

Bugs & Wish list

Tiki extremely ressource intensive | Tiki Wiki CMS Groupware :: Development

Tiki extremely ressource intensive

Status

● Open

Subject

Tiki extremely ressource intensive

Version

18.x

Category

- Release Blocker
- Bug
- Upgrade Blocker

Feature

File Gallery (elFinder or standard UI)

Resolution status

New

Submitted by

hman

Lastmod by

hman

Rating

★ ★ ↘ ↘ ★ ★ ★ ★ ↘ ↘ ★ ★ ★ ★ ↘ ↘ ★ ★ ★ ★ ↘ ↘ ★ ★ ★ ★ (0) ?

Description

I experience that my site's file galleries get slower and slower. Currently, there are just about 200 Mbyte of mostly images in there, in just about two dozen folders. Although my rented web space will of course not boast performance wonders, but it shouldn't be THAT slow. When I clear caches, just calling file galleries takes up to three minutes. If cache has not been cleared, galleries come up instantly.

I have stopped using tiki-check.php even before that, because it regularly breaks my ISPs five minute quota...

I store the images in the DB, but I don't think this really makes the issue. The DB still has almost a terabyte of available storage. And a DB's speed shouldn't depend on amount of storage taken, but on complexity of SQL requests.

I believe that most of the current slowness came when I uploaded some graphics for my co-admins that are not large in size, but large in dimensions (300 kbyte, but 14,000 x 10,000 pixels) (logo master file in PNG). For them, thumbnail creation fails. Could it be that Tiki tries to generate those thumbnails all the time, and fails?

Thanks

hman

Importance

10 high

Easy to solve?

1 difficult

Priority

10

Demonstrate Bug (Tiki 19+)

This bug has been demonstrated on show2.tikiwiki.org

SVN update

Ticket ID

7710

Created

Thursday 25 March, 2021 17:46:33 GMT-0000

by hman

LastModif

Tuesday 13 April, 2021 08:50:43 GMT-0000

Comments



hman 25 Mar 21 18:13 GMT-0000

tiki-check.php could do something even worse. Because of the above mentioned I have renamed it to tiki-check2.php (in order to prevent my co-admins to execute it by accident). I have now called tiki-check2.php, and it resulted in my entire website going offline for a quarter of an hour. My guess is that tiki-check.php brings down PHP... I would strongly advise that whatever tiki-check.php does should be behind a "are you sure" dialog with a clear warning, or better, with a pref of it's own, so I can block others from calling it by accident without renaming it.



Jonny Bradley 26 Mar 21 10:45 GMT-0000

No sure what's up there for you, tiki-check.php on a Tiki 18.x on Debian loads in under 2 seconds for me, hmm, i didn't have `fgal_enable_auto_indexing` enabled, but when i did it landed just as fast.

Maybe your database has a problem generally? I don't use the database for file storage as that used to cause issues in the olden days, so can't really test that for you. Maybe start a new separate bug report on just that?



hman 26 Mar 21 11:54 GMT-0000

Did you do your check on a local installation? I mean, local PHP, local MySQL etc? Or on rented space?

My installation is on rented space on Germany's second largest ISP, 1&1 (now they call themselves Ionos). It is not a dedicated server (too expensive), so I am pretty sure it's a virtual machine, possibly a docker instance. The MySQL DB runs separately, so PHP and MySQL talk to each other over IP.

The self-check of the DB shows no errors. The DB is about 200 MBytes. Everything besides tiki-check and accessing the file galleries after cache clear runs fine.



hman 26 Mar 21 12:00 GMT-0000

fgal_enable_auto_indexing is not enabled. And it seems that hiding the large dimensions files did not speed things up. With cleared cache, simply calling tiki-list_file_gallery.php for the first time took 174.96 seconds, using 35.85MB and 125 queries.



hman 26 Mar 21 12:18
GMT-0000

Yikes. Just ran into Allowed memory size of 134217728 bytes exhausted (tried to allocate 72 bytes) in MYDIRECTORY/tiki-18.8/installer/installlib.php on line 347.

I can increase memory a bit, BUT:
Why on earth is installer/installlib.php called when I want to access the file gallery?? And why does it use more than 128 MByte and does not report that??



hman 26 Mar 21 12:27
GMT-0000

Temporarily renaming it shows that it seems to be needed:

```
Fatal error: require_once():  
Failed opening required  
'installer/installlib.php'  
(include_path='MYHOME/vendor_bundled/vendor/pear/text_wiki_mediawiki:MYHOME/vendor_bundled/vendor/pear/xml_rpc:MYHOME/vendor_bundled/vendor/pear/pear:MYHOME/vendor_bundled/vendor/pear/xml_util:MYHOME/vendor_bundled/vendor/pear/structures_graph:MYHOME/vendor_bundled/vendor/pear/console_getopt:MYHOME/vendor_bundled/vendor/pear/archive_tar:MYHOME/vendor_bundled/vendor/pear/pear_exception:MYHOME/vendor_bundled/vendor/pear/net_socket:MYHOME/vendor_bundled/vendor/pear/xml_parser:./usr/lib/php5.6:MYHOME') in MYHOME/db/tiki-db.php on line 159
```



hman 26 Mar 21 12:57
GMT-0000

Temporarily gave it 256 MBytes. Now it took 149 seconds to show the 11 top level folders, and allegedly only 35 Mbytes got used.



hman 26 Mar 21 18:29
GMT-0000

Wow. After increasing memory to 256 it ran out of memory again. This evidences a memory leak!

