

PHP code profiling using XDebug

By

Gennady Feldman

June 16, 2009

Overview

- Introduction
- Profiling with Xdebug
- Enabling Profiling in Xdebug
- Summary
- Q & A

Introduction

- Xdebug is a powerhouse extension for PHP.
- It has a ton of features like debugging, tracing, profiling and analyzing problems with PHP code.
- Today we will focus on some of it's profiling abilities.

Profiling with Xdebug

- Xdebug can generate cachegrind files that contain profiling information.
- It can be used to find bottle-necks in your scripts.
- You need a tool to read these files.
- Some of your options are: WinCacheGrind, KCacheGrind, webgrind.
- These tools allow you to analyze script execution (execution time breakdown, # function calls, etc).

WinCacheGrind

The screenshot displays the WinCacheGrind application interface. The main window title is "WinCacheGrind - [D:\u1294\gauldong.net\web\csf\csf_pub\admin\update_db.php (cachegrind.o...". The menu bar includes File, Edit, View, Profiler, Tools, Window, and Help. The toolbar contains icons for file operations and execution. The breadcrumb shows the current file: "update_db.php" > "index.php" > "index.php".

The left pane shows a call tree for the file "D:\u1294\gauldong.net\web\csf\csf_pub\admin\update_db.php". The tree starts with "(main)" and includes several function calls such as "require_once: admin.inc.php", "client::verifysuper", "csf::getdb", "csfdb->tableexists", "php::is_null", "php::in_array", "client::issuper", "client::getusername", "csf::getauth", "php::is_null", "csf::getuser", "gduser->issuper", "gduser->exists", "csf::getuser", "gduser->getrolen", "gduser->exist", and "csf::getuser".

The right pane shows the file path and execution statistics: "D:\u1294\gauldong.net\web\csf\csf_pub\admin\update_db.php", "File: D:\u1294\gauldong.net\web\csf\csf_pub\admin\update_db.php", "Self time: - (-)", and "Cumulative time: 9.399ms (100,00%)". Below this, there are tabs for "Line by Line" and "Overall", and a "Find:" input field.

The main performance table is as follows:

Function	Avg. Self	Avg. Cum.	Total Self	Total Cum.	Calls
mdb_parser->cdatahandler	-	-	890ms	890ms	22.587
php::strtolower	-	-	107ms	107ms	20.389
php::implode	-	-	136ms	136ms	13.541
mdb_parser->endhandler	-	-	992ms	1.185ms	6.786
mdb_parser->starthandler	-	-	555ms	698ms	6.796

Below the table is a sub-table showing call details:

