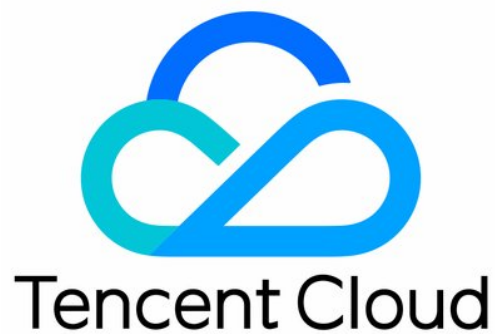


TencentDB for MariaDB

Product Introduction

Product Documentation



Copyright Notice

©2013-2019 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

 Tencent Cloud

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

Product Introduction

Overview

Strengths

Use Cases

System Architecture

Instance Types

Instance Architecture

TDSQL Edition

Product Introduction

Overview

Last updated : 2020-03-31 15:23:57

Overview

TencentDB for MariaDB is a highly secure enterprise-level cloud database dedicated to the online transaction processing (OLTP) scenario. It has always been used in Tencent's billing business. It is [compatible with MySQL syntax](#) and has various advanced features such as thread pool, audit, and remote disaster recovery while delivering easy scalability, simplicity, and high cost performance of TencentDB.

Related 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 : 2021-04-23 14:35:27

Strong Data Consistency

Strong sync replication can be configured. In the master/slave architecture, strong sync can ensure master/slave data consistency to avoid data loss in case of master/slave switchover. Of course, you can modify the configuration to disable strong sync to improve database performance.

High Security

- Anti-DDoS attacks: when you use the internet to connect to and access a TencentDB for MariaDB instance, you may suffer from DDoS attacks. When the TencentDB for MariaDB security system considers that your instance is under attacks, it will automatically enable traffic cleansing.
- System security: even in the private network, TencentDB for MariaDB is under protection by multiple layers of firewalls in order to effectively fight against various attacks and protect data security. In addition, physical servers do not allow direct login and only open ports required by specific database services to isolate 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 master/slave hot backup or two slaves for one master (where the two slaves 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

Based on the configuration of your purchased instances, TencentDB for MariaDB usually adopts the architecture of two nodes (one master and one slave) or three nodes (one master and two slaves). Each node is installed on an independent physical machine deployed across racks to ensure that database service will not be affected by network or power failure of a single device or rack.

High network availability

The physical machine of each TencentDB for MariaDB node adopts the dual-ENI dual-connection switch configuration to ensure security and reliability of the physical network. In actual use, TProxy connects to TGW; when the master node fails, TProxy can switch database routes within 200 ms at best; if TProxy fails, TGW can schedule the loads to another healthy TProxy within one second; and the switching does not change the access VIP (virtual IP) so as to eliminate the impact caused by change of physical servers.

Backup and restore services

Backup service: the backup module is responsible for periodically creating (physical) backups and storing binary files (binlogs) of TencentDB for MariaDB. Backup files will be uploaded to a distributed file cluster (HDFS) with a higher level of security. Generally, backup is initiated on slave nodes to avoid impact on services provided by the master node.

