CN REST - Bug #6836

Hazelcast "Current thread is not owner of lock!" error when calling CN create()

2015-02-03 23:19 - Andrei Buium

Status:	Closed	**Start date:**	2015-02-03
Priority:	Normal	**Due date:**	
Assignee:	Jing Tao	**% Done:**	100%
Category:	cn_metacat	**Estimated time:**	0.00 hour
Target version:			
Story Points:			

Description

Problem:

When calling the CN create() method to create a new piece of metadata, this exception is returned:
Service Failure: "Current thread is not owner of lock!"

How to reproduce:

The error can be reproduced by running through MultipartCNode.create() from the JUnit test CNAuthorizationV1IT.
(The context.cn.baseurl variable needs to be set to a valid CN. I used https://cn-sandbox-ucsb-1.test.dataone.org/cn)
testSetRightsHolder() and testIsAuthorized_*() methods will cause this error.

Details:

The error comes from edu.ucsb.nceas.metacat.dataone.CNodeService.java in the create() method.
It comes from the lock.unlock() in the finally block.

It could be that the lock.unlock() exception is hiding the actual error:
1) lock.lock() throws an exception
2) the exception is caught and in the catch block we throw the ServiceFailure with the correct message
3) in the finally block lock.unlock() throws the "Current thread is not owner of lock!" exception (because lock never got locked by this thread?)
4) the exception in step 2 is never seen or returned

History

#1 - 2015-02-03 23:21 - Andrei Buium

- Category set to cn_metacat

Changed category to cn_metacat

#2 - 2015-02-03 23:27 - Andrei Buium

- Assignee set to Jing Tao

Set Jing as the assignee so he knows about this ticket.

#3 - 2015-02-10 01:52 - Jing Tao

When I ran a new test case for the create method in my junit suite, I saw the error as well.
The code looks like:

```
try {

lock = HazelcastService.getInstance().getLock(pid.getValue());
//some code for checking if allowed
lock.lock();

} finally {
lock.unlock();
}
```

The problem is we didn't call lock.lock() immediately after getting the lock. So it can happen that the lock.unlock is called without calling lock.lock(). When I moved the code lock.lock() immediately after getLock(). The problem is gone.

I suspect you used a wrong certificate to call the create method.

#4 - 2015-02-10 01:53 - Jing Tao

- % Done changed from 0 to 30

- Status changed from New to In Progress

#5 - 2015-02-10 17:38 - Jing Tao

- Status changed from In Progress to Closed

- % Done changed from 30 to 100

When Andrei used a certificate with an CN subject, the creation method worked. The certificate with subject testRightsHolder doesn't work.