Welcome!

Profiling PHP Applications

php|tek - Chicago, US - May 25, 2011

Derick Rethans - derick@derickrethans.nl - twitter: @derickr

http://derickrethans.nl/talks.html

http://joind.in/3425

About Me

Derick Rethans

- Dutchman living in London
- PHP development
- Author of the mcrypt, input_filter, dbus, translit and date/time extensions
- Author of Xdebug
- Contributor to the Apache Zeta Components Incubator project (formerly eZ Components)
- Freelancer doing PHP (internals) development

The Internet is Full of Nonsense

A small selection of PHP optimisation tips I found on the internet

echo is faster than print

Use sprintf instead of variables contained in double quotes, it’s about 10x faster.

Use <?php ... ?> tags when declaring PHP as all other styles are depreciated, including short tags.

The Internet is Full of Nonsense

I can only say one thing:

Complete
and Utter
BULLSHIT

Do I need to optimise my code?

Before you can optimise anything:

- Find out if things are running slow
- Find out whether it is the code
- Understand your code, application and execution paths
- Find out which parts of the code are slow
- Find out what you can optimise

How does the application perform?

Benchmark the application: siege

- check ~/.siegerc and set the logfile setting
- Create a file with urls:http://derickrethans.nl/ http://derickrethans.nl/spatial-indexes-data-sqlite.html http://derickrethans.nl/using-openstreetmap-with-flickr.html http://derickrethans.nl/who.html
- run against your code: siege -c 4 -r 10 -f /tmp/urls.txt

```
Transactions:                     40 hits
Availability:                 100.00 %
Elapsed time:                   9.76 secs
Data transferred:               1.34 MB
Response time:                   0.18 secs
Transaction rate:               4.10 trans/sec
Throughput:                   0.14 MB/sec
Concurrency:                   0.73
Successful transactions:             40
Failed transactions:                  0
Longest transaction:               0.39
Shortest transaction:               0.08
```

Is it my code that is slow?

There could be multiple reasons why the application is slow:

- The database is slow
- There is lots of IO
- Your code is slow
- The system is busy with other things

$ vmstat 1

```
procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs *|dd1111|us sy|* id wa
 5  0 121248 367520 767824 3080668    0    0 11272     0 1593 5540 *|dd1111|74 22|*  3  0
 5  0 121248 290784 767960 3082980    0    0   268     0 1555 5381 *|dd1111|77 20|*  3  0
 5  0 121248 238340 768132 3084336    0    0  1364 21160 2263 6815 *|dd1111|70 21|*  3  7
 6  0 121248 170772 768300 3087100    0    0  1652     0 1802 8540 *|dd1111|71 25|*  4  0
```

CPU is (close to) fully in use: 74 + 22 ➔ your code is slow.

$ vmstat 1

```
procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id *|dd1111|wa|*
 0  1 121248  50776 314796 4156000    0    0 245800     0 3018 5394  1 11 71 *|dd1111|17|*
 1  0 121248  52112 315108 4236692    0    0 243672     0 3107 5552  1 10 71 *|dd1111|18|*
 1  1 121248  50148 315256 4318268    0    0 243096    44 3146 5463  1 11 72 *|dd1111|17|*
 1  1 121248  52120 315484 4396356    0    0 243784     0 3006 5419  1 10 72 *|dd1111|18|*
```

wait is > 10 ➔ IO is the bottleneck.

Understand your code, application and execution paths

In-code tools:

- Add timing points

External tools:

