

# Profiling with XDebug & kCacheGrind

Mike Soule

EZYield.com

[msoule@ezyield.com](mailto:msoule@ezyield.com)

# Why Profile?

- Slow websites are bad
- Save money
- Learn

# Tools Needed

- **XDebug** <http://xdebug.org>
- **Ruby-Prof** <http://ruby-prof.rubyforge.org>
- **kCacheGrind** <http://kcachegrind.sourceforge.net/cgi-bin/show.cgi>
- **WinCacheGrind** <http://sourceforge.net/projects/wincachegrind>

# Installation: XDebug

- Use PECL

```
root$ pecl install xdebug
```

- XDebug php.ini

```
zend_extension="/usr/lib/php/extensions/no-debug-  
non-zts-20060613/xdebug.so"
```

```
xdebug.default_enable = On
```

```
xdebug.show_local_vars = 1
```

```
xdebug.profiler_append = 0
```

```
xdebug.profiler_enable = 1
```

```
xdebug.profiler_output_dir = "/tmp/xdebug"
```

# Tip for PHP \*.ini Files

- PHP Compile Option:  
`--with-config-file-scan-dir=/etc/php.d`
- Add XDebug settings to:  
`/etc/php.d/xdebug.ini`

# Installation: kCacheGrind

## Linux

- Use a package manager!
- Requires KDE & Qt libraries

# Installation: kCacheGrind

## Mac OS X

- Use fink (<http://finkproject.org>)

```
root$ sudo fink install kcachegrind
```

- Select non “SSL” options for easier install
- WARNING: Lots of dependencies
- Get coffee... at B&N... while reading Atlas Shrugged
- Add /sw/bin your PATH in X11

