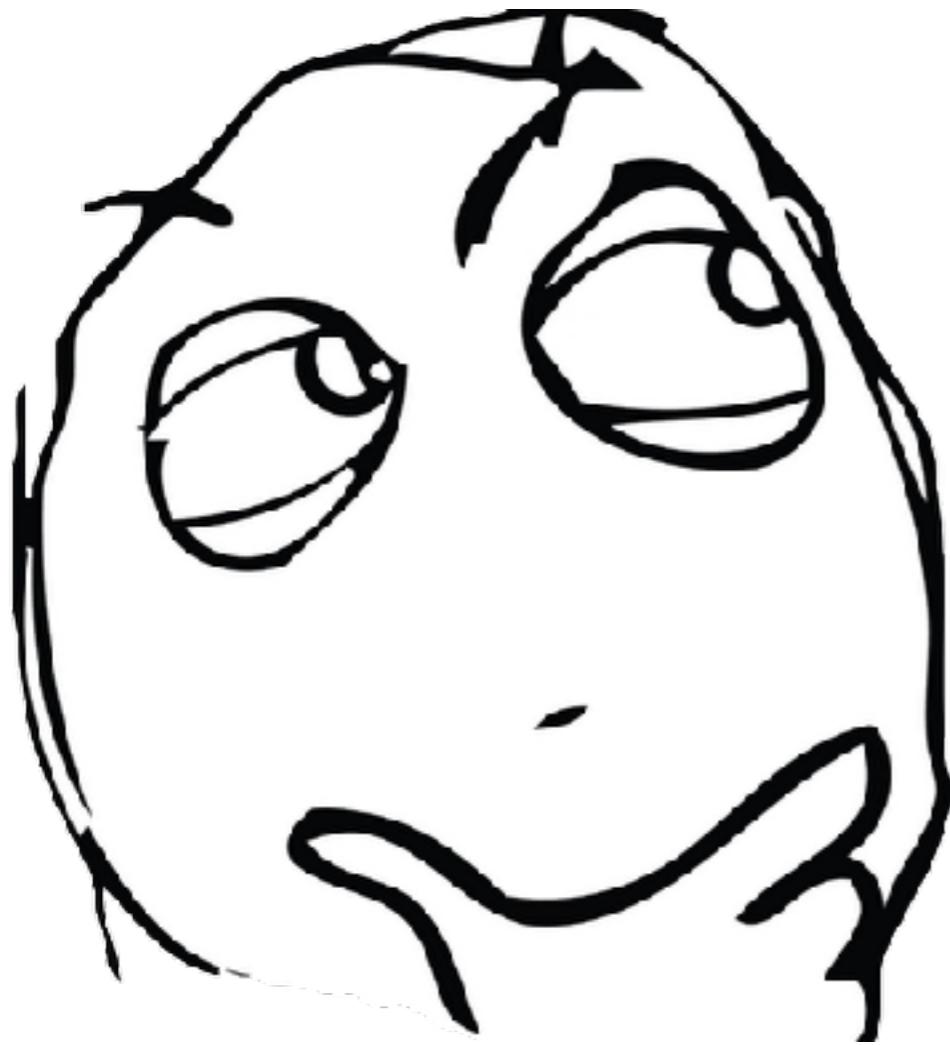


PERFORMANCE OPTIMISATION

HOW DO I GO ABOUT IT?

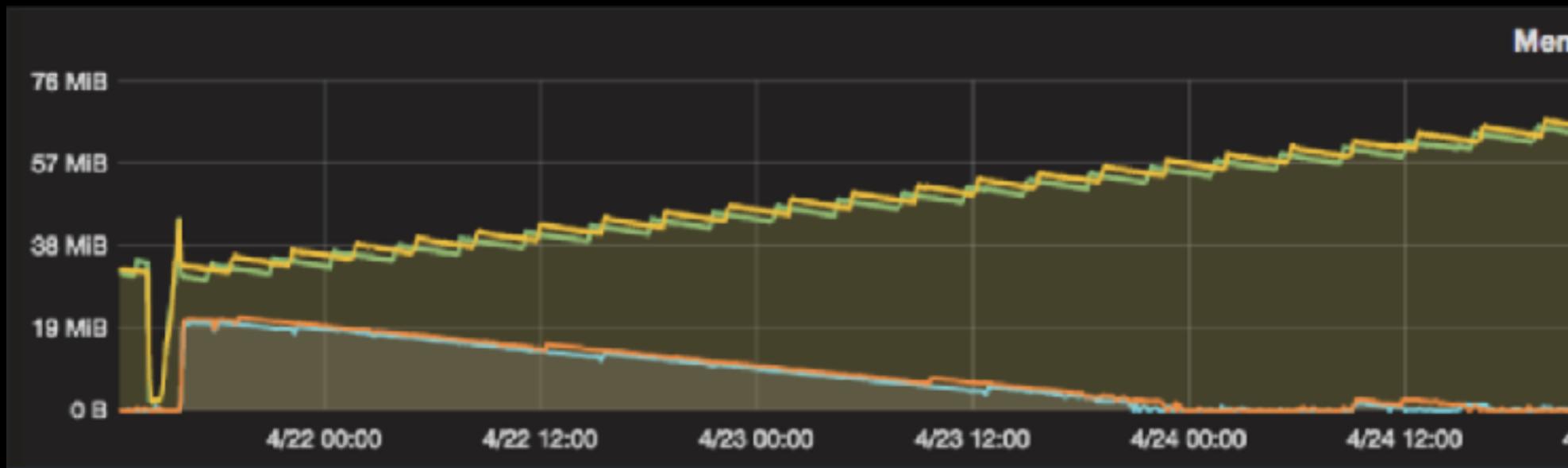


KAT ZIEŃ | @KASIAZIEN

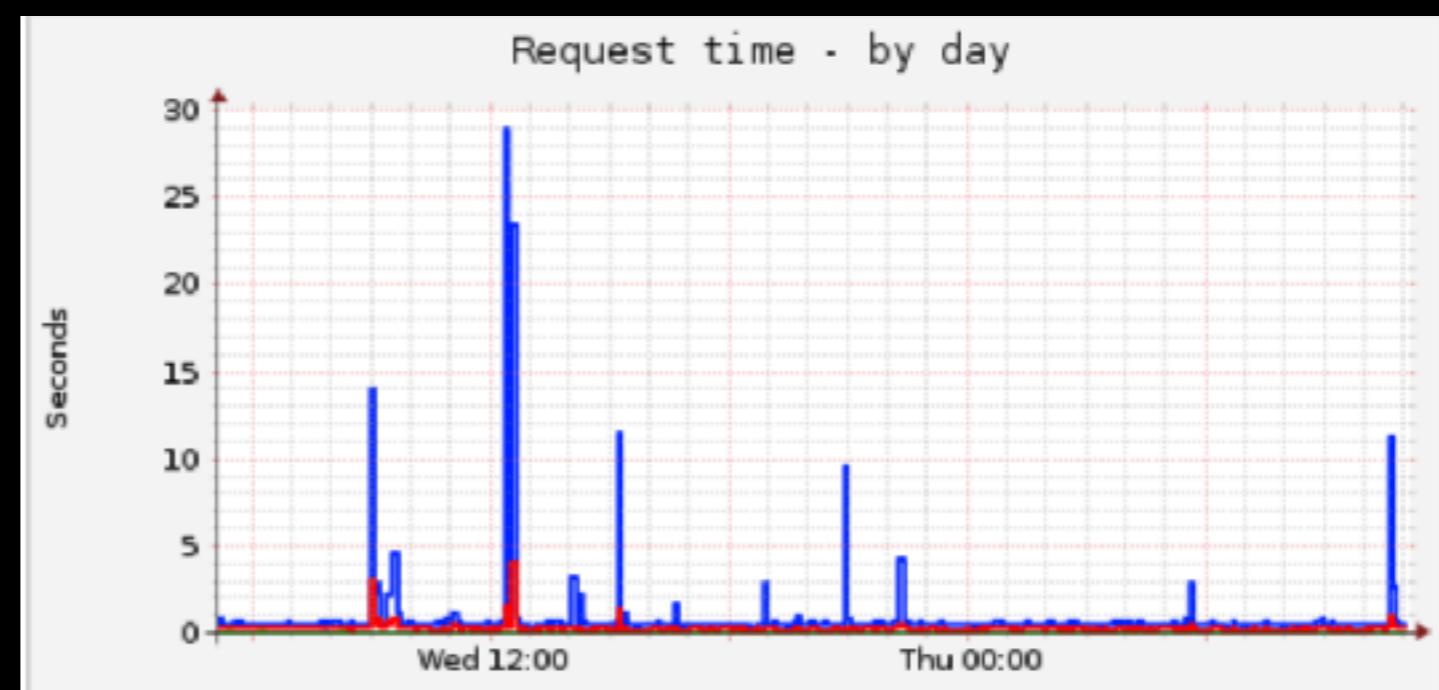
PHP UK
16 FEB 2018

W H Y

- fast is better than slow (duh)
- memory efficient is good
- saving money is good
- running out of memory is production is bad
- suddenly running out of memory in production is BAD



