TencentDB for MariaDB

Product Introduction

Contents

Product Introduction
Overview

Last updated: 2023-08-30 16:05:48

Feature Overview

TencentDB for MariaDB is a highly secure, enterprise-grade cloud database designed for OLTP scenarios, consistently utilized in Tencent's billing operations. Not only does MariaDB maintain compatibility with MySQL syntax, but it also boasts advanced features such as thread pools, auditing, and remote disaster recovery. Furthermore, it embodies the scalability, simplicity, and cost-effectiveness inherent to cloud databases.

> **Note**
> For new features of the database kernel and SQL engine, refer to Private Cloud Description. The last two digits of the private cloud version and public cloud version indicate the same kernel version when they are consistent.

Watch video

Relevant Concepts

- Instance: a TencentDB for MariaDB resource in Tencent Cloud.
- Instance type: a combination of node quantity, read/write capability, and deployment region for a TencentDB for MariaDB instance.
- Read-only instance: a TencentDB for MariaDB instance that can only be read from.
- Disaster recovery instance: a TencentDB for MariaDB instance that supports disaster recovery across AZs and regions.
- VPC: a custom virtual network space that is logically isolated from other resources.
- Security group: security access control to TencentDB for MariaDB instances by specifying IP, protocol, and port rules for instance access.
- Region and AZ: physical location of a TencentDB for MariaDB instance and other resources.
- Tencent Cloud Console: web-based UIs.

Strengths

Last updated: 2023-08-30 16:06:48

Strong data consistency

Supports configuring strong sync replication. In the primary-standby architecture, strong sync ensures strong data consistency between the primary and standby databases, preventing data loss during primary-standby switchovers. Of course, you can also improve performance by modifying the configuration to disable the strong sync feature.

High security

- Anti-DDoS attack: When users connect to and access MariaDB instances via the public network, they may be vulnerable to DDoS attacks. When the MariaDB security system detects that a user instance is under a DDoS attack, it automatically initiates traffic cleansing.
- System security: Even within the internal network, MariaDB is protected by multiple layers of firewalls, effectively defending against various malicious attacks and ensuring data security. Additionally, physical servers do not allow direct login and only open specific ports required for database services, effectively isolating risky operations.
- VPC isolation: VPCs are supported to securely isolate access requests from other devices in the private network.
- Private network risk management: The TencentDB team cannot directly access TencentDB for MariaDB physical machines or instances; instead, they must access them through the Tencent Cloud Ops management platform. Even troubleshooting has to be performed on secure devices and strictly controlled by the internal risk management system.
- Object-level permission control: TencentDB for MariaDB allows you to define table-level permissions and configure IP addresses that can access TencentDB for MariaDB instances. Access from other IP addresses will be denied.
- Database audit: Database audit can be configured. Operations of admins or users are recorded for subsequent risk management.
- Operation logs: The system records all users' operations on TencentDB for MariaDB instances through the Tencent Cloud console for future traceability.

High Availability

TencentDB for MariaDB is designed to achieve an availability of over 99.99%. It provides primary/replica hot backup or two replicas for one primary (where the two replicas are used to

implement imperceptible failover). In addition, it has features such as automatic recovery of faulty nodes, automatic backup, and rollback to facilitate smoother and securer business operations.

High physical availability

Depending on the instance configuration you purchase, MariaDB typically adopts a one-primary-one-standby two-node architecture or a one-primary-two-standby three-node architecture. Each node is installed on a physically separate, cross-rack deployed server, ensuring that database services are not affected by single device, rack network failures, or power outages.

High network availability

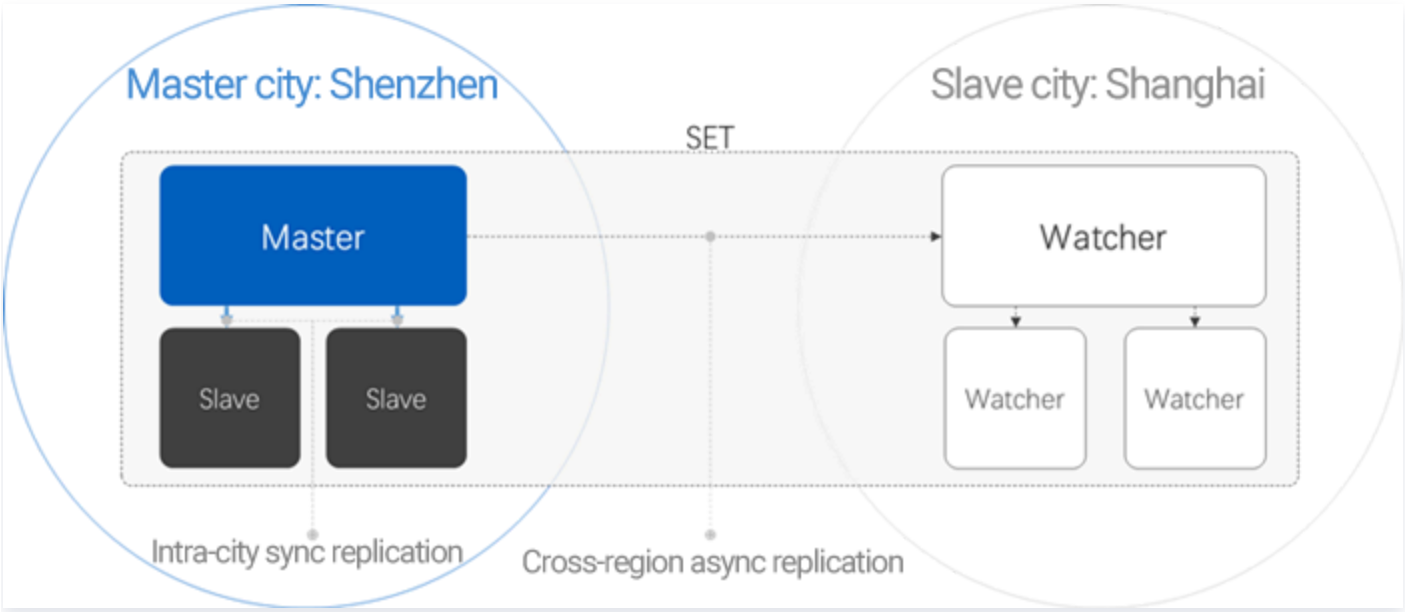
Each physical node of MariaDB is equipped with dual network cards and dual-connected switches, ensuring physical network security and reliability. In practice, TProxy connects to Tencent Cloud Gateway TGW; when the primary node fails, TProxy switches the DB route within 200ms at the fastest. If TProxy fails, TGW will load balance to other surviving TProxies within 1 second. The switchover does not change the access VIP (Virtual IP), effectively shielding the impact of physical server changes.

