TYPO3.Flow - Bug # 37363

Status:	Resolved	**Priority:**	Must have
Author:	Sebastian Wojtowicz	**Category:**	Validation
Created:	2012-05-21	**Assigned To:**	Karsten Dambekalns
Updated:	2012-06-20	**Due date:**	
PHP Version:	5.3		
Has patch:	No		
Complexity:			
Affected Flow version:	Git master		
Subject:	PersistenceManager->validateObject() ==> to many calls/loops/recursions somewhere in a subcall		

Description

In a nutshell

In my project I add objects with lots of sub sub objects. They are added in bigger numbers, like often 50, sometimes 100 even up to 1000 elements at once (it's vocabularies).
The problem is, right now it **takes already 25 seconds** on my local machine **to add only 10 elements** (with all their sub sub sub... objects of course, but still 25 seconds is way to much).
Currently I need to add 5000 vocabularies. This would take about 33 hours (if it doesn't fail at some point because of missing memory).

Testcase

Having no idea what causes it I created a small test-case at https://github.com/Allisone/Wojtowicz.AddALot

There you have a model hierarchy like this:
Elements A,B,C and D, so that As have Bs, Bs have Cs and Cs have Ds.

Via the commandline you can then call one of the following commands.
1. Create x Bs, each having a C, each C having a D
2. Create y As, each having x/y Bs, each having a C, each C having a D

I can see in the xdebug profiler output...
For 1. and x=20: 6480 calls are being made to TYPO3\FLOW3\Error\Result->mergeProperty()
For 2. and x=20 and y=4: 79656 calls are being made to TYPO3\FLOW3\Error\Result->mergeProperty()
(we get 20 Bs, each 5 Bs being added to an A)

Doing a small adjustment so that each B has 2 Cs (instead of one), I can see already 231576 mergeProperty() calls in the 2. command call.

So... I can't say exactly where the bug is, just give you the upper description, hoping that someone finds that bug, either between \TYPO3\FLOW3\Persistence\Doctrine\PersistenceManager\validateObject() and TYPO3\FLOW3\Error\Result->mergeProperty() or at some place where f.e. to many validators where added (or similar). But I think it should be related to validation...

More

For more details look at my provided testcase (+README):
https://github.com/Allisone/Wojtowicz.AddALot

Related issues:

related to TYPO3.Flow - Bug # 34154: ObjectValidator has exponential growth f...	**Resolved**	**2012-02-21**

Associated revisions

Revision 26e4b3fa - 2012-05-30 14:41 - Karsten Dambekalns

[BUGFIX] Avoid validating instances more often than needed

The GenericObjectValidator uses a static variable to store objects being
validated. Objects were still validated over and over because at the
end of isValid() they were detached again.

This change replaces that detach() call with another concept. Now
any validator implementing the new ObjectValidatorInterface can be
given an SplObjectStorage to keep track of instances already
validated.

If you reset that container when it makes sense (before you
actively validate objects) you can avoid validating single instances
more than once in one validation run but still validate at various
points on your logic flow.

If you have implemented your own object validator you should
implement the new ObjectValidatorInterface - the easiest way is
to extend the GenericObjectValidator, if you haven't done that
so far.

Change-Id: I7f437207e27593425b1fbdfc340a442e685a317f
Fixes: #37363
Related: #34154
Releases: 1.1, 1.2

Revision 632685da - 2012-05-30 15:06 - Karsten Dambekalns

[BUGFIX] Avoid validating instances more often than needed

The GenericObjectValidator uses a static variable to store objects being
validated. Objects were still validated over and over because at the
end of isValid() they were detached again.

This change replaces that detach() call with another concept. Now
any validator implementing the new ObjectValidatorInterface can be
given an SplObjectStorage to keep track of instances already
validated.

If you reset that container when it makes sense (before you
actively validate objects) you can avoid validating single instances
more than once in one validation run but still validate at various
points on your logic flow.

If you have implemented your own object validator you should
implement the new ObjectValidatorInterface - the easiest way is
to extend the GenericObjectValidator, if you haven't done that
so far.