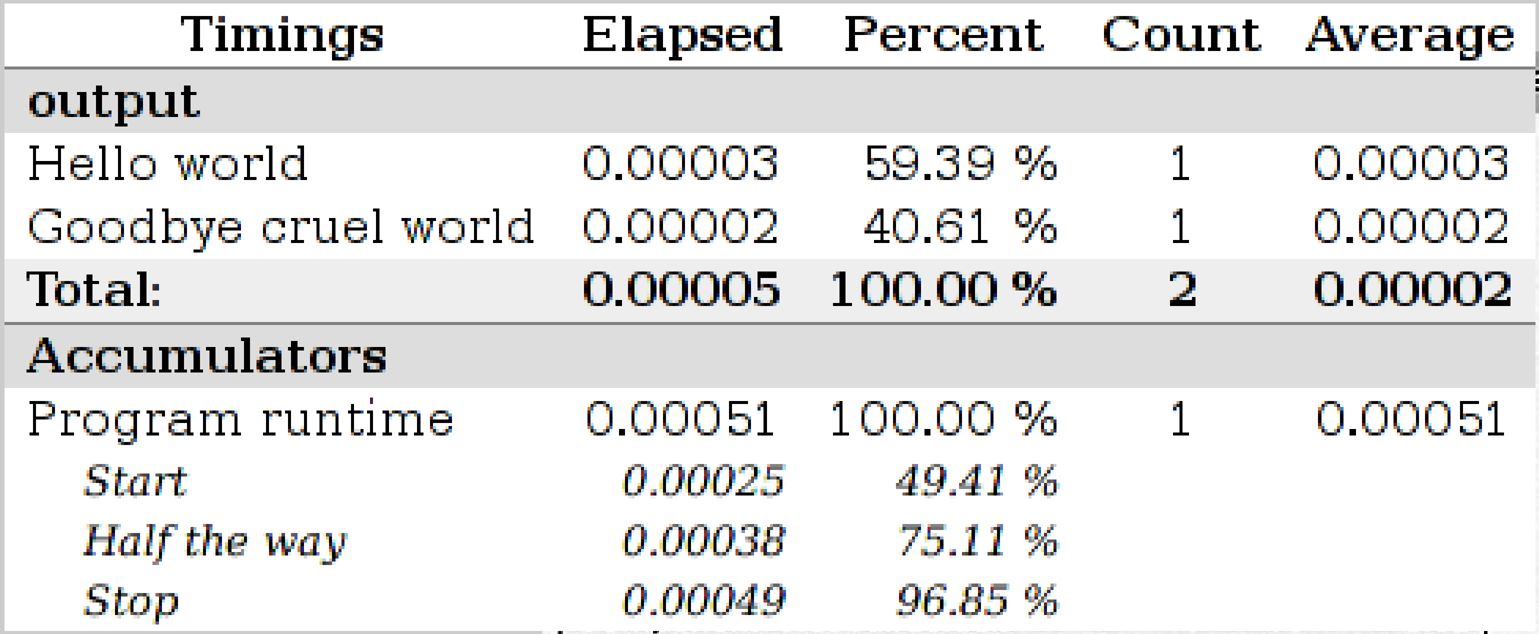
- Basic overview: inclued
- Deep details: xdebug

Timing Points

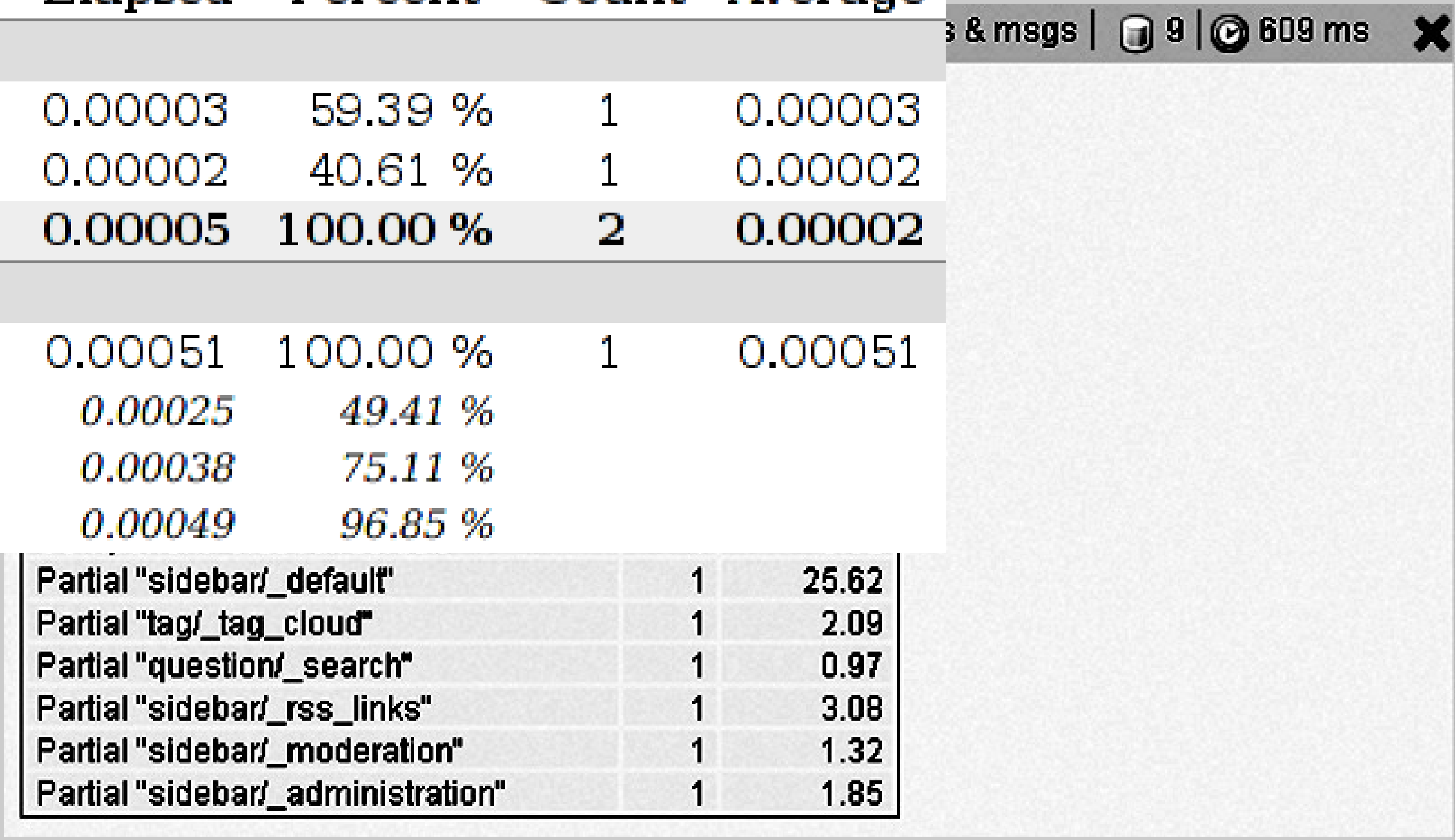
Code with analysis in mind:

