Performance Comparison between MySQL, TiDB, MariaDB

extensoData

unleash your x factor

Submitted By:

Bishal Neupane

Bishesh Kafle

23rd March, 2024

Table of Contents

1. Introduction

MySQL, MariaDB, and TiDB are all popular relational database management systems (RDBMS), but they cater to different use cases and have distinct features. Here's a comparative overview of each:

1.1 MySql

Key Features

- Maturity and Stability: Long-established with a strong community and extensive documentation.
- Replication: Supports master-slave replication and semi-synchronous replication.
- Storage Engines: Multiple storage engines like InnoDB (default) and MyISAM.
- Scalability: Can handle large databases but traditionally scales better vertically.

Use Cases

- Web applications (e.g., WordPress, Joomla)
- Online transaction processing (OLTP)
- General-purpose databases

1.2 MariaDB

Key Features

- Compatibility: Highly compatible with MySQL, intended as a drop-in replacement.
- Performance: Improved performance and additional storage engines like Aria and ColumnStore.
- Security: Enhanced security features and better handling of authentication.
- Replication: Advanced replication features, including Galera Cluster for multi-master replication.
- Storage Engines: Supports more storage engines, such as TokuDB and MyRocks.

Use Cases

- Organizations seeking an open-source alternative to MySQL
- Applications requiring advanced replication and clustering
- Users needing high performance and scalability

1.3 TiDB

Key Features

- Scalability: Horizontally scalable, designed to handle large distributed systems with ease.
- Compatibility: Compatible with MySQL, meaning existing MySQL applications can migrate with minimal changes.
- Hybrid Transactional/Analytical Processing (HTAP): Capable of handling both OLTP and OLAP workloads efficiently.
- High Availability: Built-in high availability and fault tolerance with automatic failover.
- Cloud-Native: Designed for cloud environments with features like auto-scaling and automated backup.

Use Cases

- Large-scale applications requiring distributed databases
- Systems needing both transactional and analytical processing
- Cloud-native applications

2. MySQL vs TiDB vs MariaDB

2.1 Differences

Differences	MySQL	TiDB	MariaDB
Architecture	Traditional single-node RDBMS with master-slave replication and sharding managed at the application level	Distributed SQL database designed for horizontal scalability and strong consistency.	Fork of MySQL with additional features, often used as a drop-in replacement for MySQL. Supports both single-node and multi-node deployments
Scalability	Limited to vertical scaling (increasing the capacity of a single server). Horizontal scaling is complex and typically involves manual sharding.	Built for horizontal scalability out of the box. It automatically shards data and distributes it across multiple nodes.	Supports vertical scaling and can also achieve horizontal scaling.
Consistency and Transactions	Supports ACID transactions with configurable isolation levels, but consistency can be challenging in a multi-node setup	Strong consistency with distributed transactions using the Raft consensus protocol. Designed for high availability.	Supports ACID transactions with similar isolation levels to MySQL.

Compatibility	Widely adopted SQL database. Many tools and applications support MySQL natively.	Compatible with MySQL syntax and protocols, allowing easy migration from MySQL, but with additional distributed database features.	Maintains compatibility with MySQL, but includes additional storage engines and features not available in MySQL
Features	Core RDBMS features with some NoSQL capabilities via the InnoDB storage engine. Limited by its original single-node design.	Combines traditional RDBMS features with distributed database capabilities, such as horizontal scalability and fault tolerance.	Extends MySQL's features with additional storage engines (e.g., ColumnStore for analytics), improved performance, and security enhancements.

2.2 Similarities

1. SQL Compliance:
    - All three databases support SQL and provide a similar syntax, making it easy for developers to switch between them.
2. ACID Compliance:
    - MySQL, TiDB, and MariaDB all support ACID transactions, ensuring reliable transaction processing and data integrity.
3. Open Source:
    - All three are open-source projects, with active communities contributing to their development and improvement.
4. MySQL Compatibility:
    - TiDB and MariaDB are designed to be compatible with MySQL, allowing for easier migration and integration with existing MySQL-based applications.
5. Replication and Clustering:
    - Each database offers some form of replication and clustering. MySQL has master-slave replication, TiDB has a built-in distributed architecture, and MariaDB uses Galera Cluster for synchronous replication.