Backup and restoration services

Backup Service: The backup module is responsible for periodically performing physical backups and binary file (Binlog) storage for MariaDB. Backup files are uploaded to a highly secure distributed file cluster (HDFS). Typically, backups are initiated on standby nodes to avoid impacting the services provided by the primary node.
Recovery Service: Also known as rollback recovery, the recovery module restores backup files from HDFS to temporary instances, allowing users to inspect or adjust without affecting the primary instance's operation.
Backup Download: You can dump and download backup files to a specified location, such as the more cost-effective Tencent Cloud COS.

2-region-3-DC

MariaDB two-region three-center deployment architecture: The intra-city nodes are more than 10km apart in a straight line, while the inter-city nodes are more than 100km apart. Tencent's proprietary high availability (HA) scheduling solution is employed. The schematic diagram is as follows:



High Performance

Based on PCI-E SSD, TencentDB for MariaDB has powerful I/O performance that guarantees the accessibility of database. The storage firmware adopts the NVMe protocol and is specifically designed for PCI-E SSD, bringing TencentDB for MariaDB's superior performance to full play with one single high-IO instance sustaining up to 6 TB storage, 480 GB memory, and at least 220,000 queries per second (QPS). The performance advantage allows you to support higher business concurrence with a smaller number of database instances.
All MariaDB instances use a customized kernel, modified by TencentDB's research and development team based on practical requirements, rather than the original MariaDB kernel. The default parameters have been optimized through years of production experience, and Professional DBAs continuously optimize them to ensure that MariaDB always operates based on best practices.

Powerful Features

- Multi-source replication is supported, which well sustains complex enterprise-level businesses such as frontend, middleend, backend, and data warehouse in the insurance sector.
- Higher-level storage engines such as XtraDB and TokuDB are supported, and technologies such as "group commit for the binary log" are introduced to effectively improve business performance and decrease storage usage.
- Features such as thread pool and audit logs are supported.
- The clock is accurate down to the microsecond, making it ideal for financial transaction businesses that require higher time accuracy.
- Virtual columns (function indices) are provided to improve database analysis, statistics

collection, and computing performance.

High Compatibility with MySQL

MariaDB utilizes the InnoDB storage engine and is compatible with MySQL 5.5 and 5.6. This means that the code, applications, drivers, and tools used for MySQL databases can be used with MariaDB with minimal or even no changes required.

Cost effectiveness and ease of use

- Ready-to-use support: You can customize TencentDB for MariaDB specifications through the Tencent Cloud official website. After placing an order, a MariaDB instance will be automatically generated. By using MariaDB in conjunction with Cloud Virtual Machine (CVM), you can reduce application response time and save on public network traffic costs.
- On-demand upgrades: In the early stages of your business, you can purchase smaller MariaDB instances to handle the workload. As the database pressure and data storage volume change, you can flexibly adjust the instance specifications.
- Convenient management: Tencent Cloud is responsible for the daily maintenance and management of MariaDB, including but not limited to handling hardware and software failures, updating database patches, and ensuring the normal operation of MariaDB. You can also perform database management operations such as adding, deleting, restarting, backing up, and restoring through the Tencent Cloud Console.

Scenarios

Last updated: 2023-08-30 16:07:32

Cloud-based Data Disaster Recovery (Remote Disaster Recovery)

Data is a crucial component of enterprise operations, and while digitalization brings convenience, the vulnerability of electronic data and stored information to damage and loss is also exposed. In the face of natural disasters, system failures, employee errors, and virus infections, any single incident could potentially lead to a complete interruption of business operations or even catastrophic losses. Therefore, ensuring the security and integrity of data, especially the safety and completeness of core databases, is a vital consideration for every enterprise.
Establishing a self-built remote data disaster recovery center for an enterprise typically involves significant expenses, including substantial costs for data center hardware and software, as well as ongoing maintenance investments for annual operational expenses. However, paying for such low-probability events often does not align with the financial needs of the enterprise.
Hence, by utilizing cloud databases and cloud access products, a data disaster recovery center can be directly established in the cloud, synchronizing the primary data center's data to a remote backup center in the cloud through a secure private network in real-time. This approach not only addresses the challenges of managing massive amounts of data but also offers a high cost-performance ratio.

Business System Cloudification

If your business system has not been migrated to cloud, you may encounter the following issues:

- Your business grows fast, but high costs will be incurred if you prepare servers every year based on the annual peak traffic.
- New business departments often need to launch new businesses quickly to ensure timeliness. If resource preparation and procurement are required every time, the launch efficiency will be affected.
- Almost every business system has experienced shortage of backend resources due to traffic surges.
- Many company leaders think that the IT department is a cost center and should focus on solving problems such as unstable systems or insufficient performance rather than promoting businesses.

Backed by Tencent Cloud's many years of experience, TencentDB provides the following services and resources in face of the challenges above:

- Secure and open database solution.
- Highly available scheme that adopts strong sync replication and high availability (HA) architecture to implement high-performance disaster recovery.
- Auto scaling.

Hybrid cloud

TencentDB for MariaDB supports private cloud deployment solutions, allowing for installation in user-built data centers. Business systems and data are securely synchronized through dedicated lines (or VPN), creating an easily scalable hybrid cloud architecture.

Read/Write Separation

By default, all slave TencentDB instances support read/write separation, i.e., read-only slave nodes.

- Supports read-only implementation via SQL syntax or read-only accounts.
- If your configuration has multiple slaves, the load of read-only policy will be automatically distributed across the slaves.
- You can add more slaves by upgrading the specification.

Development and Testing

You may need to maintain multiple testing environments for different software versions and even high amounts of resources for stress testing.
The traditional solution is to self-build servers and databases to this end. However, this will waste a lot of hardware resources as developers will not use testing resources all the time, causing the resources often to be idle. In contrast, with the auto scaling capability of CVM and TencentDB, you can effectively address the problems of insufficient or wasted testing resources.