PHP Fatal error: Allowed memory size of 134217728 bytes exhausted



W H E N

- development
- load testing
- monitoring production

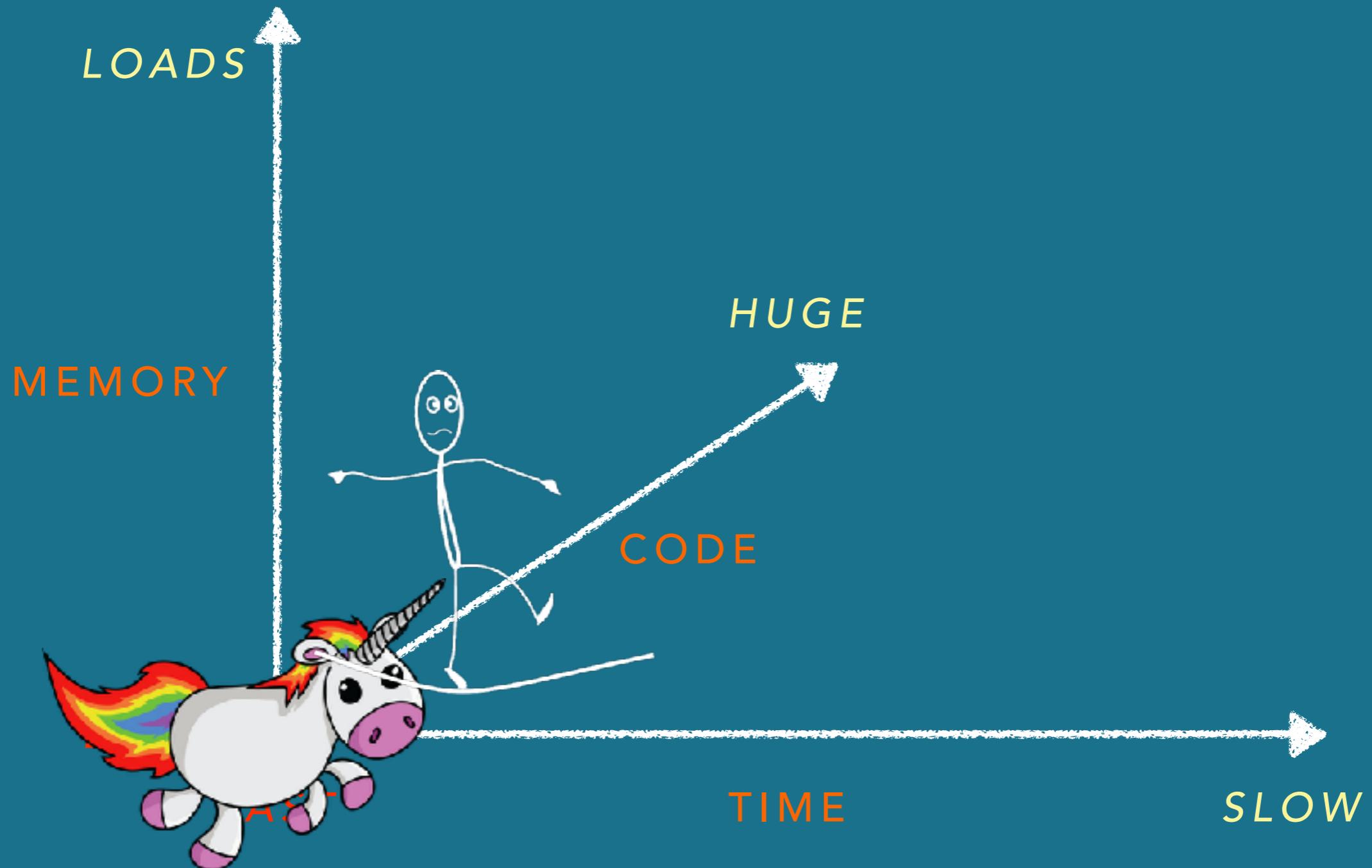
W H A T

"Do I need (to do) this?"



- execution time / no of operations
- memory usage / allocation

GOAL



- ▶ memory leaks
- ▶ timeouts
- ▶ if concurrency: race conditions, thread safety, 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++)
```

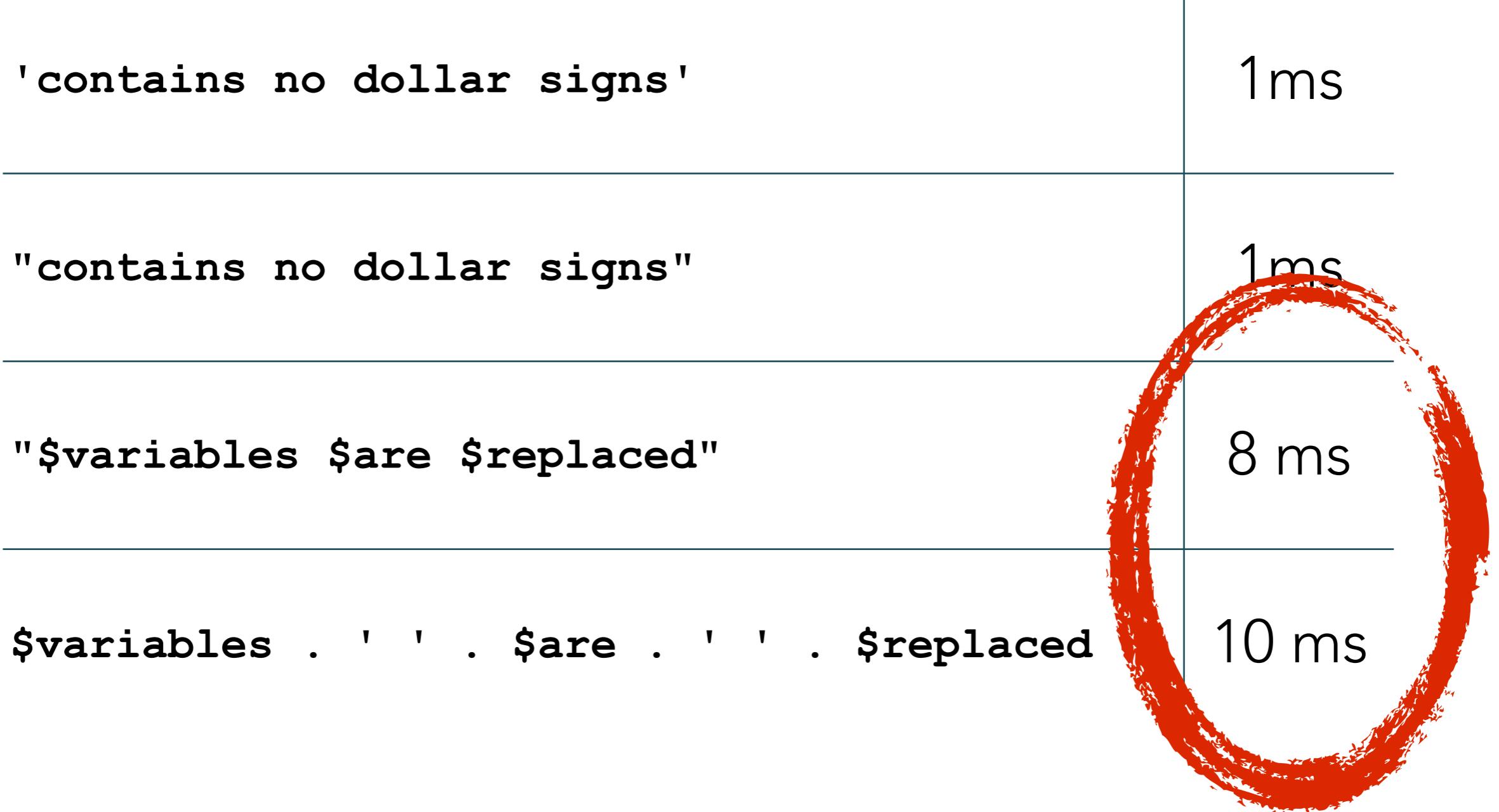
- watch out for DB queries inside loops

</sane>

- use empty(\$array) instead of count(\$array) === 0 to check if the array is empty
- use \$a === \$b instead of strcmp(\$a, \$b) === 0 for string comparison
- use single quotes instead of double quotes if you're not concatenating variables
- use regex
- don't use regex
- ...

- ▶ memory leaks
- ▶ long execution time / timeouts
- ▶ if concurrency: race conditions, thread safety, deadlocks / livelocks
- ▶ redundancy / inefficiency
- ▶ dead code / unused libraries

THE SINGLE VS DOUBLE QUOTE MYTH



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

MICRO OPTIMISATIONS

99% of the time, ignore them every time

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

TOOLS FOR PHP



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

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

- extensions

INSTALLING EXTENSIONS (YMMV)

- PEAR/PECL: `pecl install <name>`
- package managers: `brew` / `yum` / `apt-get 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

- ~~apd~~
- xdebug
- xhprof
 - PHP 7 forks: tideways or longxinH
- <https://github.com/arnaud-lb/php-memory-profiler>
- <https://github.com/BitOne/php-meminfo>

DIAGNOSTICS
BEFORE
DIAGNOSIS

Cigar

build passing coverage 100% [slsight](#) ★★★★

