RBF: Router Connection Management

lfengnan@

<This doc would be shared with the Open Source Community for more feedback as well>

Context

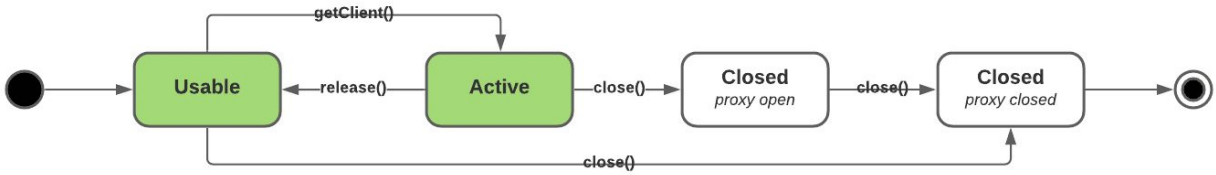
Router stands between HDFS clients and HDFS namenodes. It proxies clients' requests to namenodes and opens TCP connections to do so. To reduce the cost of frequently opening connections, it keeps a map (keyed by <ugi, namenode, protocol>) of connection pools and in each pool there are a certain number of connections. Each connection is a wrapper of RPC proxy. We have observed below problems related with Router's connection management and would like to propose a solution to it.

Problem

- High number of connections to namenodes during busy time, causing the latter to exhaust its open file descriptors.
- Low percentage of active connections among all connections.

Current state

After taking a deeper look at the current Router connection management, we believe that the problem lies in the connection (ConnectionContext.java) lifecycle. Current state machine of connection (closest to fact) is as below graph.



There are no clearly defined states in the code right now but three functions to indicate a connection's state. IsUsable(), IsActive() and IsClosed(). Some functions are used to drive the state change as described in the graph.

States:

- **Usable**: a connection is not closed and no handler is using it.
- **Active**: a connection is not closed and there is a handler using it.
- **Closed**: the closed flag is set in a connection. No more handlers can use this connection.

- ClosedProxyOpen: the RPC proxy is still there meaning the TCP connection is still open.
- ClosedProxyClosed: the RPC proxy is closed and no TCP connection is there. ***Only when the state is Closed Proxy Closed, there is no TCP connection to downstream namenodes.***

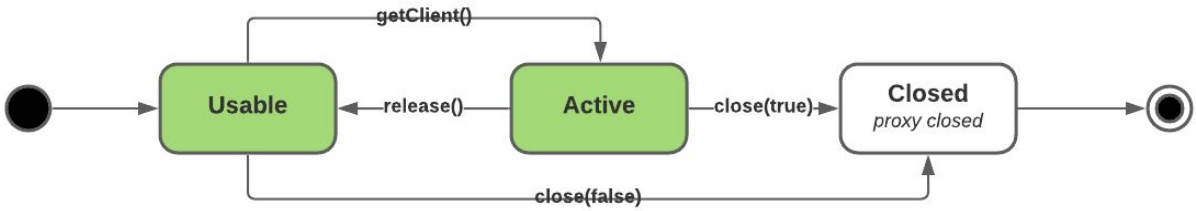
Normally a connection is switching state between Usable and Active. There is a scheduled job (30s as default period) to clean up the connections and will mark whatever selected connection as closed. (not necessarily close the RPC proxy)

The biggest problem is that when a selected connection is in Usable or Active state, it will be just transitioned into ClosedProxyOpen and wait for future cleanup. At this state, an actual TCP connection is still open, thus leaving Router keeping this unused connection for some time until this connection is picked again by the cleanup job to transition into ClosedProxyClosed. This is an effectively asynchronous connection closing.
Another problem is that the selection process is not effective enough. First of all, it is removing 1 connection at a time. Secondly, the picked connection is not necessarily at ClosedProxyClosed state, meaning the connection won't be closed for an undetermined period of time.
Furthermore, there is a potential connection leak when ConnectionManager is shutting down and all connections need to be closed gracefully. When the state is Active, it will just go into ClosedProxyClosed without the TCP connection closed. There will be no more cleanup runs since the whole ConnectionManager (even Router) process is shut down.

We believe the above factors drive the total number of connections high and the percentage of active connections low.

Proposal

We propose to make the state machine as below:



1. ClosedProxyOpen state should be removed and the closed state means no TCP connection.
2. Add a configurable value for the number of to be closed connection to control how many connections should be taken in a cleanup run. This will comply with the min active rate as well and a smaller value will be picked up.

3. Refine the picking method to make it only pick connections at Usable state, thus the connection would be closed immediately and synchronously.
4. Add a force bool parameter to the close method to close the proxy anyway. This is used when ConnectionManager is shutting down.