Infrastructure - Story #8061

develop queue-based processing system for the CN

2017-04-05 22:40 - Rob Nahf

Status:	New	**Start date:**	2017-04-05
Priority:	Normal	**Due date:**	
Assignee:	Rob Nahf	**% Done:**	0%
Category:	Architecture Design	**Estimated time:**	0.00 hour
Target version:	CCI-2.4.0		
Story Points:			

Description

The event-based mechanism for generating indexing tasks is not robust to network segregation and inefficient because it triggers indexing tasks when system metadata are loaded into Hazelcast map - not "real" events, just a data hydration from persistent storage.

Investigate using reliable queues instead. The design will want to be abstracted so that different implementations can be swapped in at a later date, so use standard messaging patterns.

RabbitMQ, ActiveMQ are potential implementations to use.
ZeroMQ is a lower-level implementation, probably a bit more complicated, but very performant.

Subtasks:

Story # 8062: Install rabbitMQ on dev CNs	**Closed**
Task # 8078: standardize task serialization for language independence	**New**
Task # 8079: prototype durable task processing for d1_index_processor	**In Progress**
Task # 8080: ioslate queue creation logic from processing logic from the queue definiti...	**In Progress**
Task # 8086: upgrade Spring dependencies	**In Progress**
Story # 8081: develop federated broker configuration for indexing	**In Progress**
Story # 8082: implement SolrCloudClient to replace HttpService to allow concurrent upda...	**New**
Story # 8084: determine the backup strategy for rabbitMQ	**New**

History

#1 - 2017-04-26 05:56 - Rob Nahf

RabbitMQ uses the terms queues, exchanges, channels, brokers, consumers, publishers.

Our processing consumers will connect to named queues via channels, and we will likely be using their high-level framework which sets up handlers in the consumers, and exception handlers in the channel (I believe).Does it make sense to abstract the channels?

#2 - 2018-01-17 18:50 - Dave Vieglais

- Sprint set to Infrastructure backlog

#3 - 2018-01-17 19:00 - Rob Nahf

I recently came across Apache Flink, which is a stream-based messaging system with deliver-exactly-once guarantees, and could be a simpler system than RabbitMQ, depending on its robustness across the WAN. It looks like it is coupled with Kafka.

Keep as a possible alternative, although development work with RabbitMQ is mostly complete.