EDIT - bug #7709

CdmTransientEntityCacher cannot handle multiple unpersisted entities of the same type

08/30/2018 12:15 PM - Andreas Kohlbecker

Status:	Feedback	**Start date:**	
Priority:	Highest	**Due date:**	
Assignee:	Katja Luther	**% Done:**	50%
Category:	cdmlib	**Estimated time:**	0:00 hour
Target version:	Release 5.16		
Severity:	normal	**Found in Version:**	

Description

the following code would fail:

```
newP1 = Person.NewInstance();
newP2 = Person.NewInstance();

pf1 = new PersonField(newP1);
cache.load(pf1);
pf2 = new PersonField(newP2);
cache.load(pf2);

assert pf1.getValue() != pf2.getValue()
```

The cacher replaces pf1 with pf2!!!

This is the cause for the problem described in issue #7641

Related issues:

Related to EDIT - bug #7641: Reference editor: Error when incerting several n...	**Closed**
Related to EDIT - task #7785: CdmUserHelper caches permission information	**Closed**
Related to EDIT - feature request #9078: Handle name parsing and deduplicatio...	**Closed**

Associated revisions

Revision c5eb9820 - 08/30/2018 01:37 PM - Andreas Müller

ref #7709 use more generics for CdmCacher (not completed yet)

Revision 67ff6af5 - 08/30/2018 01:47 PM - Andreas Müller

fix #7709 use uuid instead of id for CdmEntityCacheKey

Revision b1b42293 - 08/30/2018 02:18 PM - Andreas Müller

ref #7709 javadoc

Revision 7b2e142b - 08/30/2018 10:38 PM - Andreas Müller

ref #7709 use CdmBase constructor instead of class and id

Revision 3c5c2395 - 08/30/2018 10:39 PM - Andreas Müller

ref #7709 revert using uuid in CdmEntityCacheKey as

Revision 57e761cd - 08/30/2018 10:45 PM - Andreas Müller

ref #7709 revert using uuid in CdmEntityCacheKey as key part

Revision 7ad7b0fa - 08/30/2018 11:03 PM - Andreas Müller

ref #7709 cleanup CdmTransientEntityCacher

Revision 7b74ba3d - 08/30/2018 11:12 PM - Andreas Müller

ref #7709 use id instead of uuid for CdmEntityCacheKey in ConversationalTransientEntityCacher

Revision df75b75c - 08/30/2018 11:22 PM - Andreas Müller

ref #7709 remove NPE

Revision 9672401c - 10/12/2018 11:16 AM - Andreas Kohlbecker

ref #7709 JUnit testsuite to reproduce the test failure when running RegistrationIdentifierMinterTest and TaxonGraphHibernateListenerTest sequentially

Revision 8a790929 - 10/12/2018 01:23 PM - Andreas Kohlbecker

ref #7709 adding REGISTRATION to ClearDBDataSet.xml to solve test problems

Revision d7be01af - 06/26/2020 11:09 PM - Andreas Müller

ref #7709 rename putMethods in cacher

Revision 8991d6e9 - 06/26/2020 11:10 PM - Andreas Müller

ref #7709 rename addNewEntity to putNewEntity and check for existing entity

Revision ac3d6e3a - 06/26/2020 11:12 PM - Andreas Müller

ref #7709 use load() for newEntities

Revision b486d565 - 06/26/2020 11:21 PM - Andreas Müller

ref #7709 rename newEntity to volatileEntity

History

#1 - 08/30/2018 12:16 PM - Andreas Kohlbecker

- Related to bug #7641: Reference editor: Error when incerting several new authors into a new author team added

#2 - 08/30/2018 01:33 PM - Andreas Müller

- Status changed from New to In Progress

- Priority changed from New to Highest

- Target version changed from Unassigned CDM tickets to Release 5.3

Is this on purpose assigned to me?

If yes, please give more info what is meant by PersonField. PersonField is not a CdmBase subclass but cacher.load(T) is only defined for T instanceOf CdmBase, therefore I ask.

However, I think the critical issue is that eu.etaxonomy.cdm.cache.CdmEntityCacheKey uses (int)id as identifier, not uuid. While uuid is unchangeable and unique no matter if a CdmEntity is persisted or not the id is neither unchangable (changes when an object is persisted) nor unique (multiple volatile objects with id=0 may exist for the same class). Only when merging data from different sources the uuid may not be unique but this is not necessarily something to be handled by this chacher.

Using the id not the uuid might be the reason for certain Multiple Representation Exceptions in the past.

Disadvantage of using UUID is that it requires more memory but this is probably acceptable as the caches should not become too big usually.