Change-Id: I7f437207e27593425b1fbdfc340a442e685a317f
Fixes: #37363
Related: #34154
Releases: 1.1, 1.2

Revision fdc462b0 - 2012-05-30 15:45 - Karsten Dambekalns

[TASK] Make extending GenericObjectValidator easier

This change moves the check for already validated objects out
of the isValid() method and into validate() instead. Since in
custom validators isValid() is the method to implement, this
makes correct implementation less error-prone.

Change-Id: I16981ff9d2177cb4949d0fc3fcbd815f7c99cbc5
Related: #37363
Releases: 1.1, 1.2

Revision ed3788b3 - 2012-06-18 17:49 - Karsten Dambekalns

[TASK] Make extending GenericObjectValidator easier

This change moves the check for already validated objects out
of the isValid() method and into validate() instead. Since in
custom validators isValid() is the method to implement, this
makes correct implementation less error-prone.

Change-Id: I16981ff9d2177cb4949d0fc3fcbd815f7c99cbc5
Related: #37363
Releases: 1.1, 1.2

History

#1 - 2012-05-21 17:00 - Karsten Dambekalns

- Target version changed from 1.1 beta 2 to 1.1

#2 - 2012-05-22 00:52 - Sebastian Wojtowicz

I really have problems going through all this recursive code, so I made some effort to visualize what is going on...

https://gist.github.com/942e34ef00da89b1f459

You can see the call tree for the first command (20 Bs with 1 Cs with 1 Ds) and for the second command (4 As with 5 Bs with 1 Cs with 1 Ds).

I've ignored calls where no further recursion occurred.

I'm still not able to tell where the problem is, but at least this will keep you from loosing your head while fighting these reoccurring calls.

#3 - 2012-05-22 09:26 - Karsten Dambekalns

- Status changed from New to Needs Feedback

- Assigned To set to Karsten Dambekalns

Just to make sure: have you tried the changes listed at https://review.typo3.org/#/q/topic:34145-validation-optimisation,n,z - and if not, could you do that?

#4 - 2012-05-22 12:43 - Sebastian Wojtowicz

just found that place in the code myself and wanted to report it here, to hear what you think about the fact that this seems to be called over and over again and that it should not be instancesCurrentlyUnderValidation but some kind of wasValidated...

and now that I see the patch, I realize that I tried it already (ChristianM pointed me to it as well)... but it didn't work before, made no difference. And I tried it twice, even though I was convinced it could not effect my case because I have no validators at all.

Well... obviously I messed up, don't know how, cause this patch is working. It's the right fix. Also checked the profiler output numbers. All good now.
On one hand I could bite my ass that somehow I messed up the first 2 times I tried that patch (how ?). On the other hand, I wouldn't have known now, how to use xdebug, debug with PhpStorm, what is and how to use kCachegrind, and how the validation is done :D
Just hope I didn't wasted your time :(and that I'm better equipped for the next bug I find ;)

#5 - 2012-05-22 14:03 - Karsten Dambekalns

- Status changed from Needs Feedback to Accepted

Thanks for checking again!

#6 - 2012-05-22 14:36 - Gerrit Code Review

- Status changed from Accepted to Under Review

Patch set 9 for branch **master** has been pushed to the review server.
It is available at http://review.typo3.org/9197

#7 - 2012-05-25 09:40 - Karsten Dambekalns

- Target version changed from 1.1 to 1.1 RC1

#8 - 2012-05-30 14:15 - Gerrit Code Review

Patch set 10 for branch **master** has been pushed to the review server.
It is available at http://review.typo3.org/9197

#9 - 2012-05-30 14:43 - Gerrit Code Review

Patch set 11 for branch **master** has been pushed to the review server.
It is available at http://review.typo3.org/9197

#10 - 2012-05-30 15:07 - Gerrit Code Review

Patch set 1 for branch **FLOW3-1.1** has been pushed to the review server.
It is available at http://review.typo3.org/11737

#11 - 2012-05-30 15:38 - Karsten Dambekalns

- Status changed from Under Review to Resolved

- % Done changed from 0 to 100

Applied in changeset commit:26e4b3fa846c50fdbdc7908c11eda0756ed85f2c.

#12 - 2012-06-20 20:53 - Karsten Dambekalns

- Target version changed from 1.1 RC1 to 1.1 beta 3