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										

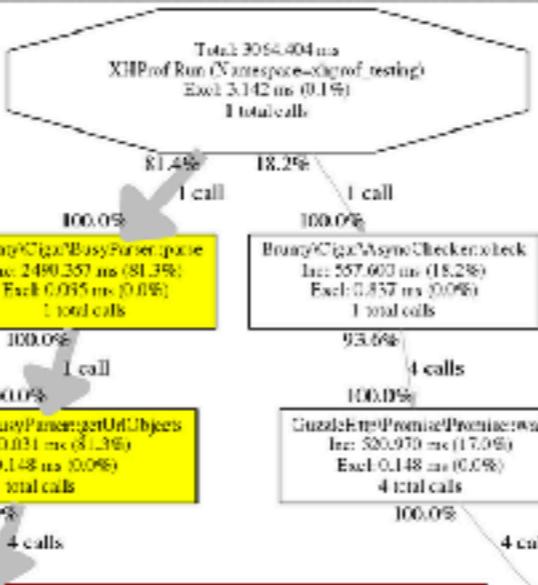
[\[View Full Callgraph\]](#)

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 %	M
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\Bus\Parser::parse	1	0.0%	2,490,357	81.3%	95	0.0%	2,481,707	94.1%	92	0.0%	
Brunty\Cigar\Bus\Parser::getUrlObjects	1	0.0%	2,490,031	81.3%	148	0.0%	2,481,464	94.0%	145	0.0%	
Brunty\Cigar\Bus\Parser::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%	157,150	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::invokeWaitEn	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::execute@1	205	12.7%	506,906	16.5%	470,370	16.2%	101,976	3.6%	45,459	1.5%	



baseline 3,064 sec



Brunty\Cigar\BusyParser::Brunty\Cigar\{closure}
Inc: 2489.883 ms (81.3%)
Excl: 2487.906 ms (81.2%)
4 total calls

1 call
100.0%
GuzzleHttp\Promise\Promise::invokeWaitIf
Inc: 520.732 ms (17.0%)
Excl: 0.089 ms (0.0%)
1 total calls

2 calls
100.0%
GuzzleHttp\Promise\Promise::allPending@1
Inc: 520.642 ms (17.0%)
Excl: 0.071 ms (0.0%)
2 total calls

1 call
100.0%
GuzzleHttp\Promise\Promise::invokeWaitIf@1
Inc: 520.553 ms (17.0%)
Excl: 0.329 ms (0.0%)
1 total calls

1 call
100.0%
GuzzleHttp\Handler\CurlMultiHandler::multiHandle
Inc: 520.024 ms (17.0%)
Excl: 12.113 ms (0.4%)
1 total calls

325 calls
100.0%
GuzzleHttp\Handler\CurlMultiHandler::tick
Inc: 506.806 ms (16.5%)
Excl: 470.372 ms (15.3%)
325 total calls

?

👉 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

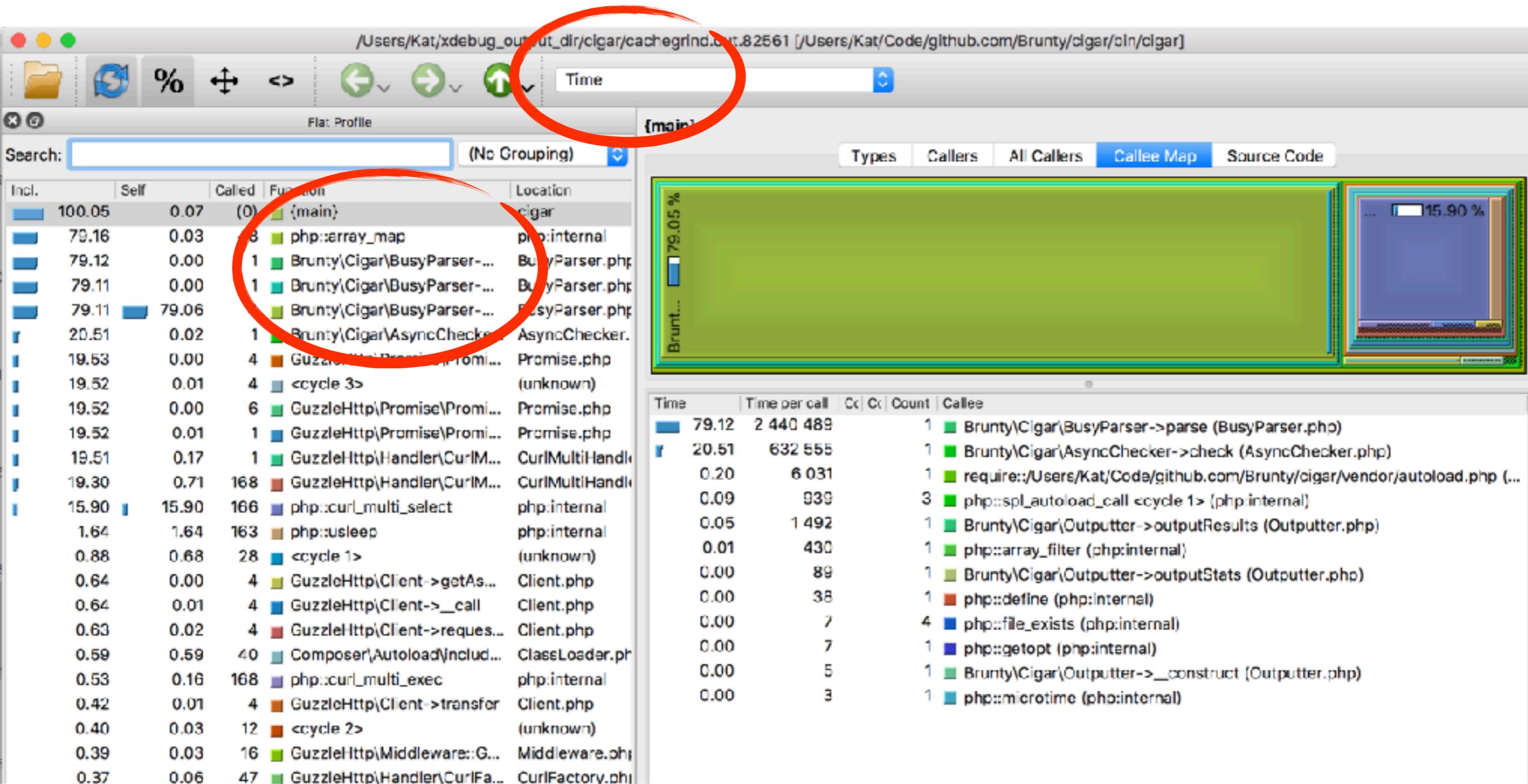
Workaround for macOS High Sierra which doesn't have dot in /usr/bin but in /usr/local/bin:

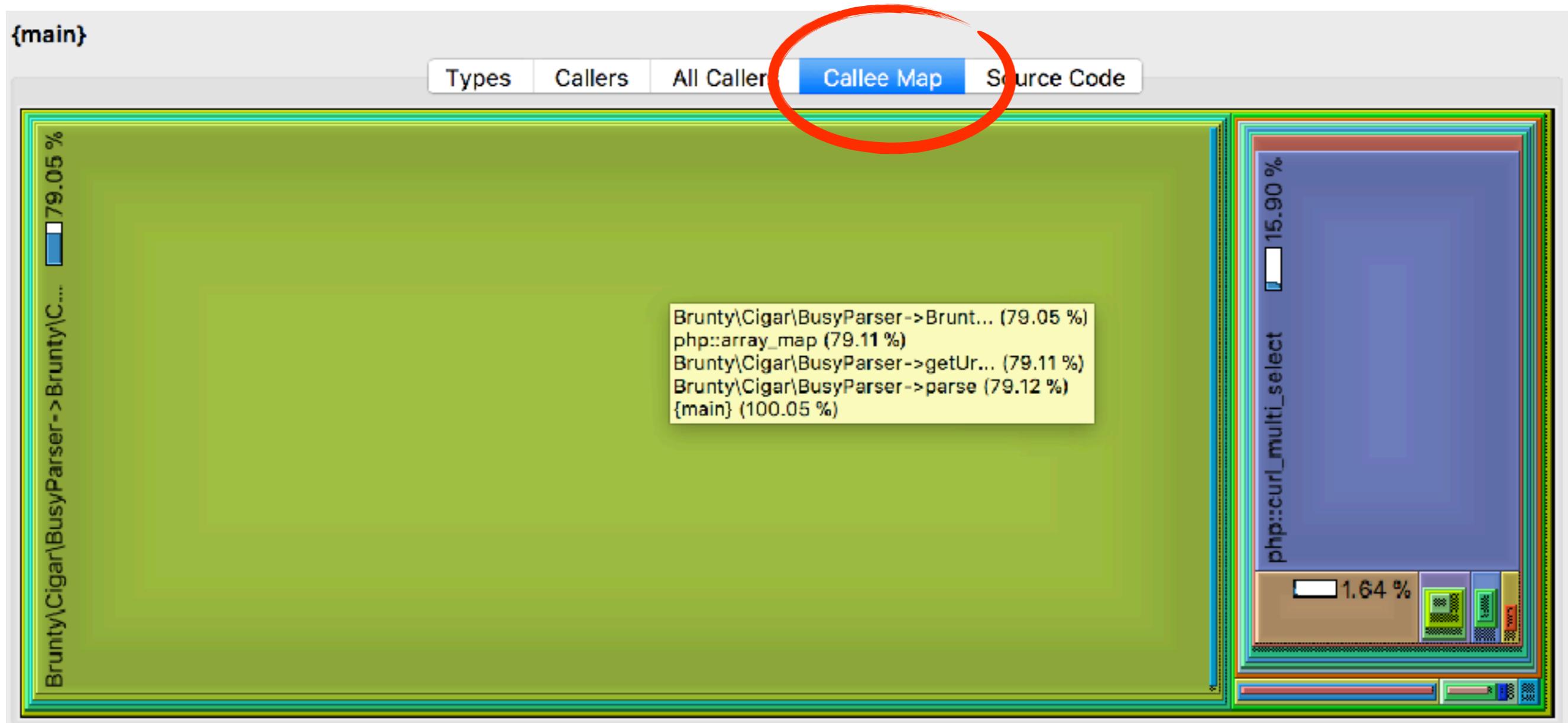
```
$ cd /Applications/qcachegrind.app/
Contents/MacOS