- Add timing points around specific events
- Check changes over time

eZ Publish:



Timings	Elapsed	Percent	Count	Average
output				
Hello world	0.00003	59.39 %	1	0.00003
Goodbye cruel world	0.00002	40.61 %	1	0.00002
Total:	**0.00005**	**100.00 %**	**2**	**0.00002**
Accumulators				
Program runtime	0.00051	100.00 %	1	0.00051
Start	*0.00025*	*49.41 %*		
Half the way	*0.00038*	*75.11 %*		
Stop	*0.00049*	*96.85 %*		



s & msgs | 9 | 609 ms

Partial "sidebar/_default"	1	25.62
Partial "tag/_tag_cloud"	1	2.09
Partial "question/_search"	1	0.97
Partial "sidebar/_rss_links"	1	3.08
Partial "sidebar/_moderation"	1	1.32
Partial "sidebar/_administration"	1	1.85

Inclued

Dumps includes/classes hiearchies

Install:

```
pecl install inclued
```

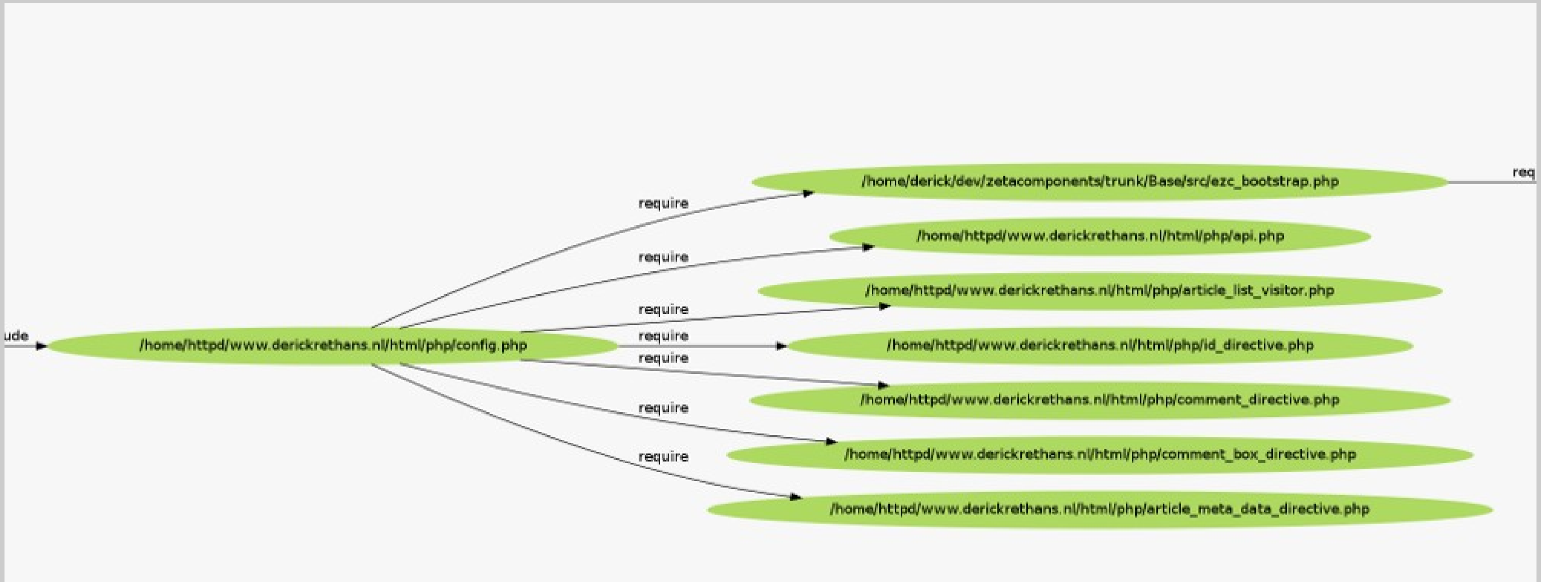
Add to php.ini:

```
extension=inclued.so
inclued.enabled=1
inclued.dumpdir=/tmp
```

