KIP-148: Add a connect timeout for client

- Status
- Motivation
- Public Interfaces
- Proposed Changes
- Compatibility, Deprecation, and Migration Plan
- Rejected Alternatives

Status

Current state:*"Under Discussion"*

Discussion thread: *here [Change the link from the KIP proposal email archive to your own email thread]*

JIRA: ⚠ Unable to render Jira issues macro, execution error.

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

Currently the kafka client will non-blocking connect to the broker, but if the broker’s OS system shuts down or crashes, and the client is connecting to that node, it will waste a long time for the client to be aware of the connection can’t be reached. If the client is connecting the node to update metadata it will not connect to another node unless the previous connection is closed, then the client can’t send or receive message until it is aware of the connection is broken and try other node. This time is determined by the TCP connection parameters in the environment, sometime maybe several minutes.

So we want to add a connecting timeout to the client in this situation, it will timeout in a specified time and try to connect to other node to update metadata to avoid this issue.

Public Interfaces

This KIP will add a configuration in the client’s configuration file(both producer and consumer):

connect.timeout.ms: the connecting time out of one connection, default is 5000.

Proposed Changes

The change is as follow:

(1)   Add a connecting time to the network client.

(2)   Whenever a connection is trying to connect, we save the connecting channel in a hash map.

(3)   At the end of every poll in the selector, we check if there are any connecting channels timeout, if so we will close these the connecting channels, then the client will try to select other node to connect.

(4)   If a channel is connected or closed, we will remove it from the connecting channel hash map.

.

Compatibility, Deprecation, and Migration Plan

The proposed change is backward compatible.

Rejected Alternatives

1. *Use connections.max.idle.ms to instead of connect.timeout.ms: it will close idle connections after the number of milliseconds specified by this config, default is 9 minutes*
   But this parameter is only work for the connection phase, if the channel is connecting, this parameter can not close the channel.

2. Enhanced request.timeout.ms to support the connecting timeout: we are using request.timeout.ms to control not only for the request(produce request or fetch request), but also for the connecting phase.
   If we only use the request.timeout for the connect and process request, it is not flexible for some case for example we need the connecting phase to a short time(5s), and the request for a longer time(40s).