3. Performance Comparisons

For comparing performance, we have taken datasets of variable rows but fixed number of columns i.e. 16.

3.1 MySQL vs TiDB

The datasets were exported using mysqldump from a MySQL server and then loaded into both MySQL and TiDB databases. The performance metrics include user time, system time, and total time taken for the loading process.

The following comparison was done in the following system:

- **OS**: Sonoma 14.5
- **Memory**: 16 GB
- **Chipset**: M1Pro

And the database versions are as follows:

- **MySQL:** 8.4.0
- **TiDB:** 8.1.0

Table1 and 2 Overview:

- **User Time**: The amount of time the CPU spent in user mode.
- **System Time**: The amount of time the CPU spent in kernel mode.
- **Total Time**: The overall time taken to complete the loading process.
- **I/O Time(**Total Time - (User Time + System Time)**):** Time spent in disk I/O and data transfer operations.
- **Speed Ratio:** Total time taken by TiDB / Total Time taken by MySQL.

Table1: Speed Comparison of loading tables using MySQL and TiDB

No. of Rows	**Database**	**User Time(s)**	**System Time(s)**	**I/O Time(s)**	**Total Time(s)**	**Speed Ratio**
100000	MySQL	0.13	0.01	0.788	0.928	3.210
	TiDB	0.13	0.01	2.839	2.979	
200000	MySQL	0.25	0.02	1.557	1.827	3.000
	TiDB	0.25	0.02	5.211	5.481	
500000	MySQL	0.62	0.03	3.886	4.536	3.017
	TiDB	0.62	0.04	13.027	13.687	
1000000	MySQL	1.23	0.05	7.628	8.908	3.066
	TiDB	1.24	0.07	26.007	27.317	
2000000	MySQL	2.45	0.1	17.587	20.137	2.736
	TiDB	2.49	0.15	52.464	55.104	
5000000	MySQL	6.14	0.23	40.33	46.700	2.941
	TiDB	6.22	0.35	130.76	137.330	

Now calculating the percentage by which MySQL is faster, we can use the following formula:

%faster = (Speed Ratio - 1) * 100

Thus we get the following results.

1. For 100000 rows, %faster = (3.210-1)*100 = **221%** faster.
2. For 200000 rows, %faster = (3.000-1)*100 = **200%** faster.
3. For 500000 rows, %faster = (3.017-1)*100 = **202%** faster.
4. For 1000000 rows, %faster = (3.066-1)*100 = **207%** faster.
5. For 2000000 rows, %faster = (2.736-1)*100 = **174%** faster.
6. For 5000000 rows, %faster = (2.941-1)*100 = **194%** faster.

On average, for insertion MySQL seems to be almost 3 times faster than that of TiDB for the provided datasets.

Table2: Speed Comparison of reading tables using MySQL and TiDB

No. of Rows	Operation	Database	User Time(s)	System Time(s)	I/O Time(s)	Total Time(s)	Speed Ratio
100000	Select count(1)	MySQL	0.01	0.00	0.08	0.018	3.000
		TiDB	0.01	0.00	0.044	0.054	
	Select *	MySQL	0.65	0.14	0.713	1.503	1.011
		TiDB	0.65	0.14	0.729	1.519	
200000	Select count(1)	MySQL	0.01	0.00	0.041	0.051	1.529
		TiDB	0.01	0.00	0.068	0.078	
	Select *	MySQL	1.30	0.27	1.429	2.999	0.995
		TiDB	1.30	0.27	1.417	2.987	
500000	Select count(1)	MySQL	0.01	0.00	0.079	0.089	1.820
		TiDB	0.01	0.00	0.152	0.162	
	Select *	MySQL	3.23	0.68	3.561	7.471	0.986
		TiDB	3.23	0.66	3.480	7.370	
1000000	Select count(1)	MySQL	0.01	0.00	0.152	0.162	1.16
		TiDB	0.01	0.00	0.178	0.188	
	Select *	MySQL	6.48	1.40	7.078	14.958	0.985 3
		TiDB	6.48	1.32	6.936	14.736	
2000000	Select count(1)	MySQL	0.01	0.00	0.254	0.264	0.950
		TiDB	0.01	0.00	0.241	0.251	
	Select *	MySQL	13.07	2.87	14.089	30.029	0.989
		TiDB	13.08	2.67	13.974	29.724	
5000000	Select count(1)	MySQL	0.01	0.01	0.540	0.560	0.641
		TiDB	0.01	0.00	0.349	0.359	
	Select *	MySQL	32.66	7.15	35.18	74.99	0.993
		TiDB	32.67	7.11	34.75	74.53	

