

TYPO3.Flow - Task #3582

Concurrency stress test

2009-06-05 10:42 - Robert Lemke

Status: Closed	Start date: 2009-06-05
Priority: Should have	Due date:
Assignee: Robert Lemke	% Done: 100%
Category: - Testing -	Estimated time: 0.00 hour
Target version: 1.0 alpha 2	Has patch:
Sprint:	Complexity:
PHP Version:	

Description

Currently we have only few experience with multiple concurrent users accessing FLOW3 at the same time. This ticket is about testing the behavior of FLOW3 when running simultaneous requests.

What we especially need to find out is

- how FLOW3 reacts if caches are empty
- how FLOW3 reacts in production context if caches are filled (theoretically shouldn't be a problem at all)
- if accessing FLOW3 in development context has any side effects on the Production context (shouldn't ...)
- how FLOW3 scales performance wise

Try to detect any race conditions so we can make them atomic.

Related issues:

Related to TYPO3.Flow - Feature #3581: Site lock / code cache flushing mechanism	Resolved	2009-06-05
Related to TYPO3.Flow - Task #3755: Concurrency stress testing and cache mech...	New	2009-06-25

History

#1 - 2009-06-25 09:33 - Robert Lemke

- Category set to - Testing -
- Status changed from New to Closed
- Assignee set to Robert Lemke
- % Done changed from 0 to 100

Did some first stress testing with concurrent requests with siege and the front page of the Blog Example.

If caches are loaded FLOW3 performs almost okay (for a first alpha) in production context:

- Concurrent users: 15
- Transaction rate: 2.53 trans/sec
- Response time: 5.16 sec

While caches were loaded in development context, basically the same:

- Concurrent users: 15
- Transaction rate: 2.64 trans/sec
- Response time: 5.14 sec

- Concurrent users: 50
- Transaction rate: 2.70 trans/sec
- Response time: 16.24 sec

However, when the caches are flushed while the site is under siege, the locking mechanism isn't doing its job 100% well. The site is locked and most requests end up in the lock message but apparently the caches are rebuilt multiple times and sometimes more than should be rebuilt which suggest some kind of race condition.

Additionally I experienced a few race conditions in the Cache File Backend which tried to either read cache entries which were already deleted or delete entries which are gone already.

We'll need to re-open [#3581](#) for alpha 3 and do some more thorough stress testing and analysis of the caching mechanisms.