

Java Client - Bug #7675

Pooling connection manager leaves sockets in CLOSE_WAIT state

2016-03-15 02:58 - Rob Nahf

Status:	Closed	Start date:	2016-03-15
Priority:	Normal	Due date:	
Assignee:	Rob Nahf	% Done:	100%
Category:	d1_libclient_java	Estimated time:	0.00 hour
Target version:	CCI-2.2.0		
Story Points:			

Description

HttpClient (4.3), to support HTTP/1.1 doesn't close socket connections until it has evidence that the remote host has closed their side. Because we do not actively manager the connection manager (don't shut it down), due to our design that hides the httpclients, sockets remain open even after an application quits. Additionally, when a new connection manager replaces an old one, its associated sockets remain.

The number of sockets left in the CLOSE_WAIT maxes out at the total number of connections in the connection pool, times the number of httpclients built during application lifetime.

We need to proactively close stale connections, as typical apache servers close their side of the HTTPS connection after 5 seconds.

Associated revisions

Revision 17777 - 2016-03-15 04:50 - Rob Nahf

refs: #7675: added custom keep-alive strategy to ensure all connections expire; refactored idleConnectionsMonitorThread to HttpConnectionMonitorService, a singleton that owns a thread that monitors any number of ConnectionManagers. Added D1Client.http.keepAlaiveDuration.sec property. Tested thread monitoring with unit test.

Revision 17777 - 2016-03-15 04:50 - Rob Nahf

refs: #7675: added custom keep-alive strategy to ensure all connections expire; refactored idleConnectionsMonitorThread to HttpConnectionMonitorService, a singleton that owns a thread that monitors any number of ConnectionManagers. Added D1Client.http.keepAlaiveDuration.sec property. Tested thread monitoring with unit test.

History

#1 - 2016-03-15 05:04 - Rob Nahf

- Status changed from In Progress to Testing

- % Done changed from 30 to 50

added singleton HttpConnectionMonitorService that runs a single monitoring thread that monitors all registered ConnectionManagers. (HttpUtils registers ConnectionManagers upon creation).

The Monitor Service will shutdown all ConnectionManagers when receiving an interrupt.

#2 - 2016-04-01 17:28 - Rob Nahf

- Status changed from Testing to Closed

- % Done changed from 50 to 100

Augmented HttpUtils.getHttpClientBuilder method with Boolean parameter that will either register the connection manager with the monitor service, or not. This is to get around one-size-fits all configuration.

Also created a `AutoCloseHttpClientInputStream` class to shutdown the `httpClient` (releasing system resources) when the last byte is read from the associated input stream.

Metacat's replication system does not consume input streams right away, so need short-lived connection managers that don't need monitoring. Metacat is using both of these features now.