Generate graphs:

```
php -dinclued.enable=0 gengraph.php -t includes -i /tmp/inclued.22439.1
dot -Tpng -o inclued-includes.png inclued.out.dot
php -dinclued.enable=0 gengraph.php -t classes -i /tmp/inclued.22439.1
dot -Tpng -o inclued-classes.png inclued.out.dot
```

Include overview: inclued-includes.png:

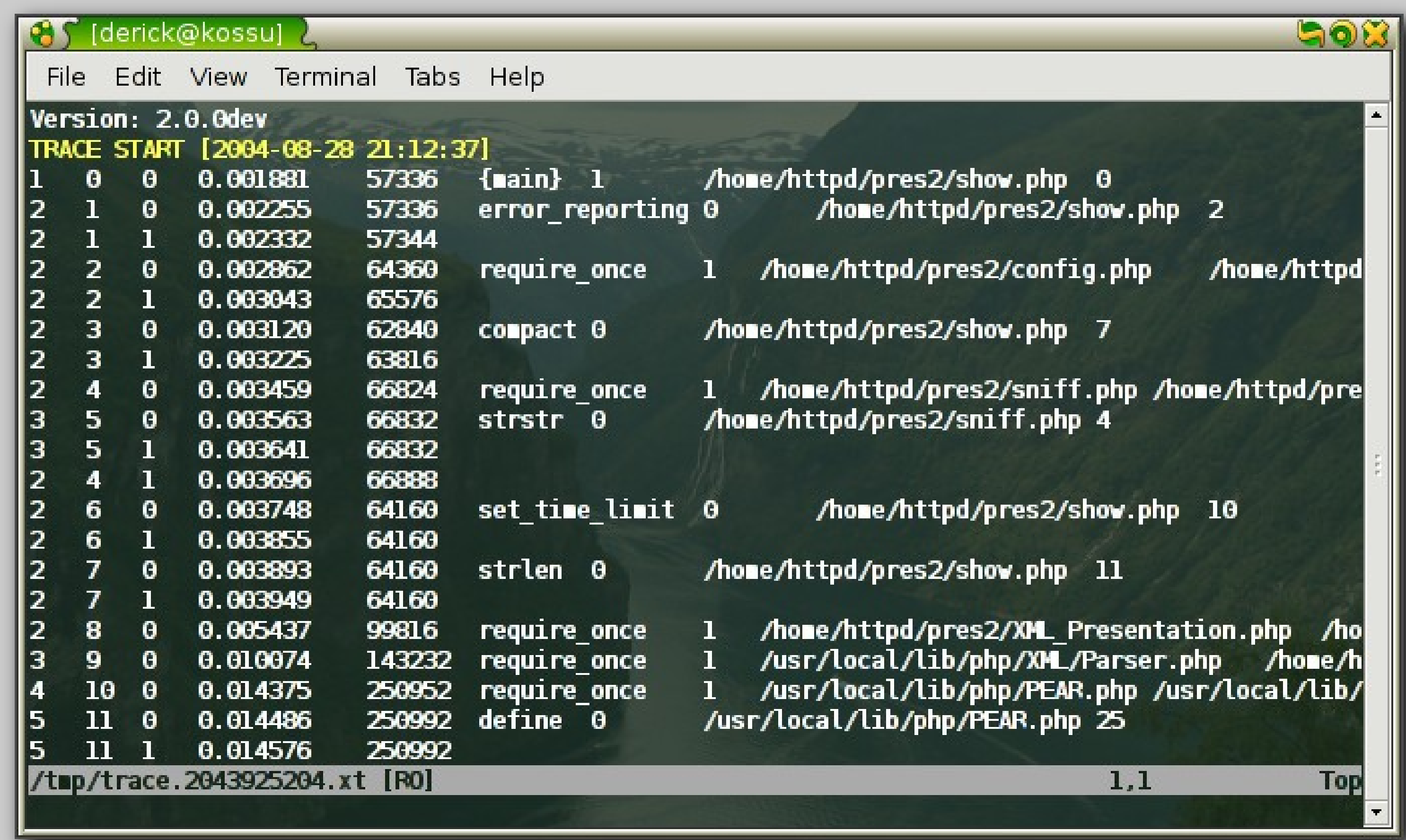

ude
/home/httpd/www.derickrethans.nl/html/php/config.php
require
/home/derick/dev/zetacomponents/trunk/Base/src/ezc_bootstrap.php
req
/home/httpd/www.derickrethans.nl/html/php/api.php
/home/httpd/www.derickrethans.nl/html/php/article_list_visitor.php
/home/httpd/www.derickrethans.nl/html/php/id_directive.php
/home/httpd/www.derickrethans.nl/html/php/comment_directive.php
/home/httpd/www.derickrethans.nl/html/php/comment_box_directive.php
/home/httpd/www.derickrethans.nl/html/php/article_meta_data_directive.php