"Fatal error: Allowed memory size of 268435456 bytes exhausted (tried to allocate 1024 bytes) in MYHOME/installer/installlib.php on line 347".

Why does it call installlib.php and why does it burn an additional 128 Mbytes??

The current memory limit of 256 MByte is so huge, actually the entirety of the database could be load into RAM, Tiki, wiki pages, images and all, even uncompressed... There is no reasonable explanation what a quarter of a gigabyte could be used for.

Normally my Tiki uses in the vicinity of 30-35 Mbytes. And even when calling file gallery it doesn't indicate higher usage...



hman 26 Mar 21
21:37 GMT-0000

This is what is done in line 347 of installlib.php, this is the function that cause the memory limit to be breached:

```
/**  
    *
```

```

@throws Exception
In case of
filesystem access
issue
    */
    function
buildPatchList()
    {
$patches = [];
foreach (['sql',
'yml', 'php' /*
"php" for
standalone PHP
scripts */] as
$extension) {
$files =
glob(dirname(__FI
LE__) .
'/schema/*_*. ' .
$extension); //
glob() does not
portably support
brace expansion,
hence the loop
if ($files ===
false) {
throw new
Exception('Failed
to scan
patches');
}
foreach ($files
as $file) {
$filename =
basename($file);
$patches[] =
substr($filename,
0, -4);
}
    }
$patches =
array_unique($pat
ches);

$installed = [];
if
($this->tableExis
ts('tiki_schema')
) {
$installed =
$this->table('tik
i_schema')->fetch
Column('patch_nam

```

```

e', []);
    }

if
(empty($installed
)) {
// Erase initial
error
$this->queries['f
ailed'] = [];
    }

Patch::$list =
[];
sort($patches);
foreach ($patches
as $patchName) {
if
(in_array($patchN
ame, $installed))
{
$status =
Patch::ALREADY_AP
PLIED;
} else {
$status =
Patch::NOT_APPLIE
D;
}
$patch = new
Patch($patchName,
$status);
$patch->optional
=
substr($patchName
, 0, 8) ==
'optional'; //
Ignore patches
starting with
"optional". These
patches have
drawbacks and
should be
manually run by
informed
administrators.
Patch::$list[$pat
chName] = $patch;
    }
}

```

The particular code is

```
$patches[] =  
substr($filename,  
0, -4);
```

but the memory leak may be elsewhere, and this is just the last drop that makes the barrel run over.

But a substr() can be memory intensive, depending on the size of \$filename...



hman 27 Mar
21 09:07
GMT-0000

This massive memory usage now explains why Tiki gets slower: I have increased memory limit from 128 to 256 MBytes and changed *nothing* in my Tiki, and memory overflowed. This proves that the *entire* 128 Mbytes get eaten up and later freed (because when my Tiki doesn't run out of memory it reports just 35 MBytes).

The slowness can thus be pinpointed to memory thrashing (i.e. processes force each other out of

1st, 2nd, 3rd level processor and possibly file system cache).

This also proves that the "memory usage" reported by Tiki is bogus. For deeper analysis we need someone who runs his/her Tiki locally. I have rented space, and no console access, so I can use no operating system tools to look into memory consumption differentiated by time.



hman 25 Mar 21 18:45 GMT-0000

In order to find out if thumbnail generation is the culprit, I have zipped those files that were large in dimensions, and the Safari pinned tab image (see the other bug report on the thumbnailer crashing on that little 4 kbyte file).



hman 25 Mar 21 18:50 GMT-0000

For reasons unknown I cannot link this bug report to "Thumbnail creation crashes with Safari pinned tabs image", the search function doesn't find that one. Even when I only search for "Thumbnail". But the other bug report clearly exists (7709).



Jonny Bradley 26 Mar 21 10:39 GMT-0000

Maybe you searched for *Thumbnail*? The title of that one is *Tumbnail* (which it finds fine, but i'm not sure it's related really)



hman 26 Mar 21 11:39 GMT-0000

Thanks. Didn't realize that. I'll change that bugs name...



luciash d' being 26 Mar 21 12:26 GMT-0000

Yes, with lots of files stored in DB it really slows down. You should switch to the filesystem storage for File Galleries if you are using many files or images.



hman 26 Mar 21 12:31 GMT-0000

I wouldn't call a couple of dozen images "lots"... And when file galleries gets called, none of them are on display! Just the top level folders... Just checked. It's 11 top level folders, and the one with the most images does contain more than I thought, but 250 image still isn't what I would call "a lot".



