# EDIT - bug #8117

## CdmFilterablePagingProviderFactory.taxonNamesWithoutOrthographicIncorrect() causes performance overhead

02/21/2019 11:32 AM - Andreas Kohlbecker

| | | | | |
|---|---|---|---|---|
| **Status:** | New | | **Start date:** | |
| **Priority:** | Highest | | **Due date:** | |
| **Assignee:** | Andreas Kohlbecker | | **% Done:** | 0% |
| **Category:** | cdm-vaadin | | **Estimated time:** | 0:00 hour |
| **Target version:** | Reviewed Next Major Release | | | |
| **Severity:** | normal | | **Found in Version:** | Release 5.5 |

**Description**

in

CdmFilterablePagingProviderFactory.taxonNamesWithoutOrthophicIncorrect()

a restriction is being used which causes the pager requests to be slow:

```
pagingProvider.addRestriction(new Restriction<>("relationsFromThisName.type", Operator.AND_NOT,
null, NameRelationshipType.ORTHOGRAPHIC_VARIANT()));
```

With this restriction the pager takes ~5 on average per call, without this restriction only ~240 ms

## Associated revisions

**Revision 8d8da57a - 02/21/2019 12:40 PM - Andreas Kohlbecker**

ref #8117 commenting orthographic corrected names filter to prevent from unclear performance probelms, see ref #7899

**Revision b599ad36 - 02/21/2019 04:46 PM - Andreas Kohlbecker**

ref #8117 commenting orthographic corrected names filter to prevent from unclear performance probelms, see ref #7899

