1.0.0

Starting and Stopping Apache Impala

Date published: 2020-11-30
Date modified: 2025-10-22

CLOUDERA

https://docs.cloudera.com/

Legal Notice

Contents

Setting Timeouts in Impala

Depending on how busy your cluster is, you might increase or decrease various timeout values. Increase timeouts if Impala is cancelling operations prematurely, when the system is responding slower than usual but the operations are still successful if given extra time. Decrease timeouts if operations are idle or hanging for long periods, and the idle or hung operations are consuming resources and reducing concurrency.

Setting Timeout and Retries for Thrift Connections to Backend Client

Impala connections to the backend client are subject to failure in cases when the network is momentarily overloaded.

About this task

To avoid failed queries due to transient network problems, you can configure the number of Thrift connection retries using the following option:

Procedure

1. Log in to the CDP web interface and navigate to the Cloudera Data Warehouse service.
2. In the Cloudera Data Warehouse service, click Virtual Warehouses in the left navigation panel.
3. Select the Impala Virtual Warehouse, click options  for the warehouse you want to set the timeout and retry options.
4. Click Edit and navigate to Impala Coordinatorunder Configurationstab.
5. Using the + sign, specify the following if the options are not already added.

   To avoid failed queries due to transient network problems, you can configure the number of Thrift connection retries using the following option:

   - The --backend_client_connection_num_retries option specifies the number of times Impala will try connecting to the backend client after the first connection attempt fails. By default, `impalad` will attempt three re-connections before it returns a failure.

   You can configure timeouts for sending and receiving data from the backend client. Therefore, if for some reason a query does not respond, instead of waiting indefinitely for a response, Impala will terminate the connection after a configurable timeout.

   - The --backend_client_rpc_timeout_ms option can be used to specify the number of milliseconds Impala should wait for a response from the backend client before it terminates the connection and signals a failure. The default value for this property is 300000 milliseconds, or 5 minutes.
6. Click Apply and restart Impala.

Increasing StateStore Timeout

If you have an extensive Impala schema, for example, with hundreds of databases, tens of thousands of tables, you might encounter timeout errors during startup as the Impala catalog service broadcasts metadata to all the Impala nodes using the StateStore service. To avoid such timeout errors on startup, increase the StateStore timeout value from its default of 10 seconds.

About this task

Increase the timeout value of the StateStore service if you see messages in the impalad log such as:

```
Connection with state-store lost
Trying to re-register with state-store
```

Procedure

1. Log in to the CDP web interface and navigate to the Cloudera Data Warehouse service.
2. In the Cloudera Data Warehouse service, click Virtual Warehouses in the left navigation panel.
3. Select the Impala Virtual Warehouse, click options  for the warehouse you want to set the timeout and retry options.
4. Click Edit and navigate to Impala Coordinatorunder Configurationstab.
5. Using the + sign, specify the following if the option is not already added.

   Specify a new timeout value larger than the current value using the option StateStoreSubscriber Timeout.
6. Click Apply and restart Impala.

Setting the Idle Query and Idle Session Timeouts

To keep long-running queries or idle sessions from tying up cluster resources, you can set timeout intervals for both individual queries, and entire sessions.

About this task

Procedure

1. Log in to the CDP web interface and navigate to the Cloudera Data Warehouse service.
2. In the Cloudera Data Warehouse service, click Virtual Warehouses in the left navigation panel.
3. Select the Impala Virtual Warehouse, click options  for the warehouse you want to set the timeout and retry options.
4. Click Edit and navigate to Impala Coordinatorunder Configurationstab.
5. In the search field, type idle.
6. In the Idle Query Timeout field, specify the time in seconds after which an idle query is cancelled.

   This could be a query whose results were all fetched but was never closed, or one whose results were partially fetched and then the client program stopped requesting further results. This condition is most likely to occur in a client program using the JDBC or ODBC interfaces, rather than in the interactive `impala-shell` interpreter. Once a query is cancelled, the client program cannot retrieve any further results from the query.

   You can reduce the idle query timeout by using the QUERY_TIMEOUT_S query option at the query level. Any non-zero value specified in this field serves as an upper limit for the QUERY_TIMEOUT_S query option.

   The value of 0 disables query timeouts.

7. In the Idle Session Timeout field, specify the time in seconds after which an idle session expires.

   A session is idle when no activity is occurring for any of the queries in that session, and the session has not started any new queries. Once a session is expired, you cannot issue any new query requests to it. The session remains open, but the only operation you can perform is to close it.