System Architecture

Last updated: 2023-08-30 16:14:48

Architecture Overview

High-availability architecture

In production systems, high availability solutions are typically required to ensure uninterrupted system operations. As the core capability of system data storage and services, the availability requirement for the database is higher than that for computing service resources. At present, high availability solutions for databases usually involve the collaboration of multiple database services. If one database fails, the remaining ones immediately take over its tasks, ensuring no service interruption or only a brief interruption. Alternatively, multiple databases can provide services simultaneously, allowing users to access any of them. If one database fails, users can immediately switch to another one.
Since databases store data, switching between multiple databases requires data synchronization. Therefore, **data synchronization technology is the foundation of high availability solutions for databases**. Currently, there are three methods of data replication:

- **Async replication**: When an application initiates an update request (including add, delete, and modify operations), the primary node (Master) immediately responds to the application after completing the corresponding operation, and asynchronously replicates data to the secondary node (Slave). In async replication mode, the unavailability of the secondary node does not affect operations on the primary node. However, the unavailability of the primary node may cause data inconsistency.
- **Strong sync replication**: When an application initiates an update request, the primary node (Master) replicates data to the secondary node (Slave) after completing the operation. The secondary node returns a success message to the primary node upon receiving the data. The primary node responds to the application after receiving the feedback from the secondary node. Data replication from the primary node to the secondary node is synchronous, so the unavailability of the secondary node will affect operations on the primary node. However, the unavailability of the primary node will not cause data inconsistency.

> **Note**
> When using "strong sync" replication, if there is a network interruption between the primary and secondary nodes or an issue with the secondary node, the primary node will be locked (hang). In this case, if there is only one primary node or one secondary node, a high availability solution cannot be implemented. A single server

> service failure would directly result in partial data loss, which does not meet the financial-grade data security requirements.

- **Semi-sync replication**: Semi-sync replication is a synchronization solution proposed by Google. Under normal circumstances, data replication adopts the strong sync replication mode. If an exception occurs when the primary node (Master) replicates data to the secondary node (Slave) (e.g., the secondary node is unavailable or there is a network issue between the two nodes), the replication mode degrades to async replication. Once the exception is resolved, async replication will revert to strong sync replication. Semi-sync replication implies that there is a smaller probability of data inconsistency when the primary node is unavailable.

Commonly used high-availability architectures

- **Shared storage solution**: Shared storage, such as SAN storage, is used. The principle of SAN is that multiple database servers share the same storage area, allowing them to "read and write" the same data. When the primary node fails, third-party high-availability software suspends the file system on the secondary node and then starts the database on the secondary node to complete the switch.
- **Log sync or streaming replication sync**: This is the most common replication mode for databases, such as MySQL. Whenever data is written, the MySQL Master Server transmits its Binary Log to the Slave through a replication thread. Upon receiving the Binary Log, the Slave writes the same data to the file system according to the Binary Log content. Currently, MySQL provides:
  - Async replication: async replication ensures quick response but cannot guarantee that binary logs actually arrive at the replica, i.e., data consistency cannot be guaranteed.
  - Semi-sync replication (sync plugin provided by Google): semi-sync replication has slower response to client requests and may downgrade to async replication in cases such as timeout. It can guarantee basic but not complete data consistency
- **Trigger-based synchronization**: Triggers are used to record data changes, which are then synchronized to another database.
- **Middleware-based synchronization**: The system does not connect directly to the underlying database but to a middleware. The middleware sends database changes to multiple underlying databases, thereby achieving data synchronization. In earlier years, due to business requirements, database performance, and synchronization mechanism issues, some software developers typically adopted similar architectures.

TencentDB for MariaDB Architecture Overview

Multi-thread asynchronous replication

You can learn about MariaDB's async multi-thread strong sync replication technology through the following video:

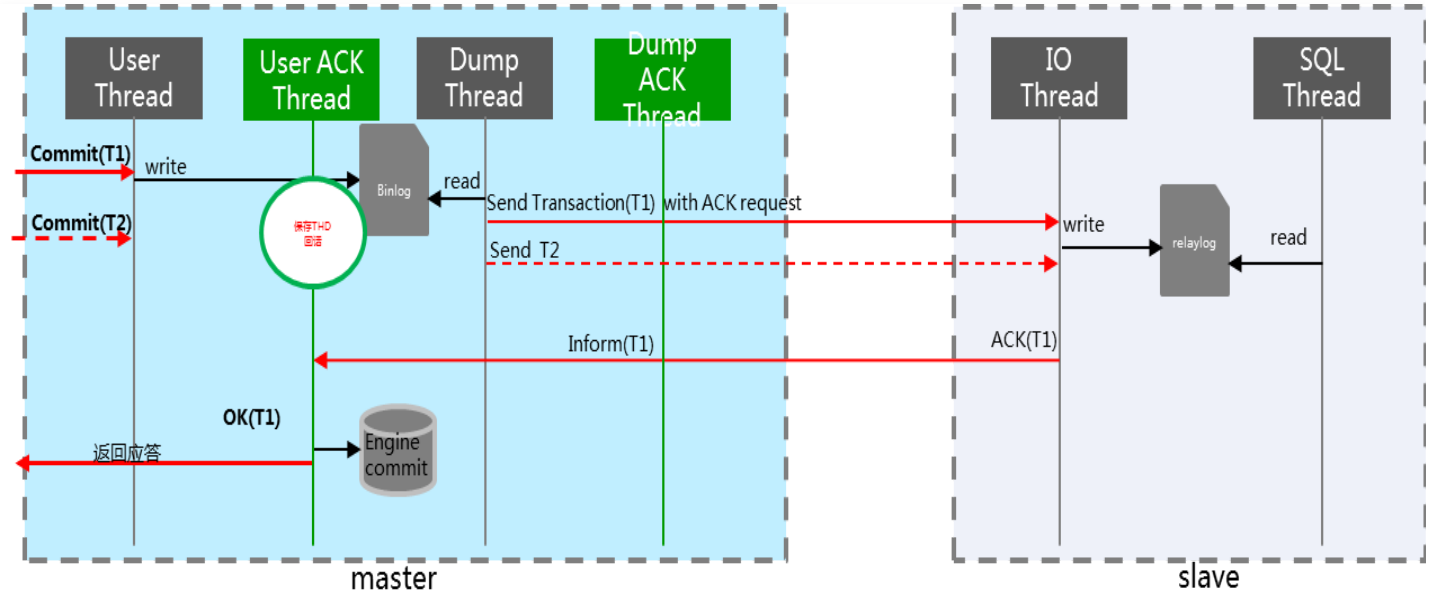
Watch video

Throughout the development of synchronization technology, async replication and semi-sync replication have been provided to cater to the general user base. These two technologies can ensure basic data synchronization when user requirements are not high, network conditions are good, and performance pressure is low. However, in most cases, using async replication and semi-sync mechanisms can often lead to data inconsistency issues, directly affecting system reliability and even resulting in lost transaction data, causing direct or indirect economic losses.