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

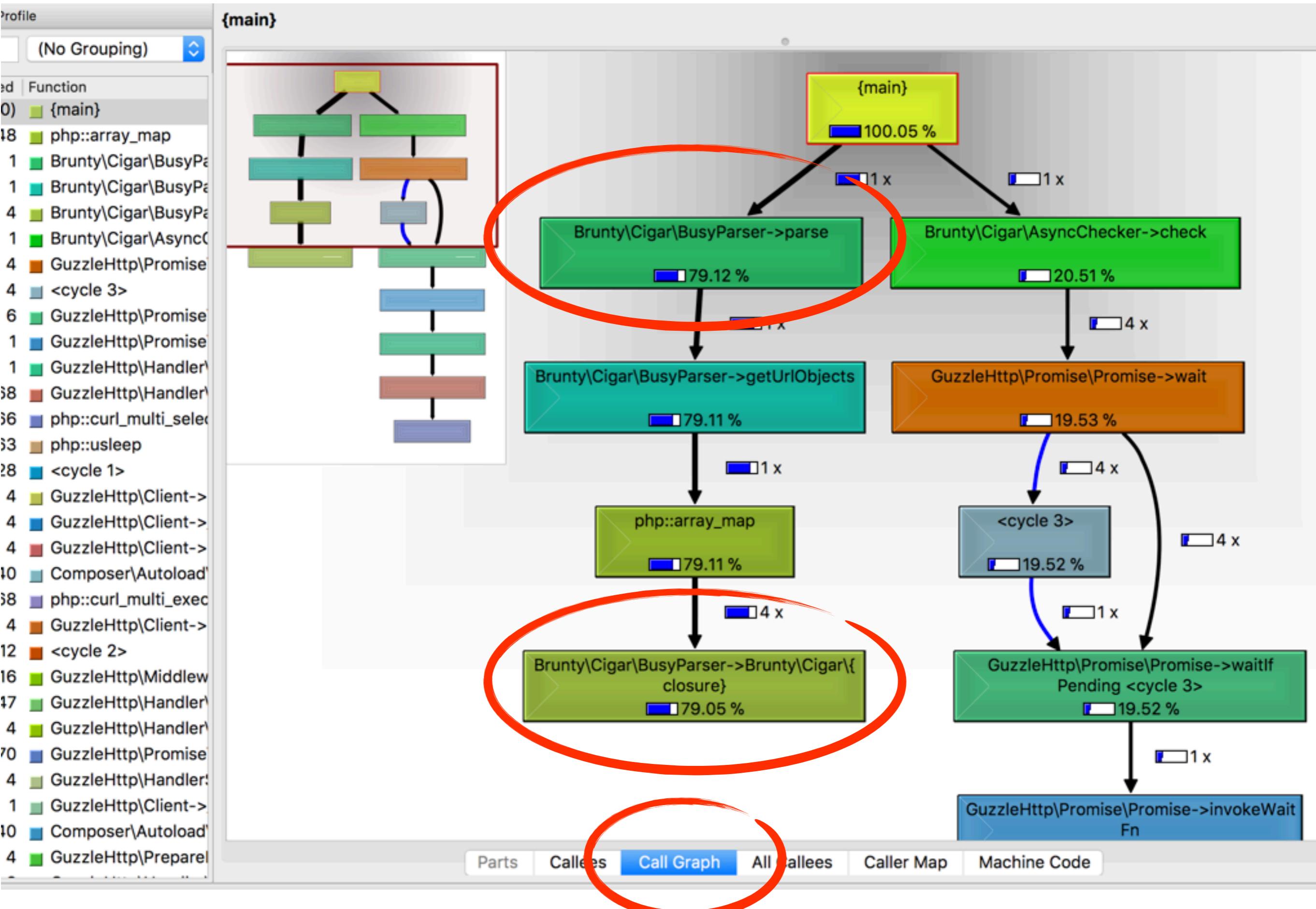
$ chmod +x qcachegrind
```

```
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
```





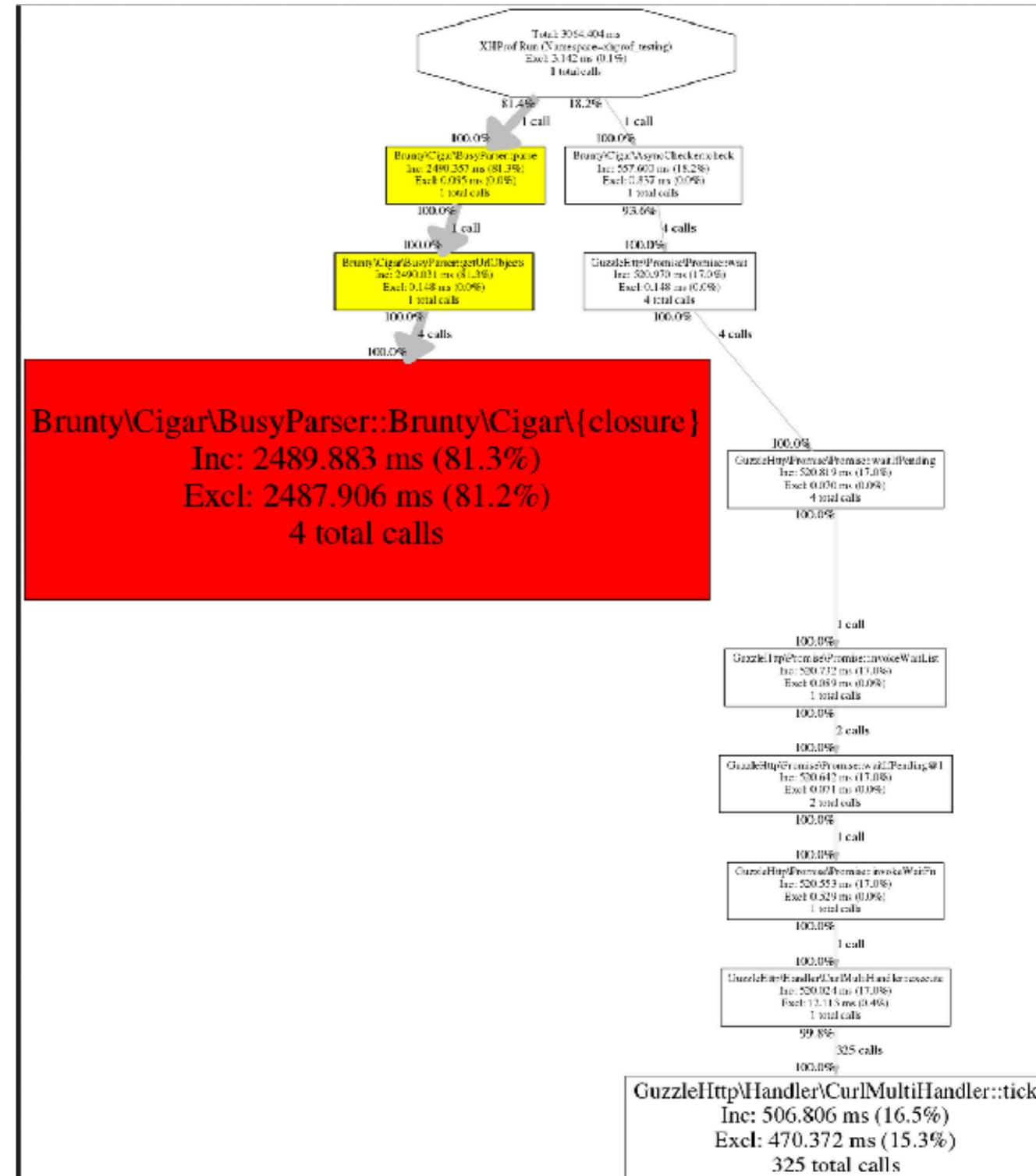
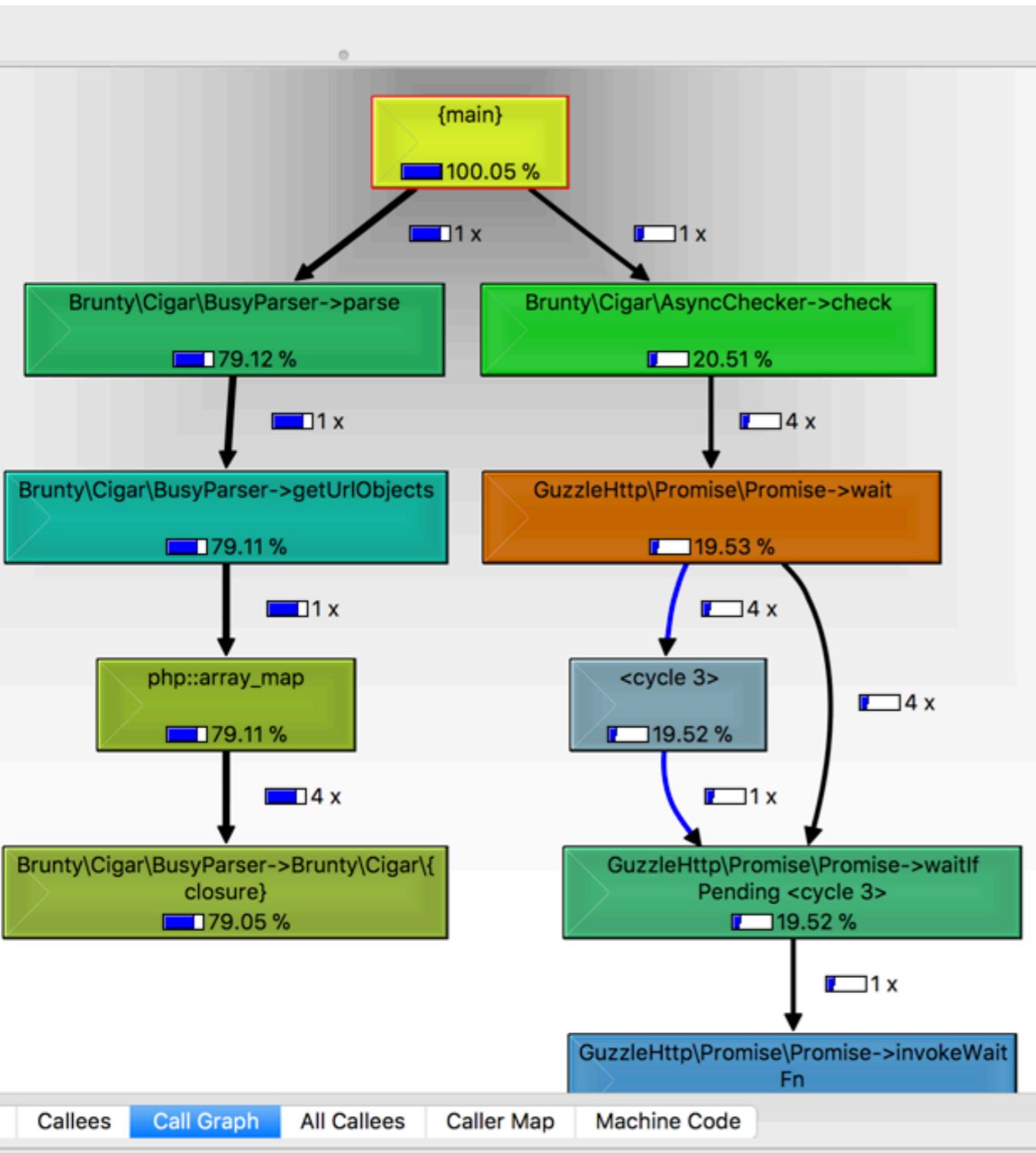
👉 look for big empty squares

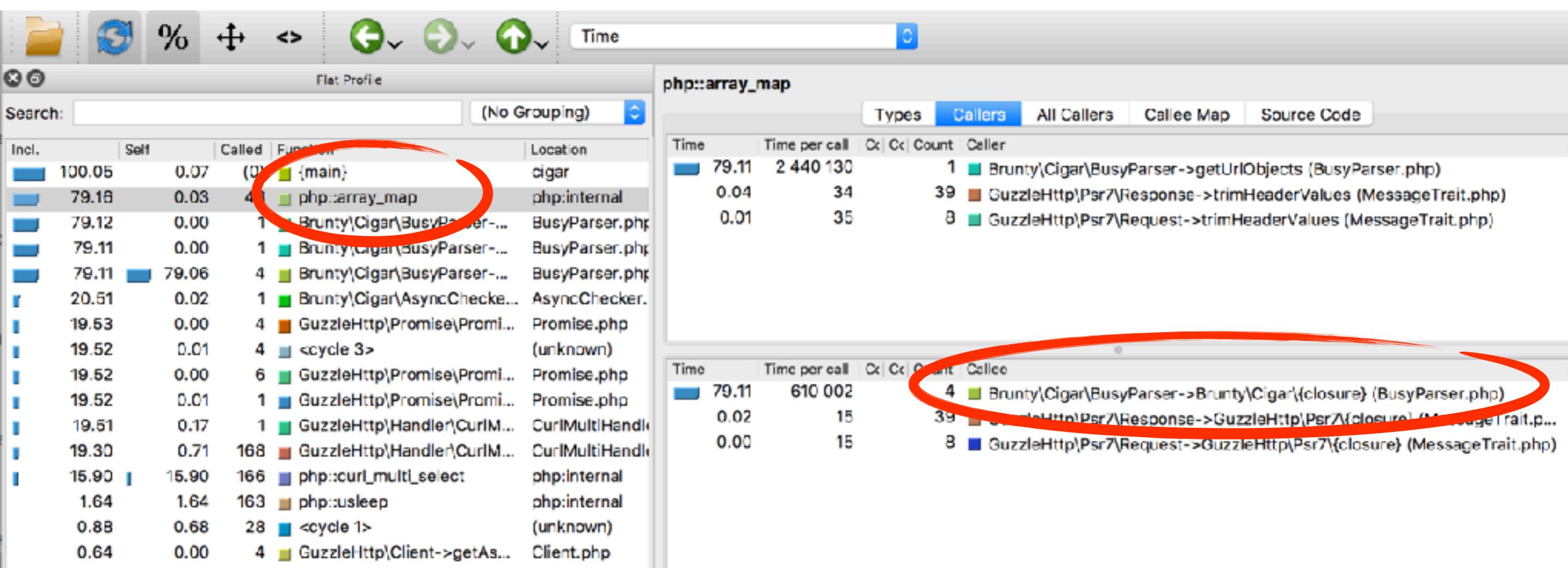


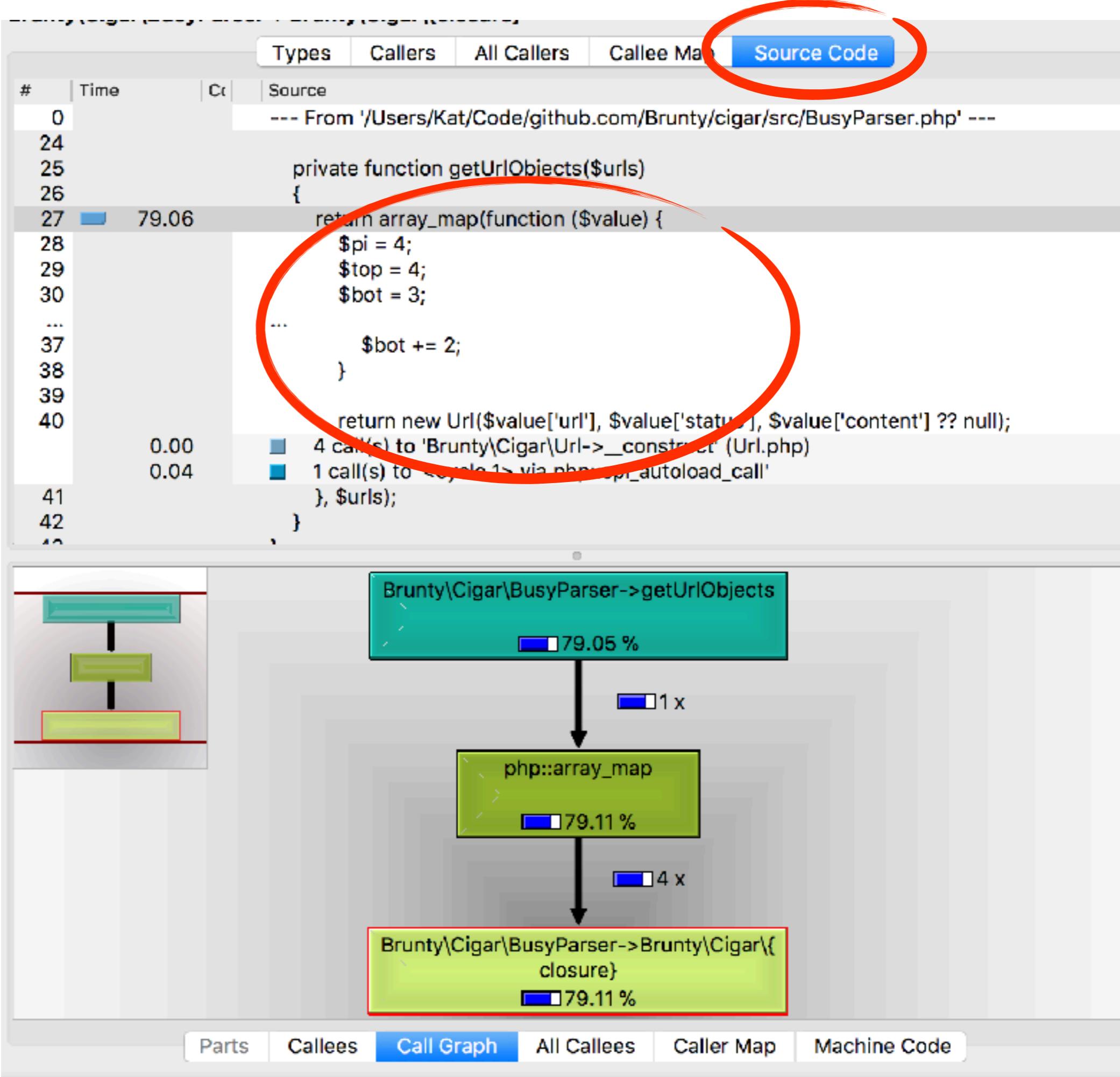
parse: 79.12%

(two separate runs)

parse: 81.4%