## History

**#1 - 02/21/2019 11:32 AM - Andreas Kohlbecker**

*- Found in Version set to Release 5.5*

**#2 - 02/21/2019 11:37 AM - Andreas Kohlbecker**

*- Description updated*

**#3 - 02/21/2019 12:24 PM - Andreas Kohlbecker**

*- File picture278-1.png added*

The query created by the restriction performs not so bad (55ms):

```
select
    this_.id as id1_499_2_, this_.created as created2_499_2_, this_.createdBy_id as created40_499_2_, this_.
uuid as uuid3_499_2_, this_.updated as updated4_499_2_, this_.updatedBy_id as updated41_499_2_, this_.
lsid_authority as lsid_aut5_499_2_, this_.lsid_lsid as lsid_lsi6_499_2_, this_.lsid_namespace as
lsid_nam7_499_2_, this_.lsid_object as lsid_obj8_499_2_, this_.lsid_revision as lsid_rev9_499_2_, this_.
protectedTitleCache as protect10_499_2_, this_.titleCache as titleCa11_499_2_, this_.acronym as
acronym12_499_2_, this_.anamorphic as anamorp13_499_2_, this_.appendedPhrase as appende14_499_2_, this_.
authorshipCache as authors15_499_2_, this_.basionymAuthorship_id as basiony42_499_2_, this_.binomHybrid as
binomHy16_499_2_, this_.breed as breed17_499_2_, this_.combinationAuthorship_id as combina43_499_2_, this_.
cultivarName as cultiva18_499_2_, this_.exBasionymAuthorship_id as exBasio44_499_2_, this_.
exCombinationAuthorship_id as exCombi45_499_2_, this_.fullTitleCache as fullTit19_499_2_, this_.
genusOrUninomial as genusOr20_499_2_, this_.homotypicalGroup_id as homotyp46_499_2_, this_.hybridFormula as
hybridF21_499_2_, this_.inBasionymAuthorship_id as inBasio47_499_2_, this_.inCombinationAuthorship_id as
inCombi48_499_2_, this_.infraGenericEpithet as infraGe22_499_2_, this_.infraSpecificEpithet as
infraSp23_499_2_, this_.monomHybrid as monomHy24_499_2_, this_.nameApprobation as nameApp25_499_2_, this_.
nameCache as nameCac26_499_2_, this_.nameType as nameTyp27_499_2_, this_.nomenclaturalMicroReference as
nomencl28_499_2_, this_.nomenclaturalReference_id as nomencl49_499_2_, this_.nomenclaturalSource_id as
nomencl50_499_2_, this_.originalPublicationYear as origina29_499_2_, this_.parsingProblem as parsing30_499_2_,
```

```
   this_.problemEnds as problem31_499_2_, this_.problemStarts as problem32_499_2_, this_.
protectedAuthorshipCache as protect33_499_2_, this_.protectedFullTitleCache as protect34_499_2_, this_.
protectedNameCache as protect35_499_2_, this_.publicationYear as publica36_499_2_, this_.rank_id as
rank_id51_499_2_, this_.specificEpithet as specifi37_499_2_, this_.subGenusAuthorship as subGenu38_499_2_,
this_.trinomHybrid as trinomH39_499_2_, rank2_.id as id2_99_0_, rank2_.created as created3_99_0_, rank2_.
createdBy_id as created41_99_0_, rank2_.uuid as uuid4_99_0_, rank2_.updated as updated5_99_0_, rank2_.
updatedBy_id as updated42_99_0_, rank2_.lsid_authority as lsid_aut6_99_0_, rank2_.lsid_lsid as lsid_lsi7_99_0_
, rank2_.lsid_namespace as lsid_nam8_99_0_, rank2_.lsid_object as lsid_obj9_99_0_, rank2_.lsid_revision as
lsid_re10_99_0_, rank2_.protectedTitleCache as protect11_99_0_, rank2_.titleCache as titleCa12_99_0_, rank2_.
termType as termTyp13_99_0_, rank2_.uri as uri14_99_0_, rank2_.idInVocabulary as idInVoc15_99_0_, rank2_.
kindOf_id as kindOf_43_99_0_, rank2_.partOf_id as partOf_44_99_0_, rank2_.symbol as symbol16_99_0_, rank2_.
symbol2 as symbol17_99_0_, rank2_.vocabulary_id as vocabul45_99_0_, rank2_.orderIndex as orderIn20_99_0_,
rank2_.rankClass as rankCla40_99_0_, representa3_.DefinedTermBase_id as DefinedT1_123_4_, representa4_.id as
represen2_123_4_, representa4_.id as id1_423_1_, representa4_.created as created2_423_1_, representa4_.
createdBy_id as createdB8_423_1_, representa4_.uuid as uuid3_423_1_, representa4_.updated as updated4_423_1_,
representa4_.updatedBy_id as updatedB9_423_1_, representa4_.language_id as languag10_423_1_, representa4_.text
 as text5_423_1_, representa4_.abbreviatedLabel as abbrevia6_423_1_, representa4_.label as label7_423_1_
   from
   TaxonName this_
   left outer join DefinedTermBase rank2_ on this_.rank_id=rank2_.id
   left outer join DefinedTermBase_Representation representa3_ on rank2_.id=representa3_.DefinedTermBase_id
   left outer join Representation representa4_ on representa3_.representations_id=representa4_.id
   where this_.id in (
       select distinct this_.id as y0_ from TaxonName this_ left
           outer join NameRelationship relationsf1_ on this_.id=relationsf1_.relatedFrom_id
               where ((1=1 and (lower(this_.titleCache) like 'Geis%')) and ((not (relationsf1_.type_id=915)
or relationsf1_.type_id is null)))
       )
   order by this_.titleCache asc limit 20
```

but the profiler reports ~5 seconds for this mysql query:

☑ Hide nested calls   ☐ Group queries by SQL type   ☐ Include prepareStatement()/prepareCall()   ☐ Hide zero time calls

| Call | Time (ms) | | Avg. Time (ms) | Invocation count |
|------|-----------|---|----------------|------------------|
| :ype_id=?) or relationsf1_.type_id is null)))) order by this_.titleCache asc limit ? | 11,421 | 22 % | 5,710 | 2 |
| :ype_id=?) or relationsf1_.type_id is null)))) limit ? | 598 | 1 % | 299 | 2 |

| Reverse Call Tree | Time (ms) | | Invocation Count |
|-------------------|-----------|---|------------------|
| ⊟ 🏴 com.mchange.v2.c3p0.impl.**NewProxyPreparedStatement.executeQuery**() NewProxyPreparedStatement.java | 11,421 | 100 % | 2 |
|   ⊟ 🔖 org.hibernate.internal.**CriteriaImpl.list**() CriteriaImpl.java:363 | | | |
|     ⊞ 🏴 eu.etaxonomy.cdm.persistence.dao.hibernate.common.**CdmEntityDaoBase.findByParamWithRestrictions**( | | | |

