Pulp - Issue #1956

"Cleaning duplicate packages" fails after upgrading from 2.6.0 to 2.8.3.

05/31/2016 12:37 PM - akegata@gmail.com

Status:	CLOSED - WONTFIX	**Start date:**	
Priority:	High	**Due date:**	
Assignee:		**Estimated time:**	0:00 hour
Category:			
Sprint/Milestone:			
Severity:	3. High	**Groomed:**	No
Version:	2.8.3	**Sprint Candidate:**	No
Platform Release:		**Tags:**	Pulp 2
OS:	RHEL 6	**Sprint:**	
Triaged:	Yes	**Quarter:**	

Description

After applying the patch in https://pulp.plan.io/issues/1952, the actual syncing of packages work, but the whole sync fails on the "Cleaning duplicate packages" step:

```
Operations:         sync
Resources:          RHEL7_test_x86_64 (repository)
State:              Failed
Start Time:         2016-05-31T10:17:00Z
Finish Time:        2016-05-31T10:31:20Z
Result:             N/A
Task Id:            d3b8443b-0e61-4027-8ded-b419b38f13f6
Progress Report:
  Yum Importer:
    Comps:
      State: FINISHED
    Content:
      Details:
        Drpm Done:  0
        Drpm Total: 0
        Rpm Done:   1
        Rpm Total:  1
      Error Details:
      Items Left:    0
      Items Total:   1
      Size Left:     0
      Size Total:    74037668
      State:         FINISHED
    Distribution:
      Error Details:
      Items Left:    0
      Items Total:   0
      State:         FINISHED
    Errata:
      State: FINISHED
    Metadata:
      State: FINISHED
    Purge Duplicates:
      Error: command SON([('mapreduce', u'units_rpm'), ('map', Code("\n
             function () {\n        var key_fields = [this.name, this.epoch,
             this.version, this.release, this.arch]\n
             emit(key_fields.join('-'), {ids: [this._id]});\n    }\n    ", {})),
             ('reduce', Code("\n    function (key, values) {\n      // collect
             mapped values into the first value to build the list of ids for
             this key/nevra\n      var collector = values[0]\n      // since
             collector is values[0] start this loop at index 1\n      // reduce
```

```
            isn't called if map only emits one result for key,\n      // so
            there is at least one value to collect\n      for (var i = 1; i <
            values.length; i++) {\n        collector.ids =
            collector.ids.concat(values[i].ids)\n      }\n      return
            collector\n    }\n    ", {})), ('out', {'inline': 1}), ('query',
            {}), ('finalize', Code("\n    function (key, reduced) {\n        if
            (reduced.ids.length > 1) {\n            return reduced;\n
            }\n        // if there's only one value after reduction, this key
            is useless\n        // undefined is implicitly returned here, which
            saves space\n    }\n    ", {}))]) on namespace pulp_database.$cmd
            failed: exception: BSONObj size: 17449063 (0x67400A01) is invalid.
            Size must be between 0 and 16793600(16MB) First element: 0: { _id:
            "2048-cli-0-0.9-4.git20141214.723738c.el5-x86_64", value: null }
      State: FAILED
Traceback:        Traceback (most recent call last):   File
                  "/usr/lib/python2.6/site-packages/celery/app/trace.py", line
                  240, in trace_task     R = retval = fun(*args, **kwargs)
                  File
                  "/usr/lib/python2.6/site-packages/pulp/server/async/tasks.py",
                  line 473, in __call__      return super(Task,
                  self).__call__(*args, **kwargs)   File
                  "/usr/lib/python2.6/site-packages/pulp/server/async/tasks.py",
                  line 103, in __call__      return super(PulpTask,
                  self).__call__(*args, **kwargs)   File
                  "/usr/lib/python2.6/site-packages/celery/app/trace.py", line
                  437, in __protected_call__      return self.run(*args,
                  **kwargs)   File
                  "/usr/lib/python2.6/site-packages/pulp/server/controllers/repo
                  sitory.py", line 810, in sync     raise
                  pulp_exceptions.PulpExecutionException(_('Importer indicated a
                  failed response')) PulpExecutionException: Importer indicated
                  a failed response
```

History

#1 - 05/31/2016 09:26 PM - bmbouter

This looks like an exception occurred during the map reduce code during the removal of duplicate NEVRA packages from the repo.

#2 - 05/31/2016 09:35 PM - bmbouter

@akegata Would you be able to post a reproducer so this issue can be investigated? Doing it with pulp-admin commands would be ideal, but curl/httpie commands would work too.