MySQL is faster with smaller datasets, but TiDB outperforms MySQL as dataset size increases.

3.2 MySQL vs MariaDB

System Description

- **OS**: Windows 11
- **Chipset**: AMD Ryzen 7 5800HS
- **RAM**: 16 GB

And the database versions are:

- **MySQL**: 8.0.36
- **MariaDB**: 11.3.2

Table 3 and 4 Overview:

- **I/O Time:** Time spent in disk I/O and data transfer operations.
- **Speed Ratio:** Time taken by MariaDB / Time taken by MySQL.

Table3: Speed Comparison of loading tables using MySQL and MariaDB

No of Rows	Database	I/O time (s)	Speed Ratio
100000	MySQL	0.25	8.596
	MariaDB	2.149	
200000	MySQL	0.578	7.350
	MariaDB	4.247	
500000	MySQL	1.155	8.940
	MariaDB	10.331	
1000000	MySQL	1.949	10.710
	MariaDB	20.882	

Now calculating the percentage by which MySQL is faster we get the following results:

1. For 100000 rows, %faster = (8.596-1)*100 = 760% faster.
2. For 200000 rows, %faster = (7.350-1)*100 = 635% faster.
3. For 500000 rows, %faster = (8.940-1)*100 = 794% faster.

4. For 1000000 rows, %faster = (10.710-1)*100 = 971% faster.

On average MySQL seems to perform 8.9 times faster than MariaDB.

Table4: Speed Comparison of reading tables using MySQL and TiDB

No. of Rows	Operation	Database	I/O time	Speed Ratio
100000	Select count(*)	MySQL	0.05	0.462
		MariaDB	0.023	
	select(*)	MySQL	0.6263	0.747
		MariaDB	0.468	
200000	Select count(*)	MySQL	0.25	0.152
		MariaDB	0.038	
	select(*)	MySQL	1.083	0.884
		MariaDB	0.958	
500000	Select count(*)	MySQL	0.63	0.152
		MariaDB	0.096	
	select(*)	MySQL	2.448	0.848
		MariaDB	2.077	
1000000	Select count(*)	MySQL	1.30	0.156
		MariaDB	0.195	
	select(*)	MySQL	4.646	0.948
		MariaDB	4.409	

It can be concluded that MariaDB has better performance overall in the read operations. In the less resource intensive work such as count() the performance can be slightly hinted with a ratio of 1.15, but in the resource intensive operations as select (*) the performance can be viewed to be almost 6 times of the counterpart.

3.3 Key Takeaways

1. MySQL's Superiority
    - MySQL demonstrates superior performance over both TiDB and MariaDB across varying dataset sizes.
    - The consistent performance advantage of MySQL suggests it is a robust choice for applications with high-performance requirements.

2. Consideration of Use Cases
    - While MySQL shows better performance in these tests, it is also important to consider other factors such as scalability, ease of use, community support, and specific use case requirements when choosing a database system.
    - TiDB, for example, offers horizontal scalability and might be more suitable for certain distributed systems use cases despite the performance gap.

3. Need for Contextual Evaluation
    - Performance metrics alone should not be the sole deciding factor. Consider the overall ecosystem, features, and long-term goals of your application to make a well-rounded decision.

Note: The results may vary from system to system but the relative difference should be similar.