



TokuDB Evaluation Guide

Overview

TokuDB's Fractal Tree indexing algorithms turn random IO into sequential IO, providing substantial performance gains for IO bound workloads. By making IO operations sequential, TokuDB can maintain a rich set of covering and clustering indexes, enabling fast queries to improve application performance and responsiveness. TokuDB typically helps applications with the following characteristics:

- Tables with more than 50M to 100M rows
- Substantial insert, update or delete requirements - TokuDB scales to tens of thousands of rows per second
- A desire to query new data in "interactive time," without waiting hours or days for batch insertions
- "Range queries" - queries that must scan multiple rows and benefit from a rich set of indexes.

This guide outlines a typical evaluation and describes the steps necessary to get a meaningful assessment of TokuDB's benefits for your applications.

Evaluation Process

Initial Data Load

After installation, the first step in any evaluation is to load data into TokuDB tables. MyISAM and InnoDB® have algorithms that optimize bulk loads into an empty table, but these algorithms do not apply for incremental inserts, updates or deletes on existing tables. TokuDB does not yet have optimizations for bulk loading into empty tables, so be sure to test incremental inserts updates and deletes into large existing tables before drawing conclusions about TokuDB's performance.



Test with large tables

Be sure to test tables that are representative of the size of tables in production. **Results from small tables are a poor predictor of performance on large tables**, especially for B-tree based storage engines since performance will drop off suddenly when the row count exceeds a certain size. We recommend testing tables with at least 50M rows, and preferably with row counts as close as possible to the row counts of production size tables.

Data Compression

TokuDB compresses the data and indexes on disk, resulting in 2x - 12x less disk space utilization as compared to InnoDB or MyISAM. TokuDB stores its data in the MySQL® data directory, under the sub-directory with the name of the database you are testing. TokuDB creates a separate sub-directory named <tablename>.tokudb/ for each table. Within that sub-directory, the file main.tokudb contains the primary table, and there are separate files named <indexname>.tokudb for each defined index. Be sure to check the disk use and compare it to the disk used by MyISAM or InnoDB.

Query Performance

After loading data, run a representative set of the most problematic queries using the existing schema. TokuDB self optimizes its data structures during use, so the first run of a query is typically slower than subsequent runs. **It's important to run each query several times** to attain representative results. Note that speed ups from data structure optimization are independent from caching effects.

Index Optimization

TokuDB is uniquely able to maintain indexes efficiently on large tables, enabling the use of additional and better indexes to improve query performance. Proper indexes often improve query times by several orders of magnitude, making index optimization a critical step in the evaluation process.

Primary table look ups are required whenever there are columns in the select clause of query that are not included in an index designed to satisfy the where clause. Look ups require many small, random IOs resulting in especially poor performance.

Covering indexes and clustering indexes provide dramatic performance improvements by eliminating the need for primary table look ups. TokuDB supports secondary clustering indexes and up to 32 columns per index, providing new options to improve performance of a broader range of queries.

Tokutek's experts are available to review schemas and sample queries to recommend indexing strategies for your application. Be sure to **optimize index definitions and re-run queries to experience the best possible performance**. For more information on indexing, see:

<http://ronaldbradford.com/blog/understanding-different-mysql-index-implementations-2009-07-22/>

<http://www.mysqlperformanceblog.com/2009/06/05/a-rule-of-thumb-for-choosing-column-order-in-indexes/>

http://blogs.tokutek.com/tokuview/covering_indexes_orders_of_magnitude_improvements/

http://blogs.tokutek.com/tokuview/introducing_multiple_clustering_indexes/

http://blogs.tokutek.com/tokuview/clustering_indexes_vs_covering_indexes/

http://blogs.tokutek.com/tokuview/long_index_keys

http://blogs.tokutek.com/tokuview/how_clustering_indexes_sometimes_help_update_and_delete_performance

Insert/Update/Delete

After optimizing the index design to improve query performance, it's critical to **test incremental inserts, updates, and/or deletes into a large, existing table to ensure data modification requirements can be met**. Define requirements for the number of rows per second to be inserted, updated or deleted both on average and during peak loads. InnoDB optimizes insert performance, so be sure to include delete testing if deletes are used in production. TokuDB is likely to demonstrate significant performance gains during this phase of the evaluation.



Summary

Complete all of the following steps to get an accurate assessment of the benefits of TokuDB for your production application:

1. Test on **large tables** (results from small tables are not indicative of performance on large tables).
2. Load an initial data set and **measure disk use**.
3. **Run queries several times** on TokuDB for data structures to be automatically optimized.
4. **Define covering and clustering indexes** for large gains in query performance.
5. **Test incremental inserts, updates and especially deletes** into large tables with indexes defined.

TokuDB's compression benefits will show up on step 2, query performance benefits become clear on step 4, and the ability to maintain indexes necessary to deliver improved query performance over time on large, growing tables is demonstrated in step 5. Data gathered during all these steps should be evaluated against productionsvn stat requirements to assess TokuDB for your application.

Contact Tokutek with questions or for technical support during the evaluation - we're here to help!