

Infrastructure - Bug #7959

ObjectListHarvestTask needs refactoring

2017-01-03 23:25 - Rob Nahf

Status:	Closed	Start date:	2017-01-03
Priority:	Urgent	Due date:	
Assignee:	Rob Nahf	% Done:	100%
Category:	d1_synchronization	Estimated time:	0.00 hour
Target version:	CCI-2.3.7	Story Points:	
Milestone:	None		
Product Version:	*		

Description

The task's execution logic is not apparent from reading through the code, and looks like it is prone to errors:

* if the objectList within the harvest time interval is not fully processed, and the list is not retrieved in chronological order, a bad lastModifiedDate could be persisted.

* setting of the endHarvestInterval value is partly in the instance properties section, and partly in the call method.

* batchSize adjustment doesn't make sense: How do you know if halving the batch size will get it under the max limit?
how do the batchSize and MemberNodes native page size interoperate?

it seems like the call should retrieve all of the objects and create a chronologically sorted objectList that could then be used to reliably set the lastHarvestedDate.

A partial refactoring was attempted before deciding it was too much to try to do in the 2.3 branch. attached.

Associated revisions

Revision 19015 - 2017-11-07 23:20 - Rob Nahf

fixes #7959: complete refactoring of ObjectListHarvestTask to properly handle setting the lastHarvestedDate when certain types of exceptions are thrown. Also no longer depends on MNs returning ObjectLists in chronological order.

Revision 19015 - 2017-11-07 23:20 - Rob Nahf

fixes #7959: complete refactoring of ObjectListHarvestTask to properly handle setting the lastHarvestedDate when certain types of exceptions are thrown. Also no longer depends on MNs returning ObjectLists in chronological order.

Revision 19018 - 2017-11-08 20:32 - Rob Nahf

fixes #7959: added unit test for SortedHarvestTimepointMap and made bug fixes on the class. Added maxHarvestSize parameter to ObjectListHarvestTask.call.

Revision 19018 - 2017-11-08 20:32 - Rob Nahf

fixes #7959: added unit test for SortedHarvestTimepointMap and made bug fixes on the class. Added maxHarvestSize parameter to ObjectListHarvestTask.call.

Revision 19023 - 2017-11-09 20:34 - Rob Nahf

refs #7959: added unit test case for ObjectListHarvestTask, mainly testing retrieval from Member Nodes and correct paging behavior & exception handling.

Revision 19023 - 2017-11-09 20:34 - Rob Nahf

refs #7959: added unit test case for ObjectListHarvestTask, mainly testing retrieval from Member Nodes and correct paging behavior & exception handling.

Revision 19024 - 2017-11-09 21:17 - Rob Nahf

refs #7959: added a unit test to test proper spooling of the identifiers retrieved to the sync Queue, and that lastHarvestedDate is updated properly.

Revision 19024 - 2017-11-09 21:17 - Rob Nahf

refs #7959: added a unit test to test proper spooling of the identifiers retrieved to the sync Queue, and that lastHarvestedDate is updated properly.

Revision 19025 - 2017-11-10 06:18 - Rob Nahf

refs #7959: added a test for processing large timepoint, and fixed the harvest implementation so that excessively large timepoints can be harvested (even if they exceed the stated capacity).

Revision 19025 - 2017-11-10 06:18 - Rob Nahf

refs #7959: added a test for processing large timepoint, and fixed the harvest implementation so that excessively large timepoints can be harvested (even if they exceed the stated capacity).

Revision 19026 - 2017-11-11 06:10 - Rob Nahf

refs #7959: added check for synchroniztion being disabled in the paged object list retrieval. Also fixed bug that would update the dateLastHarvested on empty harvests (the old ones didn't).

Revision 19026 - 2017-11-11 06:10 - Rob Nahf

refs #7959: added check for synchroniztion being disabled in the paged object list retrieval. Also fixed bug that would update the dateLastHarvested on empty harvests (the old ones didn't).

Revision 19027 - 2017-11-13 22:57 - Rob Nahf

refs #7959: bumping the 2.3 branch artifact for d1_synchronization and d1_process_daemon for beta deployment

Revision 19027 - 2017-11-13 22:57 - Rob Nahf

refs #7959: bumping the 2.3 branch artifact for d1_synchronization and d1_process_daemon for beta deployment

History

#1 - 2017-11-06 18:12 - Rob Nahf

- Target version changed from CCI-2.4.0 to CCI-2.3.6

- Priority changed from Normal to Urgent

Several GMN instances in production now sort in reverse chronological order, and we've experienced problems in STAGE for Pangaea, so need to prioritize this one.

#2 - 2017-11-06 18:19 - Rob Nahf

points to consider for partial harvest:

* even for chronologically sorted object lists, objects with the same lastModifiedDate could end up on different pages. If the latter page doesn't get loaded, and the lastHarvestDate is updated, the objects with the same time stamp will get missed, because 1 millisecond is being added to the harvest time.

* excessively large harvests (for example, resetting the lastHarvestDate) could cause a memory burden on the CN if we hold the accumulated objectList in memory.

* to minimize impact of failures while spooling onto the sync queue, we want to sort the accumulated list chronologically, and periodically update the lastHarvestDate at regular intervals while loading up the queue.

#3 - 2017-11-07 18:38 - Rob Nahf

- Target version changed from CCI-2.3.6 to CCI-2.3.7

2.3.6 is already deployed. moving to 2.3.7

#4 - 2017-11-07 23:53 - Rob Nahf

- % Done changed from 0 to 100

- Status changed from New to Closed

Applied in changeset [d1|r19015](#).

#5 - 2017-11-08 21:16 - Rob Nahf

committed a highly refactored ObjectListHarvestTask.

* removed logic that has the harvest processor wait for the synchronization queue to get smaller

(it only harvests as many as the queue has room for at the beginning of the harvest)

* handle harvesting from the theoretical member nodes who don't implement start & count (just have to get the total in the first page, otherwise fail).

* the basic strategy is to retrieve all up front, then spool to synchronization queue

Because there is no way for the CN to know if the member nodes will reliably return objectLists in ascending time order, the harvester cannot use streaming methodology to read from the member node and spool onto the synchronization queue. Instead it imposes a maximum harvest size, so the in-memory harvest does not get "too big."

Further work would be needed to avoid unnecessary work in the case of large total retrieval in comparison to the maximumHarvest size. This would be accomplished by reducing the harvest time window.

#6 - 2017-11-08 21:18 - Rob Nahf

- Status changed from Closed to In Progress

- Tracker changed from Story to Bug

- % Done changed from 100 to 30

need to test in STAGE.

#7 - 2017-11-08 21:18 - Rob Nahf

- % Done changed from 30 to 50

- Status changed from In Progress to Testing

#8 - 2017-11-09 02:26 - Rob Nahf

- Status changed from Testing to Closed

- % Done changed from 50 to 100

Applied in changeset [d1|r19018](#).

#9 - 2017-11-10 06:24 - Rob Nahf

- File deleted (ObjectListHarvestTask.java)

#10 - 2017-11-10 06:31 - Rob Nahf

How do we harvest a timepoint that is larger than the synchronization queue capacity? Normally, the harvest total would be managed to keep it under the maximum, but that would lead to 0-length harvests, and stalling synchronization because the lastHarvestedDate never increases.

Instead, changed the SortedHarvestTimepointMap so that the earliest timepoint can never be deleted, so the harvest has a chance at moving forward.

#11 - 2017-12-26 18:20 - Dave Vieglais

- Sprint set to CCI-2.3.7