**#4 - 02/21/2019 12:29 PM - Andreas Kohlbecker**

*- File picture588-1.png added*

the most time is actually consumed by

| Call Tree | Time (ms) | | Avg. Time (ms) | Own Time (ms) | Invocation Count |
|-----------|-----------|---|----------------|---------------|------------------|
| ⊟ MysqlIO.java:2954 🏴 com.mysql.jdbc.util.**ReadAheadInputStream.read**(byte[], int, int) | 11,060 | 12 % | 5,530 | < 0.1 | 2 |
|  ⊟ ReadAheadInputStream.java:173 🏴 com.mysql.jdbc.util.**ReadAheadInputStream.readFromUnderlyingS** | 11,060 | 12 % | 5,530 | 11,060 | 2 |
|     ReadAheadInputStream.java:143 🏴 java.net.**AbstractPlainSocketImpl.isConnectionResetPending**( | 0 | 0 % | 0 | 0 | 2 |
| ⊞ MysqlIO.java:3425 🏴 com.mysql.jdbc.**MysqlIO.readFully**(InputStream, byte[], int, int) | < 0.1 | 0 % | < 0.1 | < 0.1 | 2 |

Do we need to update the mysql driver for some reason?

**#5 - 02/21/2019 02:41 PM - Andreas Kohlbecker**

Correction: The sql query showing the problem is:

```
select
   this_.id as id1_499_2_, this_.created as created2_499_2_, this_.createdBy_id as created40_499_2_, this_.
uuid as uuid3_499_2_, this_.updated as updated4_499_2_, this_.updatedBy_id as updated41_499_2_, this_.
lsid_authority as lsid_aut5_499_2_, this_.lsid_lsid as lsid_lsi6_499_2_, this_.lsid_namespace as
lsid_nam7_499_2_, this_.lsid_object as lsid_obj8_499_2_, this_.lsid_revision as lsid_rev9_499_2_, this_.
protectedTitleCache as protect10_499_2_, this_.titleCache as titleCa11_499_2_, this_.acronym as
acronym12_499_2_, this_.anamorphic as anamorp13_499_2_, this_.appendedPhrase as appende14_499_2_, this_.
authorshipCache as authors15_499_2_, this_.basionymAuthorship_id as basiony42_499_2_, this_.binomHybrid as
binomHy16_499_2_, this_.breed as breed17_499_2_, this_.combinationAuthorship_id as combina43_499_2_, this_.
cultivarName as cultiva18_499_2_, this_.exBasionymAuthorship_id as exBasio44_499_2_, this_.
exCombinationAuthorship_id as exCombi45_499_2_, this_.fullTitleCache as fullTit19_499_2_, this_.
genusOrUninomial as genusOr20_499_2_, this_.homotypicalGroup_id as homotyp46_499_2_, this_.hybridFormula as
hybridF21_499_2_, this_.inBasionymAuthorship_id as inBasio47_499_2_, this_.inCombinationAuthorship_id as
inCombi48_499_2_, this_.infraGenericEpithet as infraGe22_499_2_, this_.infraSpecificEpithet as
```

```
infraSp23_499_2_, this_.monomHybrid as monomHy24_499_2_, this_.nameApprobation as nameApp25_499_2_, this_.
nameCache as nameCac26_499_2_, this_.nameType as nameTyp27_499_2_, this_.nomenclaturalMicroReference as
nomencl28_499_2_, this_.nomenclaturalReference_id as nomencl49_499_2_, this_.nomenclaturalSource_id as
nomencl50_499_2_, this_.originalPublicationYear as origina29_499_2_, this_.parsingProblem as parsing30_499_2_,
 this_.problemEnds as problem31_499_2_, this_.problemStarts as problem32_499_2_, this_.
protectedAuthorshipCache as protect33_499_2_, this_.protectedFullTitleCache as protect34_499_2_, this_.
protectedNameCache as protect35_499_2_, this_.publicationYear as publica36_499_2_, this_.rank_id as