I Do Need a Debugger

- Xdebug: An Open Source debugging tool
- About 8 years old
- Works on "every" operating system
- Version 2.1 released earlier this year
- PHP 5.1, 5.2, 5.3 and trunk

Function trace to file

Automatic readable format



```
Version: 2.0.0dev
TRACE START [2004-08-28 21:12:37]
1   0   0   0.001881    57336   {main}  1       /home/httpd/pres2/show.php  0
2   1   0   0.002255    57336   error_reporting 0       /home/httpd/pres2/show.php  2
2   1   1   0.002332    57344
2   2   0   0.002862    64360   require_once    1   /home/httpd/pres2/config.php    /home/httpd
2   2   1   0.003043    65576
2   3   0   0.003120    62840   compact 0       /home/httpd/pres2/show.php  7
2   3   1   0.003225    63816
2   4   0   0.003459    66824   require_once    1   /home/httpd/pres2/sniff.php /home/httpd/pre
3   5   0   0.003563    66832   strstr  0       /home/httpd/pres2/sniff.php 4
3   5   1   0.003641    66832
2   4   1   0.003696    66888
2   6   0   0.003748    64160   set_time_limit  0       /home/httpd/pres2/show.php  10
2   6   1   0.003855    64160
2   7   0   0.003893    64160   strlen  0       /home/httpd/pres2/show.php  11
2   7   1   0.003949    64160
2   8   0   0.005437    99816   require_once    1   /home/httpd/pres2/XML_Presentation.php  /ho
3   9   0   0.010074    143232  require_once    1   /usr/local/lib/php/XML/Parser.php   /home/h
4   10  0   0.014375    250952  require_once    1   /usr/local/lib/php/PEAR.php /usr/local/lib/
5   11  0   0.014486    250992  define  0       /usr/local/lib/php/PEAR.php 25
5   11  1   0.014576    250992
/tmp/trace.2043925204.xt [RO]                                           1,1           Top
```

```
xdebug.auto_trace=1       ; enable tracing
xdebug.trace_format=1     ; selects computerized format
xdebug.trace_options=0    ; sets extra option (1 = append)
```

Function trace

Other functionality

- HTML traces
- Tracing only parts of an application with xdebug_start_trace() and xdebug_stop_trace().
- Fetching the trace file name that is being used with xdebug_get_tracefile_name().
- Changing how much data is shown with xdebug.var_display_max_children, xdebug.var_display_max_data and xdebug.var_display_max_depth.

Trace file analysers

One bundled with Xdebug:

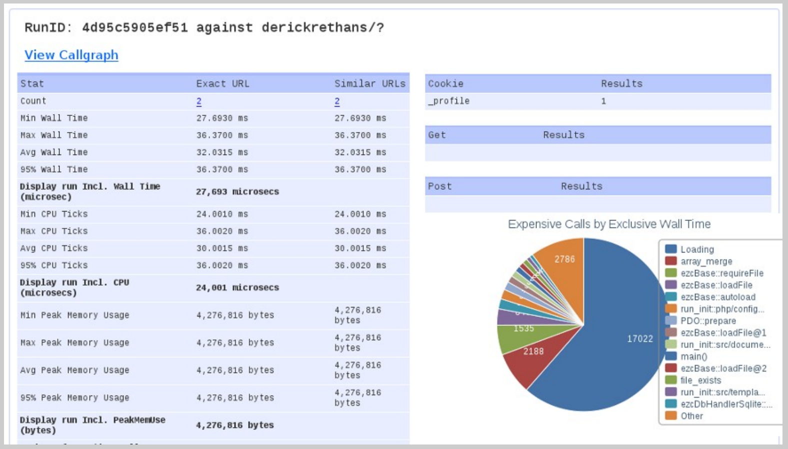
```
php ~/dev/php/xdebug/trunk/contrib/tracefile-analyser.php
...
Showing the 25 most costly calls sorted by 'time-own'.

                                                                Inclusive        Own
function                                                 #calls  time     memory  time     memory
-------------------------------------------------------------------------------------------------
array_pop                                                   715  1.0252  -139880  1.0252  -139880
preg_match                                                 2986  0.3718  1016336  0.3718  1016336
{main}                                                        1  6.4562  7335704  0.3476 -15198832
next                                                        432  0.2386        0  0.2386        0
count                                                      3302  0.2132        0  0.2123        0
ezcQuerySelectSqlite->from                                  434  0.4076   715880  0.1522  -663944
ezcDocumentRstTokenizer->tokenizeString                       1  0.4426   817840  0.1431   435888
ezcQuery::arrayFlatten                                     1303  0.1929   780360  0.1344   642792
drBlogApi->fetchMetaData                                    433  0.3938  2381928  0.1160 -1440896
ezcQuerySelect->select                                      434  0.2590   189496  0.1099  -437048
ezcQueryExpression->getIdentifier                           870  0.1937        0  0.1093  -603752
ezcQuerySelect->where                                       435  0.2155   104792  0.0917  -537512
PDO->prepare                                                434  0.0885   622072  0.0885   622072
array_key_exists                                            107  0.0884        0  0.0884        0
join                                                       1740  0.0882   258264  0.0882   258264
ezcDocumentRstParser->parse                                   4  2.3933  1427376  0.0757  -140208
ezcBase::loadFile                                            88  0.1599  5988856  0.0747  3585136
ezcDocumentRstStack->shift                                  715  1.0971   -59360  0.0719    80520
DateTime->__construct                                       866  0.0710      488  0.0710      488
```

