• You can configure how InnoDB preserves the current buffer pool state to avoid a lengthy warmup period after a server restart. For details, see Section 15.8.3.6, “Saving and Restoring the Buffer Pool State” .

**Monitoring** **the** **Buffer** **Pool** **Using** **the** **InnoDB** **Standard** **Monitor**

InnoDB Standard Monitor output, which can be accessed using SHOW ENGINE INNODB STATUS, provides metrics regarding operation of the buffer pool. Buffer pool metrics are located in the BUFFER POOL AND MEMORY section of InnoDB Standard Monitor output:

----------------------

BUFFER POOL AND MEMORY

----------------------

Total large memory allocated 2198863872

Dictionary memory allocated 776332

Buffer pool size

Free buffers

Database pages

Old database pages

Modified db pages

Pending reads 0

Pending writes: LRU 0, flush list 0, single page 0

Pages made young 4, not young 0

0.10 youngs/s, 0.00 non-youngs/s

Pages read 197, created 5523, written 5060

0.00 reads/s, 190.89 creates/s, 244.94 writes/s

Buffer pool hit rate 1000 / 1000, young-making rate 0 / 1000 not

0 / 1000

Pages read ahead 0 .00/s, evicted without access 0 .00/s, Random read

ahead 0 .00/s

LRU len: 5720, unzip\_LRU len: 0

I/O sum[0]:cur[0], unzip sum[0]:cur[0]

131072

124908

5720

2071

910

The following table describes buffer pool metrics reported by the InnoDB Standard Monitor.

Per second averages provided in InnoDB Standard Monitor output are based on the elapsed time since InnoDB Standard Monitor output was last printed.

**Table** **15.2** **InnoDB** **Buffer** **Pool** **Metrics**

|  |  |
| --- | --- |
| **Name** | **Description** |
| Total memory allocated | The total memory allocated for the buffer pool in bytes. |
| Dictionary memory allocated | The total memory allocated for the InnoDB data dictionary in bytes. |
| Buffer pool size | The total size in pages allocated to the buffer pool. |
| Free buffers | The total size in pages of the buffer pool free list. |
| Database pages | The total size in pages of the buffer pool LRU list. |
| Old database pages | The total size in pages of the buffer pool old LRU sublist. |
| Modified db pages | The current number of pages modified in the buffer pool. |
| Pending reads | The number of buffer pool pages waiting to be read into the buffer pool. |
| Pending writes LRU | The number of old dirty pages within the buffer pool to be written from the bottom of the LRU list. |
| Pending writes flush list | The number of buffer pool pages to be flushed during checkpointing. |
| Pending writes single page | The number of pending independent page writes within the buffer pool. |

|  |  |
| --- | --- |
| **Name** | **Description** |
| Pages made young | The total number of pages made young in the buffer pool LRU list (moved to the head of sublist of “new” pages). |
| Pages made not young | The total number of pages not made young in the buffer pool LRU list (pages that have remained in the “old” sublist without being made young). |
| youngs/s | The per second average of accesses to old pages in the buffer pool LRU list that have resulted in making pages young. See the notes that follow this table for more information. |
| non-youngs/s | The per second average of accesses to old pages in the buffer pool LRU list that have resulted in not making pages young. See the notes that follow this table for more information. |
| Pages read | The total number of pages read from the buffer pool. |
| Pages created | The total number of pages created within the buffer pool. |
| Pages written | The total number of pages written from the buffer pool. |
| reads/s | The per second average number of buffer pool page reads per second. |
| creates/s | The average number of buffer pool pages created per second. |
| writes/s | The average number of buffer pool page writes per second. |
| Buffer pool hit rate | The buffer pool page hit rate for pages read from the buffer pool vs from disk storage. |
| young-making rate | The average hit rate at which page accesses have resulted in making pages young. See the notes that follow this table for more information. |
| not (young-making rate) | The average hit rate at which page accesses have not resulted in making pages young. See the notes that follow this table for more information. |
| Pages read ahead | The per second average of read ahead  operations. |
| Pages evicted without access | The per second average of the pages evicted without being accessed from the buffer pool. |
| Random read ahead | The per second average of random read ahead operations. |
| LRU len | The total size in pages of the buffer pool LRU list. |
| unzip\_LRU len | The length (in pages) of the buffer pool  unzip\_LRU list. |
| I/O sum | The total number of buffer pool LRU list pages accessed. |
| I/O cur | The total number of buffer pool LRU list pages accessed in the current interval. |

