Infrastructure - Bug #2211

Use of d1_libclient_java may lead to open file descriptors

2012-01-12 01:24 - Robert Waltz

Status:	Closed	**Start date:**	2012-01-19
Priority:	Normal	**Due date:**	
Assignee:	Rob Nahf	**% Done:**	100%
Category:	d1_libclient_java	**Estimated time:**	0.00 hour
Target version:	Sprint-2012.03-Block.1.2		
Milestone:	CCI-1.0.0	**Story Points:**	
Product Version:	*		

Description

While reading some issues regarding JSolr, I came across a bug due to the use of Apache HttpClient:

https://issues.apache.org/jira/browse/SOLR-861

A comment by Jilles van Gurp on 10/Jun/10 14:24 may provide clues as to how to deal with issues.

We may wish to implement connection management of org.dataone.client.RestClient

Or implement pool management of d1_libclient_java in all components

and we may wish to create the DefaultHttpClient with a thread safe connection manager: ThreadSafeClientConnManager

Subtasks:

Task # 2234: ensure that all content input streams are consumed	**Closed**
Task # 2235: add method to D1RestClient that allows caller to close the connection	**Closed**
Task # 2236: refactor client methods to explicitly close the connection after consuming...	**Closed**
Task # 2237: refactor get() and getReplica() to buffer the input stream to allow connec...	**Closed**

Related issues:

Related to Java Client - Story #2246: refactor libclient_java connection mana...	**Closed**	**2012-07-09**	
Related to Infrastructure - Story #6494: Remove local caching of inputstreams...	**Closed**	**2014-11-10**	**2014-12-15**
Related to Infrastructure - Bug #5311: MultipartRequestResolver leaking resou...	**Closed**	**2014-05-02**	**2014-05-10**

History

#1 - 2012-01-12 01:26 - Robert Waltz

- Assignee deleted (Rob Nahf)

#2 - 2012-01-13 06:03 - Rob Nahf

- Assignee set to Rob Nahf

- Status changed from New to In Progress

The SOLR issue provided above is from 2008 and refers to httpClient 3.0.1, while RestClient uses httpClient 4.1.3. Jilles 2010 comment indicates that his project's issue due to improper connection release may have been addressed, and he was using httpClient 4.0. JIRA issue SOLR-2727

After a bit of research, I'm able to say confidently that the current RestClient / D1RestClient is using a thread safe connection manager ("SingleClientConnManager": http://hc.apache.org/httpcomponents-client-ga/httpclient/apidocs/org/apache/http/impl/conn/SingleClientConnManager.html).

While it is thread safe, it's intended for a single client context. Since there is a dichotomy of needs between ITK and CN usage of libclient, choice of connection managers could be configurable through a properties file. It's sister connection manager implementation is the above-mentioned ThreadSafeClientConnManager, which allows parallel open connections. (multiConnectionClientConnManager might have been a better name....)

At that point, we would be using a connection pool that would potentially hold open connections open longer, as does CommonsHttpSolrServer...

DefaultHttpClient is a nicely designed object, by the way, so my concern over the current client implementation releasing from the connection nicely was overstated. It wraps the response entity in BasicManagedEntity and wraps the input stream in EofSensorInputStream. The combination

releases a connection when the input stream is consumed, without having to manually call releaseConnection from the http method, or httpclient object.

One thing to find out is what happens when an unread input stream is dereferenced. We have several boolean returning methods that return empty input streams that the MNode or CNode method doesn't bother to read. We may have to or wish to close them manually.

#3 - 2012-01-13 06:26 - Rob Nahf

set up of SSL connection is currently time consuming, giving noticeable pause upon sslSetup. So, it would be beneficial to share these connections; that is, refactor the D1RestClient instances out of the methods and into a property of MNode / CNode / D1Node. (different RestClients are not sharing the same SingleClientConnManager)

#4 - 2012-01-17 15:45 - Dave Vieglais

- Position set to 1

#5 - 2012-01-19 17:06 - Rob Nahf

- % Done changed from 0 to 90

decision taken to postpone implementing any connection sharing, and simply close connections after each http call. Accordingly, I refactored the service api methods to do just that. Also squashed a bug that potentially contributed to the problem, whereby methods returning true or exception were not consuming the response content and thus not releasing connections. (setReplicationStatus was one of those).

One shortcoming of this approach is the inability to close connections on get() and getReplica(), where the calling application consumes the input stream. (consumption of the input stream does trigger a connection release, but not a connection close.)

Closing the connection after each method call is costly, and not the recommended way to do things by apache, so should be considered a short-term workaround. Multithreaded clients should be able to implement a ThreadSafeClientConnection, instead of the default one currently provided, and employ one of the connection management strategies described in the bug description.

#6 - 2012-01-19 22:33 - Rob Nahf

- Status changed from In Progress to Closed

closed this issue by solidifying the 1 http call per connection. This is not ideal, as opening and closing sockets to the same server is costly.

Will need to refactor libclient in the future to institute a connection monitoring thread (assuming that's still the strategy)