# Example:

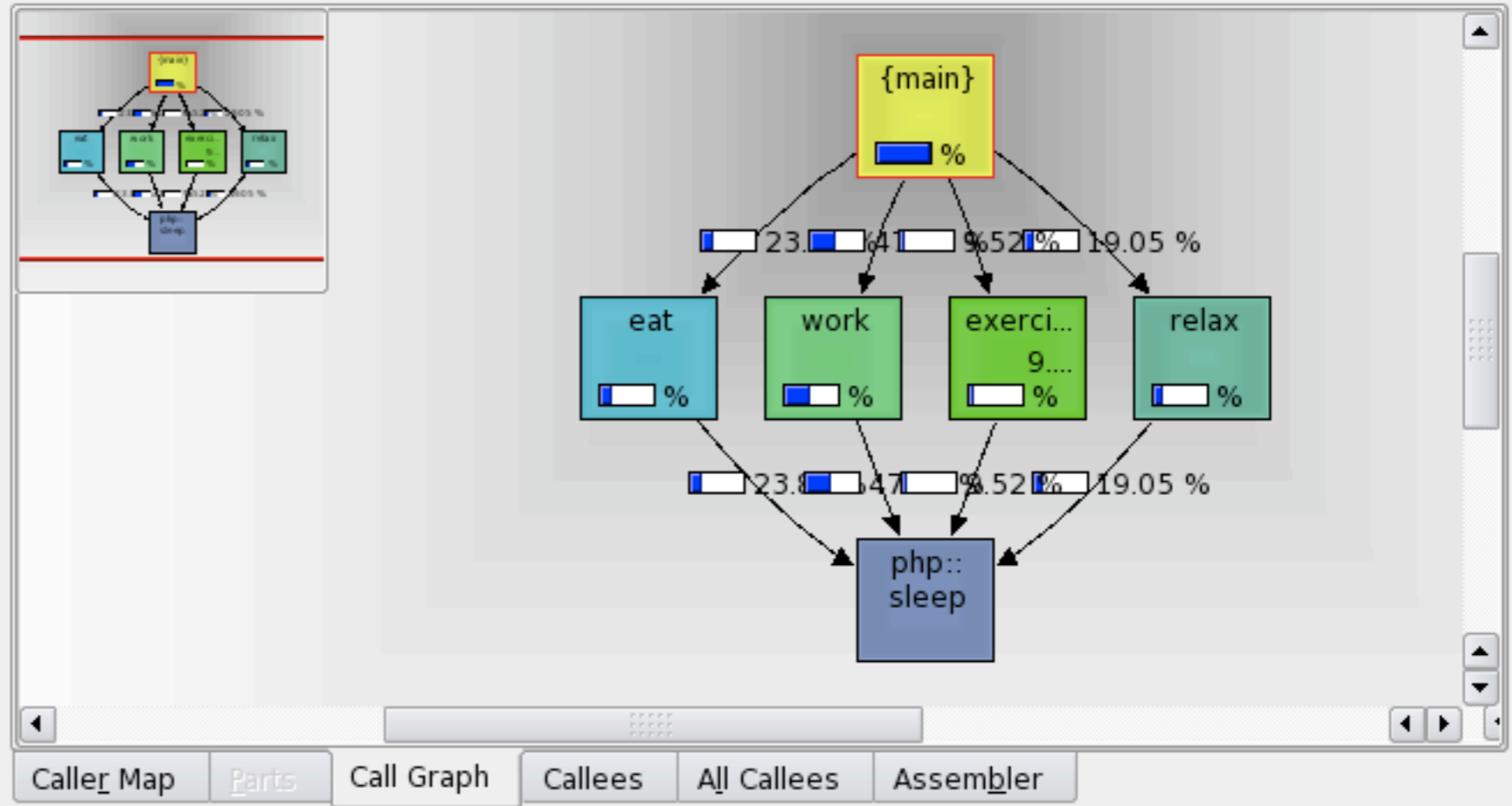
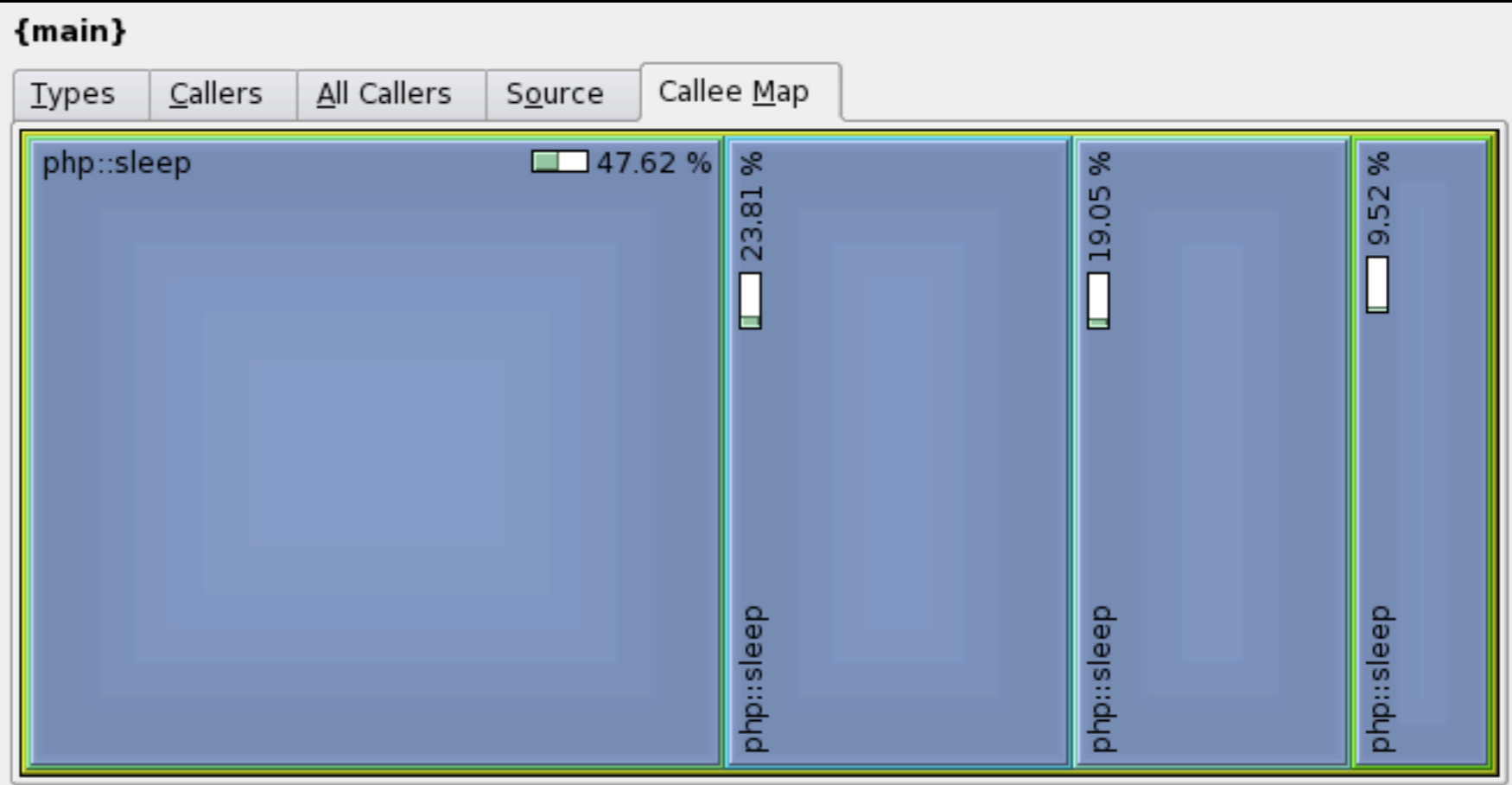
- life.php

```
1 <?php
2 eat();
3 work();
4 exercise();
5 relax();
6 function eat() { sleep(10); }
7 function work() { sleep(20); }
8 function relax() { sleep(8); }
9 function exercise() { sleep(4); }
10 ?>
```



