL

<code>if (\$var === '')</code>	3ms
<code>if (' ' === \$var)</code>	3ms
<code>\$haystack[0] === 'n'</code>	> 0 ms
<code>strpos(\$haystack, \$needle) === 0</code>	> 0 ms
<code>str_replace(\$search, \$replace, \$subject)</code>	1 ms
<code>preg_replace("/\$search/", \$replace, \$subject)</code>	4 ms

H O W

HOW???

- do I know what to improve?
- do I know what my app is doing?
- do I know what my app can handle?

- do I measure?
- do I know what to measure?
- often do I measure?

TOOLS

- monitoring
- load testing / benchmarking
 - e.g. go-wrk, k6, pewpew, jmeter, gatling+taurus,
ApacheBench, curl-loader, httpperf, vegeta, a really big
loop in your code...
- profiling
- visualising

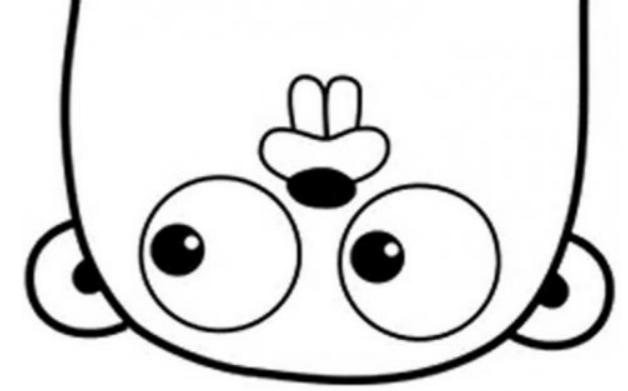
OFF THE SHELF



blackfire.io



TOOLS FOR PHP



- not built into the language :(
- roll your own?

```
$start_time = microtime(true); // 7.3: hrtime(true)  
  
// do stuff...  
  
$end_time = microtime(true); // 7.3: hrtime(true)  
  
$time_taken = $end_time - $start_time;
```

- extensions

INSTALLING EXTENSIONS (YMMV)

- package managers: brew / yum / apt-get install <name>
- PEAR/PECL: pecl install <name>
- compile from source:
 - git clone ... & cd ...
 - phpize
 - ./configure (+ any options for the extension)
 - make
 - make install
 - load the extension in .ini

INSTALL A PROFILER

- xdebug
 - + qcachegrind/kcachegrind for GUI
- xhprof
 - PHP 7 forks from tideways, patrickallaert or longxinH
-  adsr/phpspy  (Linux only at the moment, WIP)
- php-meminfo

DIAGNOSTICS
BEFORE
DIAGNOSIS

Cigar

build passing coverage 100% SLInsight ★★★★

A smoke testing tool inspired by [symm/vape](#)

Installation

Install via composer:

```
composer require brunty/cigar --dev
```

To use

Create a `.cigar.json` file that contains an array of json objects specifying the `url`, `status` and (optional) `content` to check.

```
[  
  {  
    "url": "http://httpbin.org/status/418",  
    "status": 418,  
    "content": "teapot"  
  },  
  {  
    "url": "http://httpbin.org/status/200",  
    "status": 200  
  },  
  {  
    "url": "http://httpbin.org/status/304",  
    "status": 304  
  },  
  {  
    "url": "http://httpbin.org/status/500",  
    "status": 500  
  }  
]
```

Then run `vendor/bin/cigar` to have it check each of the URLs return the status code expected.

```
> vendor/bin/cigar  
✓ http://httpbin.org/status/418 [418:418] teapot  
✓ http://httpbin.org/status/200 [200:200]  
✓ http://httpbin.org/status/304 [304:304]  
✓ http://httpbin.org/status/500 [500:500]
```

BASELINE FIRST

```
$ ./bin/cigar
✓ http://httpbin.org/status/418 [418:418] teapot
✓ http://httpbin.org/status/200 [200:200]
✓ http://httpbin.org/status/304 [304:304]
✓ http://httpbin.org/status/500 [500:500]

[4/4] passed in 3.091s
```

👉 setup xhprof

```

#!/usr/bin/env php
<?php

if (extension_loaded( name: 'xhprof' )) {
    xhprof_enable( flags: XHPROF_FLAGS_CPU + XHPROF_FLAGS_MEMORY);
}

// all the code

$xhprof_data = xhprof_disable();

$XHPROF_ROOT = "/Users/Kat/Code/github.com/longxinH/xhprof";
include_once $XHPROF_ROOT . "/xhprof_lib/utils/xhprof_lib.php";
include_once $XHPROF_ROOT . "/xhprof_lib/utils/xhprof_runs.php";

$xhprof_runs = new XHProfRuns_Default('/Users/Kat/Code/github.com/longxinH/xhprof/xhprof_html/dumps');
$run_id = $xhprof_runs->save_run($xhprof_data, "xhprof_testing");

echo "http://localhost/index.php?run={$run_id}&source=xhprof_testing\n";
exit(0);
|

```

- paste in your code
- or require/include from separate files
- or use the auto_prepend_file and auto_append_file ini directives

http://localhost/index.php?run=5a846a864c5da&source=xhprof_testing

Overall Summary											
Total Incl. Wall Time (microsec): 3,064,404 microsecs											
Total Incl. CPU (microsecs): 2,638,577 microsecs											
Total Incl. MemUse (bytes): 1,654,512 bytes											
Total Incl. PeakMemUse (bytes): 1,661,592 bytes											
Number of Function Calls: 2,366											

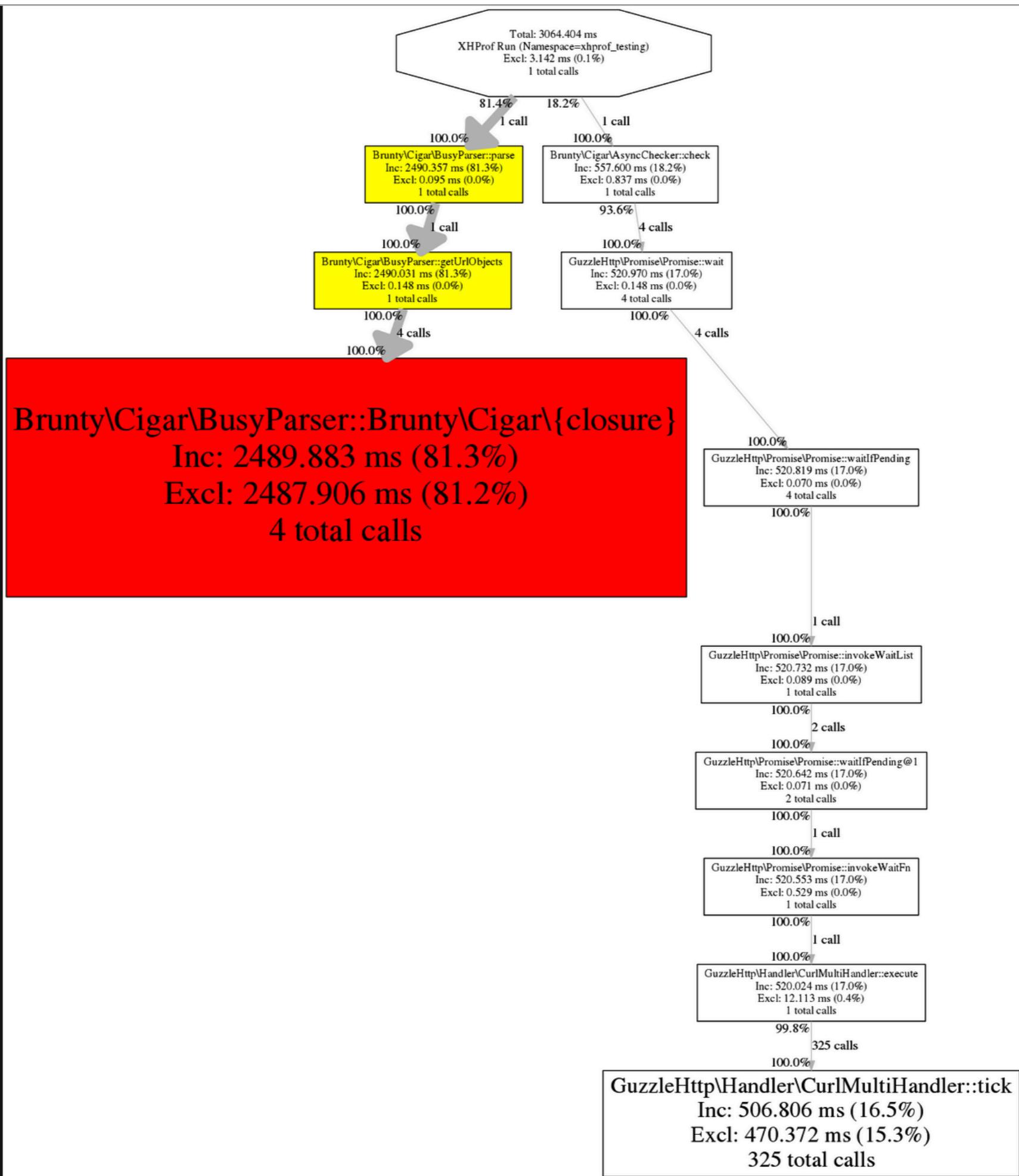
[\[View Full Callgraph\]](#)

Displaying top 100 functions: Sorted by Incl. Wall Time (microsec) [\[display all \]](#)

Function Name	Calls	Calls %	Incl. Wall Time (microsec)	IWall %	Excl. Wall Time (microsec)	EWall %	Incl. CPU (microsecs)	ICpu %	Excl. CPU (microsec)	ECPU %	N
main()	1	0.0%	3,064,404	100.0%	3,142	0.1%	2,638,577	100.0%	1,645	0.1%	1
Brunty\Cigar\BusyParser::parse	1	0.0%	2,490,357	81.3%	95	0.0%	2,481,707	94.1%	92	0.0%	
Brunty\Cigar\BusyParser::getUrlObjects	1	0.0%	2,490,031	81.3%	148	0.0%	2,481,464	94.0%	145	0.0%	
Brunty\Cigar\BusyParser::Brunty\Cigar\{closure}	4	0.2%	2,489,883	81.3%	2,487,906	81.2%	2,481,319	94.0%	2,480,537	94.0%	
Brunty\Cigar\AsyncCheck::check	1	0.0%	157,633	18.2%	837	0.0%	146,421	5.5%	794	0.0%	1
GuzzleHttp\Promise\Promise::wait	4	0.2%	520,970	17.0%	148	0.0%	115,882	4.4%	145	0.0%	
GuzzleHttp\Promise\Promise::waitIfPending	4	0.2%	520,819	17.0%	70	0.0%	115,732	4.4%	70	0.0%	
GuzzleHttp\Promise\Promise::invokeWaitList	1	0.0%	520,732	17.0%	89	0.0%	115,642	4.4%	84	0.0%	
GuzzleHttp\Promise\Promise::waitIfPending@1	2	0.1%	520,642	17.0%	71	0.0%	115,554	4.4%	68	0.0%	
GuzzleHttp\Promise\Promise::invokeWaitFn	1	0.0%	520,553	17.0%	529	0.0%	115,464	4.4%	531	0.0%	
GuzzleHttp\Handler\CurlMultiHandler::execute	1	0.0%	520,024	17.0%	12,113	0.4%	114,933	4.4%	12,052	0.5%	
GuzzleHttp\Handler\CurlMultiHandler::handle	205	12.7%	506,806	16.5%	470,370	15.3%	101,876	3.0%	65,150	2.5%	



baseline 3,064 sec



?