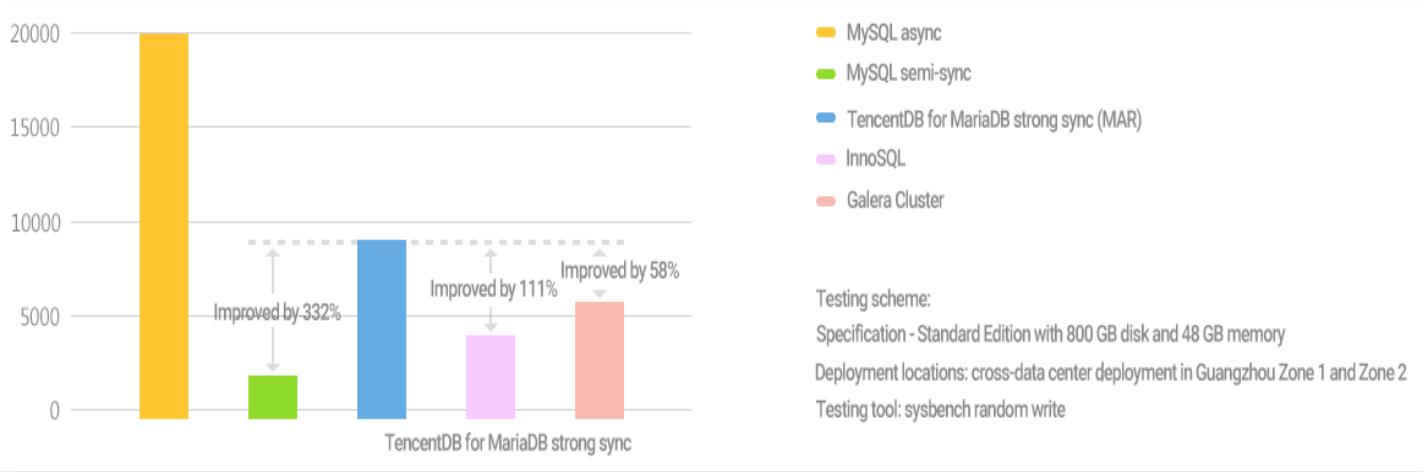
After years of accumulation in Tencent Cloud's business, a self-developed multi-thread asynchronous strong sync replication solution (Multi-thread Asynchronous Replication, MAR) has been introduced. Compared to Oracle's NDB engine, Percona XtraDB Cluster, and MariaDB Galera Cluster, MAR offers better performance, efficiency, and applicability. The characteristics of the MAR strong sync solution are as follows:

- Consistent sync replication ensures strong consistency of data between nodes.
- Complete imperceptibility to the business means that read-write separation and sync enhancement are not required at the business side.
- Asynchronization of serial sync threads and introduction of thread pool boost a substantial increase in performance.
- Cluster architecture is supported.
- Automatic member control is supported, and faulty nodes are automatically removed from the cluster.
- Automatic node joining is supported, eliminating the need for manual intervention.
- Each node contains a complete data replica, allowing for seamless switching at any time.
- There is no need to share storage devices.

With the MAR strong sync technology of Tencent, the primary will return a transaction response to the application only after data is synced to the replica as shown below:

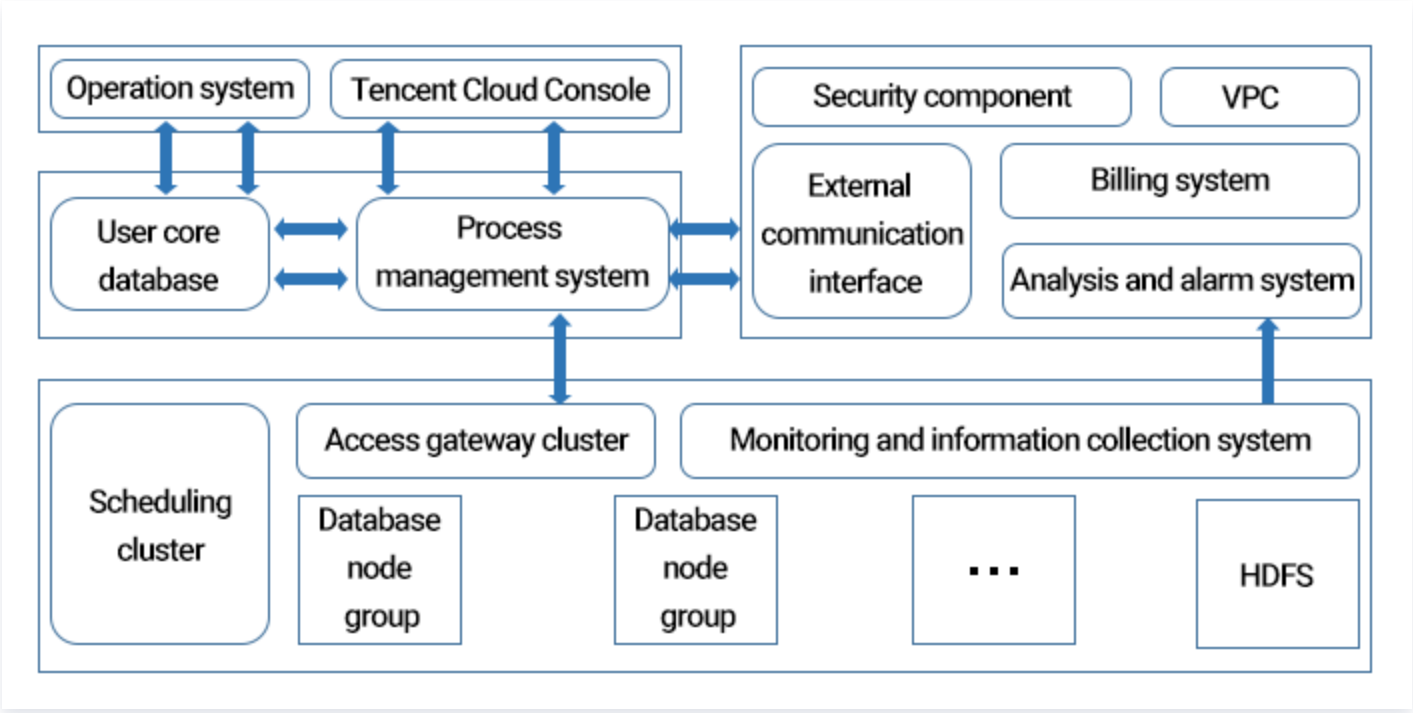


Its performance is higher than that of other mainstream sync schemes. Tests in the same cross-AZ (cross-IDC) testing configuration (with the standard sysbench use case) show that the performance of MAR is about 5 times higher than that of MySQL semi-sync replication and 1.5 times higher than that of MariaDB Galera Cluster.



