Metacat - Bug #6322

Ensure lock is obtained prior to calls to Hazelcast IMap.put()

12/20/2013 07:28 AM - Chris Jones

Status:	Rejected	**Start date:**	12/20/2013
Priority:	Normal	**Due date:**	
Assignee:	Chris Jones	**% Done:**	0%
Category:	metacat	**Estimated time:**	0.00 hour
Target version:	2.5.0	**Spent time:**	0.00 hour
Bugzilla-Id:			

Description

When modifying the system metadata table, we call systemMetadataMap.put(). since this is a distributed map in the HZ cluster, we need to call lock() first on the pid. This particularly needs to be fixed in DocumentImpl, but we should audit this call in other classes to make sure we lock and unlock correctly.

History

#1 - 01/31/2014 02:19 PM - ben leinfelder

I see quite a few hzSystemMetadata.put() calls without locks. And there are some comments like this:

```
// note: the calling subclass handles the map hazelcast lock/unlock
```

So I take it the thread that locks the pid has control of it, and any calls made further down the chain still have that lock? If that's the case, I think we have to be careful with our locking so that calls down the callstack don't try to lock a pid that was locked earlier in the stack...

#2 - 01/31/2014 03:18 PM - Matt Jones

It seems to me that we should protect against this by creating a wrapper function rather than calling hz.put() directly, where the wrapper function would 1) first check to see if the caller has the lock, and if not, tries to get a lock, and once it has a lock, then calls hz.put() for the caller. Then you wouldn't have to worry about who has the lock as much. We'd have to watch for deadlock scenarios and have a suitable lock release strategy, but this seems like it would be more reliable than just hoping that subclasses handle locks properly.

#3 - 02/07/2014 04:03 PM - ben leinfelder

- Target version changed from 2.4.0 to 2.5.0

#4 - 11/03/2015 11:28 AM - ben leinfelder

Do we have an example of something that is not working correctly because of locks or lack of locks in our code?

#5 - 11/03/2015 03:45 PM - Chris Jones

- Status changed from New to Rejected

Ben pointed out that we call lock() in the calling methods that invoke DocumentImpl methods (like CNodeService.create()). So, with that in mind, we don't need to lock pids during DocumentImpl calls. I'll reject this ticket.