👉 setup xdebug

```
$ cat /usr/local/etc/php/7.2/conf.d/ext-xdebug.ini
[xdebug]
zend_extension="/usr/local/opt/php72-xdebug/xdebug.so"
xdebug.profiler_enable=1
xdebug.profiler_output_dir=/Users/Kat/xdebug_output_dir/cigar
xdebug.profiler_output_name=cachegrind.out.%p
```

👉 install qcachegrind (mac) / kcachegrind

brew install qcachegrind

Workaround for macOS High Sierra which doesn't have dot in /usr/bin but in /usr/local/bin:

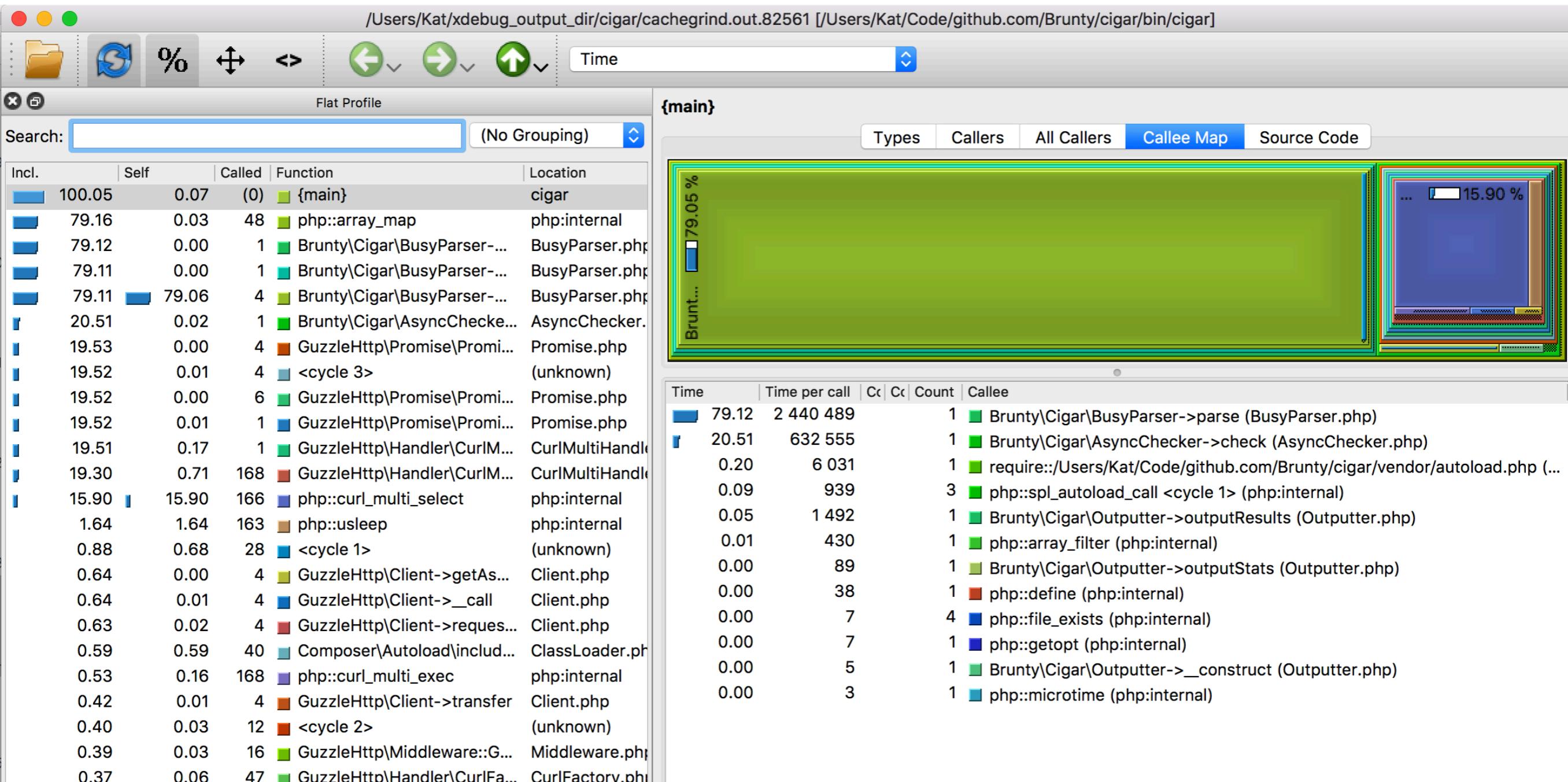
```
$ cd /Applications/qcachegrind.app/
Contents/MacOS

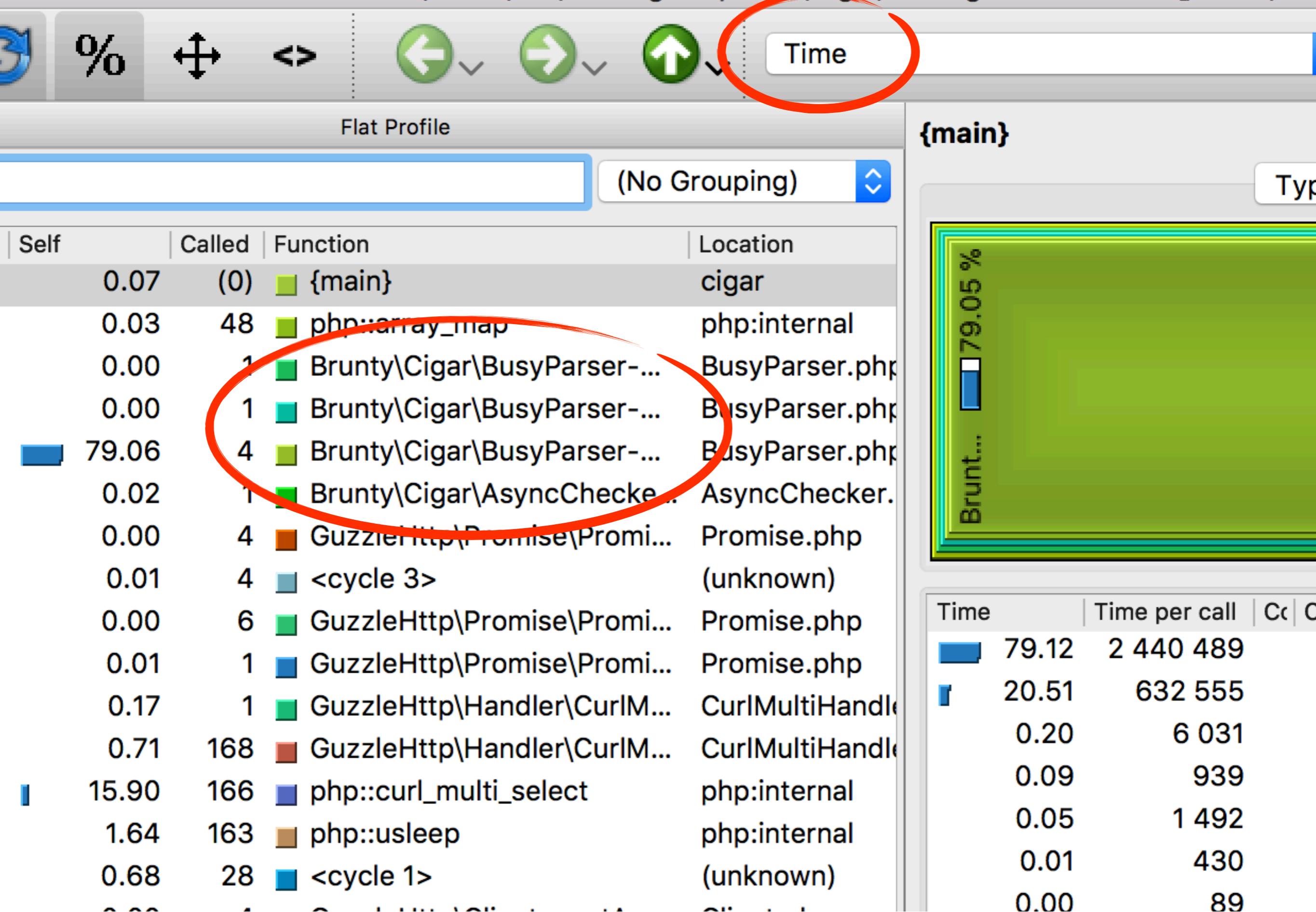
$ mv qcachegrind qcachegrind.bin
$ echo '#!/bin/bash
export PATH="$PATH:/usr/local/bin"
$(dirname $0)/qcachegrind.bin
exit 0' > qcachegrind

$ chmod +x qcachegrind
```

Kat at Dave in ~/xdebug_output_dir/cigar

```
$ ll
total 520
-rw-r--r-- 1 Kat  staff  252234 14 Feb 17:15 cachegrind.out.82561
```





