DANIEL LOCKYER

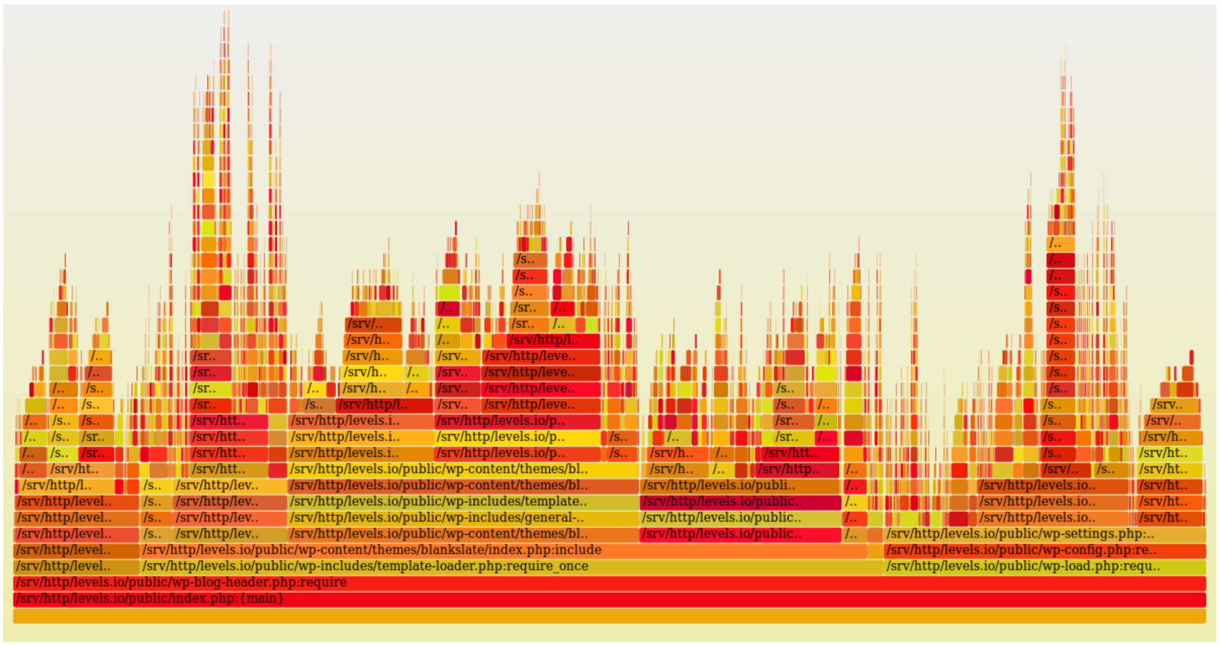
TWITTER • **HIRE ME** • EMAIL

How to generate PHP Flamegraphs

2018/09/20

In my quest to optimise the PHP sites I work on, I found that the Flame Graph repository contains a script to convert XDebug traces.

Flame graphs, created by Brendan Gregg, visualize the most frequently hit code from profiled software. The x-axis represents the profiled stack traces, ordered alphabetically, and the y-axis is the stack depth. Functions taking up more CPU time are wider.



I wrote a simple PHP script to list all the trace files and pipe them into the necessary scripts. This script is at the bottom of the page. You'll need to clone a copy of the Flame Graph repo into the same directory or change the script to alter the path. It's not safe enough to put on a production machine, as it simply takes a file path and passes it to the script. However, it's totally fine to run on your development machine.

To generate the traces, you'll need to alter the `php.ini` file. It's a similar setup to the PHP profiling I did.

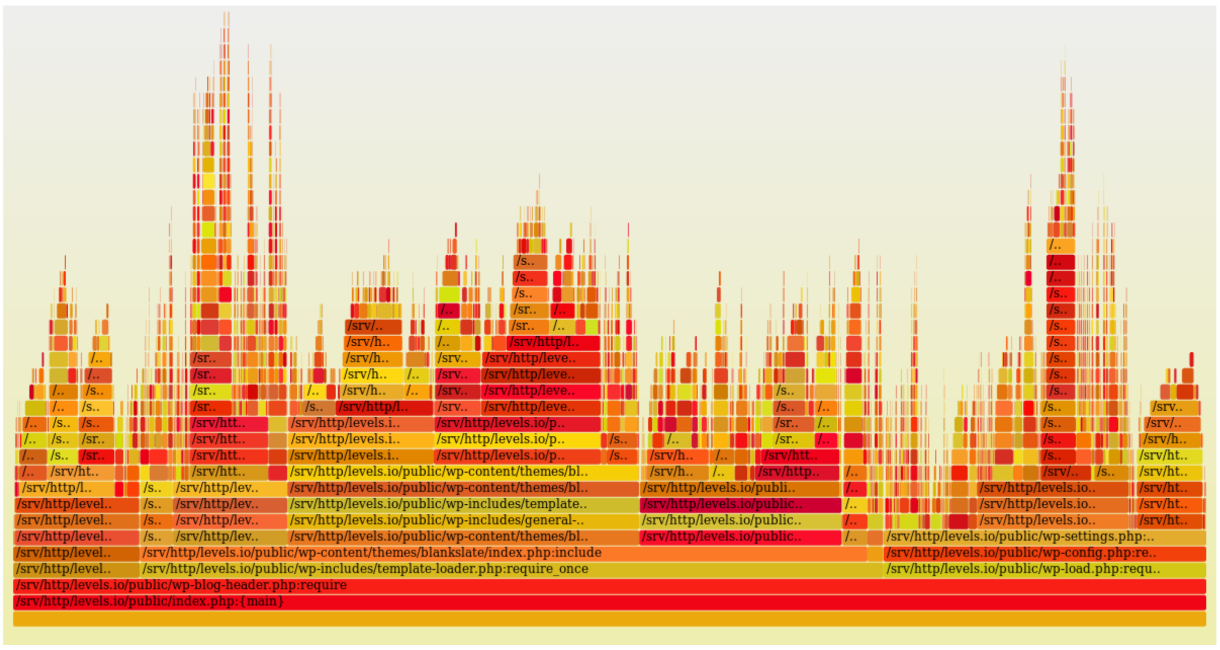
```
xdebug.trace_output_name = xdebug.trace.%t.%s
xdebug.trace_enable_trigger = 1
xdebug.trace_output_dir = /tmp
xdebug.trace_enable_trigger_value = "<secret key>"
xdebug.trace_format=1
```