rank_id51_499_2_, this_.specificEpithet as specifi37_499_2_, this_.subGenusAuthorship as subGenu38_499_2_,
this_.trinomHybrid as trinomH39_499_2_, rank2_.id as id2_99_0_, rank2_.created as created3_99_0_, rank2_.
createdBy_id as created41_99_0_, rank2_.uuid as uuid4_99_0_, rank2_.updated as updated5_99_0_, rank2_.
updatedBy_id as updated42_99_0_, rank2_.lsid_authority as lsid_aut6_99_0_, rank2_.lsid_lsid as lsid_lsi7_99_0_
, rank2_.lsid_namespace as lsid_nam8_99_0_, rank2_.lsid_object as lsid_obj9_99_0_, rank2_.lsid_revision as
lsid_re10_99_0_, rank2_.protectedTitleCache as protect11_99_0_, rank2_.titleCache as titleCa12_99_0_, rank2_.
termType as termTyp13_99_0_, rank2_.uri as uri14_99_0_, rank2_.idInVocabulary as idInVoc15_99_0_, rank2_.
kindOf_id as kindOf_43_99_0_, rank2_.partOf_id as partOf_44_99_0_, rank2_.symbol as symbol16_99_0_, rank2_.
symbol2 as symbol17_99_0_, rank2_.vocabulary_id as vocabul45_99_0_, rank2_.orderIndex as orderIn20_99_0_,
rank2_.rankClass as rankCla40_99_0_, representa3_.DefinedTermBase_id as DefinedT1_123_4_, representa4_.id as
represen2_123_4_, representa4_.id as id1_423_1_, representa4_.created as created2_423_1_, representa4_.
createdBy_id as createdB8_423_1_, representa4_.uuid as uuid3_423_1_, representa4_.updated as updated4_423_1_,
representa4_.updatedBy_id as updatedB9_423_1_, representa4_.language_id as languag10_423_1_, representa4_.text
 as text5_423_1_, representa4_.abbreviatedLabel as abbrevia6_423_1_, representa4_.label as label7_423_1_
    from
    TaxonName this_
    left outer join DefinedTermBase rank2_ on this_.rank_id=rank2_.id
    left outer join DefinedTermBase_Representation representa3_ on rank2_.id=representa3_.DefinedTermBase_id
    left outer join Representation representa4_ on representa3_.representations_id=representa4_.id
    where this_.id in (
        select distinct this_.id as y0_ from TaxonName this_ left
            outer join NameRelationship relationsf1_ on this_.id=relationsf1_.relatedFrom_id
                where ((1=1 and (lower(this_.titleCache) like '%')) and ((not (relationsf1_.type_id=915) or
relationsf1_.type_id is null)))
        )
    order by this_.titleCache asc limit 500
```

this one takes ~5s

### #6 - 03/23/2022 10:50 AM - Andreas Müller

*- Subject changed from CdmFilterablePagingProviderFactory.taxonNamesWithoutOrthophicIncorrect() casues performance overhead to CdmFilterablePagingProviderFactory.taxonNamesWithoutOrthophicIncorrect() causes performance overhead*

### #7 - 03/23/2022 10:50 AM - Andreas Müller

*- Subject changed from CdmFilterablePagingProviderFactory.taxonNamesWithoutOrthophicIncorrect() causes performance overhead to CdmFilterablePagingProviderFactory.taxonNamesWithoutOrthographicIncorrect() causes performance overhead*

## Files

| | | | |
|---|---|---|---|
| picture278-1.png | 17.9 KB | 02/21/2019 | Andreas Kohlbecker |
| picture588-1.png | 13.1 KB | 02/21/2019 | Andreas Kohlbecker |