{main}

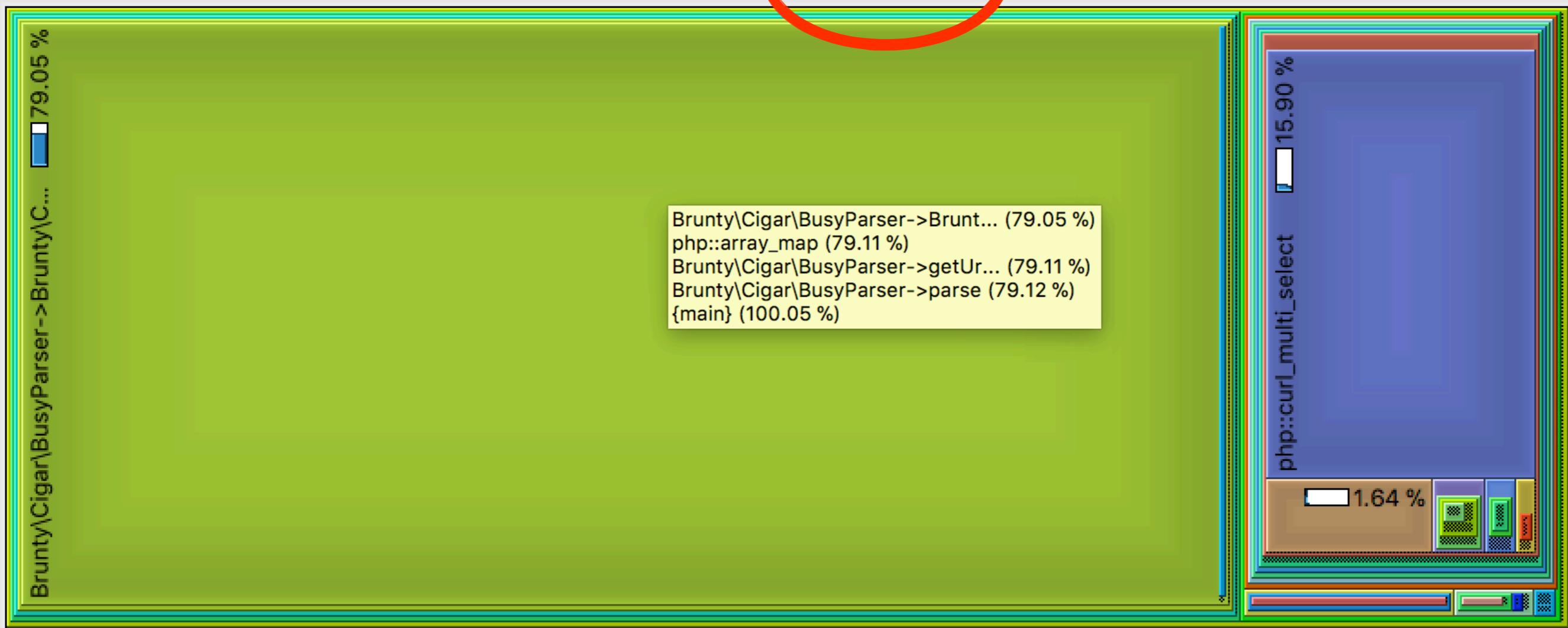
Types

Callers

All Callers

Callee Map

Source Code

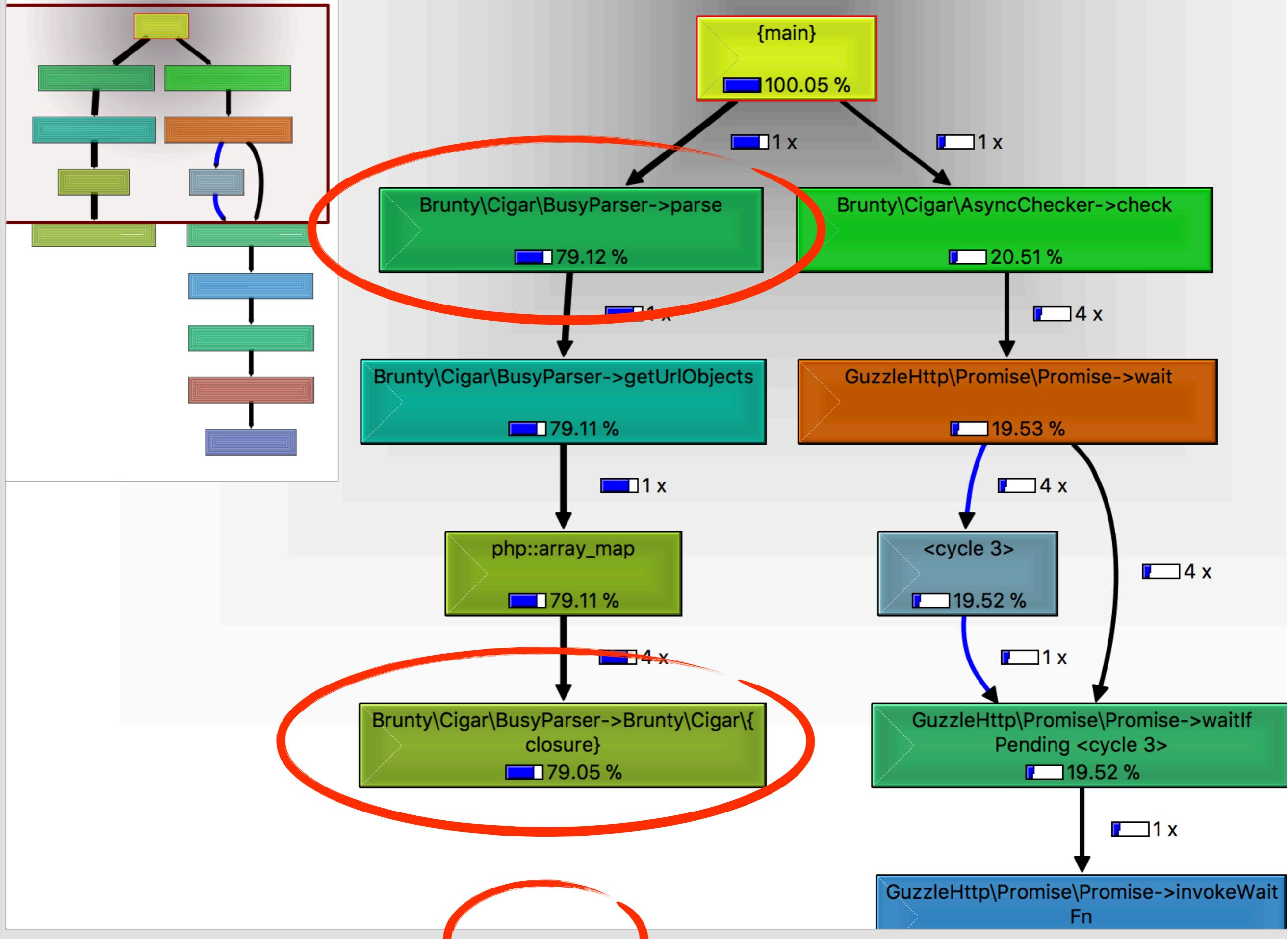


👉 look for big empty squares

[Types](#)[Callers](#)[All Callers](#)[Callee Map](#)[Source Code](#)

Brunty\Cigar\BusyParser->Brunt... (79.05 %)
php::array_map (79.11 %)
Brunty\Cigar\BusyParser->getUr... (79.11 %)
Brunty\Cigar\BusyParser->parse (79.12 %)
{main} (100.05 %)

