Cloudera Runtime 7.2.18

Tuning Apache Impala

Date published: 2020-11-30
Date modified: 2024-03-25

https://docs.cloudera.com/

Legal Notice

Contents

Setting Up HDFS Caching

Set up HDFS caching with Impala for improved performance.

Before you begin

Decide how much memory to devote to the HDFS cache on each host. The total memory available for cached data is the sum of the cache sizes on all the hosts. By default, any data block is only cached on one host although you can cache a block across multiple hosts by increasing the replication factor.

Procedure

1. Enable or disable HDFS caching through Cloudera Manager using the configuration setting Maximum Memory Used for Caching for the HDFS service.

   This control sets the HDFS configuration parameter dfs_datanode_max_locked_memory, which specifies the upper limit of HDFS cache size on each node. Set up the HDFS caching for your Hadoop cluster.

   - All the other manipulation of the HDFS caching settings, such as what files are cached, is done through the command line, either Impala DDL statements or the Linux `hdfs cacheadmin` command.
2. Using the `hdfs cacheadmin` command, set up one or more pools owned by the same user as the `impalad` daemon (typically impala).

   For example:

   ```
   hdfs cacheadmin -addPool four_gig_pool -owner impala -limit 4000000000
   ```
3. Once HDFS caching is enabled and one or more pools are available, on the Impala side, you specify the cache pool name defined by the hdfs cacheadmin command in the Impala DDL statements that enable HDFS caching for a table or partition, such as CREATE TABLE ... CACHED IN *POOL* or ALTER TABLE ... SET CACHED IN *POOL*.
4. You can use `hdfs cacheadmin -listDirectives` to get a list of existing cache pools.
5. You can use `hdfs cacheadmin -listDirectives -stats` to get detailed information about the pools.

Setting up Data Cache for Remote Reads

When Impala compute nodes and its storage are not co-located, the network bandwidth requirement goes up as the network traffic includes the data fetch as well as the shuffling exchange traffic of intermediate results. To mitigate the pressure on the network, you can enable the compute nodes to cache the working set read from remote filesystems, such as, remote HDFS data node, S3, ABFS, ADLS.

About this task

To enable remote data cache as follows.

Procedure

1. In Cloudera Manager, navigate to Clusters Impala Service .
2. In the Configuration tab, select Enable Local Data Cache to enable the local Impala Daemon data cache for caching of remote reads.
3. In Impala Daemon Data Cache Directories, add the directories Impala Daemon will use for caching of remote read data.
4. In Impala Daemon Data Cache Per Directory Capacity, specify the maximum amount of local disk space Impala will use per daemon in each of the configured directrories for caching of remote read data.

5. Click Save Changes and restart the Impala service.

Configuring Dedicated Coordinators and Executors

Configure a dedicated coordinator and a dedicated executor roles to improve scalability of Impala.

Guidelines for Dedicated Coordinators and Executors

- Dedicated coordinator:
  - Should be on an edge node with no other services running on it.
  - Does not need large local disks but still needs some that can be used for Spilling.
  - Require at least the same or even larger memory allocation than executors.
- (Dedicated)Executors:
  - Should be co-located with DataNodes.
  - The number of hosts with dedicated executors typically increases as the cluster grows larger and handles more table partitions, data files, and concurrent queries.

Procedure

1. In Cloudera Manager, navigate to Clusters Impala Configuration Role Groups .
2. Click Create to create two role groups with the following values.
   a) Group for Coordinators
      1. Group Name: Coordinators
      2. Role Type: Impala Daemon
      3. Copy from:
         - Select Impala Daemon Default Group if you want the existing configuration gets carried over to the Coordinators.
         - Select None if you want to start with a blank configuration.

   b) Group for Executors
      1. Group Name: Executors
      2. Role Type: Impala Daemon
      3. Copy from:
         - Select Impala Daemon Default Group if you want the existing configuration gets carried over to the Executors.
         - Select None if you want to start with a blank configuration.
3. In the Role Groups page, click Impala Daemon Default Group.
   a) Select the set of nodes intended to be coordinators.
      1. Click Action for Selected and select Move To Different Role Group….
      2. Select the Coordinators.

   b) Select the set of nodes intended to be Executors.
      1. Click Action for Selected and select Move To Different Role Group….
      2. Select the Executors.
4. Click Configuration. In the search field, type Impala Daemon Specialization.
5. Click Edit Individual Values.
6. For Coordinators role group, select COORDINATOR_ONLY.
7. For Executors role group, select EXECUTOR_ONLY.
8. Click Save Changes and then restart the Impala service.

Related Information

Dedicated Coordinator

Caching intermediate results

Learn about how Impala uses the intermediate results cache to improve query performance and resource efficiency.

Impala processes every query independently, because intermediate results are discarded after a query completes, subsequent queries must recompute these results even if the underlying data remains unchanged.

Caching intermediate results improves latency for repetitive work and frees up resources for other queries.

Unlike query result caching, this method allows for more granular matching of shared work between similar queries.

Caching within the database provides the following benefits:

- Data awareness – The database has direct knowledge of modifications to base tables.
- Security – The database uses authorization information to safely share cached results among users with equivalent privileges.
- Consistency – The query planner detects tables changes and prevents the use of stale results.

No lag or staleness exists for query results.

The cache key incorporates all factors that can impact the query results, including information about the base tables and any query options. When any of those factors change, a new cache entry is generated. For example, if you ingest new data into a base table, the key changes.

Administrators do not need to manually refresh or invalidate cache entries.

When the cache reaches the quota, Impala evicts cache entries to make space for new entries. You can specify the eviction policy by using the --tuple_cache_eviction_policy startup flag.

The cache supports the following eviction policies:

- **Least Recently Used (LRU)** – This is the default policy.
- **Low Inter-reference Recency Set (LIRS)** – This is a scan-resistant policy with low performance overhead.

Configuring intermediate results caching

Learn about the configurations required to enable the intermediate results cache for Impala queries.

About this task

To use the intermediate results cache, you must configure the following settings. By default, these features are disabled.

Before you begin

If the intermediate results cache storage is shared with other elements, such as the data cache or scratch space, you might need to adjust existing quotas (for example, the --data_cache startup flag) to provide sufficient space.

Procedure

1. In Cloudera Manager, click Clusters Impala Configuration
2. Search for **Impala Daemon Command Line Argument Advanced Configuration Snippet (Safety Valve)**
3. --allow_tuple_caching=true --tuple_cache=[directory_path]:[capacity]

   Add the above flags.

Example:

```
--allow_tuple_caching=true --tuple_cache=/cache/impala:20GB
```

Note: The directory path must exist on the local filesystem of each Impala daemon (coordinator and executor)

4. If you must reduce the data cache to provide space for the intermediate results cache, update the --data_cache startup flag with the new capacity.
   a) Search for **Impala Daemon Default Query Options**
   b) Add the flag enable_tuple_cache=true;
5. Click Save Changes and restart the Impala service.

Results

Impala now stores intermediate query results in the specified local directory. Subsequent queries with matching plan fragments can retrieve data from the cache, which reduces execution time and resource consumption.

What to do next

You can monitor cache hits and performance by checking the Impala Query Profile. The profile displays metrics for tuple cache hits under the relevant plan nodes.