

TYPO3 Core - Epic #95690

Performance issues when hosting a large amount of websites, and optimizations propositions

2021-10-18 14:40 - Olivier Massot

Status:	Accepted	Start date:	2021-10-18
Priority:	Should have	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Sprint Focus:			

Description

Context

My company uses Typo3 to provide a website to each of our clients, meaning that we host around 3500 websites (i.e. around 120.000 pages) on a single Typo3 installation.

We're now using Typo3 10.4. This installation is built with composer, uses php7.4 and mariadb 10.3, and is hosted on a dedicated apache server.

The general behaviour is really satisfying, but hosting that many websites and pages also presents some difficulties. The main challenges we faced were about perfs.

First of all, the website configuration system introduced with Typo3 9 and implying one yaml file per website caused a huge performance loss. The time needed to parse 3500 files is really long, and opening the "Sites" backend module can last something like 30sec.

We also had to rise the php limit about the max number of files that it can maintain open at the same time. The website and page resolution were also problematic. In their primitive form, they triggered one or two db query per site, meaning that each page displayed made around 7000 db queries each time! Not only the loading time was near to 6secs, but our hosting machine had some bad times...

We fixed these issues by:

- rebuilding an inbase website configuration (we've got a 'website' table hosting those informations). The 'pages' table got a new foreign key linking it to this new table.
- overriding the `\TYPO3\CMS\Frontend\Middleware\PageResolver` middleware to resolve the website first with one db query on this 'website' table, then a second query in the 'pages' table to find the suited page. From 7000 queries, we're now to only 2.
- also, xclassing the now named `\TYPO3\CMS\Core\Routing\PageSlugCandidateProvider` class, precisely the `getPagesFromDatabaseForCandidates` method. the way this method is designed makes the while loop to call `getSiteByPageId` once for each page matching the given 'slug'. But with 3500 websites, we've also got 3500 pages with the '/' slug..

The last problems we could'nt solve at this time are related to the backend when accessed by the super-admin. Because this admin can see all of the pages, sites, files, or users, the backend performances are really overstretched.

Reproducing and optimizing

I created a docker container reproducing the problem, and an extension that assemble the patches we made. You can find it there: <https://github.com/Opentalent/Typo3MultisitesOptim>

The docker is based on the martinhelmich/typo3:10.4 container.

An extension (populate) provides CLI command to populate the DB with as many dummy websites as requested:

```
php /var/www/html/typo3/sysext/core/bin/typo3 ot:populate 3000
```

For those who possess a blackfire licence, it also ship a blackfire docker, allowing to profile the typo3 docker, with or without the optimization measures.

Last but not least, I ran two blackfire tests against this container on my machine. These tests were run with 3510 of thoses basic websites (=108810 pages), and all caches flushed before each test :

- with optimizer extension disabled: <https://blackfire.io/profiles/1ccb80ed-08ae-40e3-8106-855039581ee8/graph>
- with optimizer extension enabled: <https://blackfire.io/profiles/23e7967a-13b4-4db4-baad-a4b51d49112c/graph>

Related issues:

Related to TYPO3 Core - Epic #93547: Collection of problems with large sites

Accepted

2021-02-19

History**#1 - 2021-10-18 19:56 - Sybille Peters**

- *Related to Epic #93547: Collection of problems with large sites added*

#2 - 2021-10-19 06:22 - Georg Ringer

- *Status changed from New to Accepted*