15.90 %
php:curl_multi_select



Parts

Callees

Call Graph

All Callees

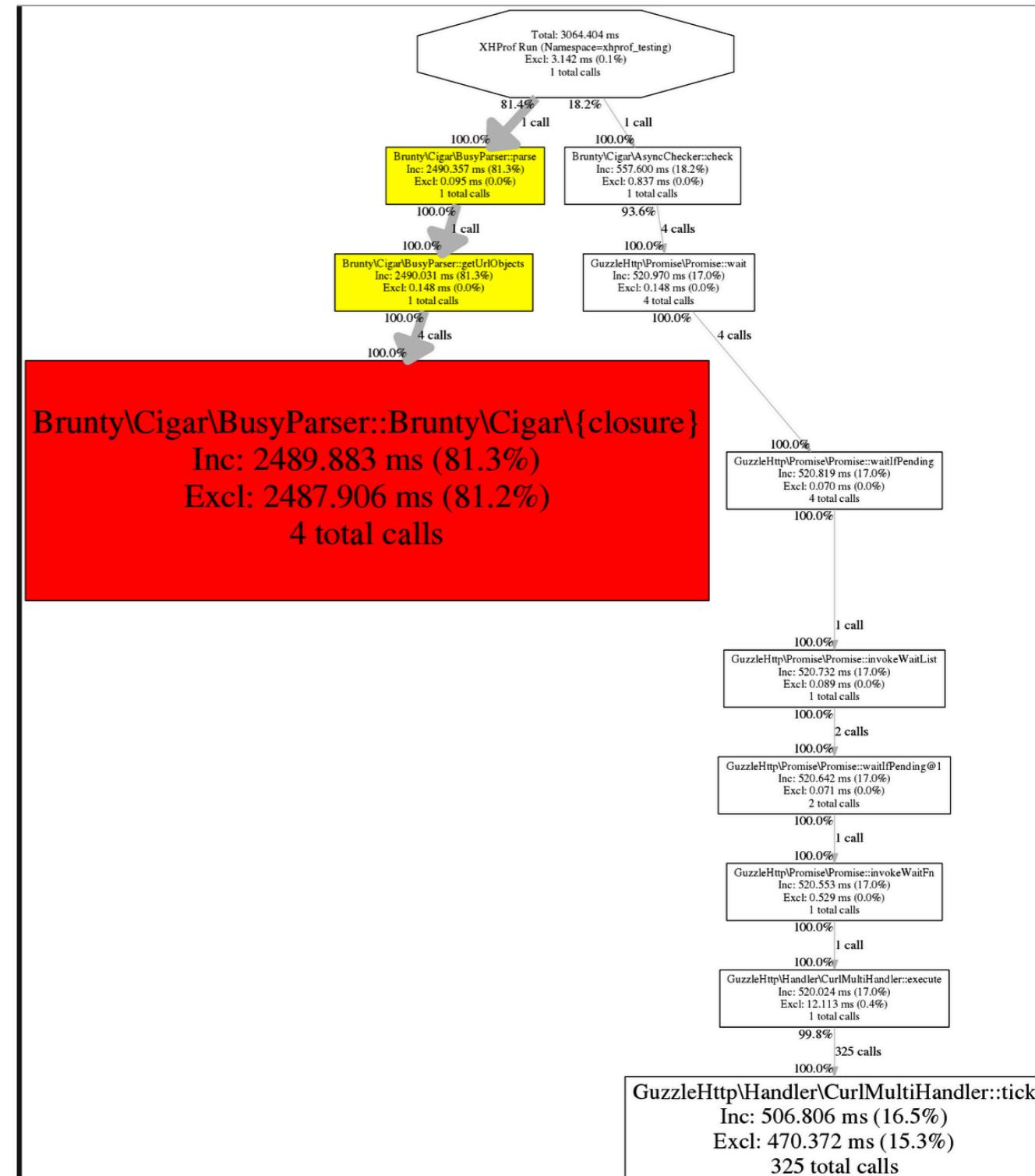
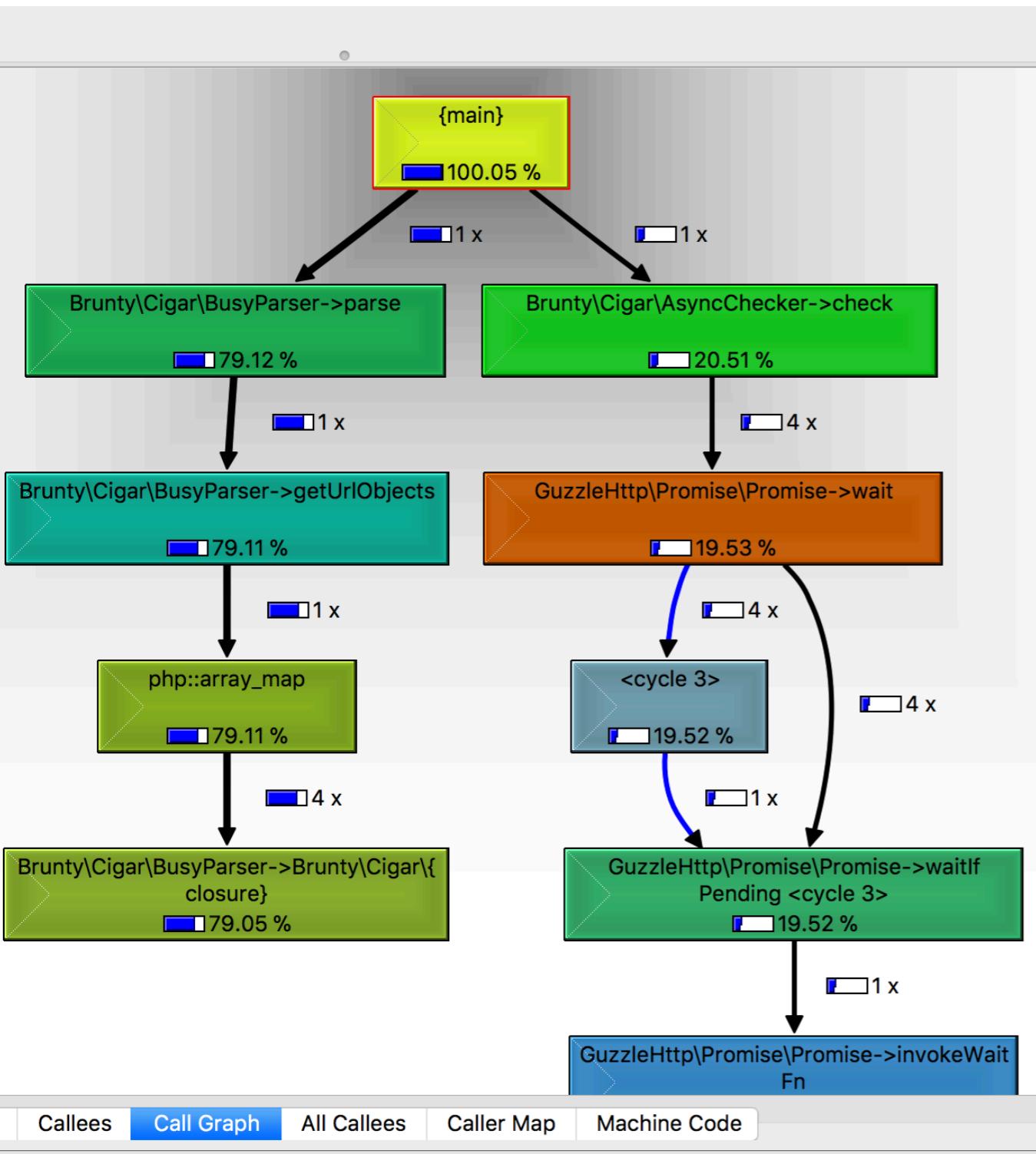
Caller Map

Machine Code

parse: 79.12%

(two separate runs)

parse: 81.4%





Flat Profile

Incl.	Self	Called	Function	Location
100.05	0.07	(0)	{main}	cigar
79.16	0.03	48	php::array_map	php:internal
79.12	0.00	1	Brunty\Cigar\BusyParser...	BusyParser.php
79.11	0.00	1	Brunty\Cigar\BusyParser...	BusyParser.php
79.11	79.06	4	Brunty\Cigar\BusyParser...	BusyParser.php
20.51	0.02	1	Brunty\Cigar\AsyncChecke...	AsyncChecker.
19.53	0.00	4	GuzzleHttp\Promise\Promi...	Promise.php
19.52	0.01	4	<cycle 3>	(unknown)
19.52	0.00	6	GuzzleHttp\Promise\Promi...	Promise.php
19.52	0.01	1	GuzzleHttp\Promise\Promi...	Promise.php

e	Time per call	Cc	Cc	Count	Callee
79.11	610 002			4	Brunty\Cigar\BusyParser->Brunty\Cigar\{closure} (BusyParser.php)
0.02	15			39	GuzzleHttp\Psr7\Response->GuzzleHttp\Psr7\{closure} (MessageTrait.p...
0.00	15			8	GuzzleHttp\Psr7\Request->GuzzleHttp\Psr7\{closure} (MessageTrait.php)

Call Tree

Legend: Function Method Constructor Destructor

Filter: (highlighted)

#	Time	Count	Source
0			--- From '/Users/Kat/Code/github.com/Brunty/cigar/src/BusyParser.php' ---
24			
25			private function getUrlObjects(\$urls)
26			{
27	79.06	1	return array_map(function (\$value) {
28			\$pi = 4;
29			\$top = 4;
30			\$bot = 3;
...			...
37			\$bot += 2;
38			}
39			
40	0.00	1	return new Url(\$value['url'], \$value['status'], \$value['content'] ?? null);
	0.04	5	4 call(s) to 'Brunty\Cigar\Url->__construct' (Url.php)
		1	1 call(s) to '<cycle 1>' via php::spl_autoload_call'
41			}, \$urls);
42			}
43			

```
private function getUrlObjects($urls)
{
    return array_map(function ($value) {
        $pi = 4;
        $top = 4;
        $bot = 3;
        $minus = true;
        $accuracy = 2000000;

        for ($i = 0; $i < $accuracy; $i++) {
            $pi += ($minus ? -($top / $bot) : ($top / $bot));
            $minus = ($minus ? false : true);
            $bot += 2;
        }

        return new Url($value['url'], $value['status'], content: $value['content'] ?? null);
    }, $urls);
}
```

