

TYPO3 Core - Bug #35684

Cache duration wrongly calculated with associated records

2012-04-04 14:31 - Francois Suter

Status:	Closed	Start date:	2012-04-04
Priority:	Must have	Due date:	
Assignee:	Francois Suter	% Done:	100%
Category:	Caching	Estimated time:	0.00 hour
Target version:	4.6.8	Complexity:	medium
TYPO3 Version:	4.6	Is Regression:	
PHP Version:	5.3	Sprint Focus:	
Tags:			

Description

In TYPO3 4.6, property config.cache was added to enable other records to influence the duration of a page's cache. The calculation happens in tslib_fe::getFirstTimeValueForRecord() but is broken.

The above-mentioned method searches for a minimum starttime and endtime (if both are defined for the given table) among records from a given table in a given pid. One condition exists to select records which have timestamps larger than the current timestamp, but it uses a OR on starttime and endtime. Example query:

```
SELECT MIN(starttime) AS starttime, MIN(endtime) AS endtime FROM tt_news WHERE pid=2070 AND (start  
time>1333539180 OR endtime>1333539180) AND tt_news.deleted=0 AND tt_news.t3ver_state<=0 AND tt_new  
s.pid<>-1 AND tt_news.hidden=0 AND (tt_news.fe_group='' OR tt_news.fe_group IS NULL OR tt_news.fe_  
group='0' OR FIND_IN_SET('0',tt_news.fe_group) OR FIND_IN_SET('-1',tt_news.fe_group))
```

This works fine as long as you have only records with a starttime, or only records with an endtime, or only records with both. But if you have records with a starttime but not endtime (or vice-versa), values of 0 will crop up in the results and the minimum will thus be 0.

A minimum of 0 actually kills this feature, because TYPO3 reverts to the page's default cache duration.

The solution is seek the minimum separately for each time field.

Related issues:

Has duplicate TYPO3 Core - Bug #38838: False PageCacheTimeout / getFirstTimeV...

Closed

2012-07-10

Associated revisions

Revision 22b5f9e - 2012-05-09 10:04 - Francois Suter

[BUGFIX] Page cache expiry calculation fails

The calculation of the page cache expiry timestamp based on property config.cache.xx = table:yy fails in some specific scenarios, name when there are records having either a start time in the future but no end time, or an end time in the future and no start time. Contrary to what is being done so far, calculation must be performed separately for each time field to avoid interferences.

Change-Id: I4e4e259083a10f7125760d14ede11ff6a68022fe

Fixes: #35684

Releases: 6.0, 4.7, 4.6

Reviewed-on: <http://review.typo3.org/10156>

Reviewed-by: Philipp Gampe

Reviewed-by: Wouter Wolters

Reviewed-by: Jigal van Hemert

Tested-by: Jigal van Hemert

Reviewed-by: Xavier Perseguers

Reviewed-by: Francois Suter

Tested-by: Francois Suter

Revision b31e36e4 - 2012-05-09 10:32 - Francois Suter

[BUGFIX] Page cache expiry calculation fails

The calculation of the page cache expiry timestamp based on property `config.cache.xx = table:yy` fails in some specific scenarios, name when there are records having either a start time in the future but no end time, or an end time in the future and no start time. Contrary to what is being done so far, calculation must be performed separately for each time field to avoid interferences.

Change-Id: I580cb6ca02c6efd9b594b31bd887cff9b9b3147c

Fixes: #35684

Releases: 6.0, 4.7, 4.6

Reviewed-on: <http://review.typo3.org/11094>

Reviewed-by: Francois Suter

Tested-by: Francois Suter

Revision 9b6fe53d - 2012-05-09 10:34 - Francois Suter

[BUGFIX] Page cache expiry calculation fails

The calculation of the page cache expiry timestamp based on property `config.cache.xx = table:yy` fails in some specific scenarios, name when there are records having either a start time in the future but no end time, or an end time in the future and no start time. Contrary to what is being done so far, calculation must be performed separately for each time field to avoid interferences.

Change-Id: I39f08a84255233952bd2fd51f28972bb1fe9affa

Fixes: #35684

Releases: 6.0, 4.7, 4.6

Reviewed-on: <http://review.typo3.org/11095>

Reviewed-by: Francois Suter

Tested-by: Francois Suter

Revision 084c4261 - 2012-06-23 23:42 - Francois Suter

[BUGFIX] Page cache expiry calculation fails

The calculation of the page cache expiry timestamp based on property `config.cache.xx = table:yy` fails in some specific scenarios, name when there are records having either a start time in the future but no end time, or an end time in the future and no start time. Contrary to what is being done so far, calculation must be performed separately for each time field to avoid interferences.

Change-Id: I4e4e259083a10f7125760d14ede11ff6a68022fe

Fixes: #35684

Releases: 6.0, 4.7, 4.6

Reviewed-on: <http://review.typo3.org/10156>

Reviewed-by: Philipp Gampe

Reviewed-by: Wouter Wolters

Reviewed-by: Jigal van Hemert

Tested-by: Jigal van Hemert

Reviewed-by: Xavier Perseguers

Reviewed-by: Francois Suter

Tested-by: Francois Suter

History

#1 - 2012-04-04 14:36 - Francois Suter

How to reproduce:

- select any page on your web site and make it depend on some other table, for example `tt_news` by setting the following configuration

```
config.cache.xx = tt_news:yy
```

where "xx" is the uid of the page whose cache duration must be influenced and "yy" is the pid where the `tt_news` items are stored.