#3 - 08/30/2018 01:39 PM - Andreas Müller

With commit c5eb9820 I introduced more generics to the cacher (not yet 100% finished). Maybe we need to adapt external apps (TaxEditor, Vaadin) afterwards.

#4 - 08/30/2018 02:06 PM - Andreas Müller

- Status changed from In Progress to Resolved

- % Done changed from 0 to 50

Applied in changeset cdmlib|67ff6af5f4a24c899385e13c671c8071d301f3e7.

#5 - 08/30/2018 02:10 PM - Andreas Müller

- Status changed from Resolved to Feedback

- Assignee changed from Andreas Müller to Andreas Kohlbecker

- % Done changed from 50 to 0

Can you please check if this implementation solves your problems and if you generally agree with the implementation?

Before closing this ticket should be reviewed by all developers I guess as it is very core functionality.

#6 - 08/30/2018 02:12 PM - Andreas Kohlbecker

Andreas Müller wrote:

> Is this on purpose assigned to me?

You are the default assignee for cdmlib, as this ticket was in "Unassigned CDM" is was not by purpose but automatically. You may want to reconsider being the default assignee.

> Using the id not the uuid might be the reason for certain Multiple Representation Exceptions in the past.

There were some other problems in cacher which where definitely causing these issues. Multiple Representation Exceptions will not occur with un-persisted entities. However, there could be situations in which the id has changed which lead to Multiple Representation Exceptions.

Your suggestion to use the uuid instead of the id as key seems like an excellent idea but might need more investigantion and testing

> Using the id not the uuid might be the reason for certain Multiple Representation Exceptions in the past.
>
> Disadvantage of using UUID is that it requires more memory but this is probably acceptable as the caches should not become too big usually.

#7 - 08/30/2018 02:17 PM - Andreas Kohlbecker

Andreas Müller wrote:

> Can you please check if this implementation solves your problems

My problem has been solved in cdm-vaadin using the cacher for un-persited entities makes not much sense as it can cause problems.
I not 100% sure but I think that the cacher should ignore un-persisted entities.

I am very busy with other tasks and will postpone this discussion for now. But I will come back to this for sure since I am the assignee of this issue now.

#8 - 08/30/2018 02:20 PM - Andreas Kohlbecker

I forgot one thing:

I am a bit frightend about the change you made in the cacher as this is touching an essential core functionality and we need to have a new release next week. It is not very likely but the switch from id to uuid can break existing functionality.

#9 - 08/30/2018 03:11 PM - Andreas Müller

I discussed with Katja the handling of un-persisted objects. She also thinks that unpersisted objects should/are not handled by the cacher. The problem is that the expected behavior and usage of the cacher is not well documented.

The main question is if unintended wrong usage of the cacher may lead to difficulties to find errors. This could be prevented by throwing an exception whenever someone tries to add an unpersisted object to the cacher. This way it is much easier to find wrong implementations as the error is thrown immediately.

We will need to investigate more on how volatile client side objects that get saved are afterwards replaced by their persistent counterpart. Somehow this happens by Cache.load(recursive=true). But in case this works not 100% save the above described exceptions may appear.

#10 - 08/30/2018 03:13 PM - Andreas Müller

Andreas Kohlbecker wrote:

> I forgot one thing:
>
> I am a bit frightend about the change you made in the cacher as this is touching an essential core functionality and we need to have a new release next week. It is not very likely but the switch from id to uuid can break existing functionality.

I agree that we should revert the commit (easy to revert) in case we can't test sufficiently. However, for now I would like to keep it for a moment as I do want to do some tests with the new implementations.
Also I can't really imagine that it creates more problems than the old implementation (but of course you never know).

#11 - 08/31/2018 12:33 AM - Andreas Müller

- Status changed from Feedback to In Progress

- Assignee changed from Andreas Kohlbecker to Andreas Müller

I reverted using UUID as key in CdmEntityCacheKey because

- the eu.etaxonomy.cdm.cache.CdmTransientEntityCacher should only be used for transient (not volatile) entities, these should always have id>0
- the eu.etaxonomy.taxeditor.remoting.cache.ConversationalTransientEntityCacher transforms CdmEntityIdentifier, which are returned by UpdateResults into CdmEntityCacheKeys, this is not possible if CdmEntityCacheKey uses uuid instead of id (Note: CdmEntityIdentifier and CdmEntityCacheKey seem to do exactly the same and should be unified)
- even if using UUID were the better choice it still needs intensive testing

#12 - 08/31/2018 12:34 AM - Andreas Müller

- Subject changed from CdmTransientEntityCacher cannot handle multiple unpersieted entities of the same type to CdmTransientEntityCacher cannot handle multiple unpersisted entities of the same type

#13 - 09/17/2018 01:30 PM - Andreas Müller

- Target version changed from Release 5.3 to Release 5.4

- Severity changed from critical to normal

We should think about throwing an exception in the moment when the cacher is used incorrectly (e.g. for volatile objects, but not using the addNewEntity method). Therefore move it to next version and don't close yet.