```
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

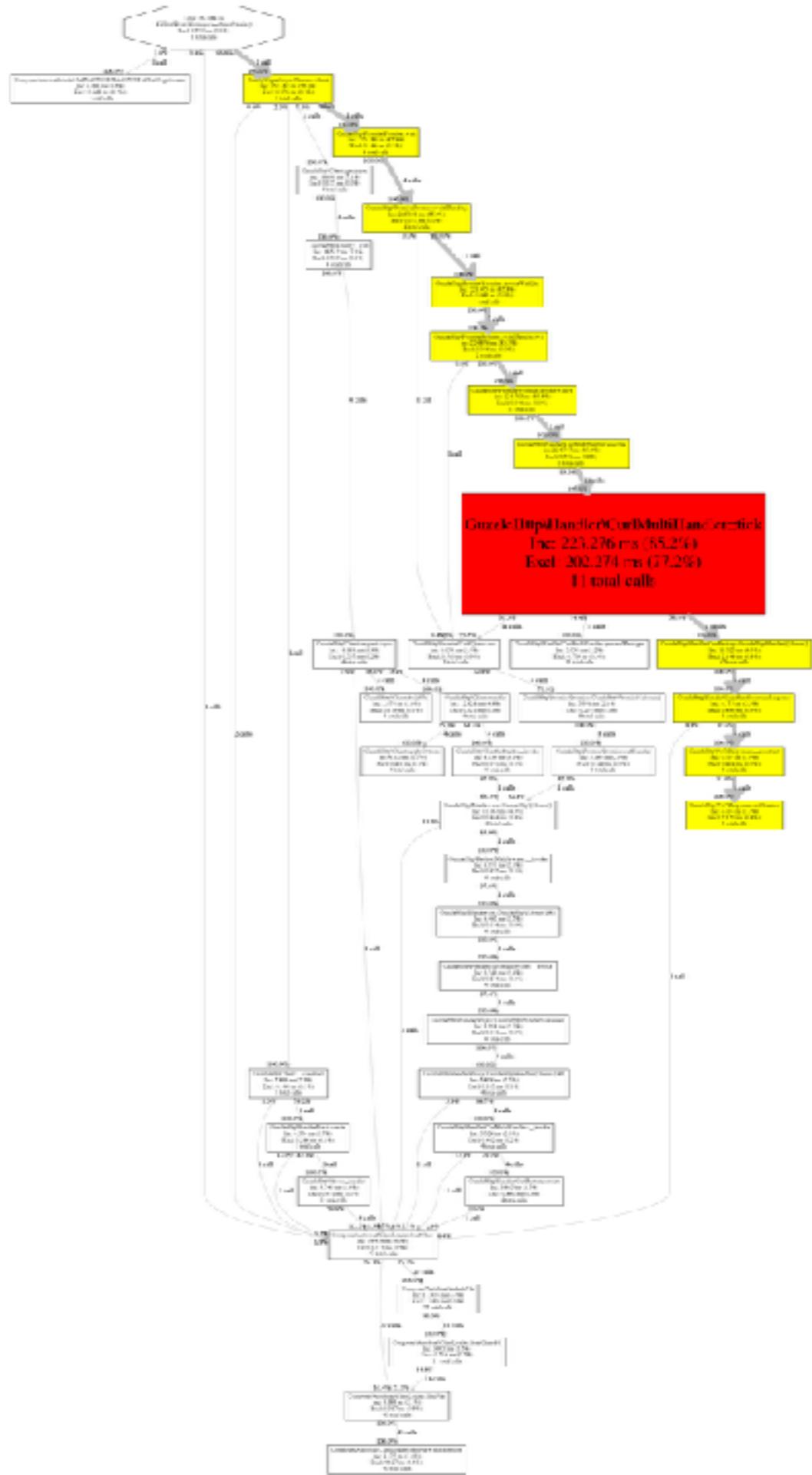
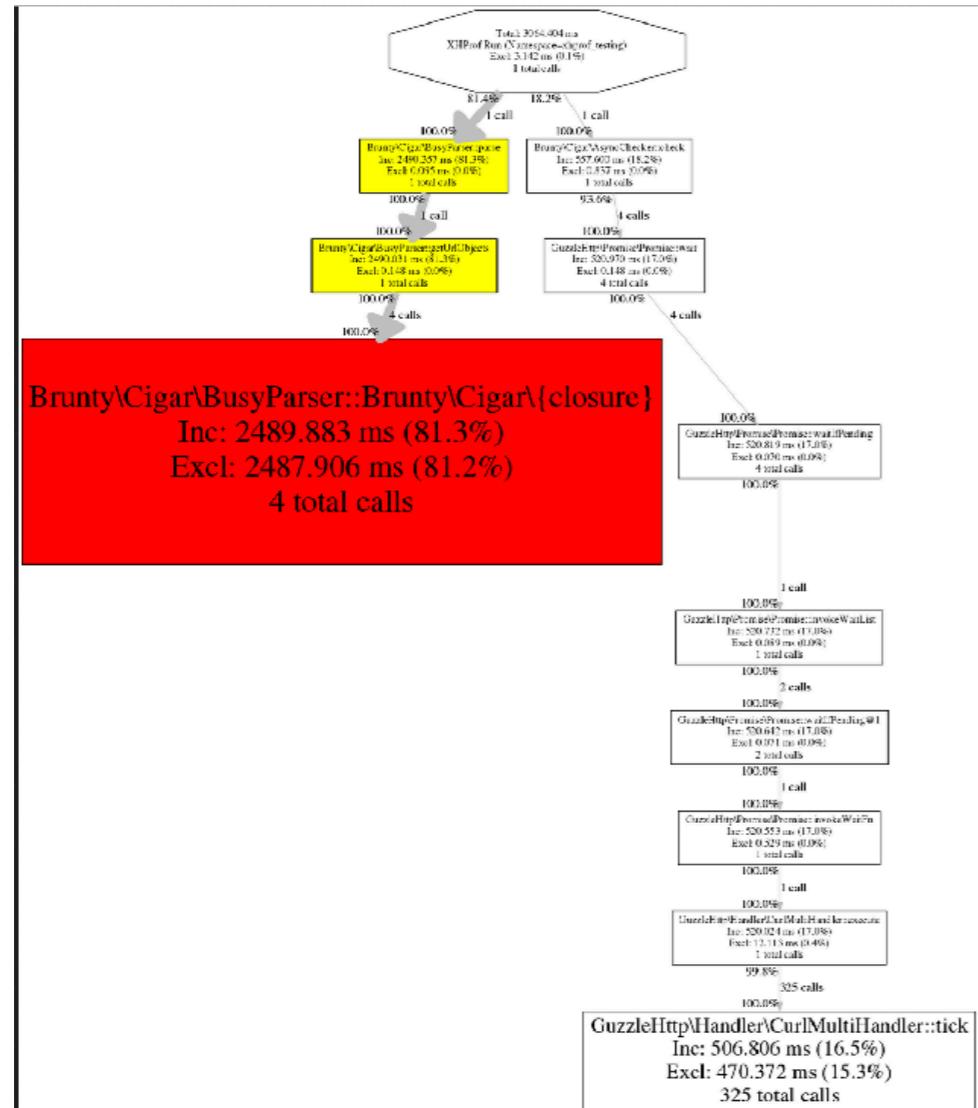
<u>Function Name</u>	<u>Calls</u>	<u>Calls %</u>	<u>Incl. Wall Time (microsec)</u>	<u>IWall %</u>	<u>Excl. Wall Time (microsec)</u>
main()	1	0.1%	262,084	100.0%	1,5
Brunty\Cigar\AsyncChecker::check	1	0.1%	252,182	96.2%	8
GuzzleHttp\Promise\Promise::wait	4	0.4%	225,184	85.9%	1
GuzzleHttp\Promise\Promise::waitIfPending	4	0.4%	225,035	85.9%	
GuzzleHttp\Promise\Promise::invokeWaitList	1	0.1%	224,945	85.8%	
GuzzleHttp\Promise\Promise::waitIfPending@1	2	0.2%	224,876	85.8%	
GuzzleHttp\Promise\Promise::invokeWaitFn	1	0.1%	224,793	85.8%	
GuzzleHttp\Handler\CurlMultiHandler::execute	1	0.1%	224,747	85.8%	5