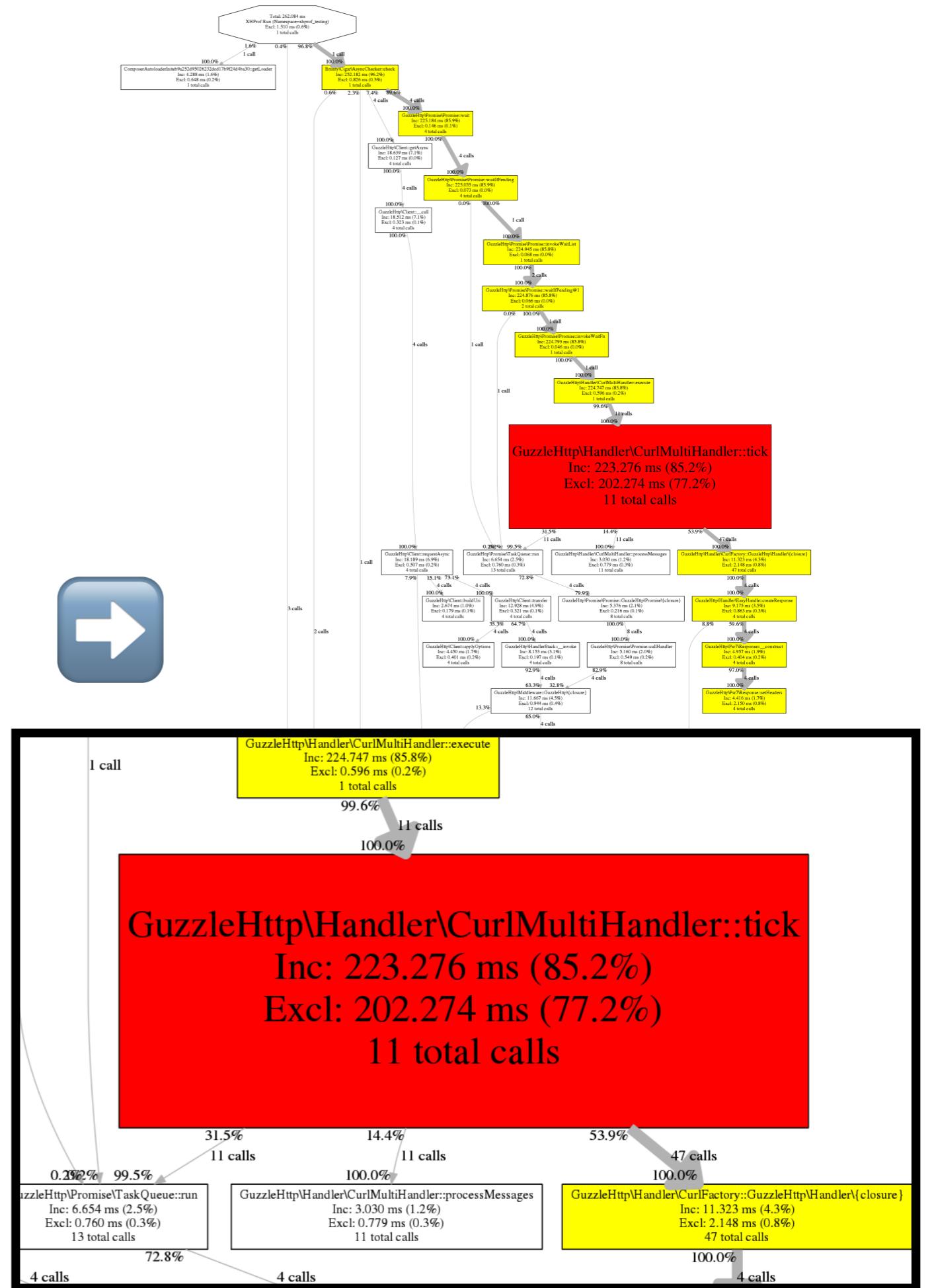
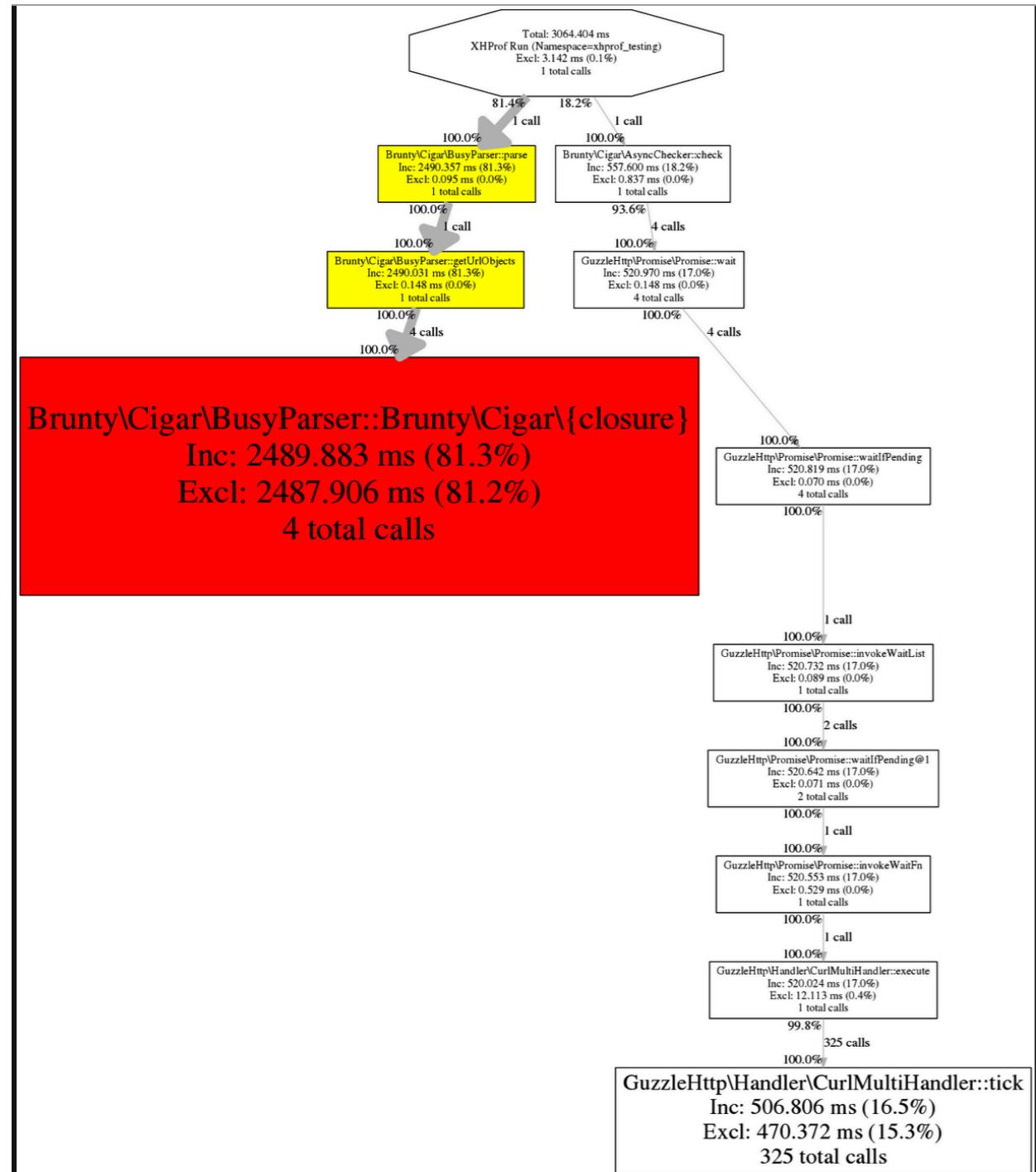


```
$ ./bin/cigar
✓ http://httpbin.org/status/418 [418:418] teapot
✓ http://httpbin.org/status/200 [200:200]
✓ http://httpbin.org/status/304 [304:304]
✓ http://httpbin.org/status/500 [500:500]

[4/4] passed in 0.311s
```



3,064 sec vs 0.311 sec



VERY NICE



GREAT SUCCESS

0.34	0.00	40	 Composer\Autoload\ClassLoader->fi...	ClassLoader.php
0.34	0.08	40	 Composer\Autoload\ClassLoader->fi...	ClassLoader.php
0.34	0.00	40	 Composer\Autoload\ClassLoader->lo...	ClassLoader.php
0.29	0.29	123	 php::strtolower	php:internal
0.28	0.28	4	 GuzzleHttp\Psr7\Request->updateHo...	Request.php
0.27	0.00	4	 GuzzleHttp\normalize_header_keys	functions.php

php::strtolower

Types	Callers	All Callers	Callee Map	Source Code
Memory	Memory per call	Cc Cc Count	Caller	
0.10	42	39	 GuzzleHttp\normalize_header_keys (functions.php)	
0.10	42	39	 GuzzleHttp\Psr7\Response->setHeaders (MessageTrait.php)	
0.03	34	16	 GuzzleHttp\Psr7\Request->hasHeader (MessageTrait.php)	
0.02	36	8	 GuzzleHttp\Psr7_caseless_remove (functions.php)	
0.02	36	8	 GuzzleHttp\Psr7\Request->setHeaders (MessageTrait.php)	
0.01	40	4	 GuzzleHttp\Psr7\Request->getHeader (MessageTrait.php)	
0.00	40	1	 GuzzleHttp\Psr7\Response->hasHeader (MessageTrait.php)	

0.19	8	■ GuzzleHttp\Psr7\Request->setHeaders	MessageTrait.php
0.17	4	■ GuzzleHttp\Handler\CurlFactory->ap...	CurlFactory.php
0.37	9	■ php::fopen	php:internal
0.00	40	■ Composer\Autoload\ClassLoader->fi...	ClassLoader.php
0.08	40	■ Composer\Autoload\ClassLoader->fi...	ClassLoader.php

php::fopen

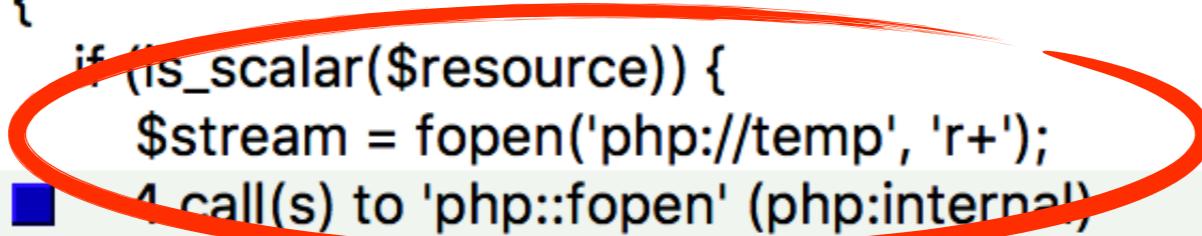
Types Callers All Callers Callee Map Source Code

Memory	Memory per call	Cc	Cc	Count	Caller
0.19	760	4	4	4	GuzzleHttp\Psr7\stream_for (functions.php)
0.15	600	4	4	4	GuzzleHttp\Handler\CurlFactory->applyHandlerOptions (Curl...
0.04	600	1	1	1	Brunty\Cigar\Parser->getFileContents (Parser.php)

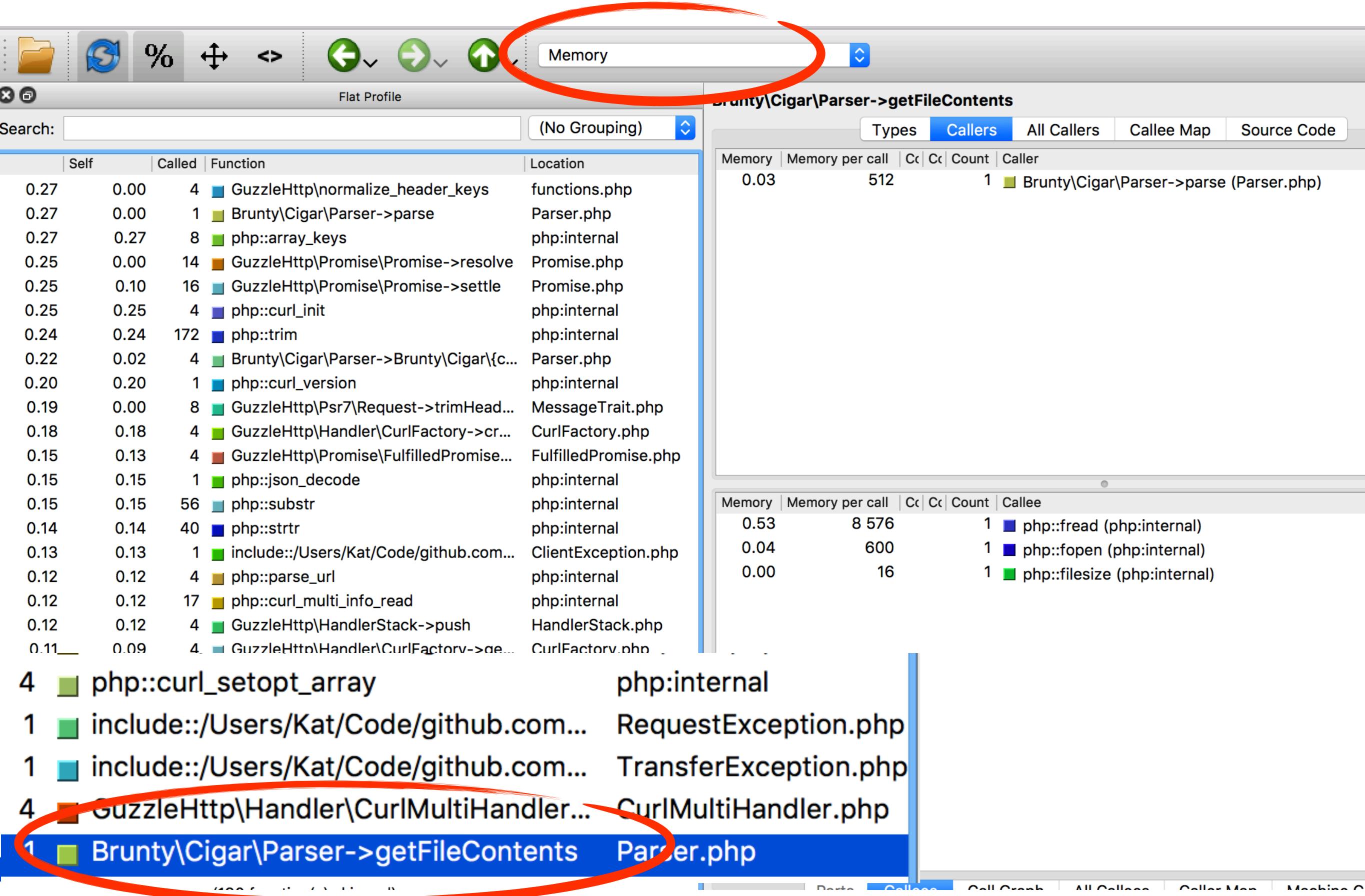
GuzzleHttp\Psr7\stream_for

Types Callers All Callers Callee Map Source Code

#	Memory	Cc	Source
76			* @throws \InvalidArgumentException if the \$resource arg is not valid.
77			*/
78	0.09		function stream_for(\$resource = "", array \$options = [])
79			{
80			if (is_scalar(\$resource)) {
81			\$stream = fopen('php://temp', 'r+');
82	0.19		4 call(s) to 'php::fopen' (php:internal)
83			if (\$resource === '') {
84			fwrite(\$stream, \$resource);
85			fseek(\$stream, 0);
86			}
			return new Stream(\$stream, \$options);
87	0.01		4 call(s) to 'GuzzleHttp\Psr7\Stream->__construct' (Stream.php)
88	6.52		1 call(s) to '<cycle 1> via php::spl_autoload_call'
			}



```
if (is_scalar($resource)) {  
    $stream = fopen('php://temp', 'r+');
```



