Infrastructure - Story #4463

Incorporate Node Replication Policy into replication processing

2014-03-14 21:47 - Skye Roseboom

Status:	In Progress		**Start date:**	2014-09-10
Priority:	Normal		**Due date:**	
Assignee:	Rob Nahf		**% Done:**	30%
Category:	d1_replication		**Estimated time:**	0.00 hour
Target version:	CCI-2.4.0			
Story Points:				

Description

Once MN replication policy are available through the Node Repository datasource, replication processing will need to be updated to honor this default replication policy when items without any policy are synchronized to the CN.

This task needs to follow redmine [#2192](#).

Subtasks:	
Task # 6372: rehabilitate ReplicationManager "unit" tests	**In Progress**
Task # 6373: utilize the new ApacheDS LDAP testing infrastructure	**Closed**
Task # 6374: Configure ReplicationManagerTestUnit with AttemptHistoryRepository	**Closed**
Task # 6375: Refactor out dependency on a working CN	**In Progress**
Task # 6376: test the new strategy against working ReplicationManager test	**New**
Task # 6378: code up the new strategy	**Testing**

Related issues:			
Related to Infrastructure - Story #2192: Implement NodeReplicationPolicy in N...	**Closed**	**2014-09-02**	**2015-01-06**
Related to Infrastructure - Story #8639: Replication performance is too slow ...	**New**	**2018-07-04**	

History

#1 - 2014-03-14 21:47 - Skye Roseboom

- Assignee deleted (Robert Waltz)

#2 - 2014-03-14 21:48 - Skye Roseboom

- Subject changed from Incorporate MN Replication Policy into replication processing to Incorporate Node Replication Policy into replication processing

#3 - 2014-03-14 21:49 - Skye Roseboom

- translation missing: en.field_remaining_hours set to 0.0

- Tracker changed from Task to Story

- Due date set to 2014-04-26

#4 - 2014-08-29 22:14 - Rob Nahf

- % Done changed from 0 to 20

- Assignee set to Rob Nahf

implementing in ReplicationManager class in trunk (v2.0).

It would be interesting to create a NodeRegistry query to return the set of possible nodes for a given object, but are probably constrained to use the hazelcast-backed map of Nodes, because that's where we are getting our systemMetadata. so I'll continue to use that structure.

Need to determine if the PreferredMemberNode list is ordered or unordered, and if we need any 'distribution' across preferred nodes.

#5 - 2014-09-10 16:40 - Rob Nahf

- Status changed from New to In Progress

this became a complicated task, because ReplicationManager tests were commented out (over a year ago), and some v2 changes have been subsequently committed. Will have to rehabilitate the tests before being able to test the new strategy. I'm creating subtasks for these.

#6 - 2014-09-24 18:15 - Robert Waltz

- Target version changed from 2014.16-Block.2.4 to CCI-1.5.0

- Due date changed from 2014-09-10 to 2014-09-24

#7 - 2014-10-02 19:46 - Robert Waltz

- Due date changed from 2014-09-24 to 2014-10-02

- Target version changed from CCI-1.5.0 to CCI-1.5.1

#8 - 2015-01-06 18:34 - Rob Nahf

- Target version changed from CCI-1.5.1 to CCI-2.0.0

- Due date changed from 2014-10-02 to 2015-01-06

#9 - 2015-05-28 23:54 - Rob Nahf

- % Done changed from 68 to 30

- Target version changed from CCI-2.0.0 to CCI-2.1.0

moving out of v2.0 release to v2.1 after review of 2.0.0 backlog at infrastructure meeting.

#10 - 2016-02-09 18:49 - Rob Nahf

- Target version changed from CCI-2.1.0 to CCI-2.2.0

#11 - 2016-03-29 17:58 - Rob Nahf

- Target version changed from CCI-2.2.0 to CCI-2.3.0

#12 - 2016-06-22 16:53 - Dave Vieglais

- Target version changed from CCI-2.3.0 to CCI-2.4.0

#13 - 2018-01-17 20:03 - Dave Vieglais

- Sprint set to Infrastructure backlog

#14 - 2018-07-04 11:19 - Dave Vieglais

- Related to Story #8639: Replication performance is too slow to service demand added