|  |  |
| --- | --- |
| **Name** | **Description** |
| I/O unzip sum | The total number of buffer pool unzip\_LRU list pages decompressed. |
| I/O unzip cur | The total number of buffer pool unzip\_LRU list pages decompressed in the current interval. |

**Notes**:

• The youngs/s metric is applicable only to old pages. It is based on the number of page accesses. There can be multiple accesses for a given page, all of which are counted. If you see very low youngs/s values when there are no large scans occurring, consider reducing the delay time or increasing the percentage of the buffer pool used for the old sublist. Increasing the percentage makes the old sublist larger so that it takes longer for pages in that sublist to move to the tail, which increases the likelihood that those pages are accessed again and made young. See Section 15.8.3.3, “Making the Buffer Pool Scan Resistant” .

• The non-youngs/s metric is applicable only to old pages. It is based on the number of page accesses. There can be multiple accesses for a given page, all of which are counted. If you do not see a higher non-youngs/s value when performing large table scans (and a higher youngs/s value), increase the delay value. See Section 15.8.3.3, “Making the Buffer Pool Scan Resistant” .

• The young-making rate accounts for all buffer pool page accesses, not just accesses for pages in the old sublist. The young-making rate and not rate do not normally add up to the overall buffer pool hit rate. Page hits in the old sublist cause pages to move to the new sublist, but page hits in the new sublist cause pages to move to the head of the list only if they are a certain distance from the head.

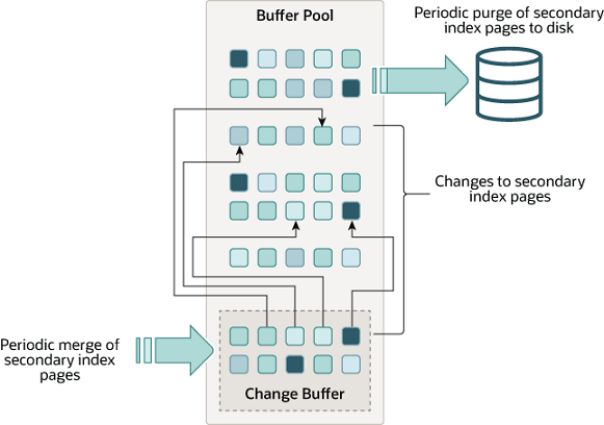
• not (young-making rate) is the average hit rate at which page accesses have not resulted in making pages young due to the delay defined by innodb\_old\_blocks\_time not being met, or due to page hits in the new sublist that did not result in pages being moved to the head. This rate accounts for all buffer pool page accesses, not just accesses for pages in the old sublist.

Buffer pool server status variables and the INNODB\_BUFFER\_POOL\_STATS table provide many of the same buffer pool metrics found in InnoDB Standard Monitor output. For more information, see Example 15.10, “Querying the INNODB\_BUFFER\_POOL\_STATS Table” .

**15.5.2** **Change** **Buffer**

The change buffer is a special data structure that caches changes to secondary index pages when those pages are not in the buffer pool. The buffered changes, which may result from INSERT, UPDATE, or DELETE operations (DML), are merged later when the pages are loaded into the buffer pool by other read operations.