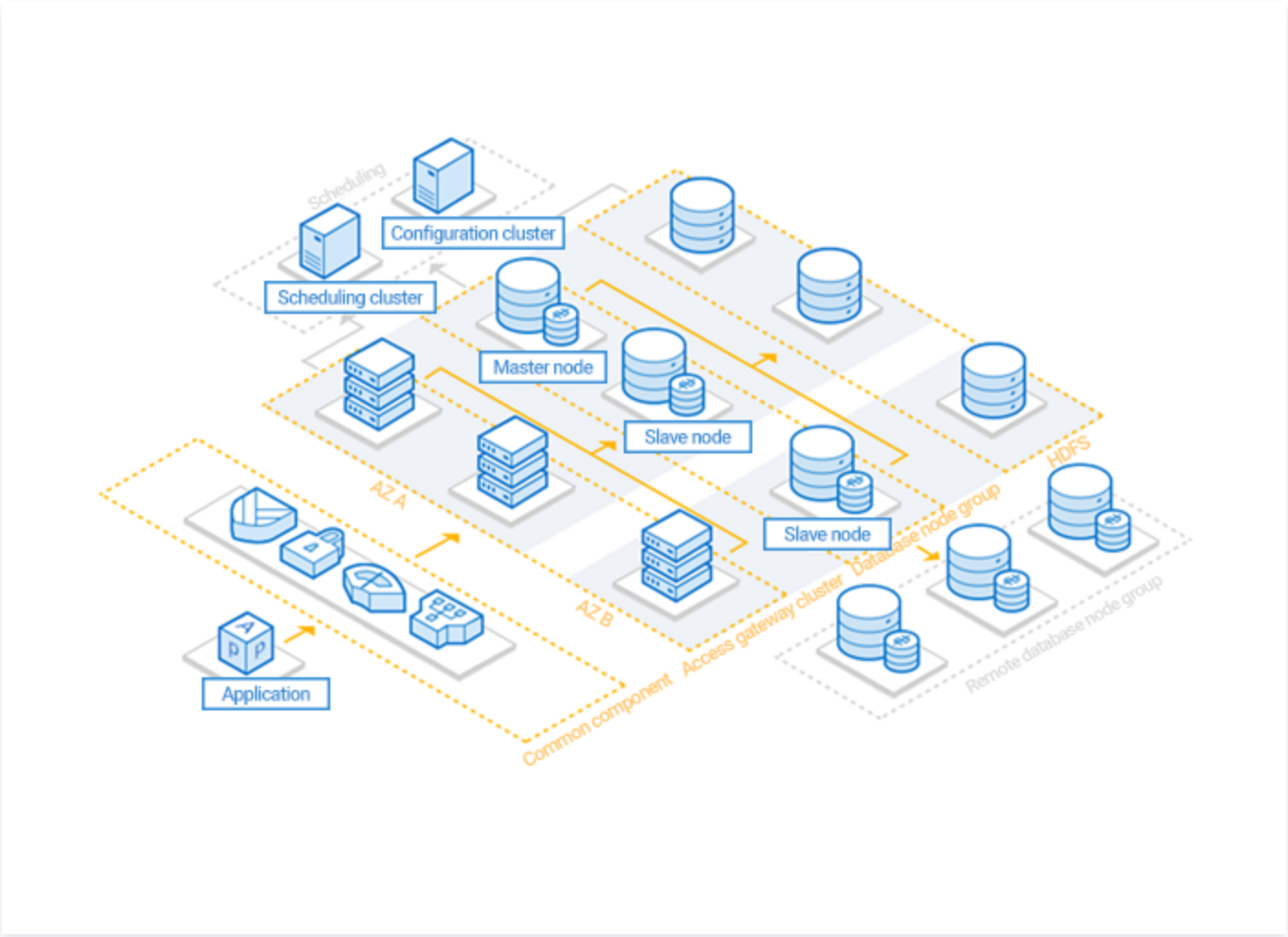
Cluster Architecture

By watching the following video, you can learn about the cluster architecture of MariaDB:
Watch video
MariaDB adopts a cluster architecture, and a standalone MariaDB system requires at least ten systems or components. The simplified architecture diagram is as follows:



The three core modules of MariaDB are: decision scheduling cluster (Tschedule), database node group (SET), and access gateway cluster (TProxy). The interaction between these three

modules is accomplished through the configuration cluster (TzooKeeper).



- **Database Node Group (SET)**: Composed of a MySQL-compatible database engine, monitoring, and information collection (Tagent), its architecture consists of "one primary node (Master), several secondary nodes (Slave_n), and several remote backup nodes (Watcher_m)". Under normal circumstances:
- It is deployed in cross-rack and cross-data center servers.
- It uses the heartbeat monitoring and information collection module (Tagent) for monitoring to ensure cluster robustness.
- In a distributed architecture and based on horizontal sharding, several shards (database node groups) provide a "logically unified and physically scattered" distributed database instance.
- **Tschedule (Decision Scheduling Cluster)**: As the management and scheduling center of the cluster, it primarily manages the normal operation of SETs, records, and distributes global database configurations, including:
- Scheduling job cluster (MariaDB Scheduler): it helps the DBA or database users automatically schedule and run various types of jobs, such as database backup, monitoring data collection, generation of different reports, and execution of business

flows. TencentDB for MariaDB combines Schedule, ZooKeeper, and operations support system (OSS) to activate specified resource plans through time windows, so as to fulfill various complex requirements of database resource management and job scheduling. Oracle uses DBMS_SCHEDULER to support similar capabilities.

- Program coordination and configuration cluster (TzooKeeper): it provides features such as configuration maintenance, election decision-making, and route sync for TencentDB for MariaDB, supports tasks like creation, deletion, and replacement for database node groups (shards), and uniformly delivers and schedules all Data Definition Language (DDL) operations. At least three TzooKeepers need to be deployed.
- Operations support system (OSS): it is a comprehensive custom business operations and management platform developed based on MariaDB. It fully takes into account database management characteristics to organically integrate network management, system management, and monitoring service.
- The scheduling cluster (Tschedule) is independently deployed in the three major Tencent Cloud data centers in China (with cross-data center deployment and remote disaster recovery).
- **Access Gateway Cluster (TProxy)**: Manages network layer connections, SQL parsing, and routing allocation (TProxy is not Tencent Cloud Gateway TGW).
- The number of deployed TProxies should be the same as that of database engines. TProxy is used to share the load and implement disaster recovery.
- It pulls the status of database nodes (shards) from the configuration cluster (TzooKeeper), provides shard routes, and implements imperceptible reads/writes.
- It records and monitors SQL execution and user access information, analyzes SQL execution efficiency, and performs security authentication to block risky operations.
- TGW is deployed on the frontend of TProxy and provides a unique virtual IP to users. This architecture greatly simplifies the communication mechanism between each node and lowers the hardware requirements, which means that even simple x86 servers can be used to set up stable and reliable databases similar to minicomputers or shared storage.