ValaXdebugTools (http://tinyurl.com/vxdebugtools) demo

Profiling

XHProf



RunID: 4d95c5905ef51 against derickrethans/?

View Callgraph

Stat	Exact URL	Similar URLs
Count	2	2
Min Wall Time	27.6930 ms	27.6930 ms
Max Wall Time	36.3700 ms	36.3700 ms
Avg Wall Time	32.0315 ms	32.0315 ms
95% Wall Time	36.3700 ms	36.3700 ms
Display run Incl. Wall Time (microsec)	**27,693 microsecs**	
Min CPU Ticks	24.0010 ms	24.0010 ms
Max CPU Ticks	36.0020 ms	36.0020 ms
Avg CPU Ticks	30.0015 ms	30.0015 ms
95% CPU Ticks	36.0020 ms	36.0020 ms
Display run Incl. CPU (microsecs)	**24,001 microsecs**	
Min Peak Memory Usage	4,276,816 bytes	4,276,816 bytes
Max Peak Memory Usage	4,276,816 bytes	4,276,816 bytes
Avg Peak Memory Usage	4,276,816 bytes	4,276,816 bytes
95% Peak Memory Usage	4,276,816 bytes	4,276,816 bytes
Display run Incl. PeakMemUse (bytes)	**4,276,816 bytes**	

Cookie	Results
_profile	1

Get	Results

Post	Results

Install:

```
pecl download xhprof-beta
tar -xvzf xhprof-0.9.2.tgz
cd xhprof-0.9.2/extension
phpize && ./configure && make install
```

In php.ini:

```
extension=xhprof.so
xhprof.output_dir=/tmp
```

Download XHGui:

```
$ cd /home/httpd
$ git clone https://github.com/preinheimer/xhprof
```

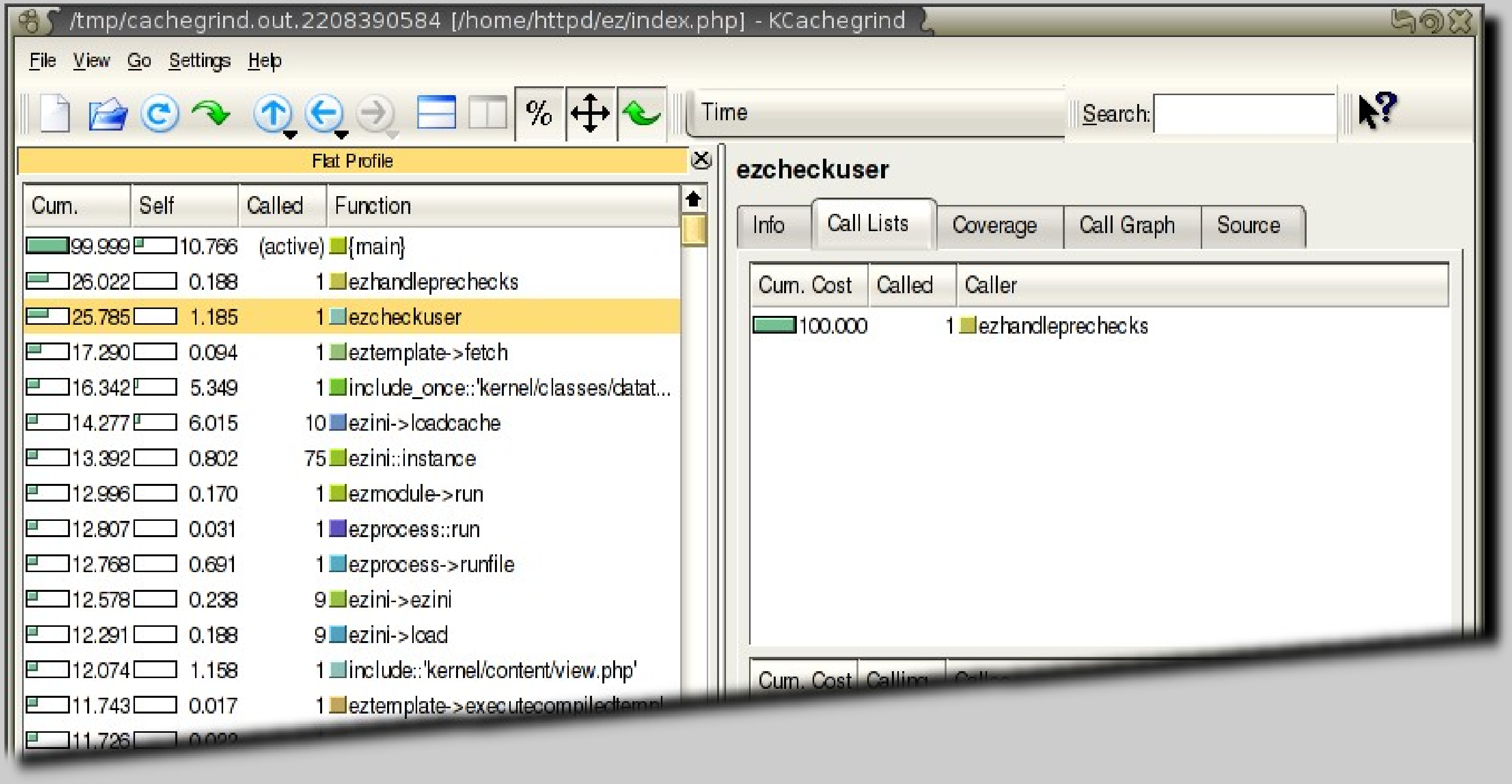
Config XHGui (xhprof/xhprof_lib/config.php:

```
$_xhprof['dbhost'] = 'localhost';
$_xhprof['dbuser'] = 'root';
$_xhprof['dbpass'] = 'root';
$_xhprof['dbname'] = 'xhprof';
$_xhprof['servername'] = 'derickrethans';
$_xhprof['namespace'] = 'MySite';
$_xhprof['url'] = 'http://xhprof/';

</VirtualHost>
```

