

Admission Control Configurations



The admission control feature is similar in some ways to the Cloudera Manager static partitioning feature, as well as the YARN resource management framework. These features can be used separately or together. This section describes some similarities and differences, to help you decide which combination of resource management features to use for Impala.

Admission control is a lightweight, decentralized system that is suitable for workloads consisting primarily of Impala queries and other SQL statements. It sets "soft" limits that smooth out Impala memory usage during times of heavy load, rather than taking an all-or-nothing approach that cancels jobs that are too resource-intensive.

Because the admission control system does not interact with other Hadoop workloads such as MapReduce jobs, you might use YARN with static service pools on CDH clusters where resources are shared between Impala and other Hadoop components. This configuration is recommended when using Impala in a *multitenant* cluster. Devote a percentage of cluster resources to Impala, and allocate another percentage for MapReduce and other batch-style workloads. Let admission control handle the concurrency and memory usage for the Impala work within the cluster, and let YARN manage the work for other components within the cluster. In this scenario, Impala's resources are not managed by YARN. The Impala admission control feature uses the same configuration mechanism as the YARN resource manager to map users to pools and authenticate them.

Admission control is an Impala feature that imposes limits on concurrent SQL queries, to avoid resource usage spikes and out-of-memory conditions on busy CDH clusters. The admission control feature lets you set an upper limit on the number of concurrent Impala queries and on the memory used by those queries. Any additional queries are queued until the earlier ones finish, rather than being cancelled or running slowly and causing contention. As other queries finish, the queued queries are allowed to proceed.

In CDH 5.7 / Impala 2.5 and higher, you can specify these limits and thresholds for each pool rather than globally. That way, you can balance the resource usage and throughput between steady well-defined workloads, rare resource-intensive queries, and ad-hoc exploratory queries.

In addition to the threshold values for currently executing queries, you can place limits on the maximum number of queries that are queued (waiting) and a limit on the amount of time they might wait before returning with an error. These queue settings let you ensure that queries do not wait indefinitely so that you can detect and correct "starvation" scenarios.

For another example, consider a memory-bound workload such as many large joins or aggregation queries. Each such query could briefly use many gigabytes of memory to process intermediate results. Because Impala by default cancels queries that exceed the specified memory limit, running multiple large-scale queries at once might require rerunning some queries that are cancelled. In this case, admission control improves the reliability and stability of the overall workload by only allowing as many concurrent queries as the overall memory of the cluster can accommodate.

One way to limit resource usage through admission control is to set an upper limit on the number of concurrent queries. This is the initial technique you might use when you do not have extensive information about memory usage for your workload. The settings can be specified separately for each dynamic resource pool.

Max Running Queries

Maximum number of concurrently running queries in this pool. The default value is unlimited for CDH 5.7 or higher. (optional) The maximum number of queries that can run concurrently in this pool. The default value is unlimited. Any queries for this pool that exceed Max Running Queries are added to the admission control queue until other queries finish. You can use MaxRunning Queries in the early stages of resource management, when you do not have extensive data about query memory usage, to determine if the cluster performs better overall if throttling is applied to Impala queries.

For a workload with many small queries, you typically specify a high value for this setting, or leave the default setting of "unlimited". For a workload with expensive queries, where some number of concurrent queries saturate the memory, I/O, CPU, or network capacity of the cluster, set the value low enough that the cluster resources are not overcommitted for Impala.

Once you have enabled memory-based admission control using other pool settings, you can still use Max Running Queries as a safeguard. If queries exceed either the total estimated memory or the maximum number of concurrent queries, they are added to the queue.

Max Queued Queries

Maximum number of queries that can be queued in this pool. The default value is 200 for CDH 5.3 or higher.

Queue Timeout

The amount of time, in milliseconds, that a query waits in the admission control queue for this pool before being canceled. The default value is 60,000 milliseconds. In the following cases, Queue Timeout is not significant, and you can specify a high value to avoid canceling queries unexpectedly:

- In a low-concurrency workload where few or no queries are queued
- In an environment without a strict SLA, where it does not matter if queries occasionally take longer than usual because they are held in admission control

You might also need to increase the value to use Impala with some business intelligence tools that have their own timeout intervals for queries.

