

Infrastructure - Task #7935

Use a dedicated hazelcast queue as the listener to generate index task

2016-11-16 21:59 - Jing Tao

Status:	New	Start date:	2016-11-16
Priority:	Normal	Due date:	
Assignee:	Jing Tao	% Done:	0%
Category:	d1_indexer	Estimated time:	0.00 hour
Target version:	CCI-2.4.0	Story Points:	
Milestone:	None		
Product Version:	*		

Description

Currently the index task generator listens to the events from the systemmetadata map. But the events from the system metadata map can come from different scenarios. For example, when tomcat starts up, hazelcast will load the system metadata from the database into the systemmetadata map. If the index generator is running, it can generate lots of index tasks which we really don't need.

If we have a dedicate queue, we can avoid this scenario. But we should call a method to put the system metadata into the queue specifically during the synchronization.

History

#1 - 2016-11-16 22:01 - Jing Tao

Besides the synchronization, what other scenario do we need to put the system metadata into the dedicated queue?

#2 - 2016-12-01 23:54 - Dave Vieglais

- Target version changed from CCI-2.3.1 to CCI-2.3.2

#3 - 2017-01-11 23:45 - Rob Nahf

The current operational structure (sync on UCSB and index on ORC) doesn't allow us to have sync generate index tasks in the local postgres DB.

In an orderly start, the index generator is started after tomcat (owner of HZ system metadata map), but tomcat restarts, and maybe even network partitioning issues might lead to a the index generator hearing a lot of spurious map add events.

Gaining this independence might be accomplished by creating a Hazelcast "SystemMetadataUpdated" list/set or map that is only added to by synchronization and the API update methods. The indexTaskGenerator would be listening to this data structure on ORC. Not sure if it solves the network partitioning issue.

#4 - 2017-02-27 18:07 - Jing Tao

- Target version changed from CCI-2.3.2 to CCI-2.4.0