Jonny Bradley 27 Mar 21 11:24 GMT-0000

I see you have now set this to be a release blocker for 18.x (although we have already released various versions of 18, 19, 20, 21 and 22 since 18.0 without issue so it's a bit late to block them)

Please recreate this on show and i will look at it. I'm afraid i think your server is broken in some way as we have no other reports of issues like this.



hman 27 Mar 21 17:37 GMT-0000

How can a server be broken "in some way" as to massively eat up memory? Also, the slowing down of file gallery has been seen by others, look at this discussion. But possibly it has been interpreted wrong.

How much memory is granted to a show instance?



hman 27 Mar 21 17:42 GMT-0000

Needs to be kicked...

Btw. d.t.o also sees big performance issues. In the last days, several times entering anything (comment or attachment alike) resulted in performance warnings and at one point it completely went down for an hour or so.



Jonny Bradley 30 Mar 21 11:23
GMT-0000

Agreed, lack of volunteers means things like this take a long time to fix sadly.



hman 12 Apr 21 18:19 GMT-0000

Set the limit to half a gigabyte, 1.5 times the total size of the database (and that is compared to uncompressed, ASCII backup size, where BLOBs are many times larger than they are in binary. And still the memory limit gets blown:

Fatal error: Allowed memory size of 536870912 bytes exhausted (tried to allocate 72 bytes) in MYHOME/tiki-18.8/installer/installlib.php on line 347. Same offending code as above!

And I still wonder why /installer/installlib.php gets called (this time it wasn't even file galleries I opened, it was view_articles... IMHO some sort of memory leak is obvious. No matter how much RAM you throw at it, it will eat it up. All of it.



hman 13 Apr 21 08:50 GMT-0000

Saw this again. I'll change the topic of this bug report, as the memory issue doesn't seem to be restricted to file galleries. Possibly it doesn't even have anything to do with file galleries at all. IMHO we would need someone with a local LAMP machine willing and able to do debugging and use some memory analysis tools. Since Tiki, when it doesn't shatter the memory limit, reports just modest memory usage of around 30-40 MBytes, these memory masses seemingly get free'ed before PHP reports memory usage...



hman 13 Apr 21 09:04 GMT-0000

A first step might be to either switch the usage of `memory_get_usage()` to `memory_get_peak_usage()` for the memory usage display in module loadstats, or to add a second report for that.



hman 13 Apr 21 10:11 GMT-0000

Hm, I see that in `/lib/smarty_tiki/function.memusage.php` there is already a branch based on PHP version to either use `memory_get_usage()` or `memory_get_peak_usage()`, so theoretically the latter should already be in force. Which makes this even more miraculous. Maybe it's better to have an 'look from outside' at the entire PHP from the OS's view on the process (which would require a real LAMP machine, not rented space as I have).



Jonny Bradley 01 Jun 21 12:56 GMT-0000

Hi hman

Just a thought... recently i fixed (in tiki 22.x in [ef6ca66f](#)) an issue that was making the file galleries on doc.t.o and dev.t.o unusably slow and it turned out to be the authentication tokens had built up to having about 2 million rows (in table `tiki_auth_tokens`) meaning the mechanism to clear out old used ones was failing... and it appears, listing files in galleries or the Since Your Last Visit module checks this table, tries to load all 2 million

rows and fails for each file, taking a few seconds for each one.

So if you can check that table in your database and let me know if that's the problem i'll backport the fix to 21.x and 18.x and meanwhile let you have the temporary workaround to clean it out (you could just truncate that table, once you've backed it up of course ;)

HTH



Jonny Bradley 01 Jun 21 12:57 GMT-0000

p.s. these tokens get generated when page printing or PDF generation is used mostly.



hman 01 Jun 21 16:00 GMT-0000

I'll look into my phpMyAdmin, thanks. Update:
SELECT * FROM `tiki_auth_tokens` yields NULL... I believe I have never used the PDF function. I enabled it, but so far never tested it...



hman 01 Jun 21 20:34 GMT-0000

There must be something else that makes Tiki shatter all memory limits... If one had a Tiki running on a local LAMP machine and not on rented space, we could have a look at memory consumption differentiated over time. Or we could run an extensive SQL log to find out what on earth is getting selected so that PHP burns twice the size of the entire uncompressed DB size in RAM... I am still quite certain that at the bottom of the issue there is a memory leak of one form or the other. It's the only explanation for memory consumption that goes WAY beyond the uncompressed total size of the DB, twofold. Meaning, even loading the entire DB into RAM without compression or deduplication would use less...

Attachments

filename

created

hits

comment

version

filetype

No attachments for this item

The original document is available at <https://dev.tiki.org/item7710-Tiki-extremely-ressource-intensive>

Show PHP error messages