In a high-concurrency workload, especially for queries with a tight SLA, long wait times in admission control can cause a serious problem. For example, if a query needs to run in 10 seconds, and you have tuned it so that it runs in 8 seconds, it violates its SLA if it waits in the admission control queue longer than 2 seconds. In a case like this, set a low timeout value and monitor how many queries are cancelled because of timeouts. This technique helps you to discover capacity, tuning, and scaling problems early, and helps avoid wasting resources by running expensive queries that have already missed their SLA.

If you identify some queries that can have a high timeout value, and others that benefit from a low timeout value, you can create separate pools with different values for this setting.

Clamp MEM_LIMIT Query Option

If this field is not selected, the MEM_LIMIT query option will not be bounded by the **Maximum Query Memory Limit** and the **Minimum Query Memory Limit** values specified for this resource pool. By default, this field is selected in CDH 6.1 and higher. The field is disabled if both **Minimum Query Memory Limit** and **Maximum Query Memory Limit** are not set.

For example, consider the following scenario:

- The cluster is running impalad daemons on five hosts.
- A dynamic resource pool has **Max Memory** set to 100 GB.
- The **Maximum Query Memory Limit** for the pool is 10 GB and **Minimum Query Memory Limit** is 2 GB. Therefore, any query running in this pool could use up to 50 GB of memory (**Maximum Query Memory Limit** * number of Impala nodes).
- Impala will execute varying numbers of queries concurrently because queries may be given memory limits anywhere between 2 GB and 10 GB, depending on the estimated memory requirements.
- For example, Impala may execute up to 10 small queries with 2 GB memory limits or two large queries with 10 GB memory limits because that is what will fit in the 100 GB cluster-wide limit when executing on five hosts.

- The executing queries may use less memory than the per-host memory limit or the **Max Memory** cluster-wide limit if they do not need that much memory. In general this is not a problem so long as you are able to execute enough queries concurrently to meet your needs.

You can combine the memory-based settings with the upper limit on concurrent queries settings discussed above. If either the maximum number of or the expected memory usage of the concurrent queries is exceeded, subsequent queries are queued until the concurrent workload falls below the threshold again.

Memory Limits and Admission Control

Each dynamic resource pool can have an upper limit on the cluster-wide memory used by queries executing in that pool.

Use the following settings to manage memory-based admission control.

Max Memory

The maximum amount of aggregate memory available across the cluster to all queries executing in this pool. This should be a portion of the aggregate configured memory for Impala daemons, which will be shown in the settings dialog next to this option for convenience. Setting this to a non-zero value enables memory based admission control.

Impala determines the expected maximum memory used by all queries in the pool and holds back any further queries that would result in **Max Memory** being exceeded.

If you specify **Max Memory**, you should specify the amount of memory to allocate to each query in this pool. You can do this in two ways:

- By setting **Maximum Query Memory Limit** and **Minimum Query Memory Limit**. This is preferred in CDH 6.1 / Impala 3.1 and greater and gives Impala flexibility to set aside more memory to queries that are expected to be memory hungry.
- By setting **Default Query Memory Limit** to the exact amount of memory that Impala should set aside for queries in that pool.

Minimum Query Memory Limit and Maximum Query Memory Limit

These two options determine the minimum and maximum per-host memory limit that will be chosen by Impala Admission control for queries in this resource pool. If set, Impala admission control will choose a memory limit between the minimum and maximum values based on the per-host memory estimate for the query. The memory limit chosen determines the amount of memory that Impala admission control will set aside for this query on each host that the query is running on. The aggregate memory across all of the hosts that the query is running on is counted against the pool's **Max Memory**.

Minimum Query Memory Limit must be less than or equal to **Maximum Query Memory Limit** and **Max Memory**.

Default Query Memory Limit

The default memory limit applied to queries executing in this pool when no explicit MEM_LIMIT query option is set. The memory limit chosen determines the amount of memory that Impala Admission control will set aside for this query on each host that the query is running on. The aggregate memory across all of the hosts that the query is running on is counted against the pool's **Max Memory**. This option is deprecated in CDH 6.1 / Impala 3.1 and higher and is replaced by **Maximum Query Memory Limit** and **Minimum Query Memory Limit**.

Do not set this if either **Maximum Query Memory Limit** or **Minimum Query Memory Limit** is set.