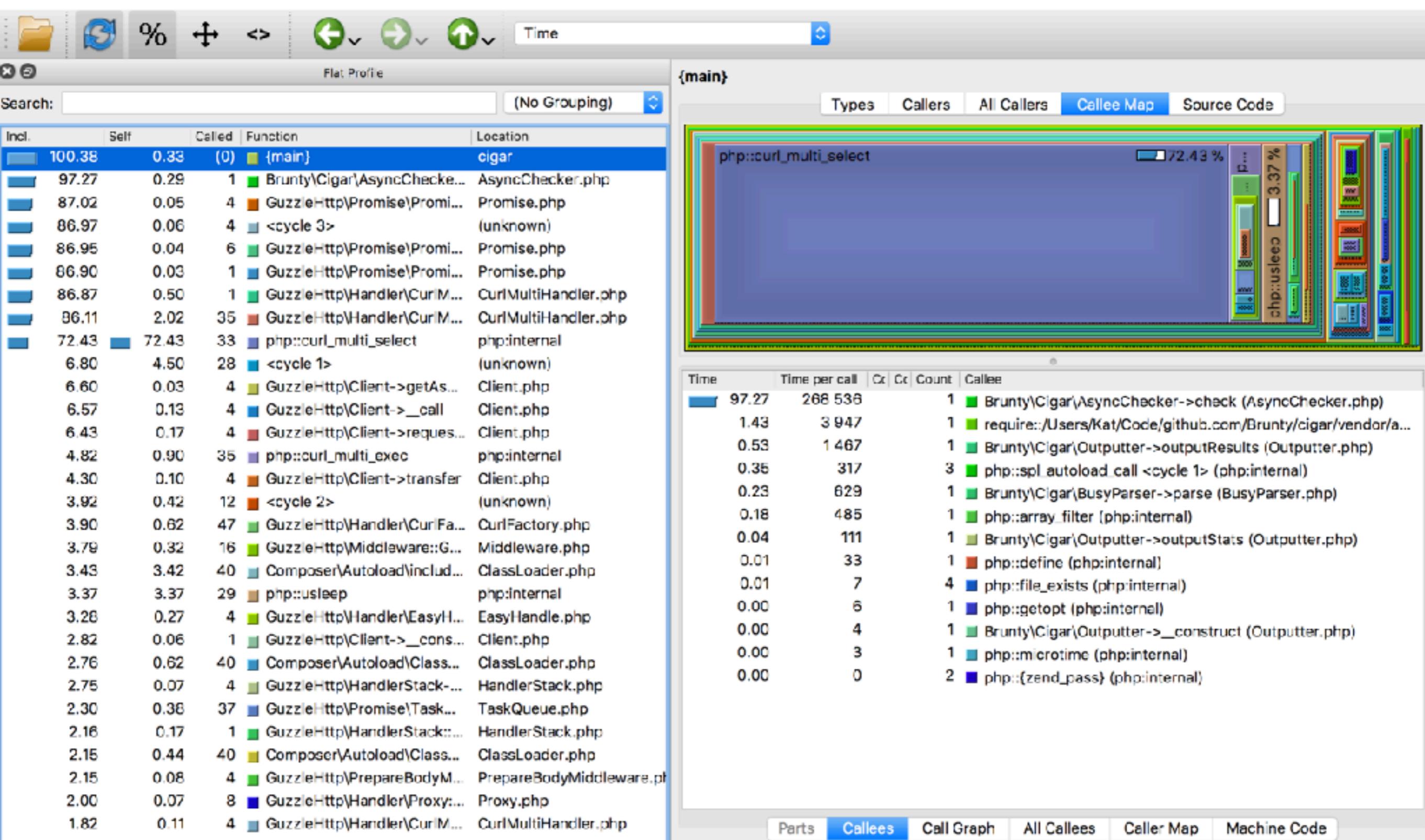


Brunty\Cigar\result::last_error	0	0.0%	0.0	0.0%
GuzzleHttp\Client::configureDefaults	1	0.1%	737	0.3%
Brunty\Cigar\BusyParser::parse	1	0.1%	723	0.3%
GuzzleHttp\normalize_header_keys	4	0.4%	709	0.3%

VS

<u>Function Name</u>	<u>Calls</u>	<u>Calls %</u>	<u>Incl. Wall Time (microsec)</u>	<u>IWall %</u>	<u>Excl. Wall Time (microsec)</u>	<u>EWall %</u>
main()	1	0.0%	3,064,404	100.0%	3,142	0.1
Brunty\Cigar\BusyParser::parse	1	0.0%	2,490,357	81.3%	95	0.0
Brunty\Cigar\BusyParser::getUrlObjects	1	0.0%	2,490,031	81.3%	148	0.0
Brunty\Cigar\BusyParser::Brunty\Cigar\{closure}	4	0.2%	2,489,833	81.3%	2,487,906	81.2
Brunty\Cigar\AsyncChecker::check	1	0.0%	557,336	18.2%	837	0.0
GuzzleHttp\Promise\Promise::wait	4	0.2%	520,970	17.0%	148	0.0
GuzzleHttp\Promise\Promise::waitIfPending	4	0.2%	520,819	17.0%	70	0.0

