Fanout

ClustrixDB is able to use multiple CPUs for query processing and operations like ALTER TABLE. This process is called Fanout.

When ClustrixDB executes a query, it is split into fragments for execution. These fragments are then passed to the various CPUs for execution. Depending on the query type and complexity, each fragment can have multiple stages of execution.

Fanout is the process by which fragments of a query are run simultaneously on multiple CPUs per node. This allows for expanded parallelism and much shorter execution times for ALTER TABLE and large OLAP queries.

Detailed information about query execution and fragmentation can be found in the documentation on the ClustrixDB Evaluation Model.

Variables that control Fanout

Fanout is controlled and configured by the following variables. Default values are appropriate for most workloads.

Name	Description	Default Value	Session Variable
Fanout Variables			
query_fanout	Enables query fanout. This takes precedence over all other fanout variables.	True	✓
query_fanout_insert_select	If enabled, INSERT INTO ... SELECT FROM ... queries will use Fanout. This takes precedence over query_fanout_all_writes.	True	✓
query_fanout_all_writes	Enable fanout for INSERT, UPDATE, and DELETE queries. Order of writes is not guaranteed.	False	✓
query_fanout_min_rows	Only queries expected to read more than this many rows will fan out.	5000	✓
Related Variables			
gtm_schedule_til	Enable the Completely Fair Scheduler.	True	
sigma_skiplist	Enable skiplist containers for aggregates and sorting.	False	

Disable Fanout

This will completely disable Fanout and each query will use only one CPU. This will affect performance and this setting should only be changed if advised by Clustrix Support.

```
sql> SET query_fanout = FALSE;
```

Table Slicing Guidelines for Optimal Fanout

To provide full parallelism with Fanout, ensure that the representations queried have enough slices. Large tables that are expected to benefit from Fanout should be sliced to the total number of CPU cores licensed for the cluster. Under-slicing reduces parallelism. For example, if three slices of a representation are present on a node, then Fanout is limited to running a maximum of three query fragments on three CPUs of that node.

Fanout Caveat

Fanout is not available for temporary tables. Temporary tables contain only one slice and reside on only one node.