Query Logging

ClustrixDB logs detailed information about significant and problematic queries. These logs are helpful to determine such things as:

- Slow queries
- Resource contention
- SQL errors
- Queries that read an unexpected number of rows
- Schema changes
- Global variable modifications
- Cluster alterations

By default, query logging is enabled and logs are stored in /data/clustrix/log/.

Each node will log information for the queries it runs while serving as the Global Transaction Manager (GTM). It is often necessary to consolidate logs from all nodes to assess cluster-wide issues. Use clx logdump to consolidate and assess logs.

Managing Query Logging

Query Types

Each entry in the query.log is categorized as one of these types. Specific logging for each query type is controlled by the global or session variable indicated.

Query Type	Description
ALTER CLUSTER	Changes made to your cluster via the ALTER CLUSTER command are always logged to the query.log automatically. This logging is not controlled by a global variable.
BAD	The query reads more rows than necessary to return the expected results. This may indicate a bad plan or missing index. Logging of BAD queries is not enabled by default (session_log_bad_queries).
DDL	The query is DDL (i.e. schema change such as CREATE, DROP, ALTER), or a SET GLOBAL or SESSION command. All DDL queries are initially logged by default (session_log_ddl).
SLOW	Query execution time exceeded the threshold specified by the variable session_log_slow_threshold_ms.
SQLERR	These database errors are things such as syntax errors, timeout notifications, and permission issues. All SQLERR queries will be logged by default (session_log_error_queries).

Global and Session Variables

Some of the variables used to control query logging may be specified by session and some are only available system-wide. To set the value for any of the variables associated with logging, use the following syntax:

```
SET [GLOBAL | SESSION]
variable_name = desired
_value;
```

These are the variables that control query and user logging. The defaults shown are generally acceptable for most installations.

Name	Description	Default Value	Session Variable
session_log_bad_queries	Log BAD queries to the query.log	false	✓
session_log_ddl	Log DDL statements to query.log	true	
session_log_error_queries	Log ERROR statements to query.log	true	
session_log_slow_queries	Log SLOW statements to query.log	true	

session_log_slow_threshold_ms	Query duration threshold in milliseconds before logging this query	10000	✓
session_log_users	Log LOGIN/LOGOUT to user.log	false	

See User Logging for additional information on session_log_users.

Reading the query.log

The Components

Each log entry begins with identifying data and includes important information to aid in troubleshooting problems on your cluster. Here is the layout of a log entry.

```
[timestamp] [hostname] clxnode INSTR [query type] [sid] [db] [user] [ac] [xid] [sql] [status] [time and breakdowns] [internal counters]
```

Identifying Information

Label	Description
timestamp	Date and time of the log entry, including the time zone. It is extremely important to have the clocks synchronized on all nodes.
hostname	Node ID and name of the host on which the entry was logged. This node served as the GTM for this transaction.
process name	The ClustrixDB process name (clxnode).
INSTR	This fixed verbiage appears before the query type in each row.
query type	Identifies the problem query: SLOW, DDL, BAD, SQLERR, ALTER CLUSTER.
SID	Session ID. Useful for grouping activity for a given session.
db	Name of the database on which the query was run.
user	User executing the query. If using statement-based replication, search for the replication account when troubleshooting statements from the master.
ac	Auto-commit indicator (Y/N). This is useful to determine if the query was used within a user-defined explicit transaction. DDL uses internally generated explicit transactions and will always be N.
xid	Transaction ID. Useful to link a session to an XID when troubleshooting locking issues.
sql	This is the text of the full query. Ellipses indicate the text has been truncated to fit within the 4KB limit.
status	Result of the query contained in brackets. For example, this could be rows affected or an error message.
time	Total elapsed time from when the query was received, compiled, and processed, to when output is returned or an error occurred. This is especially useful in analyzing SLOW queries.
Elapsed time is further broken down for any query that takes longer than one ms to execute.	
translate	Time spent in translate_dml().
prefetch	Time spent building the Sierra stub.
plan	Time spent to plan and normalize the query.
compile	Time spent in compiling Sierra.
execute	Time spent in invocation.

Internal Counters

Label	Description
reads	The number of times the database reads from a container. This may differ from the number of rows_read.
inserts	The number of times the database inserts into a container. This includes both the number of calls and the number of rows written.
deletes	The number of times the database deletes from a container. This includes both the number of calls and the number of rows deleted.

updates	The number of times the database updates a container. This includes both the number of calls and the number of rows updated.
counts	Number of calls by the query execution engine to operators BARRIER_ADD and BARRIER_FETCHADD.
rows_read	Total number of rows read to get all needed data for the query, including reads from indices. Essentially, the total number of rows processed by the last query. This may differ from from the number of rows_output by the query.
forwards	Number of rows forwarded to specific nodes.
broadcasts	Number of rows that were broadcast to all nodes.
rows_output	Total number of rows returned or output by the last query. This is usually the same as the number of rows returned from a query but may occasionally contain counts from internal processes.
semaphore_matches	Number of calls by the query execution engine to operator SEM_ACQUIRE.
fragment_executions	Number of query fragments executed for the query.
cpu_runtime_ns	This represents the aggregate total CPU time spent by all nodes to run the query.
cpu_waits	The number of times the query waited for another query to finish due to the Fair Scheduler.
cpu_waittime_ns	The amount of time spent waiting for CPU due to the Fair Scheduler.
barriers	Number of barriers created for the query. This is used to synchronize message communication between nodes.
barrier_forwards	Number of barriers created to synchronize messaging for forwarded rows.
barrier_flushes	Number of flush operations performed on barriers.
bm_fixes	Number of attempted page fixes by the Buffer Manager.
bm_loads	Number of pages loaded from disk by the Buffer Manager.
bm_waittime_ns	Nanoseconds spent blocked on Buffer Manager page fixes.
lockman_waits	Count of the number of times that the query had to wait for a lock to be released by another query.
lockman_waittime_ms	The total time spent waiting for other queries to release locks on needed rows.
trxstate_waits	Number of calls to trxstate_check that had to block.
trxstate_waittime_ms	Milliseconds spent blocked in trxstate_check.
wal_perm_waittime_ms	Milliseconds spent waiting because the WAL is more than 75% full.
bm_perm_waittime_ms	Milliseconds spent waiting for the Buffer Manager to grant write permission for pages.
sigmas	The number of sigma containers used by the query.
sigma_fallbacks	The number of sigma containers that ran out of memory and had to fall back to disk.
row_count	The total number of rows updated, inserted or deleted by the last query.
found_rows	The number of rows affected by the last statement, but not necessarily output by that statement . A value of 0 or -1 means no rows were found.
insert_id	Not currently being used, always displayed as 0.
fanout	Y/N indicator that tells if fanout was used for this query.
attempts	Number of attempts to automatically retry the query execution after it failed.

When ClustrixDB logs queries to query.log, the semicolon is stripped off. This means that any comments that are included with the statement are not logged.