You can then browse to the URL of the page you want to check out and add `?XDEBUG_TRACE=<secret key>`.

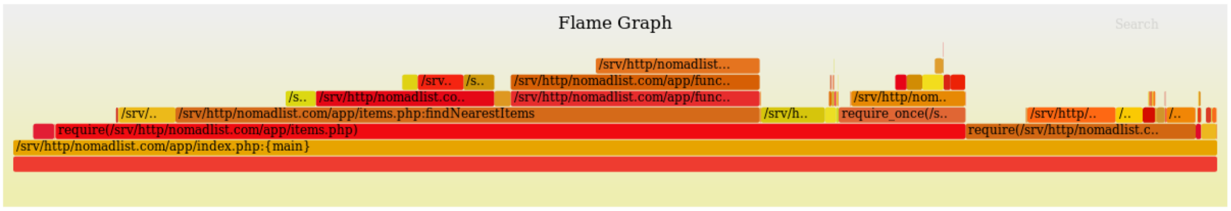
Finally, load up the PHP script, choose the trace file that was just generated and you should see the associated flame graph. The labels might seem a bit odd but according to the original pull request, “Each millionth of a second spent in a function is treated as one sample”.

When using flame graphs to optimise your code, you want to look for the widest stack frames. This indicates something taking up more CPU time. To dig down into the actual lines of code that are hot, you could check out my PHP profiling post.

Here is one for levels.io, a WordPress blog.



Here is one for Nomad List, a framework-less PHP application.



The images above were produced with a modified version of `stackcollapse-xdebug.php` that prepends/append the filename. I plan on pushing this upstream soon.

```
diff --git a/stackcollapse-xdebug.php b/stackcollapse-xdebug.php
index 6548903..0ed2eb8 100755
--- a/stackcollapse-xdebug.php
+++ b/stackcollapse-xdebug.php
@@ -152,7 +152,11 @@ if ($do_time) {
```

```
                addCurrentStackToStacks($current_stack, $time - $prev_start_time,
$stacks);
                array_pop($current_stack);
            } else {
-               $func_name = $parts[5];
+               if (in_array($parts[5], ["require", "require_once", "include",
 "include_once"])) {
+                   $func_name = $parts[5] . "(" . $parts[7] . ")";
+               } else {
+                   $func_name = $parts[7] . ":" . $parts[5];
+               }
```

If you're looking to upgrade the performance of your server code, email me at hi@daniellockyer.com!

```
<!DOCTYPE html>
<html>
    <head>
        <title>XDebug Flame Graph</title>
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1, max
        <style>
            label {cursor: pointer;}
            svg{width:100%;}
        </style>
    </head>
    <body>
        <h1>XDebug Flame Graph</h1>
        <form method="POST" class="load">
            <label for="file">File:</label>
            <select name="file" id="file">
                <?php
                    $dir = ini_get('xdebug.trace_output_dir');
                    if (!$dir) $dir = '/tmp/';

                    $files = glob("$dir/*.xt");
                    foreach ($files as $file) {
                        $checked = ($file == $_REQUEST['file']) ? 'selected="se
                        echo '<option value="' . htmlspecialchars($file) . '" '
                    }
                ?>
            </select>
            <button type="submit">Load</button>
            <br/>
            <p>Files from <code>xdebug.trace_output_dir = <?php echo htmlspecia
        </form>
        <?php
            if (!empty($_REQUEST['file'])) {
                $file = $_REQUEST['file'];
                if (!file_exists($file)) { echo "input file does not exist"; re
                if (!is_readable($file)) { echo "cannot read input file"; retur
                passthru(__DIR__.'/FlameGraph/stackcollapse-xdebug.php '.$file.
            }
        ?>
    </body>
</html>
```

xdebugfg.php hosted with ❤ by **GitHub** **view raw**

If you're looking to upgrade the performance of your server code, email me at hi@daniellockyer.com!

Tweet

Subscribe to hear more from me

Email address

Subscribe