#14 - 10/09/2018 05:02 PM - Andreas Kohlbecker

- Blocks task #7785: CdmUserHelper caches permission information added

#15 - 10/14/2018 08:09 PM - Andreas Kohlbecker

Andreas Müller wrote:

> - the eu.etaxonomy.taxeditor.remoting.cache.ConversationalTransientEntityCacher transforms CdmEntityIdentifier, which are returned by UpdateResults into CdmEntityCacheKeys, this is not possible if CdmEntityCacheKey uses uuid instead of id (Note: CdmEntityIdentifier and CdmEntityCacheKey seem to do exactly the same and should be unified)

Note that the CdmTransientEntityCacher currently handles volatile entities, see :

```
    //map for volatile entities (id=0)
    private final Map<UUID, CdmBase> newEntitiesMap = new HashMap<>();
```

we would need to remove this map if we want to avoid the CdmTransientEntityCacher handling volatile entities. Before that we need to examine why Cherian added this and if it is needed for some reason!!!

#16 - 10/23/2018 12:47 PM - Andreas Müller

- Target version changed from Release 5.4 to Release 5.5

#17 - 11/14/2018 02:51 PM - Andreas Kohlbecker

- Blocks deleted (task #7785: CdmUserHelper caches permission information)

#18 - 11/14/2018 02:51 PM - Andreas Kohlbecker

- Related to task #7785: CdmUserHelper caches permission information added

#19 - 02/14/2019 10:38 PM - Andreas Müller

- Target version changed from Release 5.5 to Release 5.6

#20 - 02/19/2019 05:04 PM - Andreas Müller

- Target version changed from Release 5.6 to Reviewed Next Major Release

#21 - 06/26/2020 05:44 PM - Andreas Müller

- Target version changed from Reviewed Next Major Release to Release 5.18

- % Done changed from 0 to 20

To me it is not clear anymore, why the CdmTransientEntityCacher should handle only transient but not volatile objects.

The main purpose of the cache should be to avoid duplicate entries in the sessions object graph for equal entities. To achieve this object graphs coming from outside (e.g. from the server) are to be merged into the sessions object graph in a way that the state of these outside objects is merged onto the objects in the session (if update is selected).
Now, in the outside objects there can be objects which are equal to objects in the session. This is the case if these objects have been sent outside before and are returning from there as volatile (or as persistent) objects but not as identical objects. This is e.g. the case if you send an object to the server for doing there some computation like duplicate resolving of attached objects and then you get it back from the server. The returning object is equal but not identical. Especially if the returning graph contains some related non-volatile objects.

So I think we need to find a way to handle transient and volatile objects in #7709#note-9 after discussion with KL.

#22 - 06/26/2020 06:48 PM - Andreas Müller

By the way, there is already a UUID based cacher in cdmlib-services (eu.etaxonomy.cdm.api.cache.CdmCacher)

#23 - 06/30/2020 03:11 PM - Andreas Müller

- Status changed from In Progress to Resolved

- Assignee changed from Andreas Müller to Katja Luther

- % Done changed from 20 to 50

This should be fixed now. The transient entity cacher handles now both transient and volatile objects via the same UI method load(). Volatile objects are handled via UUID while transient objects still use id as identifier. If a volatile object becomes transient in the meanwhile (don't know if there are use-cases for this) it is moved to the transient cache while loading.
Also the handling of terms which should be managed in the permanent cache (term cache) is improved in a way that they are always put in this cache and never should be loaded to the transient cache.
Also loading is interrupted when an object that should be handled in the permanent cache is loaded during recursion. This is to avoid loading large graphs of terms (e.g. rank with rank vocabulary with language with language vocabulary...). Terms do not need to be deduplicated necessarily as they are not cascade (but we need to check if this creates problems in places where we expect terms to be updated, if such places exist).

#24 - 06/30/2020 03:14 PM - Andreas Müller

Please decide who wants to review this ticket.

By the way: I adapted the session inspection dialog in TaxEditor in such a way that terms (objects from permanent cache) are not showing up there anymore if the session is about ordinary cdm entities. That makes it muche easier to use the view (only the relevant information is shown now - helped me a lot fixing #9078).

