

# **TDSQL for MySQL**

## **General References**

### **Product Documentation**



## Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

# Contents

## General References

Performance Comparison Data for Strong Sync

# General References

## Performance Comparison Data for Strong Sync

Last updated : 2024-01-06 17:35:36

This document provides a performance comparison between a TDSQL shard and open-source MySQL (not optimized) for your reference.

### Testing Environment for Comparison

**Hardware:** 24-core CPU; 128 GB memory; 1.8 TB SSD

**Network environment:** LAN with an average network latency of 0.80 ms

**Operating system:** CentOS 7.0

**Data volume:** 10 tables. Each of them has 2,180,000 data rows of about 5.2 GB. InnoDB buffer: 30 GB

**Open-source version:** MySQL 5.7.17 community version (not optimized; semi-sync enabled)

**TDSQL shard version:** MySQL 5.7 (strong sync enabled; thread pool enabled by default) with the following parameters:

`thread_pool_max_threads=2000`

`thread_pool_oversubscribe = 10`

`thread_pool_stall_limit = 50`

`thread_handling = 2`

### Detailed Comparison Data

#### 1. Data initialization parameters



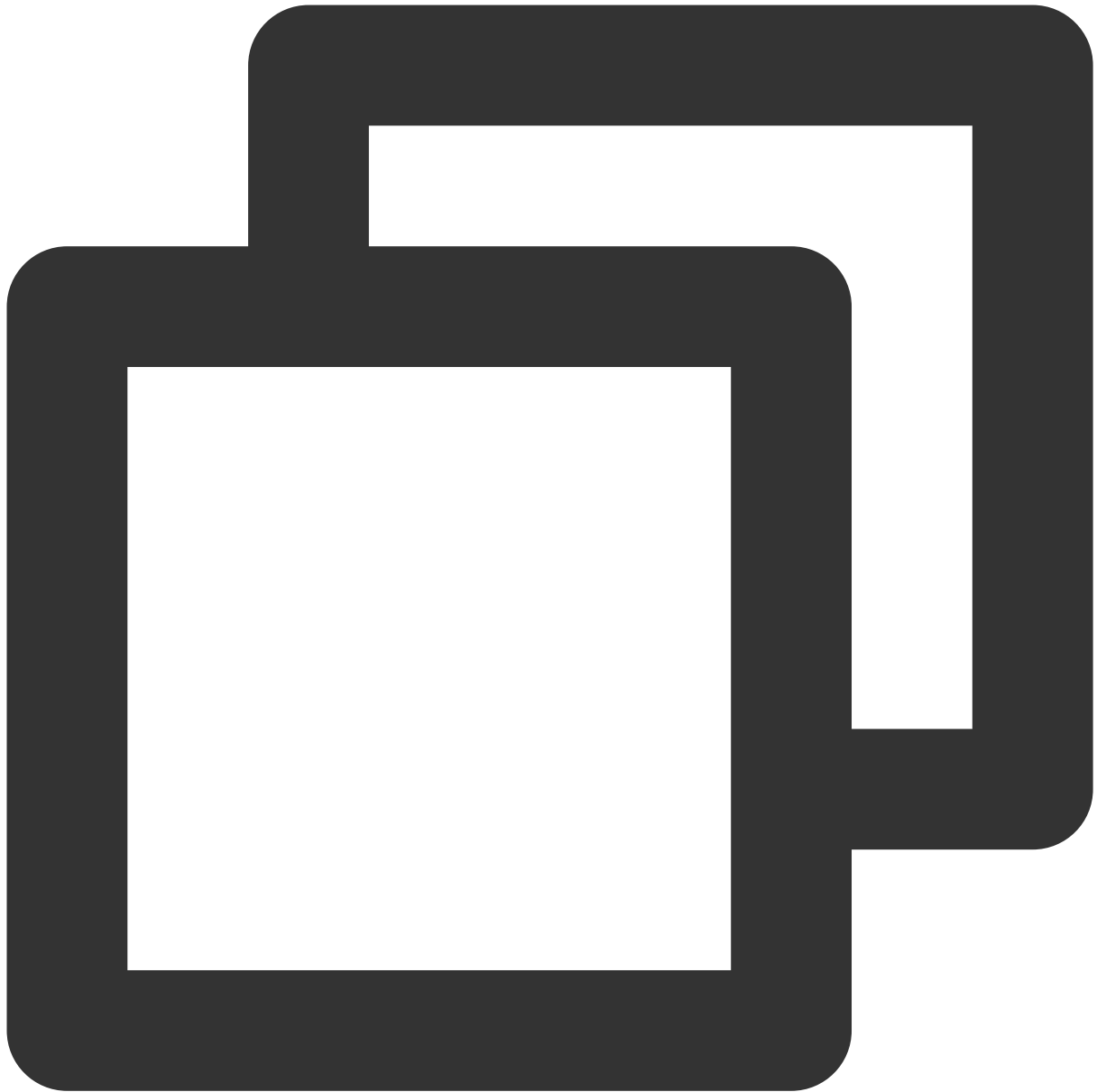
```
create database caccts ;  
./sysbench --num-threads=500 --test=./tests/db/oltp.lua.bak --oltp-table-size=21800
```

## 2. Non-index update (UPDATE)



```
./sysbench --num-threads=500 --test=./tests/db/update\\_non\\_index.lua --oltp-tabl
```

### 3. Read-only (SELECT)



```
./sysbench --num-threads=500 --test=./tests/db/select.lua --oltp-table-size=2180000
```

#### 4. Hybrid test



```
./sysbench\\_orig --num-threads=500 --test=./tests/db/oltp\\_new.lua --oltp-read-on
```

### Read request (READ) test results

Concurrency	Version	QPS	Average Response Time (ms)	99% Response Time (ms)
50	Open-source MySQL	304585	0.16	0.26



50	TDSQL	330695	0.15	0.24
100	Open-source MySQL	407443	0.24	0.48
100	TDSQL	484640	0.2	0.72
200	Open-source MySQL	433401	0.57	1
200	TDSQL	498215	0.55	1.22
500	Open-source MySQL	428542	1.16	2.42
500	TDSQL	494874	1.01	2.61
1000	Open-source MySQL	412775	2.4	6.3
1000	TDSQL	478393	2.08	4.21

### Write request (write) test results

Concurrence	Version	QPS	Average Response Time (ms)	99% Response Time (ms)
50	Open-source MySQL	14816	3.37	4.82
50	TDSQL	28925	1.73	2.55
100	Open-source MySQL	25046	3.99	6.91
100	TDSQL	43466	2.3	4
200	Open-source MySQL	32690	6.12	10.86
200	TDSQL	54045	3.7	7.27
500	Open-source MySQL	37192	13.44	21.1
500	TDSQL	70370	7.25	15.52
1000	Open-source	35447	28.2	40.47

	MySQL			
1000	TDSQL	69890	14.35	30.73

### Hybrid scenario (OLTP) test results

Concurrency	Version	QPS	Average Response Time (ms)	99% Response Time (ms)
50	Open-source MySQL	63806	4.7	7.13
50	TDSQL	162883	1.84	3.45
100	Open-source MySQL	102516	5.85	11.4
100	TDSQL	173974	3.58	6.64
200	Open-source MySQL	124550	9.64	18.92
200	TDSQL	208128	5.76	11.9
500	Open-source MySQL	125386	23.93	39.68
500	TDSQL	232543	13.58	27.81
1000	Open-source MySQL	121765	49.29	80.71
1000	TDSQL	226130	27.76	54.78