Brunty\Cigar\Parser->getFileContents

Types

Callers

All Callers

Callee Map

Source Code

Memory | Memory per call | Cc| Cc| Count | Caller

0.03

512

1

Brunty\Cigar\Parser->parse (Parser.php)

```
private function getFileContents($filename)
{
    $file = fopen($filename, mode: 'r');

    $contents = fread($file, filesize($filename));

    fclose($file);

    return $contents;
}
```

Memory | Memory per call | Cc| Cc| Count | Callee

0.53

8 576

1

php::fread (php:internal)

0.04

600

1

php::fopen (php:internal)

0.00

16

1

php::filesize (php:internal)

Brunty\Cigar\Parser->getFileContents

Types

Callers

All Callers

Callee Map

Source Code

Memory | Memory per call | Cc| Cc| Count | Caller

0.03

496

1

■ Brunty\Cigar\Parser->parse (Parser.php)

vs 512 before

Memory | Memory per call | Cc| Cc| Count | Callee

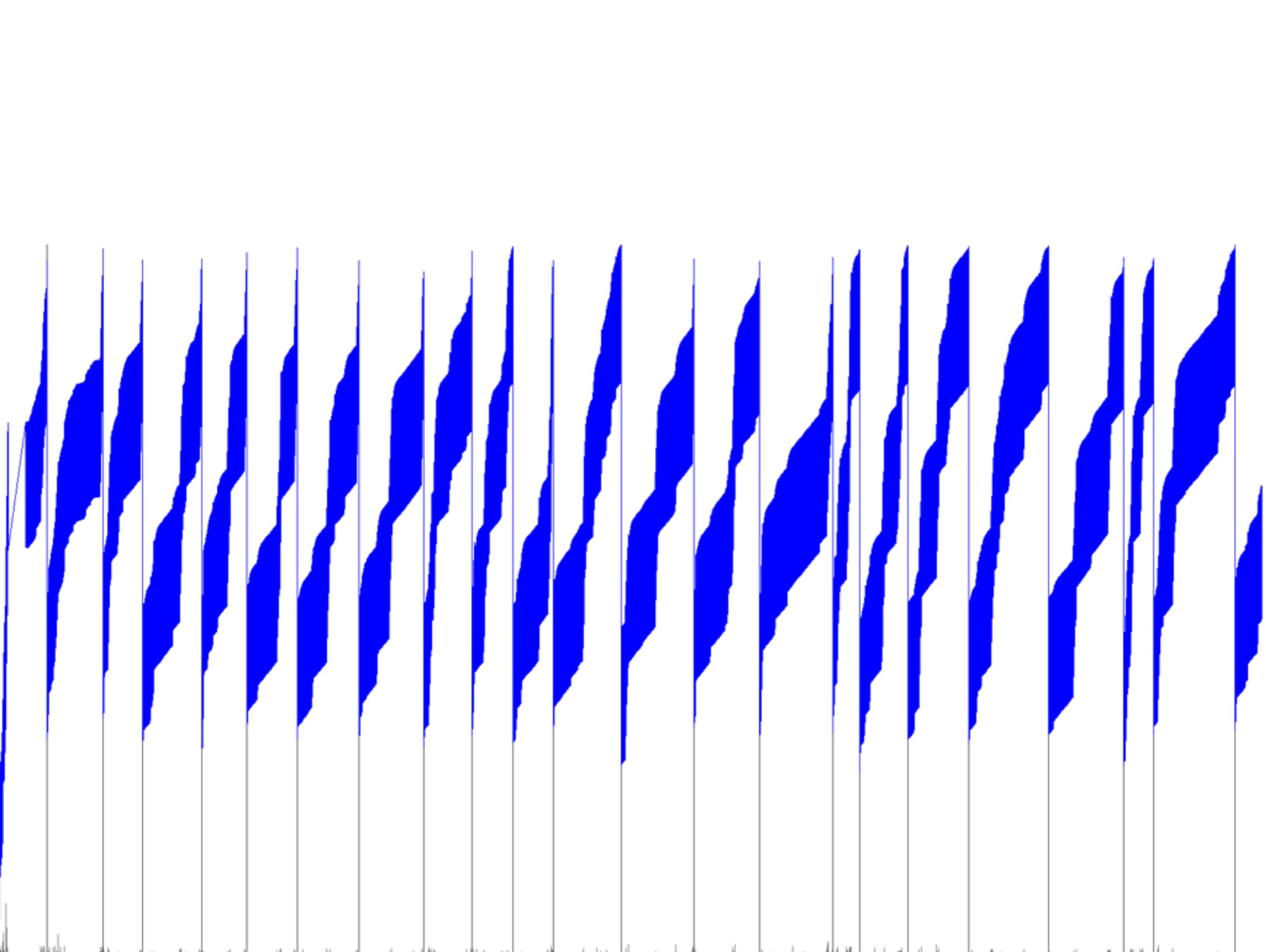
0.03

496

1

■ php::file_get_contents (php:internal)





👉 <https://github.com/BitOne/php-meminfo>

```
$outputter->outputResults($results);
$outputter->outputStats($passedResults, $results, $start);

meminfo_dump(fopen( filename: '/tmp/cigar-memory-before.json' , mode: 'w' ));

if (count($passedResults) !== count($results)) {
    exit(1);
}

exit(0);
```

👉 inspect using php-meminfo's analyser

```
$ ./bin/analyzer summary /tmp/cigar-memory-before.json
```

Type	Instances Count	Cumulated Self Size (bytes)
string	44	2379
array	15	1080
integer	11	176
Brunty\Cigar\Url	8	576
Brunty\Cigar\Result	8	576
boolean	6	96
null	6	96
unknown	3	48
float	2	32
Brunty\Cigar\Outputter	1	72
Brunty\Cigar\Parser	1	72

/bin/cigar (~ index.php)

```
$parser = new Parser();

try {
    $domains = $parser->parse($file);
} catch (\Throwable $e) {
    $outputter->writeErrorLine('Unable to parse .cigar.json file');
    exit(1);
}

$results = (new AsyncChecker())->check($domains);
$passedResults = array_filter($results, function (Result $result) {
    return $result->hasPassed();
});

$outputter->outputResults($passedResults);
```

```
$ ./bin/analyzer query -v -f "class=Brunty\Cigar\Parser" /tmp/cigar-memory-before.json
+-----+-----+-----+
| Item ids | Item data | Children |
+-----+-----+-----+
| 0x108a63300 | Type: object | |
| | Class: Brunty\Cigar\Parser | |
| | Object Handle: 2 | |
| | Size: 72 B | |
| | Is root: Yes | |
| | Execution Frame: <GLOBAL> | |
| | Symbol Name: parser | |
+-----+-----+-----+
```

```
$ ./bin/analyzer -v ref-path 0x108a63300 /tmp/cigar-memory-before.json
```

```
Found 2 paths
```

```
Path from 0x108a633a0
```

```
+-----+  
| Id: 0x108a63300 |  
| Type: object |  
| Class: Brunty\Cigar\Parser |  
| Object Handle: 2 |  
| Size: 72 B |  
| Is root: Yes |  
| Execution Frame: <GLOBAL> |  
| Symbol Name: parser |  
| Children count: 0 |  
+-----+
```

```
    ^  
    |  
    parser  
    |  
    |
```

```
+-----+  
| Id: 0x108a633a0 |  
| Type: array |  
| Size: 72 B |  
| Is root: Yes |  
| Execution Frame: <GLOBAL> |  
| Symbol Name: GLOBALS |  
| Children count: 23 |  
+-----+
```

```
Path from 0x108a63300
```

```
+-----+  
| Id: 0x108a63300 |  
| Type: object |  
| Class: Brunty\Cigar\Parser |  
| Object Handle: 2 |  
| Size: 72 B |  
| Is root: Yes |  
| Execution Frame: <GLOBAL> |  
| Symbol Name: parser |  
| Children count: 0 |  
+-----+
```

```
    ^  
    |  
    <self>  
    |
```

```
+-----+  
| Id: 0x108a63300 |  
| Type: object |  
| Class: Brunty\Cigar\Parser |  
| Object Handle: 2 |  
| Size: 72 B |  
| Is root: Yes |  
| Execution Frame: <GLOBAL> |  
| Symbol Name: parser |  
| Children count: 0 |  
+-----+
```

```
$ ./bin/analyzer query -v -f "id=0x108a633a0" /tmp/cigar-memory-before.json
```