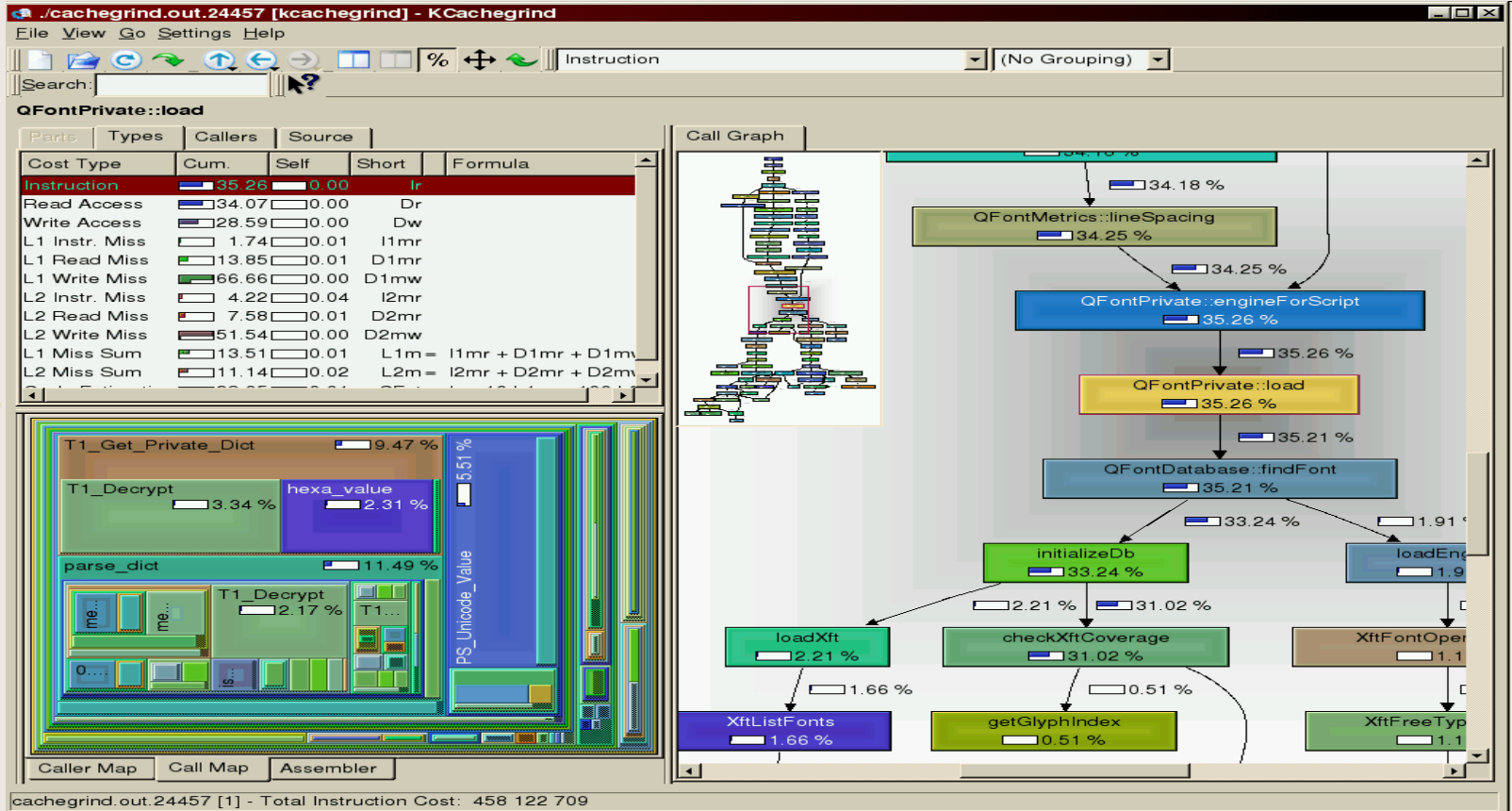
Num.	Self	Cum.	Called by	Called from
1	-	-	php::xml_parse	php:internal (505)
2	-	-	php::xml_parse	php:internal (505)
3	-	-	php::xml_parse	php:internal (505)

The bottom pane shows a list of cache files:

File Name	Title	Modified	Size
cachegrind.out.468619628	D:\u1294\gauldong.net\web\csf\csf_pub\admin\update_db.php	19/01/2005 21:54	9.638.95z
cachegrind.out.1043930140	D:\u1294\gauldong.net\web\index.php	19/01/2005 21:37	2.612.03E
cachegrind.out.3762595594	D:\u1294\gauldong.net\web\member\index.php	19/01/2005 23:55	2.190.65E

At the bottom left, it states "Allocated memory: 8.026.456 bytes".

KCacheGrind Screenshot



WebGrind Screenshot

The screenshot shows the WebGrind browser profiling tool interface. The browser address bar displays `http://localhost/webgrind/`. The tool title is `webgrind v0.8` with the subtitle `profiling in the browser`. The current profile is for `musikfreak/index.php` (cachegrind) in `percent` mode. A progress bar indicates that 418 different functions were called in 148 milliseconds, with 109 functions shown.

