

BIEN 3 - Task #902

Task # 887 (Rejected): fix disk space leak that fills the disk and crashes the import

fix bug that causes joining on the wrong columns in the import

04/23/2014 03:52 PM - Aaron Marcuse-Kubitza

<div>Status: Resolved</div> <div>Priority: Normal</div> <div>Assignee: Aaron Marcuse-Kubitza</div> <div>Category:</div> <div>Target version:</div> <div>Activity type:</div>	<div>Start date:</div> <div>Due date:</div> <div>% Done: 100%</div> <div>Estimated time: 0.00 hour</div>
<div>Description</div> <div>bug fixed in r14074</div> <div>issue</div> <div><ul style="list-style-type: none">• in some queries, the columns being joined on are completely the wrong set (columns that are not even used in any unique index)• this creates unusual (and perhaps unstable) query plans (#884), which in turn trigger a disk space leak in Postgres/Ubuntu (#887) and make the query hang indefinitely• the problem occurs 8 min into the import ("03:00:28" - "02:52:30"; see snapshot below > Mac > ARIZ.omoccurrences)• was not an issue in the last successful import• this is definitely the cause of the incorrect query plans (#884), but is <i>not</i> the cause of the disk space leak (#887), as that is a Postgres bug <i>triggered by</i> the unusual query plans (no query plan should ever cause a disk space leak or make the query hang indefinitely).• if this is caused by a Postgres bug, it may be the same bug that caused the disk space leak (#887)• it is more likely caused by a Postgres bug than our import code, because the changes we made since the last successful import (2014-2-2) do not look like they would cause radical changes in the joins• it is almost certainly not an Ubuntu bug¹, because it also occurs on Mac OS X</div> <div>¹ other than perhaps the shared BSD code between Linux and Mac, but this is unlikely unless they cross-pollinate system updates</div> <div>.</div> <div>test case</div> <div>see the following pg_stat_activity snapshots (note the JOIN columns in the EXPLAIN output below each query):</div> <div><div>config: Mac OS X 10.8 (on starscream) with the installer version of Postgres 9.3.1 (not the standalone .app)</div><div>import command: \$0 # nested shell to contain the env changes unset TMOUT # TMOUT causes shell to exit even with background processes unset n # clear any limit set in .profile (unless desired) unset log # allow logging output to go to log files export version=test_import declare -ax inputs; inputs=(inputs/ARIZ/) # or NY #declare -ax inputs; inputs=(inputs/{ARIZ,SALVIAS}/) # SALVIAS was part of the original test case . bin/import_all # once problem occurs, clean up: kill % # cancel after_import() . bin/stop_imports sudo pkill -f postgres sudo pkill -f postgres # sometimes the first pkill doesn't kill everything</div><div>backend_start 2014-05-09 02:52:29.987578-07 query_start 2014-05-09 03:00:28.378299-07 /*ARIZ.omoccurrences*/ CREATE TEMP TABLE "location_pkeys" AS</div></div>	

```

SELECT
  "in#6"."occurrenceID"
, "location"."location_id" AS "out.location_id"
FROM "in#6"
JOIN "location" ON COALESCE("location"."elevation_m", CAST('NaN
' AS double precision)) = COALESCE("in#6"."_alt(1=verbatimElevation, 2=_avg(1=minimumElevat.result
::double", CAST('NaN' AS double precision))
/* EXPLAIN:
Merge Join  (cost=86617.89..9234073.27 rows=609750474 width=8)
  Merge Cond: ((COALESCE("in#6"."_alt(1=verbatimElevation, 2=_avg(1=minimumElevat.result::double",
'NaN'::double precision)) = (COALESCE(location.elevation_m, 'NaN'::double precision)))
  ->  Sort  (cost=28258.34..28857.47 rows=239654 width=12)
        Sort Key: (COALESCE("in#6"."_alt(1=verbatimElevation, 2=_avg(1=minimumElevat.result::doubl
e", 'NaN'::double precision))
        ->  Seq Scan on "in#6"  (cost=0.00..6844.54 rows=239654 width=12)
  ->  Sort  (cost=58359.55..59631.70 rows=508859 width=12)
        Sort Key: (COALESCE(location.elevation_m, 'NaN'::double precision))
        ->
*/