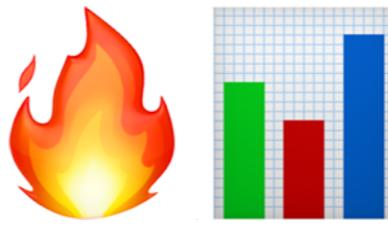
Item ids	Item data	Children
0x108a633a0	Type: array Size: 72 B Is root: Yes Execution Frame: <GLOBAL> Symbol Name: GLOBALS	<u>_GET: 0x108a63100</u> <u>_POST: 0x108a63120</u> <u>_COOKIE: 0x108a63140</u> <u>_FILES: 0x108a63160</u> <u>argv: 0x108a63180</u> <u>argc: 0x108a631a0</u> <u>_ENV: 0x108a631c0</u> <u>_REQUEST: 0x108a631e0</u> <u>_SERVER: 0x108a63200</u> <u>start: 0x108a63220</u> <u>autoloadFileDirectory: 0x108a63240</u> <u>options: 0x108a63260</u> <u>content: 0x108a63280</u> <u>version: 0x108a632a0</u> <u>outputter: 0x108a632c0</u> <u>file: 0x108a632e0</u> <u>parser: 0x108a63300</u> <u>domains: 0x108a63320</u> <u>e: 0x108a63340</u> <u>results: 0x108a63360</u> <u>passedResults: 0x108a63380</u> <u>GLOBALS: 0x108a633a0</u> <u>__composer_autoload_files: 0x108a633c0</u>

```
try {
    $domains = (new Parser())->parse($file);
} catch (\Throwable $e) {
    $outputter->writeErrorLine('Unable to parse .cigar.json');
    exit(1);
}
```

```
$ ./bin/analyzer summary /tmp/cigar-memory-after.json
```

Type	Instances Count	Cumulated Self Size (bytes)
string	44	2379
array	15	1080
integer	11	176
Brunty\Cigar\Url	8	576
Brunty\Cigar\Result	8	576
boolean	6	96
null	6	96
unknown	3	48
float	2	32
Brunty\Cigar\Outputter	1	72





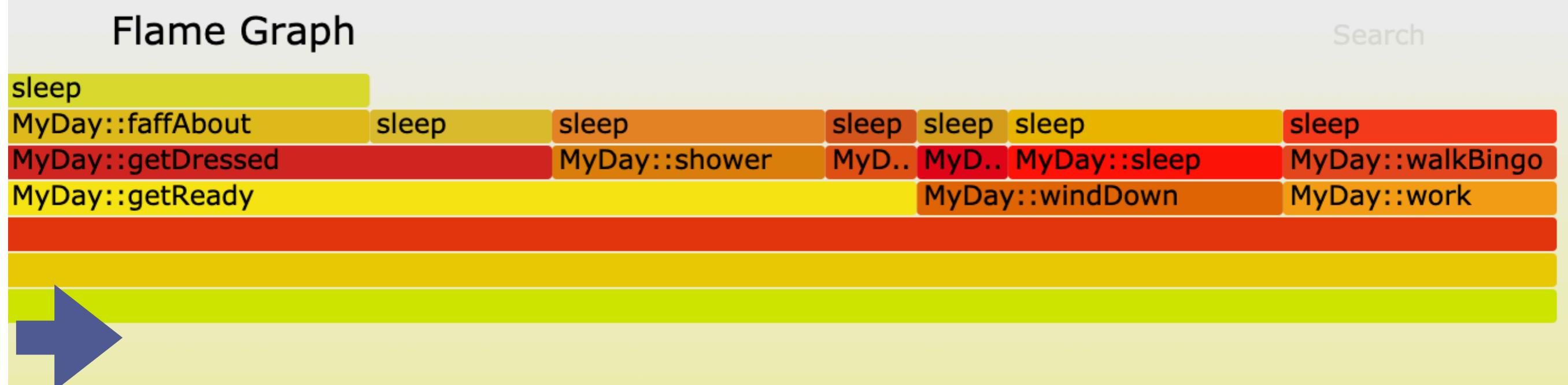
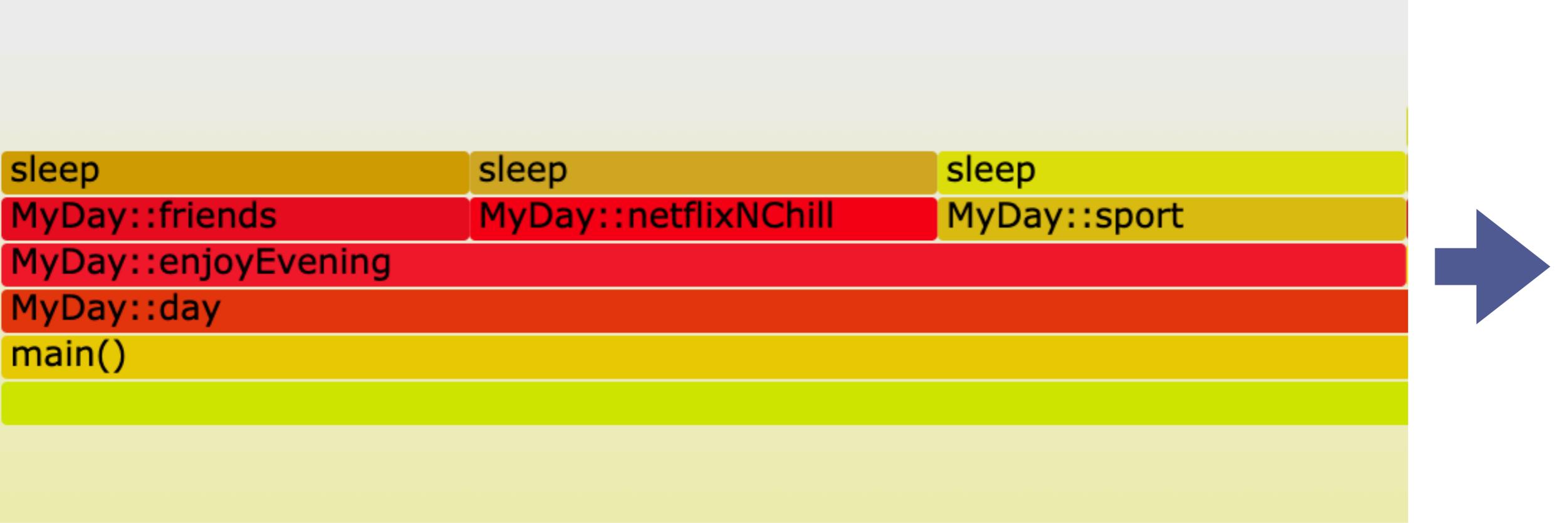
GENERATING FLAMEGRAPHS

- phpspy, or...
 - `xhprof_sample_enable()` / `xhprof_sample_disable()`
 - [`https://github.com/msonnabaum/xhprof-flamegraphs`](https://github.com/msonnabaum/xhprof-flamegraphs)
& [`https://github.com/brendangregg/FlameGraph`](https://github.com/brendangregg/FlameGraph)
- ```
$./xhprof-sample-to-flamegraph-stacks path-to-
your.xhprof | path-to-your-flamegraph.pl >
xhprof-flamegraph.svg
```