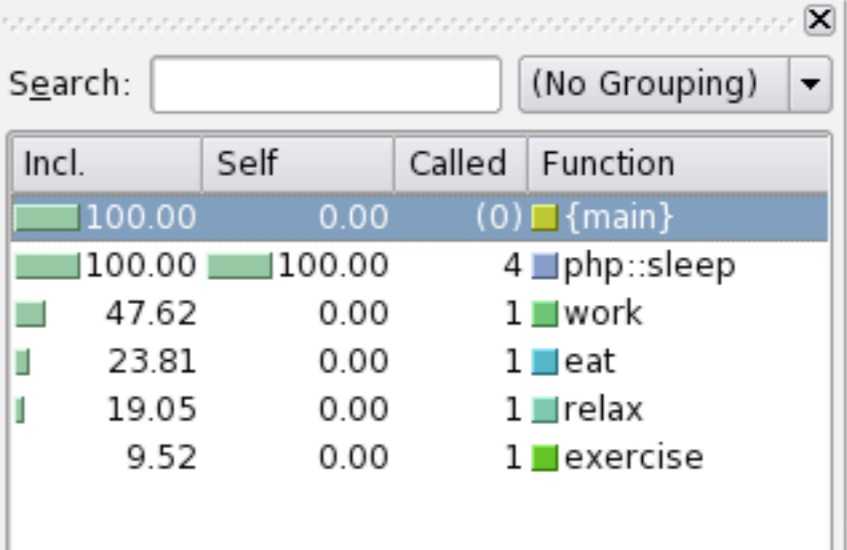
Search:  (No Grouping) ▾

Incl.	Self	Called	Function
100.00	0.00	(0)	{main}
100.00	100.00	4	php::sleep
47.62	0.00	1	work
23.81	0.00	1	eat
19.05	0.00	1	relax
9.52	0.00	1	exercise



# Function Calls

- Lists called functions
- Group by file, class etc.
- ‘php::’ = native php function
- ‘Incl.’ = relative exec time
- ‘Self’ = time taken by function - other func calls
- ‘Called’ = number of times called