Instance Types

Last updated: 2023-08-30 16:15:11

A database instance is a standalone database environment that runs in the Tencent Cloud. A database instance can contain multiple user-created databases and can be accessed using the same client tools and applications as those for a standalone database instance.
There are four types of instances available in TencentDB for MariaDB.

Instance Type	Description	SDK
Source instance	An instance that can be read from and written to	The primary instance can be attached to a disaster recovery instance, enabling remote disaster recovery capabilities.
Primary Instance (Dedicated)	Primary Instance in Dedicated Cluster Mode	Database instances created within a dedicated cluster possess all the features of a primary TencentDB instance.
Disaster recovery instance	An instance that supports disaster recovery across AZs and regions	A disaster recovery instance is read-only during sync with a master instance. It can actively stop the sync and be promoted to a master instance for read/write access. It must reside in a different region from the master instance
Temp Instance	A temporary instance generated by the rollback feature.	Used for data validation and restoration after rollback.

Instance Architecture

Last updated: 2023-08-30 16:15:33

Instance Architecture	Description	Node	Advantage
Standard Edition (one source and one replica)	Each shard provides a high-availability architecture based on source/replica active-active deployment.	Two nodes: One source node and one replica node	• Supports read-only replicas In the one-primary-one-replica architecture, the read-only replica is suitable for lightweight read tasks only. Avoid high-load tasks such as large transactions to prevent impacting backup tasks and replica server availability. • Automatic node failover. • Default sampling granularity for monitoring: Once every 5 minutes. • Maximum backup duration: 30 days. • Operational log backup: 60 days • Supports database audit where audit logs can be retained for 15 days and an unlimited number of rules which can be configured.
Standard Edition (one source and two replicas)	Each shard provides a high-availability architecture based on source/replica multi-site active-active deployment.	Three nodes: One primary node and two replica nodes	
Finance Edition (one source and one replica)	Each shard provides a high-availability architecture based on source/replica active-active deployment.	Two nodes: One source node and one replica node	• Supports other deployment schemes. To use other schemes, contact your Tencent Cloud sale rep. • Supports read-only replicas with intelligent load balancing when replicas are read-only. • Automatic node failover. • Default sampling granularity for monitoring: Once every 1 minute. • Maximum backup duration: 3,650 days. You need to submit
Finance	Each shard	Three nodes:	

Edition (one source and two replicas)	provides a high-availability architecture based on source/replica multi-site active-active deployment.	One primary node and two replica nodes	a ticket for application. • Operational log backup: Default 60 days, with archived storage for 1 year. • Supports database audit where audit logs can be retained for 15 days. • Provides assistance for regulatory compliance.

TDSQL Edition

Last updated: 2023-08-30 16:15:57

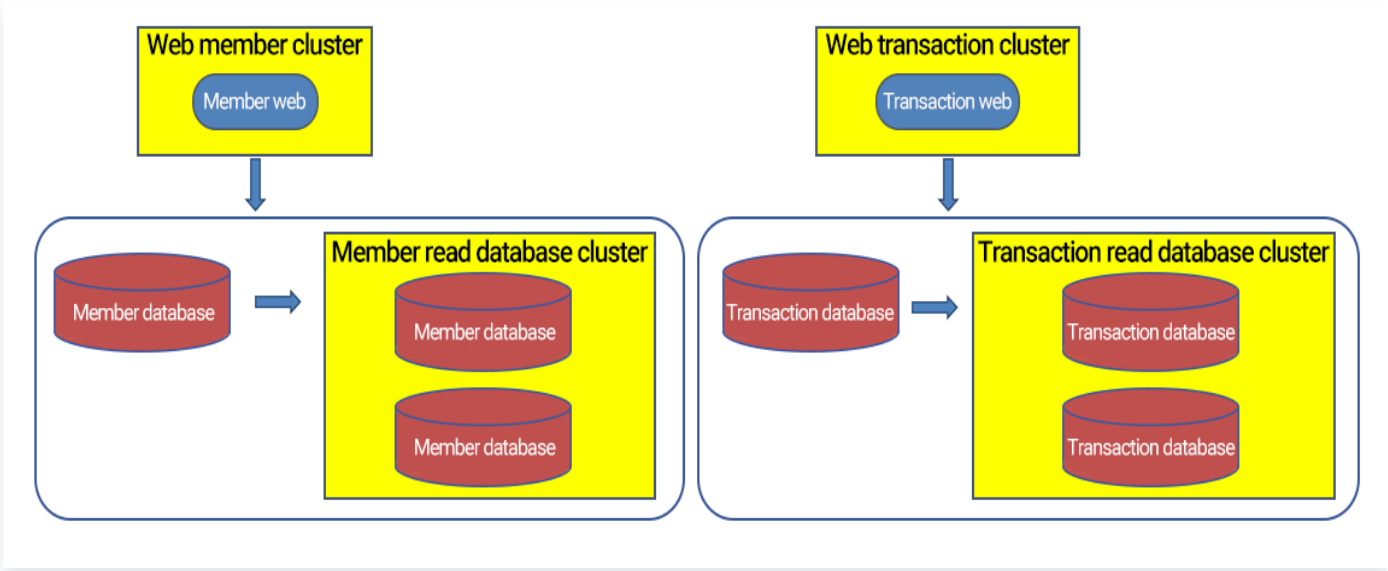
TDSQL for MySQL is a distributed edition of TencentDB for MariaDB. You can use TDSQL for MySQL if you anticipate that your business will grow so rapidly that the stand-alone edition, i.e. MariaDB, will no longer be able to support it.

