Find out where your PHP code is slowing down (Performance Issue)

Asked 16 years, 3 months ago Modified 9 years, 7 months ago

Viewed 33k times Part of PHP Collective

31

Here's my first question at SO.

I have a internal application for my company which I've been recently ask to maintain. The applications is built in PHP and its fairly well coded (OO, DB Abstraction, Smarty) nothing WTF-ish.

The problem is the applications is **very slow**.

How do I go about finding out what's slowing the application down? I've optimized the code to make very few DB queries, so I know that it is the PHP code which is taking a while to execute. I need to get some tools which can help me with this and need to devise a strategy for checking my code.

I can do the checking/strategy work myself, but I need more PHP tools to figure out where my app is crapping up.

Thoughts?

PHP php mysql performance profiling

Share

Improve this question

Follow

edited Sep 11, 2008 at 2:48

Chris Upchurch

15.5k ● 6 ● 52 ● 66

asked Sep 11, 2008 at 2:47

Rushi

1,662 ● 3 ● 17 ● 24

12 Answers

Sorted by: Highest score (default)

40

I've used XDebug profiling recently in a similiar situation. It outputs a full profile report that can be read with many common profiling apps (Can't give you a list though, I just used the one that came with slackware).

Share Improve this answer

Follow

edited Sep 20, 2008 at 22:12

Polsonby

22.9k ● 19 ● 60 ● 74

answered Sep 11, 2008 at 3:06

Champo

3,419 ● 30 ● 32

9

As Juan mentioned, xDebug is excellent. If you're on Windows, WinCacheGrind will let you look over the reports.

Share Improve this answer

Follow

answered Sep 11, 2008 at 14:58

MattBelanger

5,350 ● 6 ● 38 ● 35

I have a windows machine lying around that I can put to good use – Rushi Sep 11, 2008 at 15:27

2 KCacheGrind (KDE/Linux) is much better. – blueyed Feb 19, 2010 at 1:44

6

Watch this presentation by Rasmus Lerdorf (creator of PHP). He goes into some good examples of testing PHP speed and what to look for as well as some internals that can slow things down. XDebug is one tool he uses. He also makes a very solid point about knowing what performance cost you're getting into with frameworks.

Video: http://www.archive.org/details/simple_is_hard

Slides (since it's hard to see on the video): http://talks.php.net/show/drupal08/1

Share Improve this answer

Follow

answered Sep 11, 2008 at 19:37

Patrick Hogan

2,108 ● 4 ● 21 ● 28

4

There are many variables that can impact your application's performance. I recommend that you do not instantly assume PHP is the problem.

First, how are you serving PHP? Have you tried basic optimization of Apache or IIS itself? Is the server busy processing other kinds of requests? Have you taken advantage of a PHP code accelerator? One way to test whether the server is your bottleneck is to try running the application on another server.

Second, is performance of the entire application slow, or does it only seem to affect certain pages? This could give you an indication of where to start analyzing performance. If the entire application is slow, the problem is more likely in the underlying server/platform or with a global SQL query that is part of every request (user authentication, for example).

Third, you mentioned minimizing the number of SQL queries, but what about optimizing the existing queries? If you are using MySQL, are you taking advantage of the various strengths of each storage system? Have you run EXPLAIN on your most important queries to make sure they are properly indexed? This is critical on queries that access big tables; the larger the dataset, the more you will notice the effects of poor indexing. Luckily, there are many articles such as this one which explain how to use EXPLAIN.

Fourth, a common mistake is to assume that your database server will automatically use all of the resources available to the system. You should check to make sure you have explicitly allocated sufficient resources to your database application. In MySQL, for example, you'll want to add custom settings (in your my.cnf file) for things like key buffer, temp table size, thread concurrency, innodb buffer pool size, etc.

If you've double-checked all of the above and are still unable to find the bottleneck, a code profiler like Xdebug can definitely help. Personally, I prefer the Zend Studio profiler, but it may not be the best option unless you are already taking advantage of the rest of the Zend Platform stack. However, in my experience it is very rare that PHP itself is the root cause of slow performance. Often, a code profiler can help you determine with more precision which DB queries are to blame.

Share Improve this answer

Follow

answered Sep 15, 2008 at 16:28

giltotherescue

768 ● 6 ● 9

You're right too. I actually looked at the database end of things and they seemed fine. Apache was an issue, but it was fixed. I will install an op-code cache program too. But still profiling your app helps improve general programming anyways. – Rushi Sep 21, 2008 at 4:42

Also You could use APD (Advanced PHP Debugger).

