Core - Bug # 55294

Status:	Closed	**Priority:**	Must have
Author:	Jan-Erik Revsbech	**Category:**	Caching
Created:	2014-01-24	**Assigned To:**	Benjamin Mack
Updated:	2015-02-28	**Due date:**	
TYPO3 Version:	6.2		
PHP Version:			
Complexity:	hard		
Is Regression:	No		
Sprint Focus:			
Subject:	Under heavy load, two request can simultaneously write to same cache file - should use locking		

Description

On sites with rather high load, we have witnessed a rare problem, where two request is apparently writing to the same cache file at the same time. This happens if the cache is cleared, and two requests both try to populate the cache_core cache, which is i SimpleFileBackend cache.

In some cases the file might be a file that is only needed by an extensions, and pages using that will fail, in some cases the file might be one that is also used by the backend, which means the backend is effectively broken, and its not possible to clear cache (other that manually deleteing the typo3temp/Cache/Code directory).

The problem is much worse on a slow filesystem, and I believe that it is due to SimpleFileBackend nt using a filelock before writing the file.

The fix should be very easy. Simply either issue a flock directly in SimpelFileBackend, og use TYPO3 locking mechanism.

Related issues:

related to Core - Feature # 47712: Refactor Locking	**Resolved**	**2013-02-08**
related to Core - Bug # 55099: PHP Warnings after clearing configuration cach...	**Resolved**	**2014-01-17**
duplicated by Core - Bug # 63135: Need a stopgap fix for SimpleFileBackend ca...	**Resolved**	**2014-01-24**

History

#1 - 2014-01-24 13:31 - Markus Klein

- Status changed from New to Accepted

- Complexity set to medium

#2 - 2014-02-28 01:19 - Markus Klein

- Priority changed from Should have to Must have

- Complexity changed from medium to hard

We're investigating a general locking system for all cache assesses to solve this independent of the CacheBackend used.

#3 - 2014-03-11 23:59 - Markus Klein

- Status changed from Accepted to Needs Feedback

Which cashed are affected for you Jan-Erik?

If it is the classloader caches, then the fix is pending already in #54906.

If other caches are affected please tell us which, because each of them needs careful manual investigation.

We can't employ a general and safe locking mechanism to all caches.

#4 - 2014-07-10 14:56 - Alexander Opitz

- Status changed from Needs Feedback to Closed

- Target version deleted (6.2.0)

No feedback within the last 90 days => closing this ticket.

If you think that this is the wrong decision or experience this issue again, then please write to the mailing list typo3.teams.bugs with issue number and an explanation or open a new ticket and add a relation to this ticket number.

#5 - 2014-11-21 20:41 - Gabe Blair

We are experiencing the same issue on a relatively high-traffic site using TYPO3 6.1. This is harsh when it happens, because it brings down the entire site with a PHP error, when a PHP cache file is partially or doubly written. This issue has proven quite difficult to reproduce for development/testing purposes, but it definitely is real.

The SimpleFileBackend uses file_put_contents() to create the temp file that eventually becomes (https://github.com/TYPO3/TYPO3.CMS/blob/master/typo3/sysext/core/Classes/Cache/Backend/SimpleFileBackend.php#L238). However, file_put_contents() is not locking, by default (http://us.php.net/file_put_contents). Can anyone think of a downside to using the "LOCK_EX" flag in file_put_contents() to allow PHP to handle the locking?

Obviously there is some activity at the moment around cache locking in general. But how about we at least use native PHP support for locking as a stopgap measure for SimpleFileBackend until a more holistic locking mechanism can be developed overall for caches (http://wiki.typo3.org/Blueprints/LockingForCaches)?

#6 - 2014-11-21 20:52 - Markus Klein

- Status changed from Closed to Needs Feedback

Dear Gabe,

as author of the mentioned blueprint I have to tell you, that I stopped working in this direction for the time being.
The main issue is that there's no guaranteed native way of locking available in PHP that works across all platforms.

Generally I'm not aware of the locking mechanisms employed in version 6.1, but I can tell you that 6.2 does a way better job already.
I can only encourage you to upgrade to 6.2 as soon as possible.

For version 7 my only intention is to replace the current locking API with a more flexible one, which allows registering of other locking technologies, so the use can choose which locking technology fits best on his environment.
Besides that we still face the problem in various caches that we need to know the state of a lock. This information is not available with all locking mechanisms. The ones where this is available are not safe to use in all environments, which could be a cause for your problems, but that is just a guess.

Regards
Markus

#7 - 2015-02-01 00:44 - Benjamin Mack

- Assigned To set to Benjamin Mack

- Target version set to 7.3 (Packages)

#8 - 2015-02-28 18:30 - Benjamin Mack

- Status changed from Needs Feedback to Resolved

Fixed with composer class loader in 6.2 and master

#9 - 2015-02-28 19:00 - Alexander Opitz

The answer is wrong as only core classes are loaded by the new class loader and not the extension ones ... but closed as 3 month no response from author.

#10 - 2015-02-28 19:45 - Markus Klein

- Status changed from Resolved to Closed

- Target version deleted (7.3 (Packages))