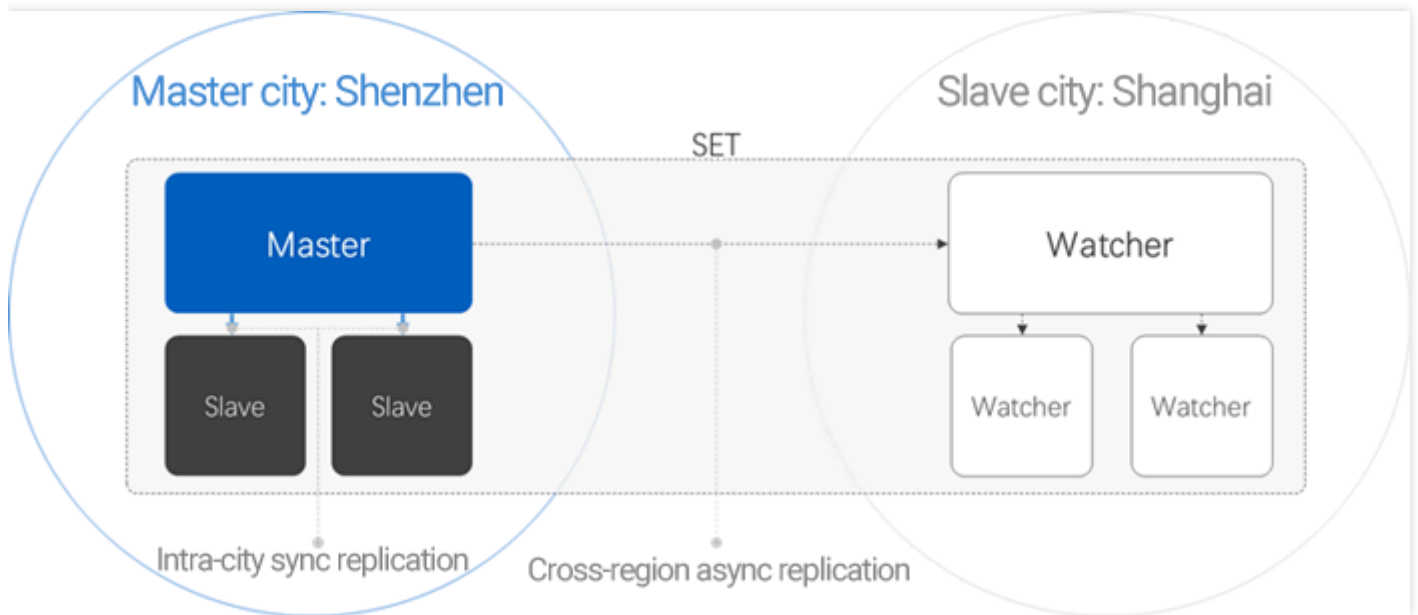
Restoration service: it is also known as rollback, where the restoration module restores backup files in HDFS to the temp instance for you to check or adjust without affecting operations of the master instance.

Backup download: you can transfer or download backup files to a specified location, e.g., lower-priced COS.

2-region-3-DC

2-region-3-DC deployment architecture of TencentDB for MariaDB: the straight-line distance between nodes in the same city is greater than 10 KM, and that between nodes in different cities is greater than 100 KM. This architecture

can be implemented by Tencent's proprietary high availability (HA) scheduling scheme as shown below:



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 kernels of TencentDB for MariaDB instances are not native ones but optimized by the TencentDB development team according to actual business needs. In addition, the default parameters are fine-tuned through Tencent Cloud's many years of production practices and continuously improved by professional DBAs, so that TencentDB for MariaDB can always run based on the 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

TencentDB for MariaDB uses the InnoDB storage engine and is highly compatible with MySQL 5.5/5.6, which means that code, applications, drivers, and tools that apply to MySQL databases can be directly used in TencentDB for MariaDB with no or only slight change required.

Cost Effectiveness and Ease of Use

- Out-of-the-box usage: you can customize TencentDB for MariaDB specifications at Tencent Cloud's official website. Once an order is placed, a TencentDB for MariaDB instance will be generated automatically. You can use it with CVM to reduce internet traffic fees and application response time.
- On-demand upgrade: at the initial stage of your business, you can purchase low-specced TencentDB for MariaDB instances to cope with business pressure. As database pressure increases and data volume changes, you have the flexibility to adjust instance specifications.
- Easy management: Tencent Cloud is responsible for routine maintenance and management of TencentDB for MariaDB instances, including without limitation software and hardware troubleshooting and database patch update, to ensure normal operations of the instances. You can also perform management operations in the Tencent Cloud Console on your own, such as adding, dropping, restarting, backing up, and restoring databases.

Use Cases

Last updated : 2020-04-02 11:31:40

Cloud-based Data Disaster Recovery (Remote Disaster Recovery)

Data is an important part of enterprise operations. Although the information technology brings convenience, it also reveals that electronic data and stored information are very vulnerable to damage or loss. Any incident, such as natural disaster, system failure, faulty operation, or virus, can cause complete interruption of business operations or even disastrous loss. Therefore, ensuring security and integrity of data, especially that in core databases, is a top priority of every enterprise.