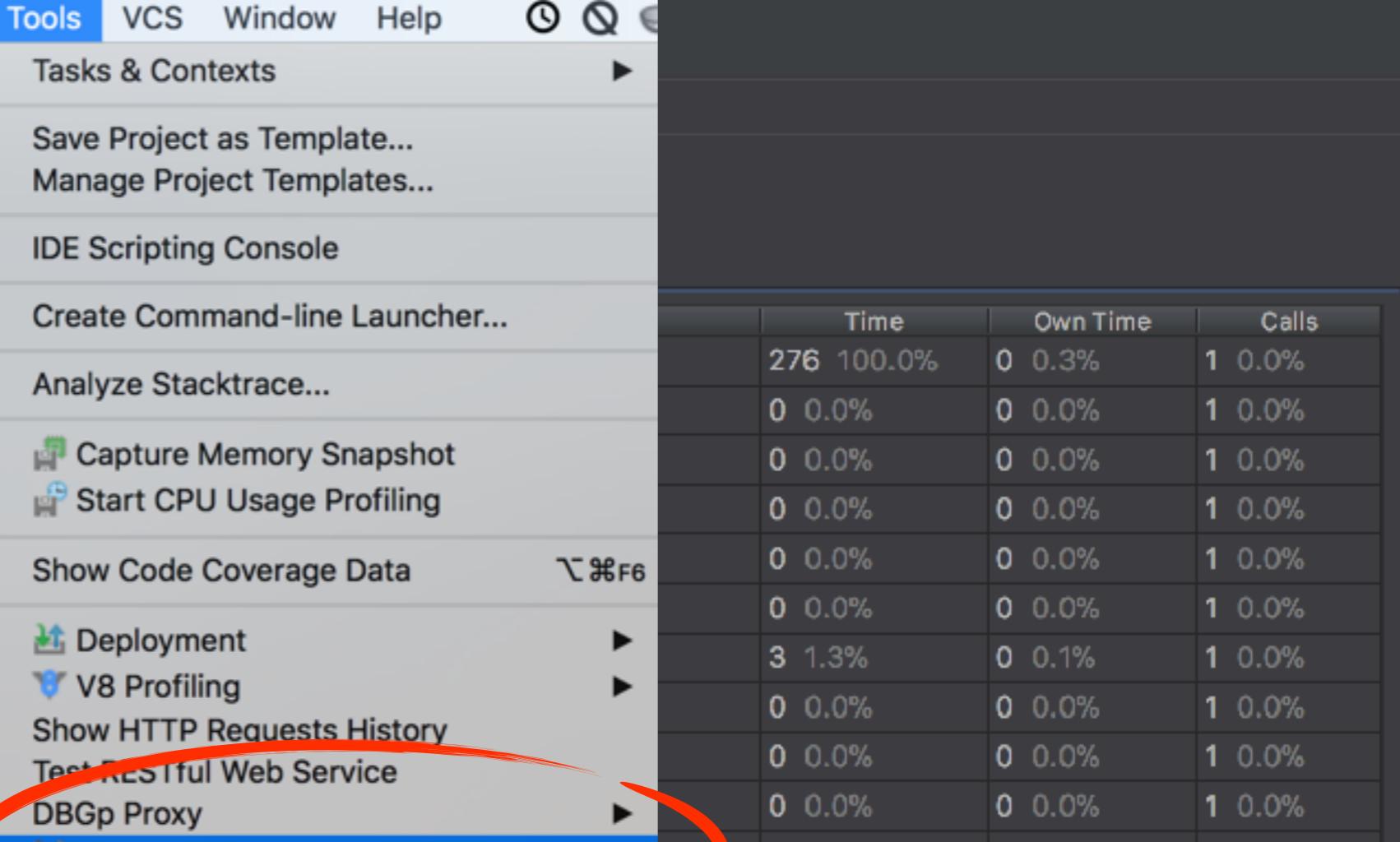




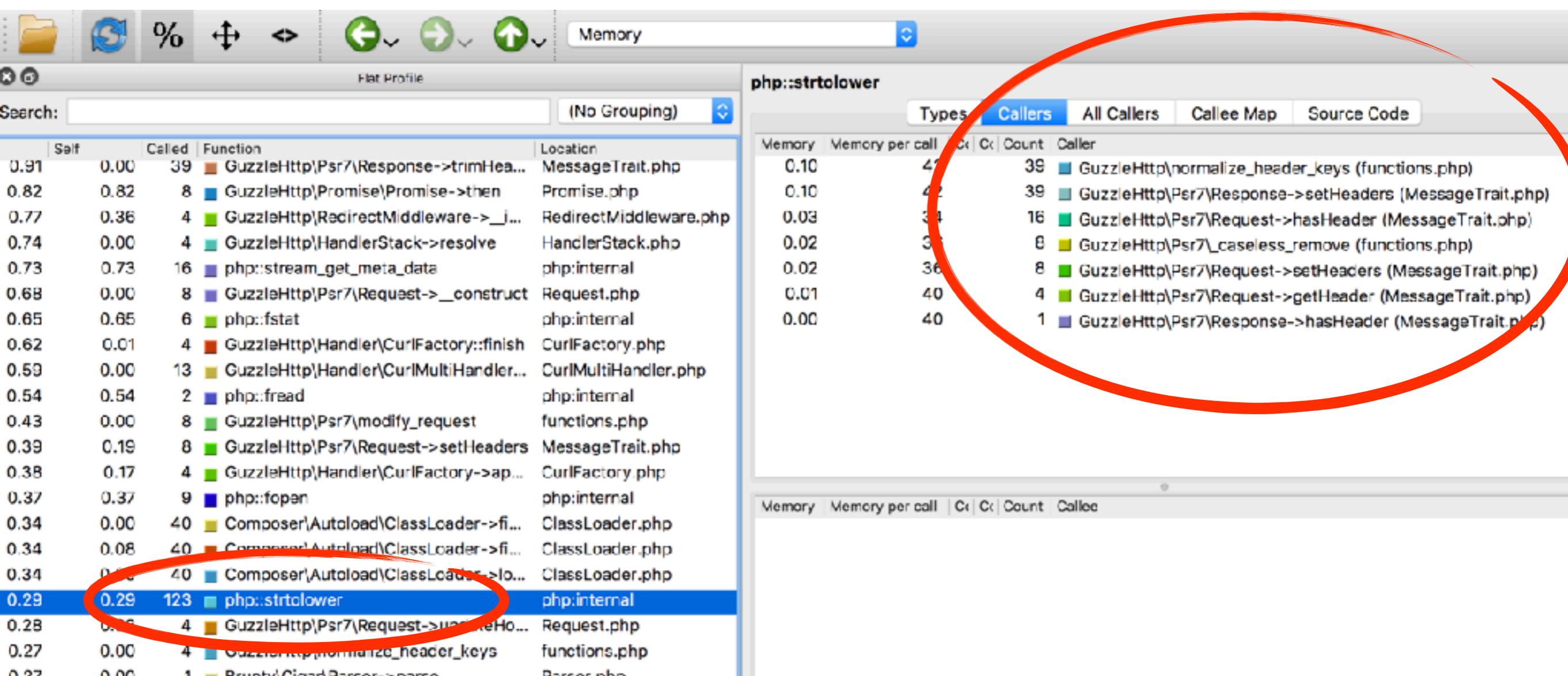
VERY NICE

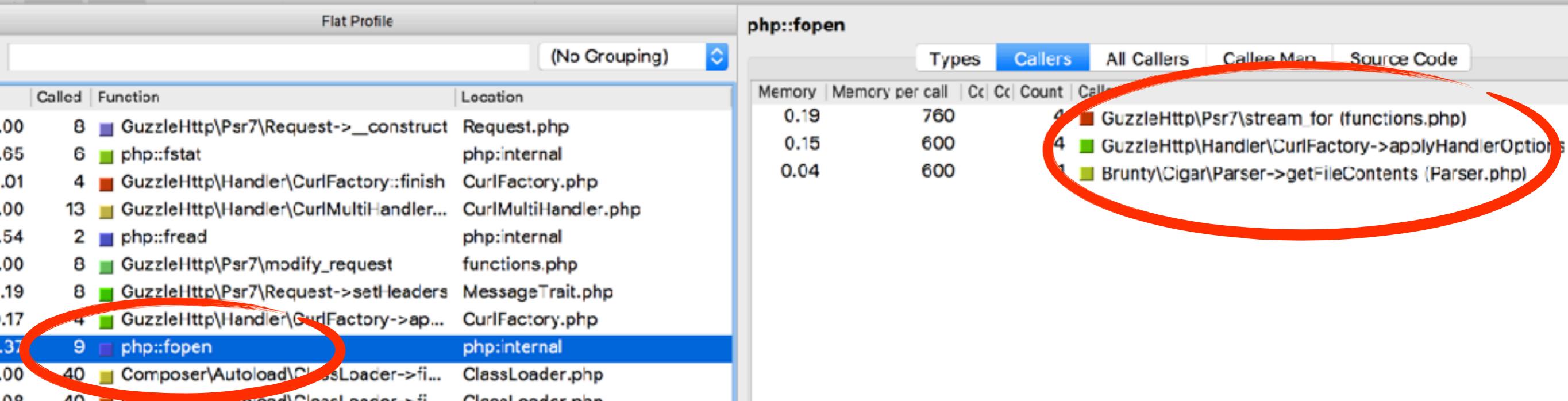


GREAT SUCCESS



	Time	Calls
cigar	276 100.0%	1 0.0%
autoload.php	3 1.3%	1 0.0%
file_exists	0 0.0%	1 0.0%
{zend_pass}	0 0.0%	1 0.0%
file_exists	0 0.0%	1 0.0%
spl_autoload_call	0 0.1%	1 0.0%
array_filter	0 0.2%	1 0.0%
Brunty\Cigar\Outputter->__construct	0 0.0%	1 0.0%
Brunty\Cigar\Outputter->outputStats	0 0.0%	1 0.0%
{zend_pass}	0 0.0%	1 0.0%
getopt	0 0.0%	1 0.0%
file_exists	0 0.0%	1 0.0%
spl_autoload_call	0 0.1%	1 0.0%





GuzzleHttp\Psr7\stream_for

#	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	■ 4 call(s) to 'php::fopen' (php:internal)
83			if (\$resource !== '') {
84			fwrite(\$stream, \$resource);
85			fseek(\$stream, 0);
86			}
87			return new Stream(\$stream, \$options);
88	0.01	4	■ 4 call(s) to 'GuzzleHttp\Psr7\Stream->__construct' (Stream.php)
89	6.52	1	■ 1 call(s) to '<cycle 1> via php::spl_autoload_call'
90			}

Memory

Brunty\Cigar\Parser->getFileContents

Types Callers All Callers Callee Map Source Code

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

Memory	Memory per call	Cr	Cc	Count	Caller
0.03	512	1	1	1	Brunty\Cigar\Parser->parse (Parser.php)

