Infrastructure - Task #2178

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

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

2012-01-09 22:14 - Chris Jones

Status:	Rejected	**Start date:**	2012-01-09
Priority:	Normal	**Due date:**	
Assignee:	Chris Jones	**% Done:**	0%
Category:	Metacat	**Estimated time:**	0.00 hour
Target version:	Sprint-2012.01-Block.1.1		
Milestone:	CCI-1.0.0	**Story Points:**	
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 - 2012-01-10 16:07 - Chris Jones

- Status changed from New to In Progress

#2 - 2012-01-17 18:04 - Chris Jones

- Status changed from In Progress to Rejected

Using lock.tryLock() in Metacat would require a task-based queue for all API calls to the REST API implementation so that if the lock fails, the task would be queued and retried later. This would be a major change to the way Metacat handles D1 REST requests. Since only Metacat is locking the pid for a given object, the chance of lock contention is very low, and blocking locks should work fine. Rejecting this for now.