Function	Invocation Count	Total Self Cost	Total Inclusive Cost
▶ Zend_Loader::loadClass	11	7.51	20.11
▶ Zend_Controller_Front->dispatch	1	5.89	62.75
▶ include::/Library/WebServer/Documents/musikfreak/app/views/scripts/news/index.phtml	1	5.12	8.07
▶ Zend_Cache::factory	1	3.77	7.42
▶ php::unserialize	4	3.24	3.24
▶ include_once::/Library/WebServer/Documents/zend/library/Zend/Controller/Front.php	1	3.20	10.29
▶ require_once::/Library/WebServer/Documents/zend/library/Zend/Controller/Router/Rewrite.php	1	3.04	3.78
▶ require_once::/Library/WebServer/Documents/zend/library/Zend/Controller/Action.php	1	2.57	2.57
▶ {main}	1	2.47	99.99
▶ require_once::/Library/WebServer/Documents/zend/library/Zend/View.php	1	2.22	2.49
▶ Zend_Controller_Action_Helper_ViewRenderer->getInflector	3	2.01	10.72
▶ require_once::/Library/WebServer/Documents/zend/library/Zend/Cache/Backend/File.php	1	1.55	1.55
▶ require_once::/Library/WebServer/Documents/zend/library/Zend/Controller/Action/Helper/Abstract.php	1	1.45	4.02
▶ require_once::/Library/WebServer/Documents/zend/library/Zend/Filter/Inflector.php	1	1.41	2.12
▶ include_once::/Library/WebServer/Documents/zend/library/Zend/Config/Ini.php	1	1.40	1.41
▶ Zend_Loader::loadFile	1	1.39	1.44
▶ Zend_Loader_PluginLoader->load	6	1.28	3.45
▶ require_once::/Library/WebServer/Documents/zend/library/Zend/Controller/Request/Http.php	1	1.28	1.62
▶ require_once::/Library/WebServer/Documents/zend/library/Zend/Controller/Dispatcher/Standard.php	1	1.27	1.71
▶ php::crc32	2	1.18	1.18
▶ Zend_Controller_Dispatcher_Abstract->_formatName	7	1.17	1.25
▶ Zend_Controller_Front->getDispatcher	8	1.15	2.95
▶ require_once::/Library/WebServer/Documents/zend/library/Zend/Controller/Action/HelperBroker.php	1	1.15	5.78

Enabling Profiling

- `xdebug.profiler_append`
 - Default: 0
- `xdebug.profiler_enable`
 - Default: 0
- `xdebug.profiler_output_dir`
 - Default: /tmp
- `xdebug.profiler_output_name`
 - Default: `cachegrind.out.%p`

Summary

- Xdebug is a very powerful extension with a ton of great features.
- We only covered one tiny thing that this extension can do.
- This is a MUST in development environment.
- Come see me speak at NYPHP August 25, 2009 covering Xdebug from A to X.

Resources

- Xdebug: <http://www.xdebug.org>
- WinCacheGrind:
<http://sourceforge.net/projects/wincachegrind/>
- KCacheGrind:
<http://kcachegrind.sourceforge.net/>
- WebGrind: <http://code.google.com/p/webgrind/>
- My blog: <http://www.gena01.com/blog/>



October 14 - 16, 2009

New World Stages, New York City

[HOME](#) | [REGISTER](#) | [SPEAKERS & SESSIONS](#) | [SPONSOR](#) | [PRESS](#) | [LOCATION](#) | [CONTACT](#)

Win A Free Ticket!

Register Now 

www.ManageT.me

**\$50-off with rate code:
NYWebPerformance**

**Learn from the
greatest minds in IT!**

Zeev Suraski

CTO of Zend Technologies
Co-Creator of PHP

Carol Rizzo

Technology Executive
Former-CTO of Citigroup, AIG & Kaiser
Permanente

John A. Glowacki Jr.

Corporate VP & CTO of CSC

Marco Tabini

CEO of Marco Tabini & Associates
Publisher of php|architect and Python
Magazine



Ron Zahavi

Chief Business Architect, Unisys Corporation



Michael "Monty" Widenius

CEO of Monty Program Ab
Founder and Creator of MySQL



Gary Vaynerchuk

Founder of Wine Library TV
Internet Celebrity & Social Media Sommelier



David Teten

CEO of Teten Advisors
Serial Entrepreneur

Q & A