   The default value of 0 specifies sessions never expire.

   You can override this setting with the IDLE_SESSION_TIMEOUT query option at the session or query level.
8. Click Apply and restart Impala.

Results

Impala checks periodically for idle sessions and queries to cancel. The actual idle time before cancellation might be up to 50% greater than the specified configuration setting. For example, if the timeout setting was 60, the session or query might be cancelled after being idle between 60 and 90 seconds.

Adjusting Heartbeat TCP Timeout Interval

Using the TCP flag, you can prevent the Statestore from waiting indefinitely for a response from the subscribers that fail to respond to the heartbeat RPC within the set period.

About this task

This flag statestore_heartbeat_tcp_timeout_seconds defines the time that may elapse before a heartbeat RPC connection request from a Statestore server to an Impalad or a Catalog server (subscribers) should be considered dead.

You can increase the flag value if you see intermittent heartbeat RPC timeouts listed in the statestore's log. You may find the max value of "statestore.priority-topic-update-durations" metric on the statestore to get an idea of a reasonable value to be used in this flag.

Note: The priority topic updates are only small amounts of data that take little time to process, similar to the heartbeat complexity.

Procedure

1. Log in to the CDP web interface and navigate to the Data Warehouse service.
2. In the Cloudera Data Warehouse service, click Virtual Warehouses in the left navigation panel.
3. Select the Impala Virtual Warehouse, click options  for the warehouse you want to adjust the Heartbeat TCP Timeout.
4. Click Edit and navigate to Impala Statestoredunder Configurationstab.
5. Using the + sign, specify the following if the options are not already added.
6. In the Add Customs Configuration field, add the flag statestore_heartbeat_tcp_timeout_seconds with an appropriate value.
7. You can also control the maximum number of consecutive heartbeat messages an impalad can miss before being declared failed by the statestore by adding this flag statestore_max_missed_heartbeats. Typically, you will not have to change this value.
8. Click Apply and restart Impala.

Setting Impala query cancellation on shut down

Learn how to configure Impala to cancel running queries before the shut down deadline, ensuring a controlled shut down and preventing resource leaks.

About this task

When the graceful shutdown deadline is reached, the Impala Daemon exits immediately, leaving queries unfinished. This can lead to unreleased resources, such as scratch files in remote storage.

To improve this process, a new state has been added to the graceful shut down procedure. Before reaching the shut down deadline, Impala attempts to cancel any remaining running queries within a configurable time limit, shutdown _query_cancel_period_s.

- The default value is 60 seconds.
- If the specified value exceeds 20% of the total shut down deadline, it is automatically capped at that value.
- If this value is set to 0, the shutdown query cancellation state is disabled, and Impala waits directly until the shut down deadline.

How query cancellation works during shut down

1. The shut down process is initiated.
2. Before the shutdown deadline, Impala enters a query cancellation phase if shut down query cancellation state is enabled.
3. Any remaining running queries are canceled within the shutdown_query_cancel_period_s duration.
4. If all queries are successfully canceled within this period, Impala shuts down immediately.
5. If queries are still running when the deadline is reached, Impala shuts down with those queries still active.

To configure query cancellation before the shut down deadline, follow these steps:

Procedure

1. Log in to the Cloudera web interface and navigate to the Cloudera Data Warehouse service.
2. In the Cloudera Data Warehouse service, click Virtual Warehouses in the left navigation panel.
3. Select the Impala Virtual Warehouse, click options  for the warehouse you want to set the query cancellation before the shut down deadline.
4. Click Edit and go to the Configurations tab.
5. To configure shut down query cancellation state for Coordinator.
   a) Go to Impala Coordinator.
   b) Under Configuration files, select the flagfile.
      If it is not already present with an appropriate value.
   c) Click on Add Custom Configuration.
   d) Add the flag shutdown_query_cancel_period_s with an appropriate value.
   e) Click Add.

6. To configure shut down query cancellation state for Coordinator.
   a) Go to Impala Executor.
   b) Under Configuration files, select the flagfile.
      If it is not already present with an appropriate value.
   c) Click on Add Custom Configuration.
   d) Add the flag shutdown_query_cancel_period_s with an appropriate value.
   e) Click Add.
7. Click Apply Changes and restart Impala.

Results

Configuring this setting, Impala attempts to cancel running queries before the shut down deadline, ensuring a more controlled shut down process.