Overview

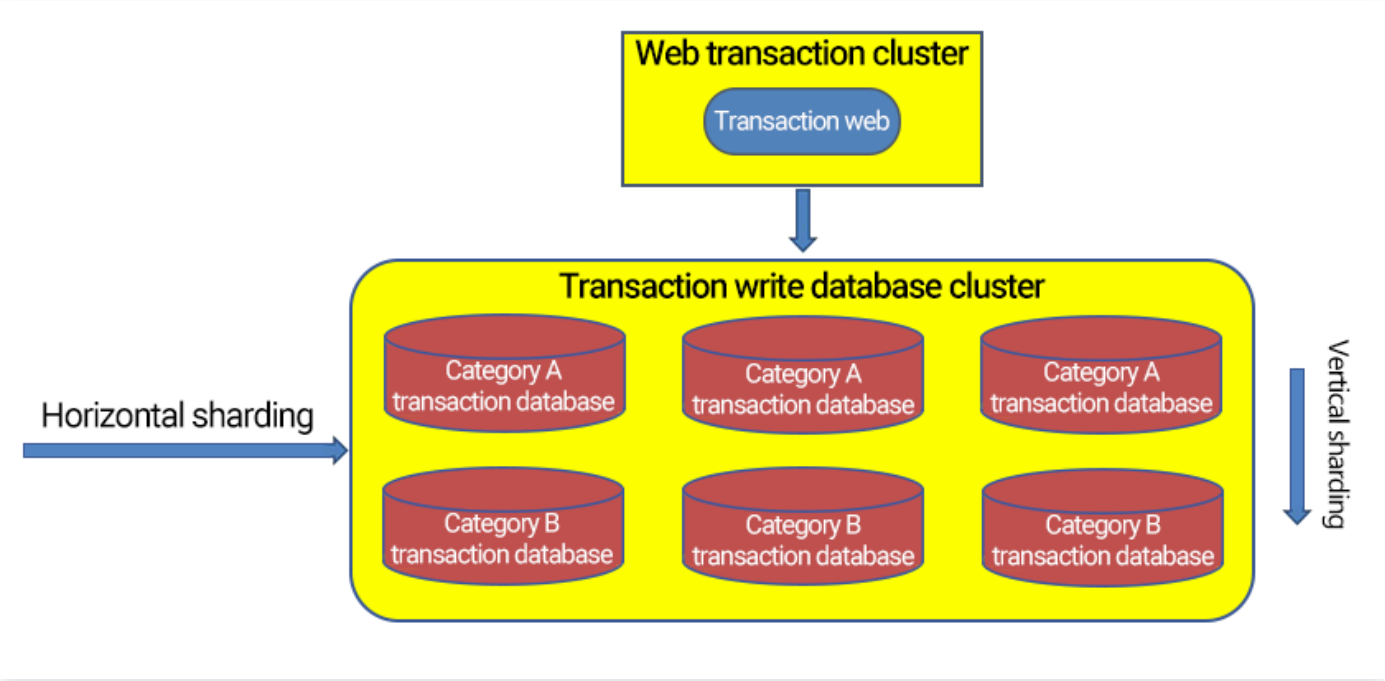
Data sharding

In high-performance concurrent internet architectures, performance bottlenecks often occur at the database server, especially when the business (users) reaches a scale of millions. By implementing reasonable data sharding, database performance and scalability issues can be effectively resolved. Database sharding is considered from two dimensions: vertical sharding (functional partitioning) and horizontal sharding.

- **Vertical sharding** is functional partitioning, which is closely related to the business and has a straightforward implementation approach. For example, an e-commerce platform can divide data by function into member databases, product databases, transaction databases, and logistics databases. However, vertical sharding cannot completely solve pressure issues, as a single database server has limited load and capacity, which will inevitably become a bottleneck as the business grows. The common solution to these problems is horizontal sharding.



- **Horizontal sharding** involves distributing a table's data across multiple physically independent database servers according to a certain rule. These "independent" database "shards" form a logically complete database instance.



Sharding rules

Relational databases are two-dimensional models, and data sharding usually requires finding a shard key to determine the splitting dimension, followed by defining rules to implement database partitioning. Identifying suitable sharding rules requires a comprehensive consideration of the business. Here are some common sharding rules:

1. Based on chronological order, such as splitting by year, with one shard for 2015 and another for 2016.
   - Advantages: simple and easy to query.
   - Disadvantages: The server performance for the current (e.g., 2016) hot data may be insufficient, while the storage performance for cold data is idle.
2. Based on the modulo of user ID, where the value ranges of the ID field after modulo operation are spread across different databases.
   - Advantages: The performance is relatively balanced and all data of the same user is in the same database.
   - Disadvantages: This may lead to data skew (for example, when a merchant system is designed, the data of one merchant of large-scale business mall may be more than that of thousands of merchants of small-scale business).
3. Based on the modulo of primary key, where the value ranges of the primary key field after modulo operation are spread across different databases.
   - Advantages: The performance is relatively balanced, data skew seldom occurs, and all data of the same primary key is in the same database.

- Disadvantages: Data is randomly distributed, and some business logics may require cross-shard join operations that are not supported directly.

In terms of sharded data source management, there are currently two modes:

1. Client mode: The data sources of multiple shards are managed by the configuration in the business program module, and the reading, writing, and data integration of the shards are performed within the business program.
2. Middleware proxy mode: A middleware proxy is built on the frontend of the sharding databases which are imperceptible to the frontend application.

TencentDB for TDSQL

Automatic horizontal sharding (sharding databases and splitting tables)

TDSQL (TencentDB for TDSQL) is a distributed database deployed on Tencent Cloud's public cloud that is compatible with MySQL protocols and syntax, and supports automatic horizontal sharding. In a distributed database, the business accesses a complete logical database table, while the backend evenly splits the table across multiple physical shard nodes. Currently, TDSQL is deployed by default in a primary-replica architecture and provides a comprehensive solution for disaster recovery, backup, restoration, monitoring, and migration, suitable for TB or PB scale massive database scenarios.