Profiling

KCacheGrind's Flat Profile and Call List


/tmp/cachegrind.out.2208390584 [/home/httpd/ez/index.php] - KCachegrind
File View Go Settings Help
Time
Search:
Flat Profile
ezcheckuser
Info
Call Lists
Coverage
Call Graph
Source

Cum.	Self	Called	Function
99.999	10.766	(active)	{main}
26.022	0.188	1	ezhandleprechecks
25.785	1.185	1	ezcheckuser
17.290	0.094	1	eztemplate->fetch
16.342	5.349	1	include_once::'kernel/classes/datat...
14.277	6.015	10	ezini->loadcache
13.392	0.802	75	ezini::instance
12.996	0.170	1	ezmodule->run
12.807	0.031	1	ezprocess::run
12.768	0.691	1	ezprocess->runfile
12.578	0.238	9	ezini->ezini
12.291	0.188	9	ezini->load
12.074	1.158	1	include::'kernel/content/view.php'
11.743	0.017	1	eztemplate->executecompiledtempl
11.726	0.022		

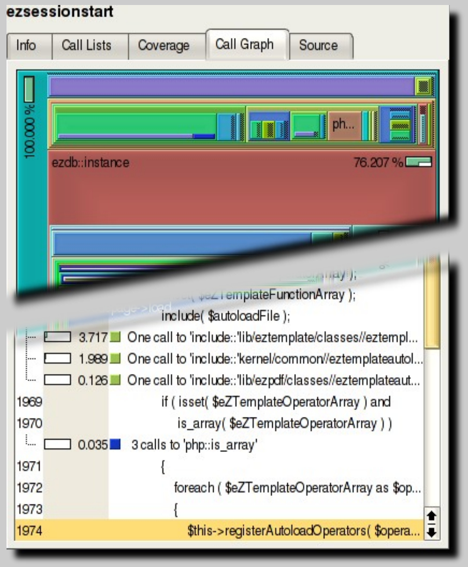
Cum. Cost	Called	Caller
100.000	1	ezhandleprechecks

Cum. Cost | Calling | Calle

```
xdebug.profiler_enable=1                 ; enable profiler
xdebug.profiler_output_dir=/tmp           ; output directory
xdebug.profiler_output_name=cachegrind.out.%p
```