```

config: Ubuntu 14.04 (on vegbiendev) with official Ubuntu version of Postgres 9.3

```

import command:
declare -ax inputs=(inputs/{ARIZ,SALVIAS}/) # SALVIAS because it's part of the test case
export version=test_import
. bin/import_all

```

```

/*ARIZ.omoccurrences*/      CREATE TEMP TABLE "location_pkeys" AS
SELECT
  "in#6"."occurrenceID"
, "location"."location_id" AS "out.location_id"
FROM "in#6"
JOIN "location" ON COALESCE("location"."elevation_m", CAST('NaN
' AS double precision)) = COALESCE("in#6"."_alt(1=verbatimElevation, 2=_avg(1=minimumElevat.result
::double", CAST('NaN' AS double precision))
/* EXPLAIN:
Merge Join  (cost=52039.20..3508151.98 rows=230343442 width=8)
  Merge Cond: ((COALESCE(location.elevation_m, 'NaN
'::double precision)) = (COALESCE("in#6"."_alt(1=verbatimElevation, 2=_avg(1=minimumElevat.result
:double", 'NaN'::double precision)))
  ->  Sort  (cost=23780.86..24261.44 rows=192230 width=12)
        Sort Key: (COALESCE(location.elevation_m, 'NaN'::double precision))
        ->  Seq Scan on location  (cost=0.00..6910.30 rows=192230 width=12)
  ->  Sort  (cost=28258.34..28857.47 rows=239654 width=12)
        Sort Key: (COALESCE("in#6"."_alt(1=verbatimElevation, 2=_avg(1=minimumElevat.result::doubl
e", 'NaN'::double precision))
        -
*/

```

config: Ubuntu 12.04 (on vegbiendev) with compatibility version of Postgres 9.3

```

import command:
. bin/import_all # all datasources

```

```

/*NVS.StemObservation*/      CREATE TEMP TABLE "stemobservation_pkeys" AS
SELECT
  "in#11"."row_num"
, "stemobservation"."stemobservation_id" AS "out.stemobservation_id"
FROM "in#11"
JOIN "stemobservation" ON COALESCE("stemobservation"."height_m", CAST('NaN' AS double precision))
= COALESCE("in#11"."NVS.StemObservation.Height::double", CAST('NaN' AS double precision))
/* EXPLAIN:
Merge Join  (cost=1045217.21..95909185.15 rows=6323041431 width=8)
  Merge Cond: ((COALESCE(stemobservation.height_m, 'NaN'::double precision)) = (COALESCE("in#11"."
NVS.StemObservation.Height::double", 'NaN'::double precision)))
  ->  Sort  (cost=30039.56..30492.76 rows=181279 width=12)
        Sort Key: (COALESCE(stemobservation.height_m, 'NaN'::double precision))

```

```
-> Seq Scan on stemobservation (cost=0.00..14206.79 rows=181279 width=12)
-> Materialize (cost=1015177.64..1050057.81 rows=6976033 width=12)
-> Sort (cost=1015177.64..1032617.73 rows=6976033 width=12)
    Sort Key: (COALESCE("in#11"."NVS.StemObs

/*CVS.taxon_observation.***/      CREATE TEMP TABLE "location_pkeys" AS
SELECT
"in#5"."row_num"
, "location"."location_id" AS "out.location_id"
FROM "in#5"
JOIN "location" ON ("location"."accesslevel" = "in#5"."CVS.taxon_observation.**.accessRights::acce
sslevel" OR ("location"."accesslevel" IS NULL AND "in#5"."CVS.taxon_observation.**.accessRights::a
ccesslevel" IS NULL))
/* EXPLAIN:
Nested Loop (cost=0.00..83049883712.45 rows=27683225000 width=8)
  Join Filter: ((location.accesslevel = "in#5"."CVS.taxon_observation.**.accessRights::accesslevel
") OR ((location.accesslevel IS NULL) AND ("in#5"."CVS.taxon_observation.**.accessRights::accessle
vel" IS NULL)))
    -> Seq Scan on location (cost=0.00..175778.45 rows=5536645 width=8)
    -> Materialize (cost=0.00..35434.00 rows=1000000 width=8)
        -> Seq Scan on "in#5" (cost=0.00..30434.00 rows=1000000 width=8)
*/
```

fix

- 1. check if [r13361](#) fixed this (see explanation in [r13392](#))no, problem still occurs
- 2. see proposed steps in [#905](#)