#25 - 06/30/2020 03:15 PM - Andreas Müller

- Related to feature request #9078: Handle name parsing and deduplication on server side added

#26 - 06/30/2020 03:26 PM - Andreas Müller

- Target version changed from Release 5.18 to Release 5.16

#27 - 06/30/2020 04:26 PM - Andreas Kohlbecker

Andreas Müller wrote:

> Please decide who wants to review this ticket.
>
> By the way: I adapted the session inspection dialog in TaxEditor in such a way that terms (objects from permanent cache) are not showing up there anymore if the session is about ordinary cdm entities. That makes it muche easier to use the view (only the relevant information is shown now - helped me a lot fixing #9078).

excellent! The same should be applied to the EntityCacheDebuggerComponent in cdm-vaadin.
Can you point me to the according changeset?

#28 - 06/30/2020 04:53 PM - Andreas Müller

Andreas Kohlbecker wrote:

> Andreas Müller wrote:
>
> > Please decide who wants to review this ticket.
> >
> > By the way: I adapted the session inspection dialog in TaxEditor in such a way that terms (objects from permanent cache) are not showing up there anymore if the session is about ordinary cdm entities. That makes it muche easier to use the view (only the relevant information is shown now - helped me a lot fixing #9078).
>
> excellent! The same should be applied to the EntityCacheDebuggerComponent in cdm-vaadin.
> Can you point me to the according changeset?

This is already the case. See line 52 in the class. You may implement a checkbox to switch this on and off via UI if you want.

#29 - 07/01/2020 09:31 AM - Andreas Kohlbecker

Andreas Müller wrote:

> Andreas Kohlbecker wrote:
>
> > Andreas Müller wrote:
> >
> > > Please decide who wants to review this ticket.
> > >
> > > By the way: I adapted the session inspection dialog in TaxEditor in such a way that terms (objects from permanent cache) are not showing up there anymore if the session is about ordinary cdm entities. That makes it muche easier to use the view (only the relevant information is shown now - helped me a lot fixing #9078).
> >
> > excellent! The same should be applied to the EntityCacheDebuggerComponent in cdm-vaadin.
> > Can you point me to the according changeset?
>
> This is already the case. See line 52 in the class. You may implement a checkbox to switch this on and off via UI if you want.

Excellent! I left an according comment in the code.

#30 - 07/01/2020 11:57 AM - Katja Luther

For terms we have to have a deeper look because inspecting the volatileCache shows a lot of terms of type UNKNOWN.

Another thing I recognized, when entering a nomenclatural reference in the freetext editor for every keystroke a new volatile person is created although the author is already correctly parsed and persisted.

After adding a new Synonym to a taxon there were 3767 entities in the volatileEntitiesMap.

#31 - 07/01/2020 11:57 AM - Katja Luther

- Status changed from Resolved to Feedback

- Assignee changed from Katja Luther to Andreas Müller

#32 - 07/01/2020 12:15 PM - Andreas Müller

Katja Luther wrote:

> Another thing I recognized, when entering a nomenclatural reference in the freetext editor for every keystroke a new volatile person is created although the author is already correctly parsed and persisted.
>
> After adding a new Synonym to a taxon there were 3767 entities in the volatileEntitiesMap.

That's interesting. These new persons (and probably also names) are probably generated during TaxEditor side parsing. The problem is, that during parsing new volatile objects are generated. By default, ALL newly created CdmBase objects are automatically put to the current session as new objects. This has not really changed and probably happened also before. Even if 3767 new entities is really a lot.
Generally these objects do not really matter as long as they are not connected with the root graph. However, it might be interesting to find a way to not store undwanted objects in the cache. Currently a listener is triggered by a constructor call. Maybe it is possible to explicitly tell the following pipeline somehow that certain objects do not need to be added to the cache (or should be removed from e.g. by analyzing the parsed name before server side deduplication and remove all the objects found in the name from the cache again). But as I said this is not really critical.

#33 - 07/01/2020 12:16 PM - Andreas Müller

- Assignee changed from Andreas Müller to Katja Luther

Katja Luther wrote:

> For terms we have to have a deeper look because inspecting the volatileCache shows a lot of terms of type UNKNOWN.

When exactly did this happen. When you edited terms via the term editor or what else did you do TaxEditor side?

#34 - 07/01/2020 01:21 PM - Katja Luther

Andreas Müller wrote:

Katja Luther wrote:

> For terms we have to have a deeper look because inspecting the volatileCache shows a lot of terms of type UNKNOWN.

When exactly did this happen. When you edited terms via the term editor or what else did you do TaxEditor side?

No, I did nothing with terms. I started the editor and added a new synonym. I can do some more debugging to find out why these empty terms are loaded in the volatileEntityMap.