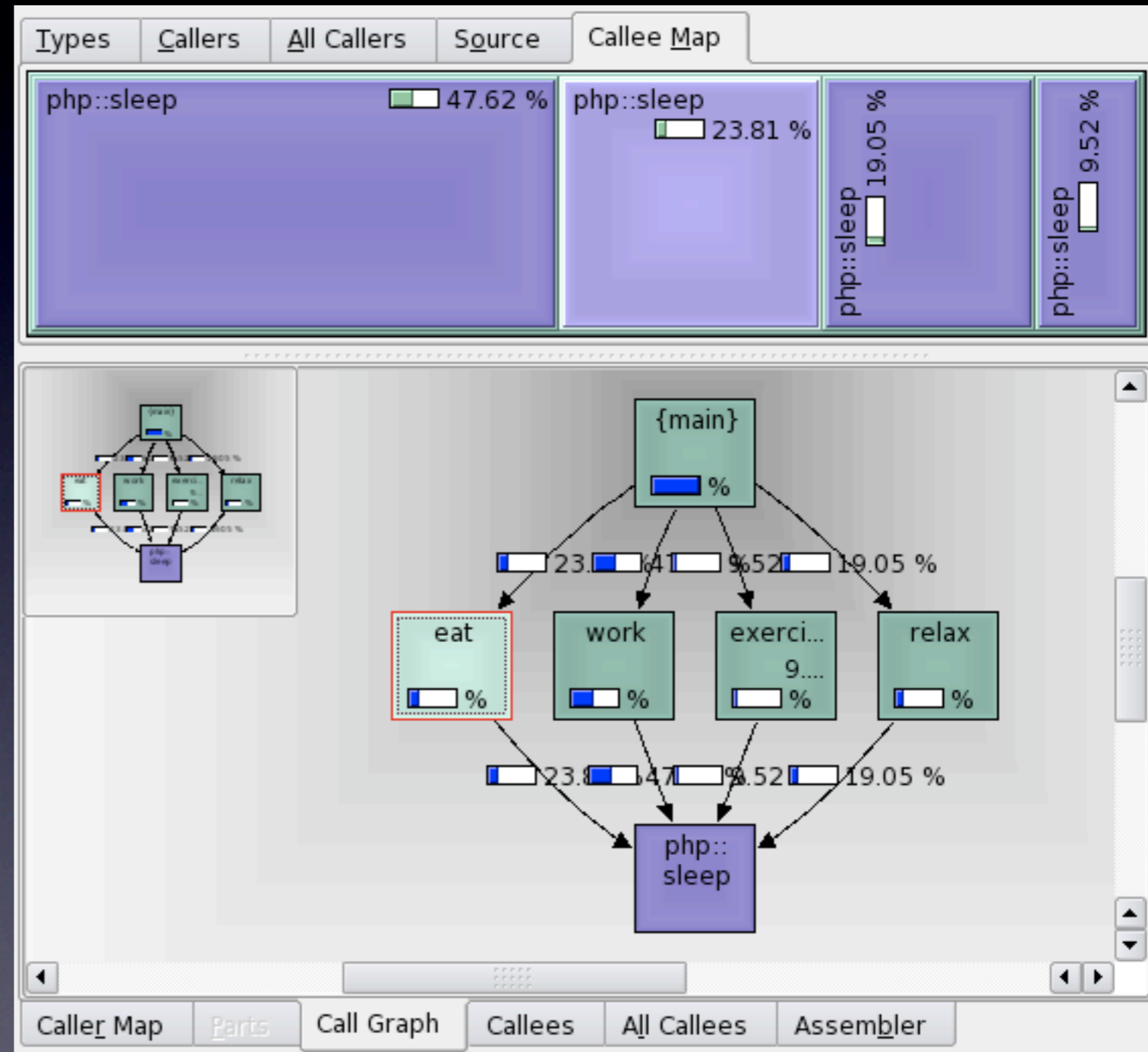


The screenshot shows a window titled 'Function Calls' with a search bar and a dropdown menu set to '(No Grouping)'. Below is a table with columns: 'Incl.', 'Self', 'Called', and 'Function'. The data is as follows:

Incl.	Self	Called	Function
100.00	0.00	(0)	{main}
100.00	100.00	4	php::sleep
47.62	0.00	1	work
23.81	0.00	1	eat
19.05	0.00	1	relax
9.52	0.00	1	exercise

# Graphs

- ‘Callee Map’ shows proportional time
- ‘Call Graph’ shows relationships between function calls



# Tips

- Profiles are relative vs benchmarks which are controlled
- Add profile output to your repository for future comparison
- Tools such as apache bench and http\_load are useful for testing load capability and consistency across your site



# We're Hiring

- We're looking for PHP developers!
- Send resumes to [techsupport@ezyield.com](mailto:techsupport@ezyield.com)

# Profiling with XDebug & kCacheGrind

Mike Soule

EZYield.com

[msoule@ezyield.com](mailto:msoule@ezyield.com)