## Flame Graph

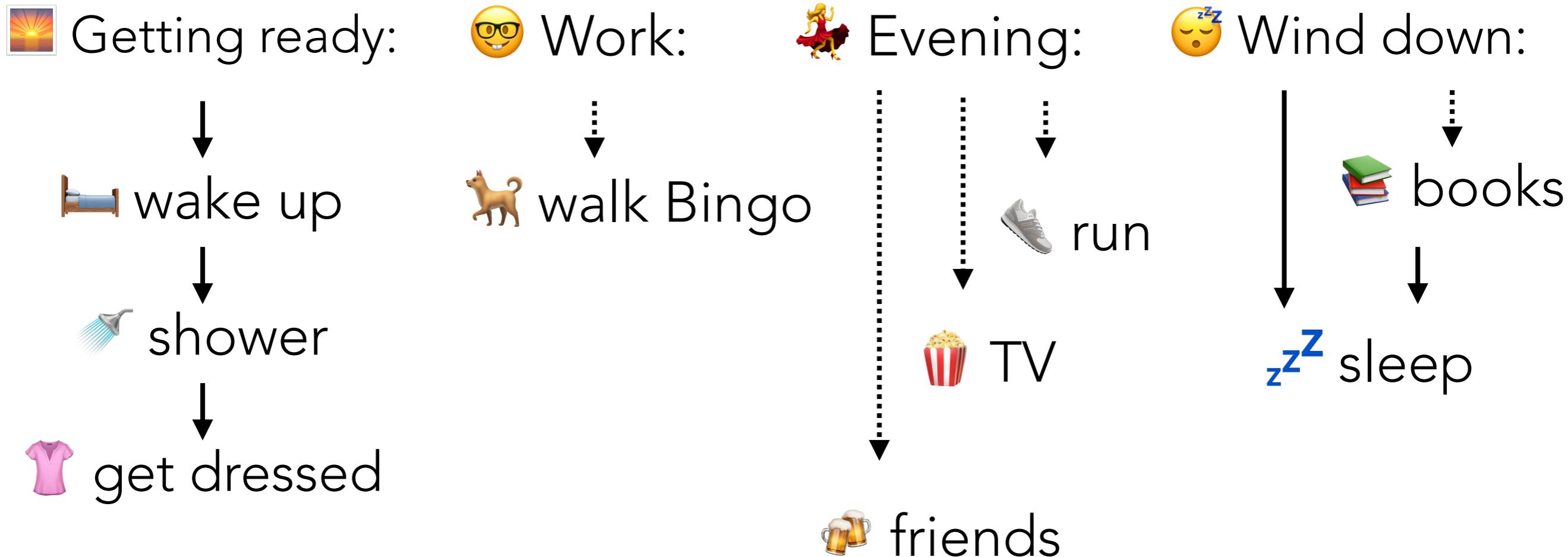
Search



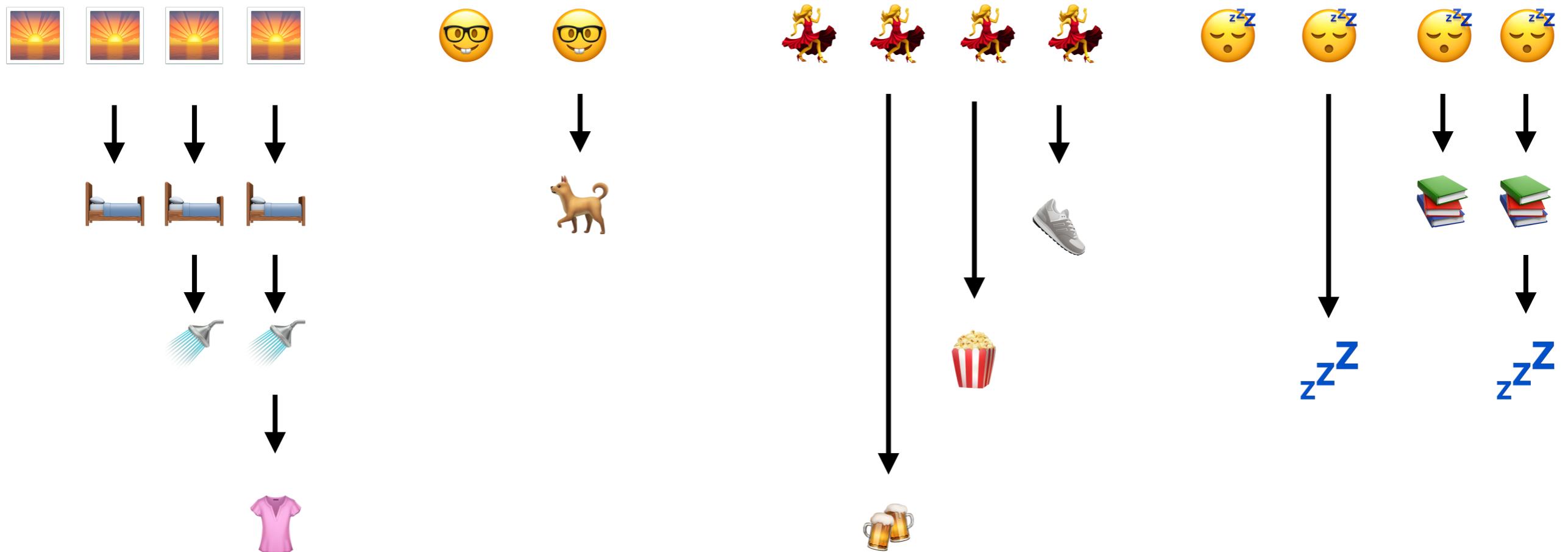


# UNDERSTANDING FLAMEGRAPHS

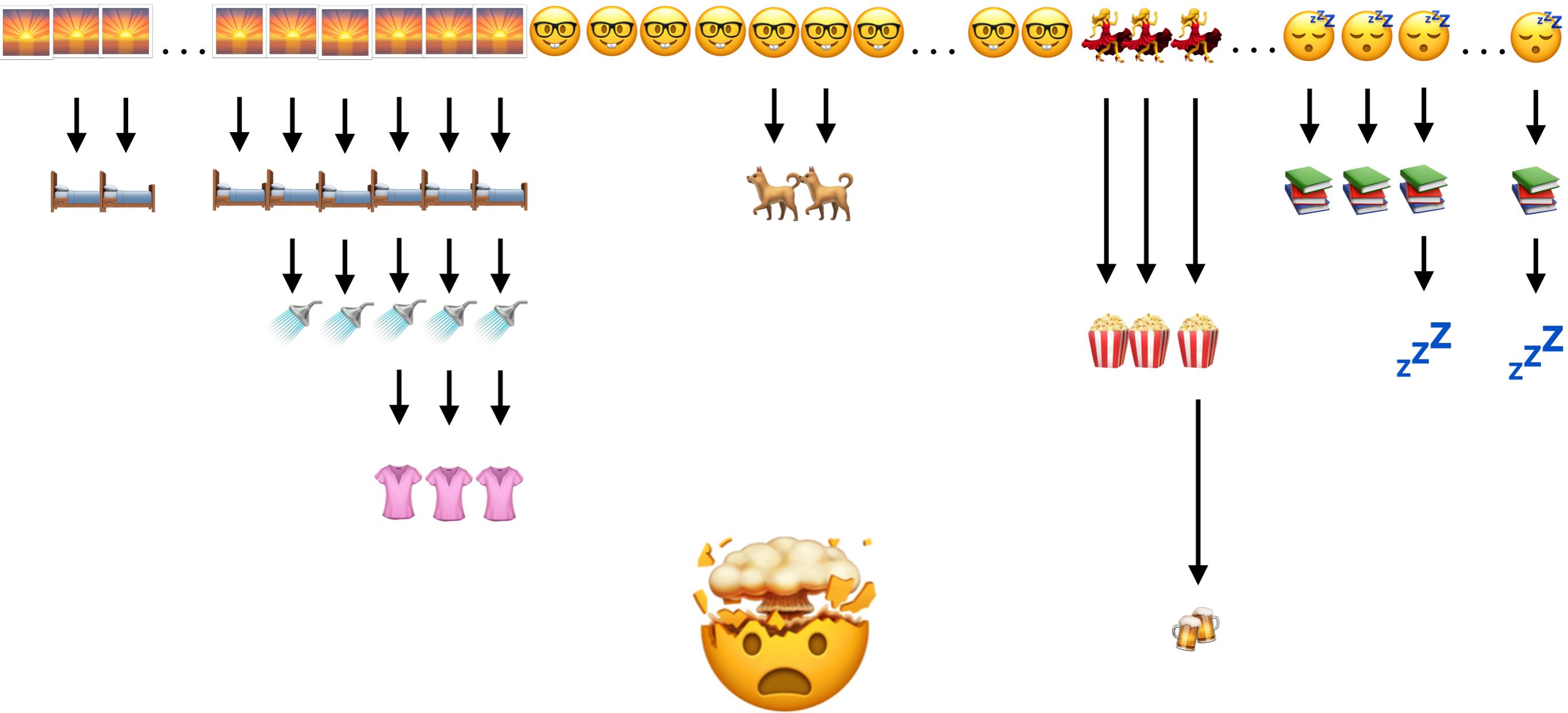
My day:



# POSSIBLE STACK TRACES



# POSSIBLE STACK TRACES



## Flame Graph

## Flame Graph

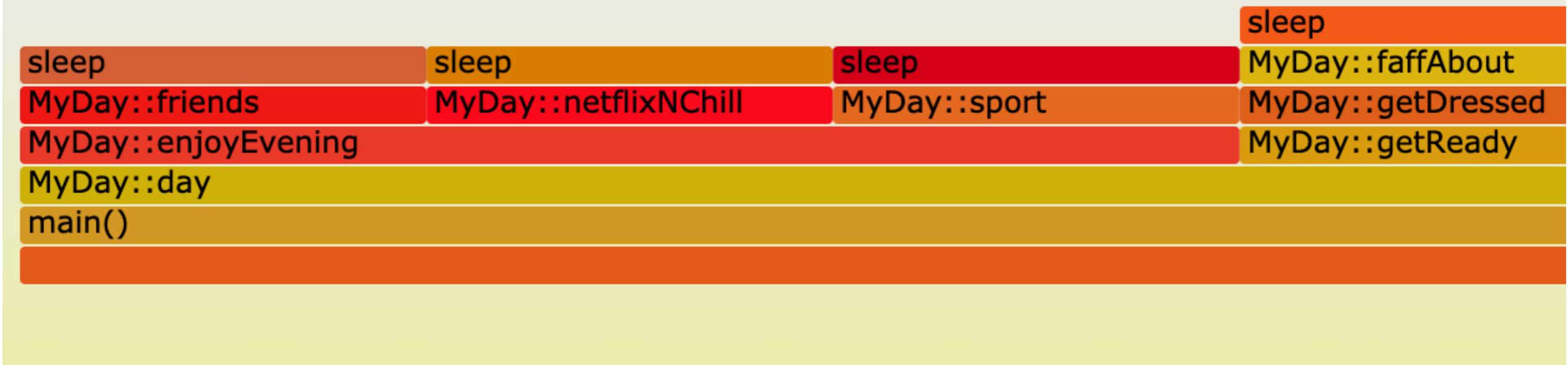


me Graph

Search

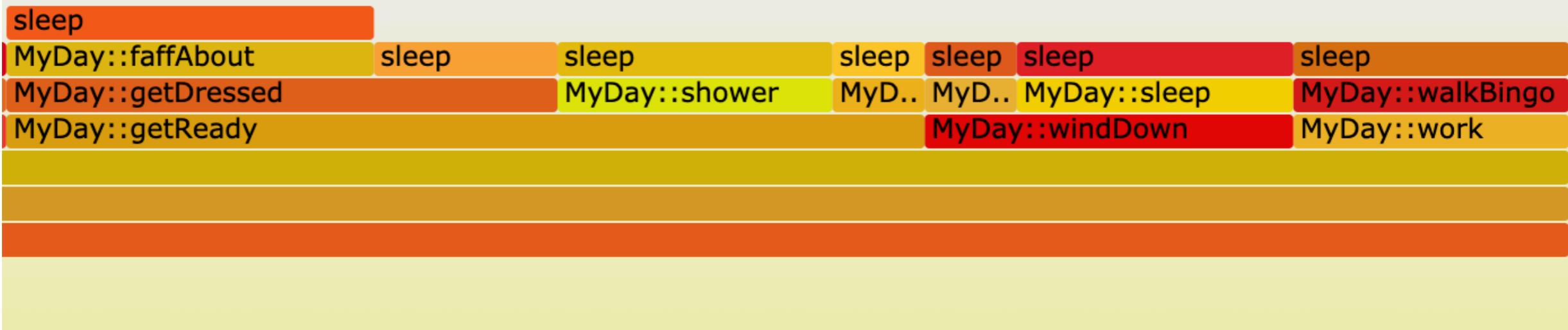


## Flame Gra



## Flame Graph

Search





# TAKEAWAYS

"Do I need (to do) this?"



# PRIORITISE

PRIORITISE

CUSTOMISE

PRIORITISE

CUSTOMISE

MEASURE

PRIORITISE

CUSTOMISE

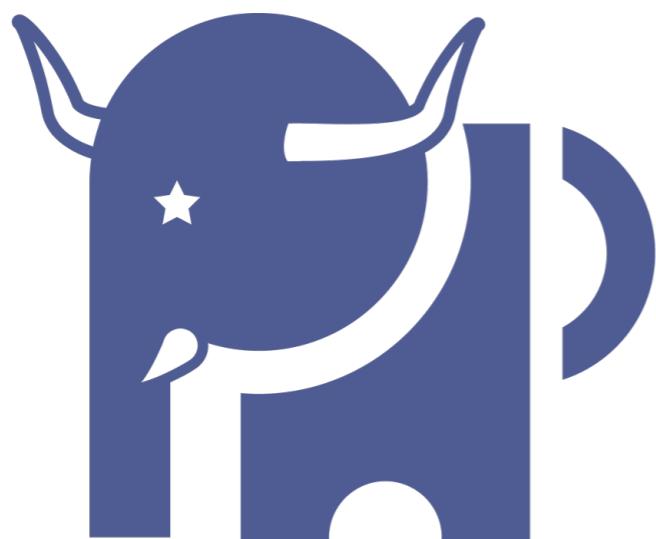
MEASURE

ITERATE

# THANK YOU!

JOIND.IN:

[HTTPS://JOIND.IN/TALK/09F01](https://joind.in/talk/09f01)



LONGHORN  
PHP CONFERENCE



TWITTER:  
@KASIAZIEN