#3 - 06/03/2016 04:41 PM - dkliban@redhat.com

- Priority changed from Normal to High

- Severity changed from 2. Medium to 3. High

- Triaged changed from No to Yes

#4 - 06/03/2016 05:07 PM - akegata@gmail.com

Not sure if I can reproduce this from a clean install. It happens on most of our current repos though.

#5 - 06/07/2016 04:39 PM - semyers

```
BSONObj size: 17449063 (0x67400A01) is invalid.
            Size must be between 0 and 16793600(16MB)
```

If I understand the error right, the "reduced" document returned as the result of a mapreduce is too large for mongo. This is unfortunate, and you might not have a way around this other than upgrading to mongo 2.6 from SCL[0].

The main limiting factor is the number of documents in the units_rpm collection, so it's possible that cleaning up orphans[1] might immediately get you past this issue. Removing unused rpm repos, if any, before purging orphans would also help. That might not help, but even if it does it's just a temporary relief and not a fix. This was tested with a *lot* of content units, but it looks like you might have *a lot more*. Mongo 2.6 introduced new features that let you work on huge collections without breaking mongo, which is why I think that upgrading is the solution here.

[0]: https://www.softwarecollections.org/en/scls/rhscl/rh-mongodb26/
[1]: pulp-admin orphan remove --type rpm

#6 - 06/07/2016 09:32 PM - akegata@gmail.com

> The main limiting factor is the number of documents in the units_rpm collection, so it's possible that cleaning up orphans[1] might immediately get you past this issue. Removing unused rpm repos, if any, before purging orphans would also help. That might not help, but even if it does it's just a temporary relief and not a fix. This was tested with a *lot* of content units, but it looks like you might have *a lot more*. Mongo 2.6 introduced new features that let you work on huge collections without breaking mongo, which is why I think that upgrading is the solution here.

I just tried a pulp-admin orphan remove --all. The syncs still fail with the same error.
It's puzzling to me why this happened after upgrading from pulp 2.6 to 2.8.3. It's happened on both our stand alone installations. Is the units_rpm collection something that was added after 2.6?
I would have guessed we have a big, but not enormous, number of rpm's. Would it be of interest to see how many content units we have?

Anyway, our plan is to upgrade to mongodb 3.2 on one of the machines tomorrow, and on the other one the day after if everything works fine. I'll get back with an update after the upgrade.

#7 - 06/07/2016 10:15 PM - semyers

akegata@gmail.com wrote:

> I just tried a pulp-admin orphan remove --all. The syncs still fail with the same error.
> It's puzzling to me why this happened after upgrading from pulp 2.6 to 2.8.3. It's happened on both our stand alone installations. Is the units_rpm collection something that was added after 2.6?
> I would have guessed we have a big, but not enormous, number of rpm's. Would it be of interest to see how many content units we have?

This change was introduced in 2.8 and fixes a data integrity issue that could result in unpredictable RPM publishes. The units_rpm collection is not new, the data integrity check is.

I would love to know how many units are in your units_rpm collection.

#8 - 06/07/2016 10:22 PM - akegata@gmail.com

> This change was introduced in 2.8 and fixes a data integrity issue that could result in unpredictable RPM publishes. The units_rpm collection is not new, the data integrity check is.
>
> I would love to know how many units are in your units_rpm collection.

Alright, that makes sense then.
How do I retrieve that value?

#9 - 06/07/2016 10:29 PM - semyers

If you're on the mongo shell (run mongo pulp_database on the pulp server), you can run this db query to get the rpm unit count db.units_rpm.count(). Feel free to PM me (smyers) in IRC if you have trouble.

#10 - 06/08/2016 02:09 PM - akegata@gmail.com

We upgraded one of the servers to mongodb 3.2 as planned today. After the upgrade this issue is no longer present, everything seems to work as expected.

For reference, count db.units_rpm.count() says we have 248366 rpm's.

#11 - 04/12/2019 09:34 PM - bmbouter

- Status changed from NEW to CLOSED - WONTFIX

#12 - 04/12/2019 09:39 PM - bmbouter

Pulp 2 is approaching maintenance mode, and this Pulp 2 ticket is not being actively worked on. As such, it is being closed as WONTFIX. Pulp 2 is still accepting contributions though, so if you want to contribute a fix for this ticket, please reopen or comment on it. If you don't have permissions to reopen this ticket, or you want to discuss an issue, please reach out via the developer mailing list.

#13 - 04/15/2019 10:29 PM - bmbouter

- Tags Pulp 2 added