3

It's quite easy to make it work.

```
$ php apd-test.php

$ pprofp -l pprof.SOME_PID

Trace for /Users/martin/develop/php/apd-test/apd-test.
Total Elapsed Time = 0.12
Total System Time  = 0.01
Total User Time    = 0.07


         Real         User        System              s
%Time (excl/cumm)  (excl/cumm)  (excl/cumm) Calls    c
Usage Name
--------------------------------------------------------
-------
71.3 0.06 0.06  0.05 0.05  0.01 0.01  10000  0.0000
in_array
27.3 0.02 0.09  0.02 0.07  0.00 0.01  10000  0.0000
my_test_function
 1.5 0.03 0.03  0.00 0.00  0.00 0.00      1  0.0000
apd_set_pprof_trace
 0.0 0.00 0.12  0.00 0.07  0.00 0.01      1  0.0000
```

There is a nice tutorial how to compile APD and make profiling with it : http://martinsikora.com/compiling-apd-for-php-54

Share Improve this answer Follow

edited Feb 6, 2015 at 9:57

answered Mar 30, 2013 at 9:51

Fedir RYKHTIK

9,954 ● 7 ● 59 ● 69

While this link may answer the question, it is better to include the essential parts of the answer here and provide the link for reference. Link-only answers can become invalid if the linked page changes. – Lee Taylor Feb 6, 2015 at 3:21

The answer is "Use APD". Additional link it's for those, who has problems with the installation. – Fedir RYKHTIK Feb 6, 2015 at 9:54

2

phpED (http://www.nusphere.com/products/phped.htm) also offers great debugging and profiling, and the ability to add watches, breakpoints, etc in PHP code. The integrated profiler directly offers a time breakdown of each function call and class method from within the IDE. Browser plugins also enable quick integration with Firefox or IE (i.e. visit slow URL with browser, then click button to profile or debug).

It's been very useful in pointing out where the app is slow in order to concentrate most coding effort; and it avoids wasting time optimising already fast code. Having tried Zend and Eclipse, I've now been sold on the ease of use of phpED.

Bear in mind both Xdebug and phpED (with DBG) will require an extra PHP module installed when debugging against a webserver. phpED also offers (untried by me) a local debugging option too.

Share Improve this answer answered Sep 11, 2008 at 21:02

Follow

RobS
21 • 2

2

Xdebug profile is definitely the way to go. Another tip - WincacheGrind is good, but not been updated recently. http://code.google.com/p/webgrind/ - in a browser may be an easy and quick alternative.

Chances are though, it's still the database anyway. Check for relevant indexes - and that it has sufficient memory to cache as much of the working data as possible.

Share Improve this answer

Follow

answered Sep 13, 2008 at 0:24

Alister Bulman
35.1k • 9 • 74 • 111

1

ifs its a large code base try apc if you're not already.

http://pecl.php.net/package/APC

Share Improve this answer

Follow

answered Sep 11, 2008 at 20:32

Brendon-Van-Heyzen
2,493 • 2 • 24 • 24

Hey there. Were you te review your question, would you leave it as it is? You probably know what to do however, so I'll leave you do it. Please include relevant parts of your link :)
– Félix Adriyel Gagnon-Grenier Feb 5, 2015 at 18:16

1

you can also try using the register_tick_function function in php. which tells php to call a certain function periodcally through out your code. You could then keep track of which function is currently running and the amount of time between calls. then you could see what's taking the most time.
http://www.php.net/register_tick_function

Share Improve this answer

Follow

answered Sep 12, 2008 at 19:11

SeanDowney

17.7k ● 20 ● 83 ● 90

0

We use Zend Development Environment (windows). We resolved a memory usage spike yesterday by stepping through the debugger while running Process Explorer to watch the memory/cpu/disk activity as each line was executed.

Process Explorer: http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx.

ZDE includes a basic performance profiler that can show time spent in each function call during page requests.

Share Improve this answer

Follow

answered Sep 11, 2008 at 3:47

Mark

7,041 ● 1 ● 42 ● 39

0

I use a combination of PEAR Benchmark and log4php.

At the top of scripts I want to profile I create an object that wraps around a Benchmark_Timer object. Throughout the code, I add in `$object->setMarker("name");` calls, especially around suspect code.

The wrapper class has a destroy method that takes the logging information and writes it to log4php. I typically send this to syslog (many servers, aggregates to one log file on one server).

In debug, I can watch the log files and see where I need to improve things. Later on in production, I can parse the log files and do performance analysis.

It's not xdebug, but it's always on and gives me the ability to compare any two executions of the code.

Share Improve this answer

Follow

answered Sep 11, 2008 at 14:50

Gary Richardson

16.4k ● 10 ● 54 ● 48

Even though I'm going with XDebug i will check out Pear Benchmark. I can use it in other projects – Rushi Sep 11, 2008 at 15:28

0

You can also look at the HA Proxy or any other load balancing solution if your server degraded performance is the cause of the application slow processing. server.

Share Improve this answer

Follow

answered May 13, 2015 at 8:38

Asuquo12

835 ● 18 ● 27