Profiling

KCacheGrind's Call Graph and Source Annotations



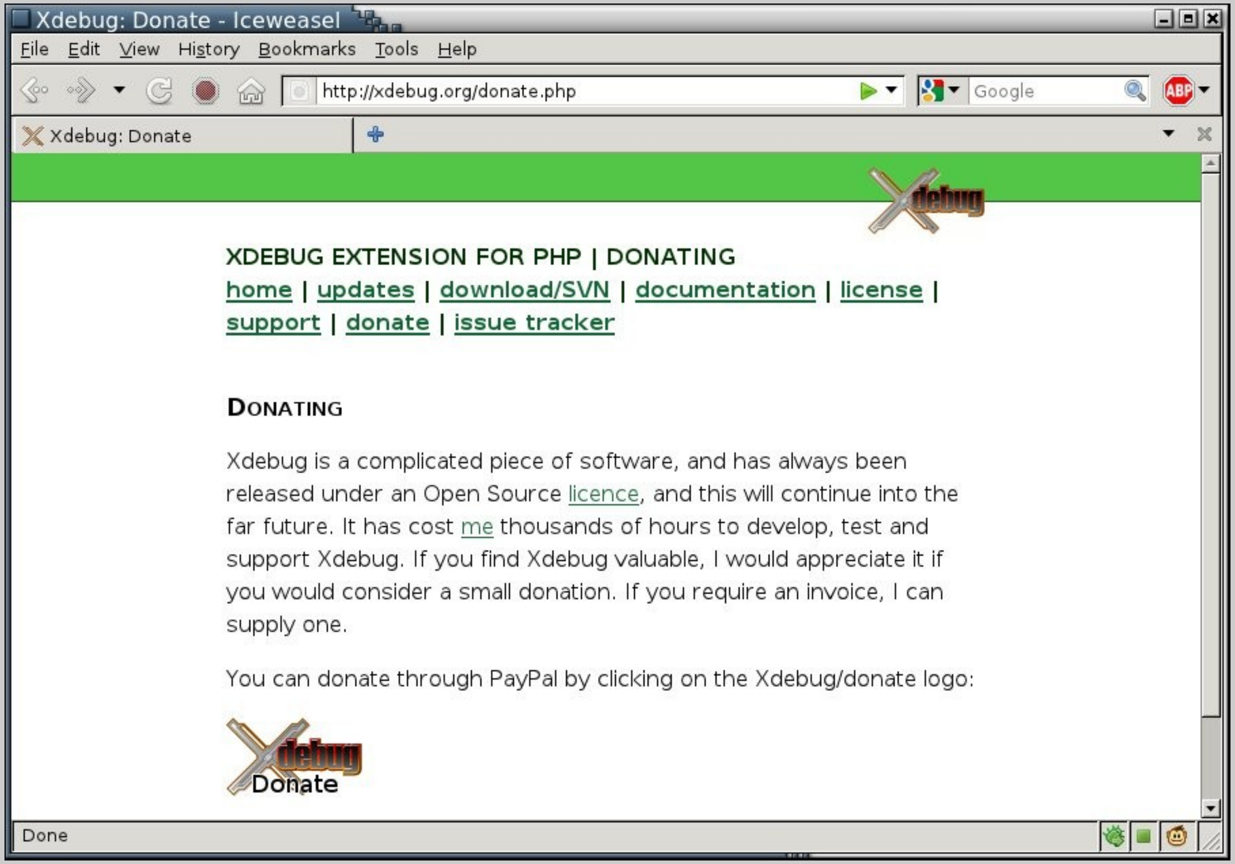
```
               $eZTemplateFunctionArray );
          include( $autoloadFile );
   3.717  One call to 'include::'lib/eztemplate/classes//eztempl...
   1.989  One call to 'include::'kernel/common//eztemplateautol...
   0.126  One call to 'include::'lib/ezpdf/classes//eztemplateaut...
          if ( isset( $eZTemplateOperatorArray ) and
               is_array( $eZTemplateOperatorArray ) )
   0.035   3 calls to 'php::is_array'
          {
              foreach ( $eZTemplateOperatorArray as $op...
              {
                  $this->registerAutoloadOperators( $opera...
```

- Call graph
- Area shows time spend
- Stacked to show callees
- Source annotations
- Number of calls
- Total time per function

demo

Xdebug

- It's Open Source and free (as in "free beer")
- Working on Xdebug takes up a lot of spare time
- I don't have a lot of spare time


Xdebug: Donate - Iceweasel
File Edit View History Bookmarks Tools Help
http://xdebug.org/donate.php
Google
Xdebug: Donate
Done

XDEBUG EXTENSION FOR PHP | DONATING

home | updates | download/SVN | documentation | license | support | donate | issue tracker

DONATING

Xdebug is a complicated piece of software, and has always been released under an Open Source licence, and this will continue into the far future. It has cost me thousands of hours to develop, test and support Xdebug. If you find Xdebug valuable, I would appreciate it if you would consider a small donation. If you require an invoice, I can supply one.

You can donate through PayPal by clicking on the Xdebug/donate logo:

Donate

Resources

- Siege: http://www.joedog.org/index/siege-home
- Apache Zeta Components: http://incubator.apache.org/zetacomponents/
- Inclued: http://uk2.php.net/inclued
- ValaXdebugTools: http://tinyurl.com/vxdebugtools
- XHProf: http://mirror.facebook.net/facebook/xhprof/
- XHGui: https://github.com/preinheimer/xhprof
- Xdebug: http://xdebug.org
- If you like Xdebug: http://xdebug.org/donate.php
- These slides: http://derickrethans.nl/talks.html