**Figure** **15.3** **Change** **Buffer**



Unlike clustered indexes, secondary indexes are usually nonunique, and inserts into secondary indexes happen in a relatively random order. Similarly, deletes and updates may affect secondary index pages that are not adjacently located in an index tree. Merging cached changes at a later time, when affected pages are read into the buffer pool by other operations, avoids substantial random access I/O that would be required to read secondary index pages into the buffer pool from disk.

Periodically, the purge operation that runs when the system is mostly idle, or during a slow shutdown, writes the updated index pages to disk. The purge operation can write disk blocks for a series of index values more efficiently than if each value were written to disk immediately.

Change buffer merging may take several hours when there are many affected rows and numerous secondary indexes to update. During this time, disk I/O is increased, which can cause a significant slowdown for disk-bound queries. Change buffer merging may also continue to occur after a transaction is committed, and even after a server shutdown and restart (see Section 15.21.3, “Forcing InnoDB Recovery” for more information).

In memory, the change buffer occupies part of the buffer pool. On disk, the change buffer is part of the system tablespace, where index changes are buffered when the database server is shut down.

The type of data cached in the change buffer is governed by the innodb\_change\_buffering variable. For more information, see [Configuring Change Buffering](#_bookmark1). You can also configure the maximum change buffer size. For more information, see [Configuring the Change Buffer Maximum Size](#_bookmark2).

Change buffering is not supported for a secondary index if the index contains a descending index column or if the primary key includes a descending index column.

For answers to frequently asked questions about the change buffer, see Section A.16, “MySQL 8.0 FAQ: InnoDB Change Buffer” .

**Configuring** **Change** **Buffering**

When INSERT, UPDATE, and DELETE operations are performed on a table, the values of indexed columns (particularly the values of secondary keys) are often in an unsorted order, requiring substantial I/O to bring secondary indexes up to date. The change buffer caches changes to secondary index entries when the relevant page is not in the buffer pool, thus avoiding expensive I/O operations by not immediately reading in the page from disk. The buffered changes are merged when the page is loaded into the buffer pool, and the updated page is later flushed to disk. The InnoDB main thread merges buffered changes when the server is nearly idle, and during a slow shutdown.

Because it can result in fewer disk reads and writes, change buffering is most valuable for workloads that are I/O-bound; for example, applications with a high volume of DML operations such as bulk inserts benefit from change buffering.

However, the change buffer occupies a part of the buffer pool, reducing the memory available to cache data pages. If the working set almost fits in the buffer pool, or if your tables have relatively few secondary indexes, it may be useful to disable change buffering. If the working data set fits entirely within the buffer pool, change buffering does not impose extra overhead, because it only applies to pages that are not in the buffer pool.

The innodb\_change\_buffering variable controls the extent to which InnoDB performs change buffering. You can enable or disable buffering for inserts, delete operations (when index records are initially marked for deletion) and purge operations (when index records are physically deleted). An update operation is a combination of an insert and a delete. The default innodb\_change\_buffering value is all.

Permitted innodb\_change\_buffering values include:

• **all**

The default value: buffer inserts, delete-marking operations, and purges.

• **none**

Do not buffer any operations.

• **inserts**

Buffer insert operations.

• **deletes**

Buffer delete-marking operations.

• **changes**

Buffer both inserts and delete-marking operations.

• **purges**

Buffer the physical deletion operations that happen in the background.

You can set the innodb\_change\_buffering variable in the MySQL option file (my.cnf or my.ini) or change it dynamically with the SET GLOBAL statement, which requires privileges sufficient to set global system variables. See Section 5.1.9.1, “System Variable Privileges” . Changing the setting affects the buffering of new operations; the merging of existing buffered entries is not affected.

**Configuring** **the** **Change** **Buffer** **Maximum** **Size**

The innodb\_change\_buffer\_max\_size variable permits configuring the maximum size of the change buffer as a percentage of the total size of the buffer pool. By default, innodb\_change\_buffer\_max\_size is set to 25. The maximum setting is 50.