Self-built data centers for off-site disaster recovery are usually expensive, as a great amount of hardware and software resources are required and continuous maintenance incurs high OPS expenses. Paying for incidents with small probability is certainly not in line with the principle of company operations.

TencentDB and Tencent Cloud's access products can be used to establish a data disaster recovery center directly in the cloud, which can sync data in the master IDC to the remote in-cloud backup center in real time over the secure private network. This scheme not only solves the problems of managing massive amounts of data, but also is highly cost-effective.

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 in your own data center. Your business systems and data will be synced securely over Direct Connect (or VPN) to implement 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.

- Read-only access can be implemented through SQL syntax or read-only account.
- 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 : 2021-04-29 15:01:16

Architecture Overview

High-availability architecture

In a production system, a high-availability scheme is usually required to ensure continuous operation of the system. As a database is the core of system data storage and service, its availability requirement is higher than that of computing services. Currently, in most database high-availability schemes, multiple databases are used together, so that whenever a database fails, another one can immediately take its place to eliminate or minimize service interruption; or, multiple databases provide services at the same time, so that users can access any of them, and whenever a database fails, access requests will be immediately directed to another one.

As a database records data in it, to switch between multiple databases, the data in them must be synchronous.

Therefore, **data sync is the foundation of database high-availability scheme**. Currently, data replication can be implemented in the following three modes:

- **Async replication:** an application initiates an update request such as adding, deletion, or modification. After completing the corresponding operation, the primary node (primary) responds to the application immediately and replicates data to the replica node (replica) asynchronously. Therefore, in async replication mode, an unavailable replica does not affect operations on the primary database, while an unavailable primary may cause data inconsistency.
- **Strong sync replication:** an application initiates an update request. After completing the operation, the primary replicates data to the replica immediately. After receiving the data, the replica returns a success message to the primary. Only after receiving the message from the replica will the primary respond to the application. The data is replicated synchronously from the primary to the replica. Therefore, an unavailable replica will affect operations on the primary database, while an unavailable primary will not cause data inconsistency.

Note :

When you perform strong sync replication, the primary database will be hanged if it is disconnected from the replica database or the replica database fails. If there is only one primary or replica database, the high-availability scheme is unavailable, because if only one single server is used, part of data will be lost completely when a failure occurs, which does not meet the requirements for finance-level data security.

- **Semi-sync replication:** proposed by Google, this sync method works in the following way: generally, strong sync replication is employed. When an exception occurs with the data replication from the primary to the replica (for example, the replica becomes unavailable or an exception occurs with the network connection between the two

nodes), the replication will be downgraded to async replication. When the replication returns to a normal state, strong sync replication will be restored from async replication. In semi-sync replication, there is a slight chance that a primary failure may cause data inconsistency.

Commonly used high-availability architectures

- **Shared storage scheme:** shared storage such as SAN storage is used. In SAN, multiple database servers share the same storage region, so that they can read/write the same copy of data. When the primary database fails, a third-party high-availability application can mount the file system to a replica database and start the replica to complete primary/replica switch.
- **Log sync or streaming replication sync:** this is the most common database replication mode for various databases such as MySQL. Whenever data is written, MySQL primary server will transfer its binary logs to the replica through the replication thread; then, the replica will write the same data as in the binary logs to the file system. Currently, MySQL provides the following replication modes:
 - 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 sync:** a trigger is used to record data changes which are then synced to another database.
- **Middleware-based sync:** the system is connected to a middleware program instead of multiple underlying databases, which sends data changes to underlying databases for data sync. In earlier years, due to issues such as business requirements, database performance, and sync mechanisms, some software developers usually used this architecture or similar ones.

