

Cloudera Runtime 7.2.17

## Phoenix Performance Tuning

Date published: 2020-02-29

Date modified: 2021-09-09

The Cloudera logo is displayed in a bold, orange, sans-serif font. The word "CLOUDERA" is written in all caps, with a stylized 'E' that has a horizontal bar extending to the right.

<https://docs.cloudera.com/>

# Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

# Contents

|                                |          |
|--------------------------------|----------|
| <b>Performance tuning.....</b> | <b>4</b> |
|--------------------------------|----------|

## Performance tuning

You can use the following configuration properties to tune Phoenix to work optimally on your cluster. You can tune your Phoenix deployment by configuring certain Phoenix specific properties that are configured both on the client and server side hbase-site.xml files. For a full list of Phoenix Tuning properties that are available, see the [Apache Phoenix tuning guide](#).

**Table 1:**

| Property                                | Description   | Default    |
|---|---|------------|
| phoenix.query.threadPoolSize            | The number of threads in client-side thread pool executor. As the number of machines/cores in the cluster grows, this value should be increased.  | 128        |
| phoenix.query.queueSize                 | Max queue depth of the bounded round robin backing the client side thread pool executor, beyond which an attempt to queue additional work is rejected. If zero, a SynchronousQueue is used instead of the bounded round-robin queue. The default value is 5000.   | 5000       |
| phoenix.stats.guidepost.width           | The server-side parameter that specifies the number of bytes between guideposts. A smaller amount increases parallelization, but also increases the number of chunks which must be merged on the client side. The default value is 100 MB.  | 104857600  |
| phoenix.stats.guidepost.per.region      | The server-side parameter that specifies the number of guideposts per region. If set to a value greater than zero, then the guidepost width is determined by MAX_FILE_SIZE of table/phoenix .stats.guidepost.per.region. Otherwise, if not set, then the phoenix.stats.guidepost.width parameter is used. No default value.   | None       |
| phoenix.stats.updateFrequency           | The server-side parameter that determines the frequency in milliseconds for which statistics will be refreshed from the statistics table and subsequently used by the client. The default value is 15 min.  | 900000     |
| phoenix.query.spoolThresholdBytes       | Threshold size in bytes after which results from parallelly executed query results are spooled to disk. Default is 20 mb.   | 20971520   |
| phoenix.query.maxSpoolToDiskBytes       | Threshold size in bytes up to which results from parallelly executed query results are spooled to disk above which the query will fail. Default is 1 GB.  | 1024000000 |
| phoenix.query.maxGlobalMemoryPercentage | Percentage of total heap memory (i.e. Runtime.getRuntime().maxMemory()) that all threads may use. Only coarse grain memory usage is tracked, mainly accounting for memory usage in the intermediate map built during group by aggregation. When this limit is reached, the clients' block attempting to get more memory, essentially throttling memory usage. Defaults to 15% | 15         |

| Property                                       | Description  | Default   |
|--|--|-----------|
| phoenix.query.maxGlobalMemorySize              | Max size in bytes of total tracked memory usage. By default it is not specified, however, if present, the lower of this parameter and the phoenix.query.maxGlobalMemoryPercentage will be used.  |           |
| phoenix.query.maxGlobalMemoryWaitMs            | The maximum amount of time that a client will block while waiting for more memory to become available. After this amount of time, an InsufficientMemoryException error is displayed. The default value is 10 seconds.  | 10000     |
| phoenix.query.maxTenantMemoryPercentage        | The maximum percentage of phoenix.query.maxGlobalMemoryPercentage that a tenant is allowed to consume. After this percentage, an InsufficientMemoryException error is displayed. Default is 100%   | 100       |
| phoenix.mutate.maxSize                         | The maximum number of rows that may be batched on the client before a commit or rollback must be called.   | 500000    |
| phoenix.mutate.batchSize                       | The number of rows that are batched together and automatically committed during the execution of an UPSERT SELECT or DELETE statement. This property may be overridden at connection time by specifying the UpsertBatchSize property value. Note that the connection property value does not affect the batch size used by the coprocessor when these statements are executed completely on the server side. | 1000      |
| phoenix.query.maxServerCacheBytes              | Maximum size (in bytes) of a single subquery result (usually the filtered result of a table) before compression and conversion to a hash map. Attempting to hash an intermediate subquery result of a size bigger than this setting will result in a MaxServerCacheSizeExceededException. Default 100MB.   | 104857600 |
| phoenix.coprocessor.maxServerCacheTimeToLiveMs | Maximum living time (in milliseconds) of server caches. A cache entry expires after this amount of time has passed since last access. Consider adjusting this parameter when a server-side IOException("Could not find a hash cache for joinId") happens. Getting warnings like "Earlier hash cache(s) might have expired on servers" might also be a sign that this number should be increased.             | 30000     |
| phoenix.query.useIndexes                       | Client-side property determining whether or not indexes are considered by the optimizer to satisfy a query. Default is true  | true      |
| phoenix.index.failure.handling.rebuild         | Server-side property determining whether or not a mutable index is rebuilt in the background in the event of a commit failure. Only applicable for indexes on mutable, non-transactional tables. Default is true   | true      |
| phoenix.groupby.maxCacheSize                   | Size in bytes of pages cached during GROUP BY spilling. Default is 100Mb   | 102400000 |
| phoenix.groupby.estimatedDistinctValues        | Number of estimated distinct values when a GROUP BY is performed. Used to perform initial sizing with the growth of 1.5x each time reallocation is required. Default is 1000   | 1000      |

| Property                                       | Description   | Default  |
|--|---|----------|
| phoenix.distinct.value.compress.threshold      | Size in bytes beyond which aggregate operations which require tracking distinct value counts (such as COUNT DISTINCT) will use Snappy compression. Default is 1Mb   | 1024000  |
| phoenix.index.maxDataFileSizePerc              | The percentage used to determine the MAX_FILESIZE for the shared index table for views relative to the data table MAX_FILESIZE. The percentage should be estimated based on the anticipated average size of a view index row versus the data row. Default is 50%. | 50       |
| phoenix.coprocessor.maxMetaDataCacheTimeToLive | Time in milliseconds after which the server-side metadata cache for a tenant will expire if not accessed. Default is 30mins   | 180000   |
| phoenix.coprocessor.maxMetaDataCacheSize       | Max size in bytes of total server-side metadata cache after which evictions will begin to occur based on least recent access time. Default is 20Mb  | 20480000 |
| phoenix.client.maxMetaDataCacheSize            | Max size in bytes of total client-side metadata cache after which evictions will begin to occur based on least recent access time. Default is 10Mb  | 10240000 |
| phoenix.sequence.cacheSize                     | Number of sequence values to reserve from the server and cache on the client when the next sequence value is allocated. Only used if not defined by the sequence itself. Default is 100   | 100      |