

Infrastructure - Task #2177

Story # 2166 (Closed): Hazelcast cluster errors need to be isolated

Use Lock.tryLock() (not Lock.lock()) in d1_replication

2012-01-09 22:13 - Chris Jones

Status:	Closed	Start date:	2012-01-09
Priority:	Normal	Due date:	
Assignee:	Chris Jones	% Done:	100%
Category:	d1_replication	Estimated time:	0.00 hour
Target version:	Sprint-2012.01-Block.1.1	Story Points:	
Milestone:	CCI-1.0.0		
Product Version:	*		
Description			
In order to avoid potential deadlocks in Hazelcast, convert calls to Lock.lock() to Lock.tryLock() and introduce a timeout for the try. Also introduce a queue structure to re-process operations that need to be re-tried if the lock fails.			

History

- #1 - 2012-01-09 22:13 - Chris Jones
- Subject changed from Use Lock.tryLock() (not Lock.lock()) d1_replication to Use Lock.tryLock() (not Lock.lock()) in d1_replication
- #2 - 2012-01-10 16:07 - Chris Jones
- Status changed from New to In Progress
- #3 - 2012-01-20 14:05 - Chris Jones
- Status changed from In Progress to Closed

I've created a ReplicationEventListener class that queues events for processing. entryAdded and entryUpdated events trigger a tryLock() on an event string based on the incoming identifier. Whichever CN instance gets the lock first will queue the identifier and createAndQueueTasks() will be called for it.