- make sure to have at least `tt_news` item that has a starttime but not endtime and another item which has a endtime but no starttime
- clear the cache and view your page (while logged out of the backend)
- check the cache duration in table `cf_cache_pages`. It should be expiring in 24h (if that's your default duration)
- apply the patch, clear the cache again and view the page

- check the cache table again: the duration should correspond to the smallest start or end time in your tt_news set of items

#2 - 2012-04-04 15:08 - Francois Suter

Note: both starttimes and endtimes must be in the future for this bug to happen.

#3 - 2012-04-04 15:23 - Gerrit Code Review

- Status changed from Accepted to Under Review

Patch set 1 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/10156>

#4 - 2012-05-04 13:25 - Francois Suter

Moving Jigal's comment from the Core mailing list and answering:

Let's discuss the approach for finding the cache lifetime here instead of in gerrit reviews.

Sure.

Meaning of values:

- starttime = 0 : no starttime or infinite
- starttime > 0 : timestamp
- endtime = 0 : no endtime or +infinite
- endtime > 0 : timestamp

Right.

For determining the lifetime we have to find CEs with:
endtime in the future OR no endtime
or:
endtime > now OR endtime = 0

```
SELECT * FROM $table WHERE endtime > $now OR endtime = 0  
(plus the enableFields we already have)
```

For all these records we

- ignore the endtime if it is equal to zero (+infinite)
- take the minimum of all the start times which are in the future and all end times

I agree so far.

```
SELECT  
MIN(  
MIN(  
CASE WHEN base.starttime <= $now THEN NULL  
ELSE base.starttime  
END  
)  
FROM (  
SELECT starttime, endtime  
FROM $table  
WHERE endtime > $now OR endtime = 0  
) AS base
```

(both NULLIF and CASE are ANSI SQL)
(for the subquery the enableFields which were already calculated in the current code must be added of course)

I haven't checked the performance of this query, but this can be a start to find the correct values.

Well, this is the sticking point, of course. The query works fine, but I really wonder if it makes us gain anything. I know my patch introduces a second query where there was only one before, but I don't think this can be avoided. Indeed your proposal also uses 2 queries, one being a sub-select. Furthermore your proposal makes use of several functions, whereas the current code (event with my patch) just calls on MIN. Instinctively I would tend to think that your proposal is less efficient.

I tried to test this locally but - with only a few hundred tt_news records at hand - all these queries amount to peanuts in processing time. So we could

say that having a single query is more efficient if it takes about the same time as the current query, which would be doubled. However I wonder how functions like NULLIF and CASE scale with an increasing number of records. And are they supported by DBAL?

#5 - 2012-05-04 22:02 - Jigal van Hemert

Tested with:

```
SELECT SQL_NO_CACHE
  MIN(NULLIF(base.endtime,0)),
  MIN(
    CASE WHEN base.starttime <= UNIX_TIMESTAMP() THEN NULL
    ELSE base.starttime
    END
  )
FROM (
  SELECT starttime, endtime
  FROM tt_content
  WHERE (endtime > UNIX_TIMESTAMP() OR endtime = 0) AND deleted=0 AND hidden=0
) AS base
```

(cache disabled, deleted and hidden added for enableFields)

On four different databases (and different servers) with tt_content tables ranging from 7,000 - 20,000 records the queries took between 0.01 and 0.06 seconds to run.

Reading and testing François' patch shows that this is a correct solution. Taking into consideration that the query only has to look at the records in a single pid I don't think we'll notice the difference between one or two queries. The complexity of the single query and the chance that it might not be DBAL compatible causes me to favour François' solution.

#6 - 2012-05-04 22:10 - Francois Suter

Jigal, thanks a lot for this extensive reviewing.

#7 - 2012-05-09 10:30 - Gerrit Code Review

Patch set 1 for branch **TYPO3_4-7** has been pushed to the review server.
It is available at <http://review.typo3.org/11094>

#8 - 2012-05-09 10:31 - Gerrit Code Review

Patch set 1 for branch **TYPO3_4-6** has been pushed to the review server.
It is available at <http://review.typo3.org/11095>

#9 - 2012-05-09 14:30 - Francois Suter

- Status changed from Under Review to Resolved

- % Done changed from 0 to 100

Applied in changeset [9b6fe53d0564b7ecf645df74f8e2b6cf7c8df28a](https://review.typo3.org/11094).

#10 - 2012-07-31 15:47 - Alexander Opitz

I also found this problem, but as I didn't find this bug (I searched this forge and I also searched via google, but didn't find this bug) I created my own issue entry [#38838](#)

I think the proposed patch in [#38838](#) works better than this one, as it only uses one SQL query.

#11 - 2017-10-24 14:07 - Riccardo De Contardi

- Status changed from Resolved to Closed