Subtasks:

Task # 905: narrow down the cause of the import bug (incorrect join columns and disk sp... Resolved

Associated revisions

Revision 14074 - 07/15/2014 05:44 PM - Aaron Marcuse-Kubitza

bugfix: lib/sql_io.py: put_table(): handle_MissingCastException(): when updating join_cols, don't add new entry for join_cols[out_col], only update existing one. this fixes #902 (import bug), and with #902 fixed, #887 (disk space leak) should no longer occur.

History

#1 - 04/23/2014 03:55 PM - Aaron Marcuse-Kubitza

- Description updated

#2 - 04/25/2014 05:53 AM - Aaron Marcuse-Kubitza

- Description updated

#3 - 04/25/2014 05:55 AM - Aaron Marcuse-Kubitza

- Description updated

#4 - 04/25/2014 05:59 AM - Aaron Marcuse-Kubitza

- Description updated

#5 - 04/25/2014 06:01 AM - Aaron Marcuse-Kubitza

- Description updated

#6 - 04/25/2014 06:10 AM - Aaron Marcuse-Kubitza

- Description updated

#7 - 04/25/2014 06:11 AM - Aaron Marcuse-Kubitza

- Description updated

#8 - 05/02/2014 10:57 PM - Aaron Marcuse-Kubitza

- Description updated

#9 - 05/03/2014 12:43 AM - Aaron Marcuse-Kubitza

- Description updated

#10 - 05/03/2014 12:43 AM - Aaron Marcuse-Kubitza

- Description updated

#11 - 05/09/2014 06:52 PM - Aaron Marcuse-Kubitza

- Description updated

#12 - 05/09/2014 07:15 PM - Aaron Marcuse-Kubitza

- Description updated

#13 - 05/09/2014 07:19 PM - Aaron Marcuse-Kubitza

- Description updated

#14 - 05/09/2014 07:28 PM - Aaron Marcuse-Kubitza

- Description updated

#15 - 05/09/2014 07:29 PM - Aaron Marcuse-Kubitza

- Description updated

#16 - 05/09/2014 07:33 PM - Aaron Marcuse-Kubitza

- Description updated

#17 - 05/09/2014 11:43 PM - Aaron Marcuse-Kubitza

- Description updated

#18 - 05/09/2014 11:44 PM - Aaron Marcuse-Kubitza

- % Done changed from 0 to 20

#19 - 05/09/2014 11:46 PM - Aaron Marcuse-Kubitza

- Description updated

#20 - 05/10/2014 12:20 AM - Aaron Marcuse-Kubitza

- Description updated

#21 - 05/10/2014 12:21 AM - Aaron Marcuse-Kubitza

- Description updated

#22 - 05/28/2014 02:51 PM - Aaron Marcuse-Kubitza

- Parent task set to #905

#23 - 05/28/2014 02:59 PM - Aaron Marcuse-Kubitza

- Parent task changed from #905 to #887

#24 - 07/10/2014 11:11 AM - Aaron Marcuse-Kubitza

- Description updated

#25 - 07/15/2014 03:28 PM - Aaron Marcuse-Kubitza

- Description updated

#26 - 07/15/2014 03:29 PM - Aaron Marcuse-Kubitza

- Description updated

#27 - 07/15/2014 05:27 PM - Aaron Marcuse-Kubitza

- Description updated

#28 - 07/15/2014 05:48 PM - Aaron Marcuse-Kubitza

- *Description updated*

- *Status changed from New to Resolved*