The history of TDSQL dates back to 2004 when Tencent's internet value-added services began to explode, and the explosive growth in business volume brought tremendous pressure on MySQL databases to expand. At that time, the database partitioning mechanism was introduced to solve the problem – large tables were pre-split into multiple sub-tables based on ShardKey and distributed across different physical machine nodes. Today, the backend storage data volume of TDSQL is quite substantial. For example, TDSQL for "Mi Master" supports 10 billion accounts across various channels, with nearly 900 million users and daily transaction amounts exceeding 1 billion RMB.
TDSQL has the following advantages to easily sustain massive amounts of business requests:

- **Automatic table sharding**: TDSQL supports automatic database and table sharding, combined with a unified data scheduling mechanism, to achieve on-demand capacity scaling. As TDSQL's gateway masks the internal details of database and table sharding, developers no longer need to worry about data partitioning, request routing, etc. They only need to initialize the shard key and directly program against the logical tables, focusing on implementing business logic, which significantly reduces the complexity of the application.
- **Automatic failover**: Whether it's the Internet of Things, big data, or payment services, any business that stores massive amounts of data has high availability requirements for its backend storage databases. The common solution is that failover requires business detection and cooperation, deep coupling with business programs, and a complex

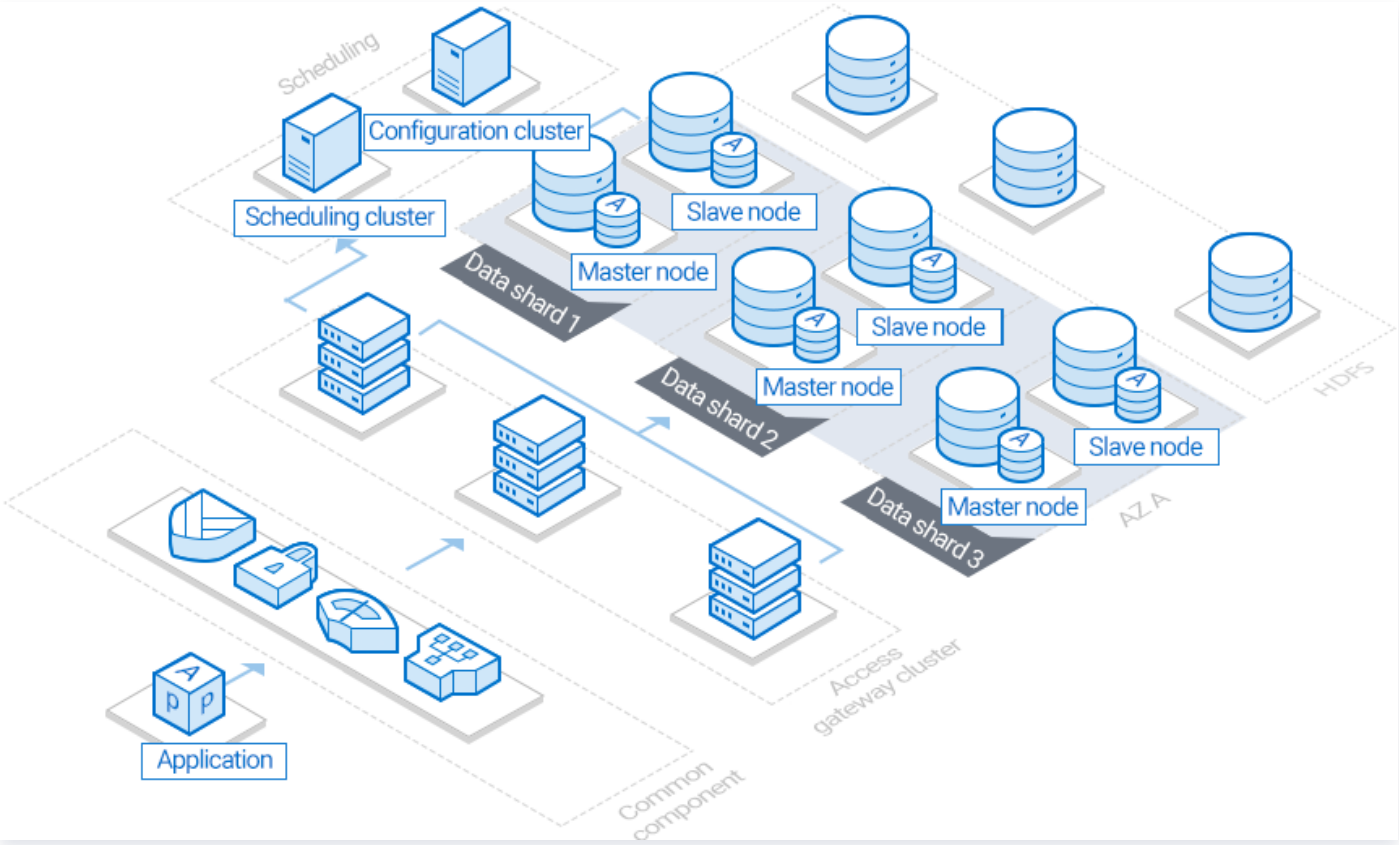
switching process that may even require manual intervention. After the business is restored, manual repair of any erroneous data that may have occurred during the switch is necessary, making maintenance very labor-intensive. TDSQL data nodes and gateways implement multi-point disaster recovery, automatically detecting the running status of instances. When the primary node is found to be unavailable, the system automatically triggers the master-replica failover process, ensuring high availability of the database in the event of host failure, network failure, or IDC failure. This failover process is completely transparent to the business and does not require manual intervention, ensuring a seamless user experience while greatly simplifying maintenance tasks.

- **High data consistency assurance**: If you have zero tolerance for data loss or corruption, TDSQL innovatively employs a multi-threaded strong sync replication mechanism based on MySQL's original async and semi-sync replication. This ensures that at least two copies of each transaction are made within the cluster before responding to the user. Then, a series of switching mechanisms are used to guarantee that data will not be lost or corrupted during node failure and subsequent switchovers.
- **Clustered management and automatic scaling**: The volume of business requests may surge several times or even dozens of times due to the launch of new features or marketing activities. Previously, DBAs had to be aware of business trends and manually scale the database in advance. Typically, the expansion process for most distributed databases is complex, with numerous manual operations prone to errors. TDSQL, on the other hand, implements automatic deployment, capacity scaling, backup and recovery, point-in-time rollback, and multi-dimensional monitoring at the cluster level. When scaling is required, DBAs only need to initiate the scaling process with a single click, and TDSQL will automatically complete the task. This clustered operation system significantly improves DBA efficiency while reducing the risk of errors caused by manual operations.

TDSQL architecture

By watching the following video, you can learn about the architecture of the distributed edition:
Watch video

The architecture of a TDSQL instance is as follows:



Data sharding: Compatible with the performance of open-source MySQL database engine, monitoring, and information collection (Tagent).

> **Note**
> In TDSQL, each shard is configured with two nodes by default, i.e., one primary and one secondary, and each TDSQL instance has at least two shards.

Scheduling Cluster: As the management and scheduling center of the cluster, it primarily manages the normal operation of SETs and records and distributes global database configurations.

Access Gateway Cluster (TProxy): Manages network layer connections, SQL parsing, and routing allocation, which can be understood as middleware for open-source distributed databases.

> **Note**
> In order to prevent the proxies from becoming a bottleneck, the number of proxies generally should equal to the number of shards.

Backup Cluster: Tencent Cloud's database backup cluster.

Note
TDSQL backups are stored for 7 days by default.