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

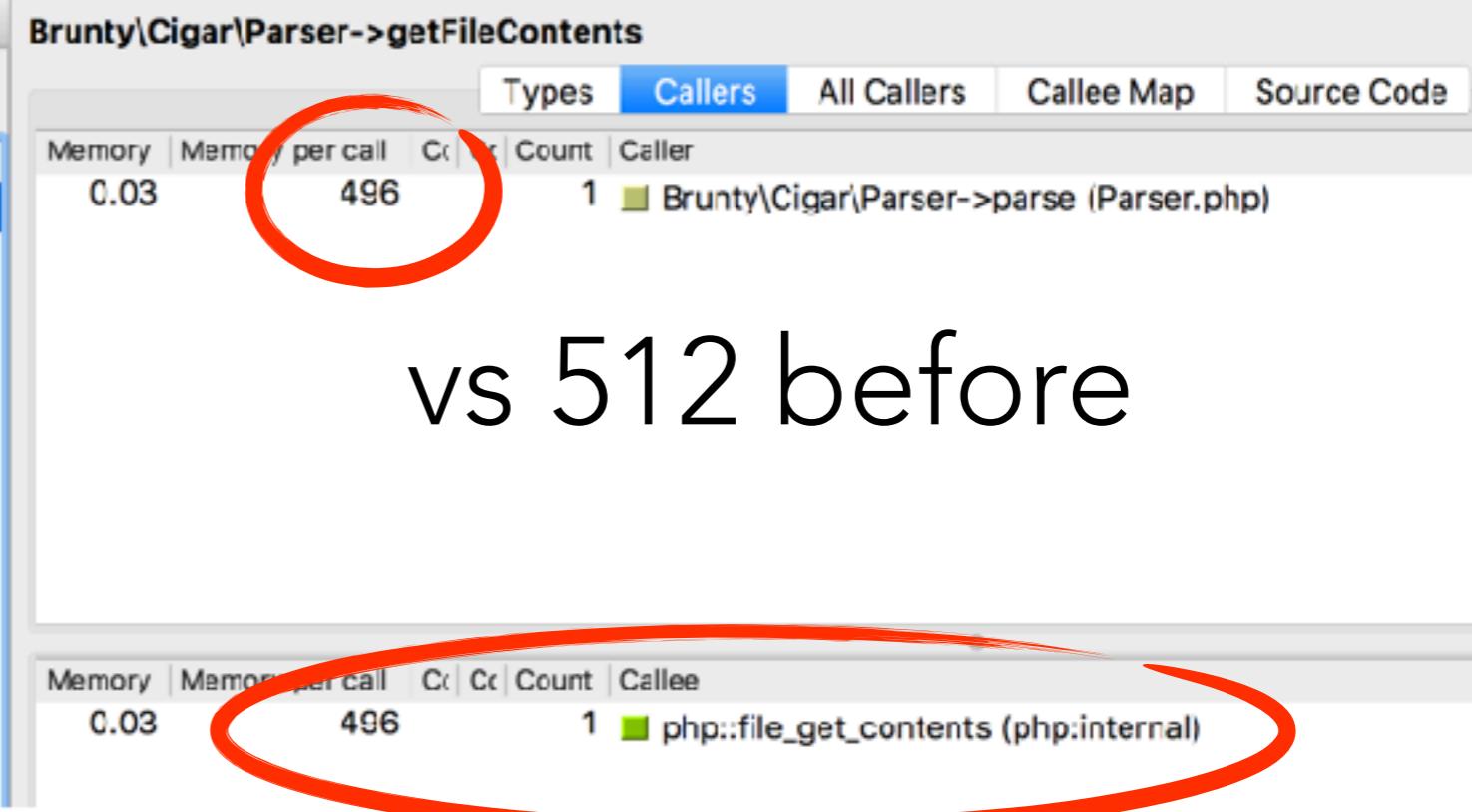
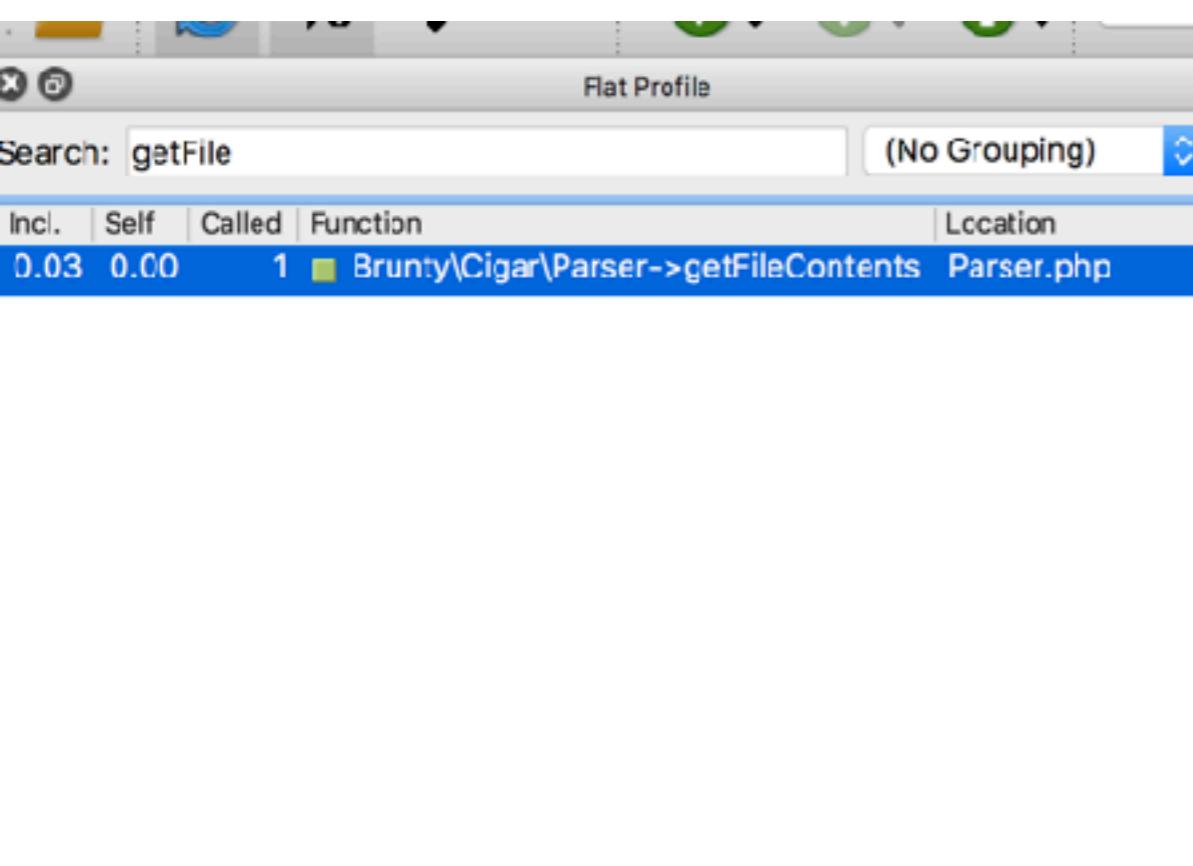
Memory	Memory per call	Cr	Cc	Count	Callee
0.53	8 576	1	1	1	php::fread (php:internal)
0.04	600	1	1	1	php::fopen (php:internal)
0.00	16	1	1	1	php::filesize (php:internal)

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

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

    fclose($file);

    return $contents;
}
```



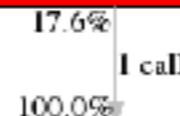
vs 512 before



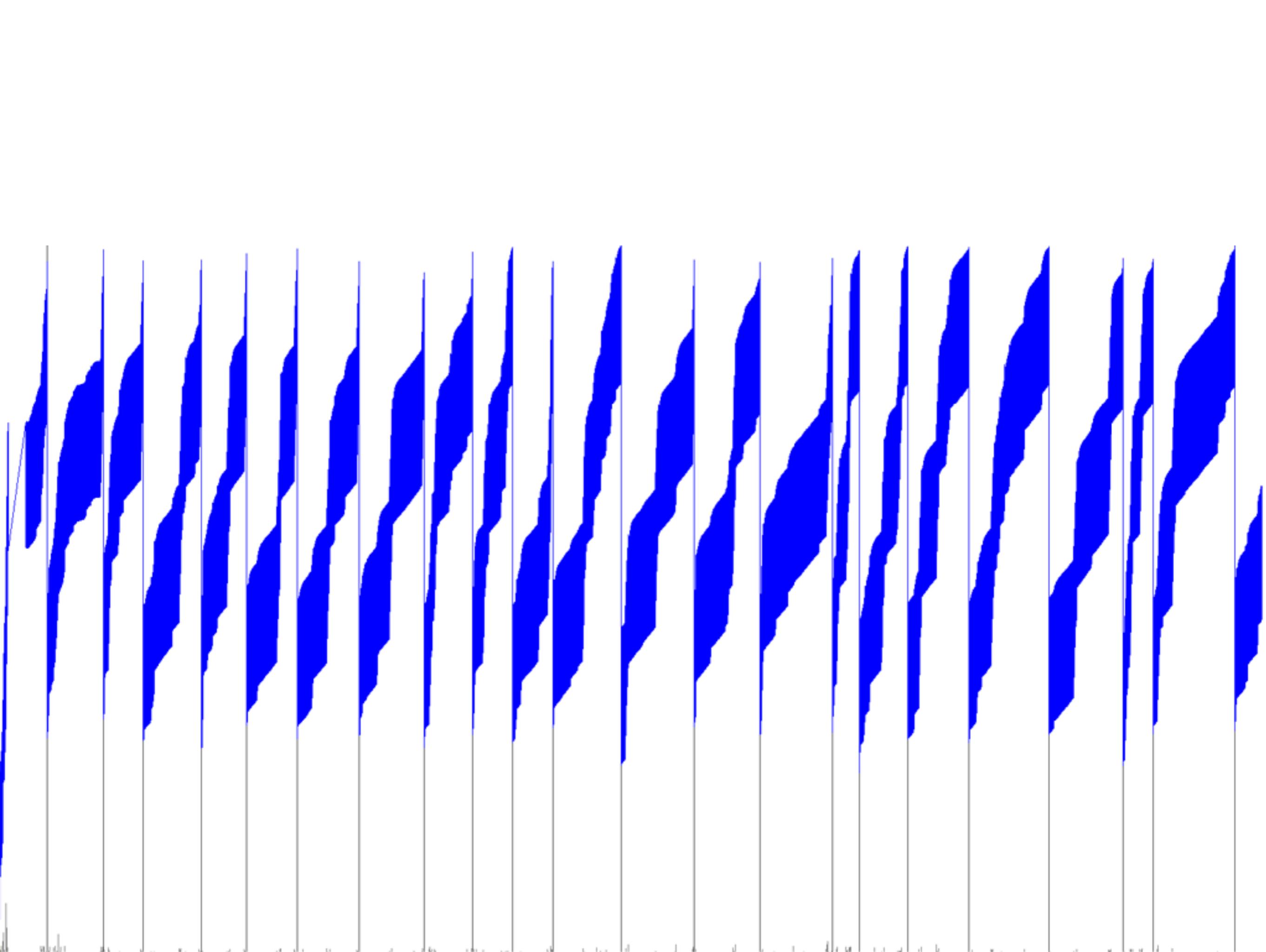
Parent/Child report for Brunty\Cigar\BusyParser::getFileContents [[View Callgraph](#)]

<u>Function Name</u>	<u>Calls</u>	<u>Calls%</u>	<u>Incl. Wall Time (microsec)</u>	<u>IWall%</u>	<u>Incl. CPU (microsecs)</u>	<u>ICpu %</u>	<u>Incl. MemUse (bytes)</u>	<u>IMemUse %</u>
<i>Current Function</i>								
Brunty\Cigar\BusyParser::getFileContents	1	5.9%	93	0.0%	96	0.1%	1,080	0.1%
Exclusive Metrics for Current Function			93	100.0%	96	100.0%	1,080	100.0%
<i>Parent function</i>								
Brunty\Cigar\BusyParser::parse	1	100.0%	93	100.0%	96	100.0%	1,080	100.0%

Brunty\Cigar\BusyParser::parse
 Inc: 0.723 ms (0.3%)
 Excl: 0.196 ms (0.1%)
 1 total calls



Brunty\Cigar\BusyParser::getFileContents
 Inc: 0.093 ms (0.0%)
 Excl: 0.093 ms (0.0%)
 1 total calls



👉 <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

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	_GET: 0x108a63100 _POST: 0x108a63120 _COOKIE: 0x108a63140 _FILES: 0x108a63160 argv: 0x108a63180 argc: 0x108a631a0 _ENV: 0x108a631c0 _REQUEST: 0x108a631e0 _SERVER: 0x108a63200 start: 0x108a63220 autoloadFileDirectory: 0x108a63240 options: 0x108a63260 content: 0x108a63280 version: 0x108a632a0 outputter: 0x108a632c0 file: 0x108a632e0 parser: 0x108a63300 domains: 0x108a63320 e: 0x108a63340 results: 0x108a63360 passedResults: 0x108a63380 GLOBALS: 0x108a633a0 __composer_autoload_files: 0x108a633c0

(new Parser())->parse(...

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





TAKEAWAYS

PRIORITISE

PRIORITISE

CUSTOMISE

PRIORITISE

CUSTOMISE

MEASURE

PRIORITISE

CUSTOMISE

MEASURE

ITERATE

THANK YOU



[HTTPS://JOIND.IN/TALK/BA977](https://join.d.in/talk/BA977)

@KASIAZIEN