

duration on different processor architectures. For more information, see [Section 15.8.8, “Configuring Spin Lock Polling”](#).

- [InnoDB](#) parallel read thread performance for large data sets was improved in MySQL 8.0.17 through better utilization of read threads, through a reduction in read thread I/O for prefetch activity that occurs during parallel scans, and through support for parallel scanning of partitions.

The parallel read thread feature is controlled by the `innodb_parallel_read_threads` variable. The maximum setting is now 256, which is the total number of threads for all client connections. If the thread limit is reached, connections fall back to using a single thread.

- The `innodb_idle_flush_pct` variable, introduced in MySQL 8.0.18, permits placing a limit on page flushing during idle periods, which can help extend the life of solid state storage devices. See [Limiting Buffer Flushing During Idle Periods](#).
- Efficient sampling of [InnoDB](#) data for the purpose of generating histogram statistics is supported as of MySQL 8.0.19. See [Histogram Statistics Analysis](#).
- As of MySQL 8.0.20, the doublewrite buffer storage area resides in doublewrite files. In previous releases, the storage area resided in the system tablespace. Moving the storage area out of the system tablespace reduces write latency, increases throughput, and provides flexibility with respect to placement of doublewrite buffer pages. The following system variables were introduced for advanced doublewrite buffer configuration:

- `innodb_doublewrite_dir`

Defines the doublewrite buffer file directory.

- `innodb_doublewrite_files`

Defines the number of doublewrite files.

- `innodb_doublewrite_pages`

Defines the maximum number of doublewrite pages per thread for a batch write.

- `innodb_doublewrite_batch_size`

Defines the number of doublewrite pages to write in a batch.

For more information, see [Section 15.6.4, “Doublewrite Buffer”](#).

- The Contention-Aware Transaction Scheduling (CATS) algorithm, which prioritizes transactions that are waiting for locks, was improved in MySQL 8.0.20. Transaction scheduling weight computation is now performed a separate thread entirely, which improves computation performance and accuracy.

The First In First Out (FIFO) algorithm, which had also been used for transaction scheduling, was removed. The FIFO algorithm was rendered redundant by CATS algorithm enhancements.