TencentDB for MariaDB Architecture Overview

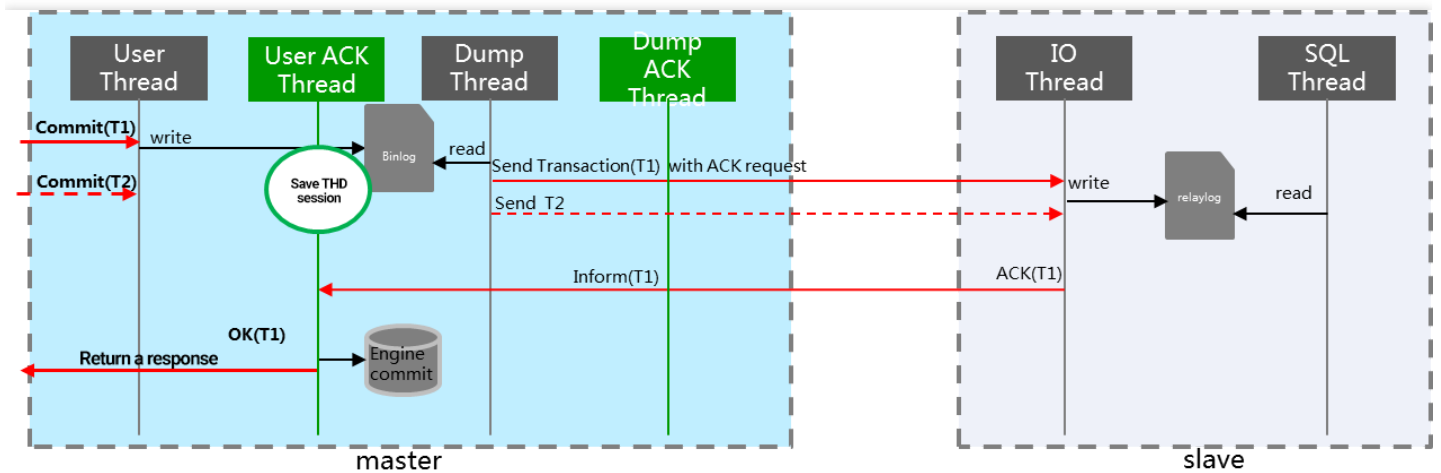
Multi-thread asynchronous replication

In the development of sync technology, schemes such as async and semi-sync replication emerged. These two schemes are oriented to general users and can ensure basic data sync when the user requirements are not high, network condition is good, and performance pressure is low. However, they generally tend to cause data inconsistency, which directly affects system reliability and even leads to transaction data loss, thus incurring direct or indirect economic loss.

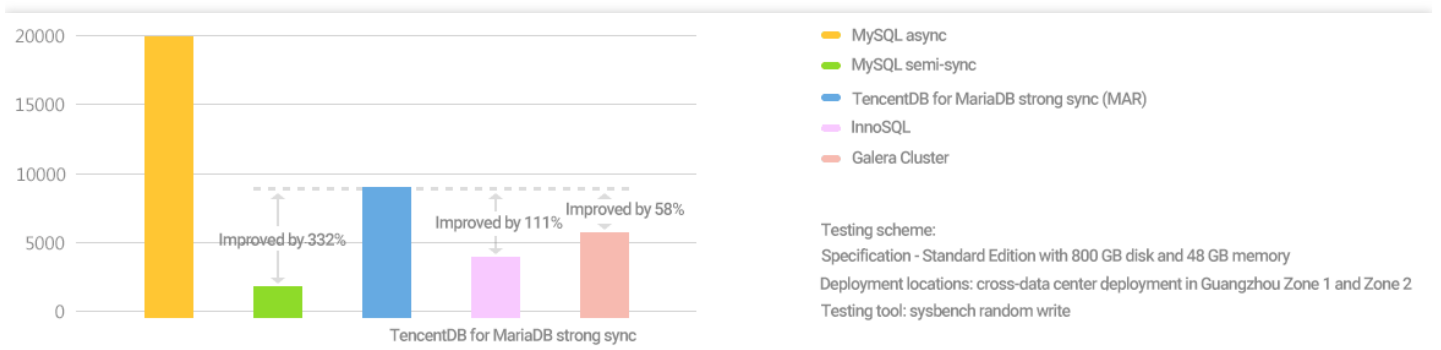
Backed by many years' experience in business operations, Tencent Cloud has developed an MAR (Multi-thread Async Replication) scheme for database sync. MAR outperforms the NDB engine of Oracle, Percona XtraDB Cluster, and MariaDB Galera Cluster in terms of performance, efficiency, and availability. The MAR strong sync scheme has the following advantages:

- Consistent sync replication ensures strong consistency of data between nodes.
- Complete transparency to the service. Read/write separation or synchronization enhancement is not required for the service.
- 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.
- Nodes can be automatically added without human intervention.
- Each node contains a complete replica of the data and the primary and replica nodes can switch 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:

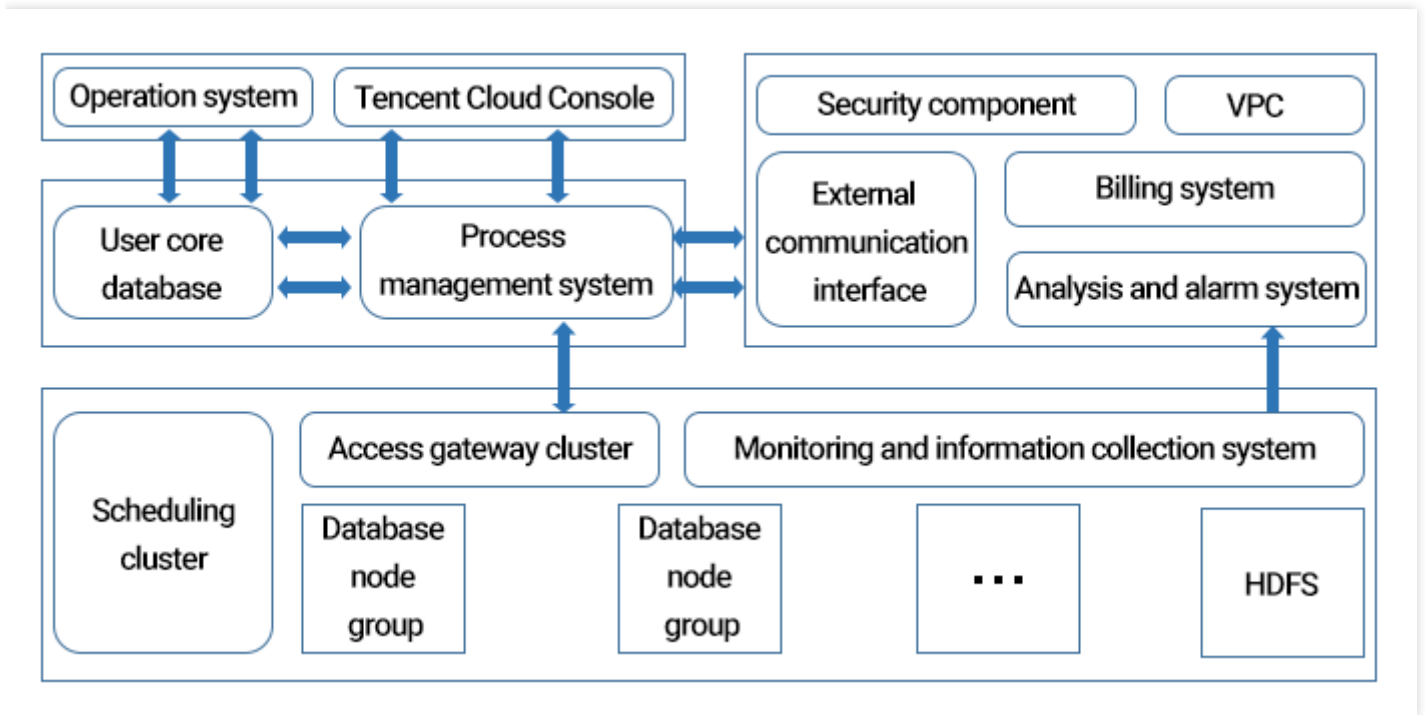


Its performance is higher than that of other mainstream sync schemes. Tests in the same cross-AZ (cross-IDC) testing configuration (with the standard sysbench use case) show that the performance of MAR is about 5 times higher than that of MySQL semi-sync replication and 1.5 times higher than that of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



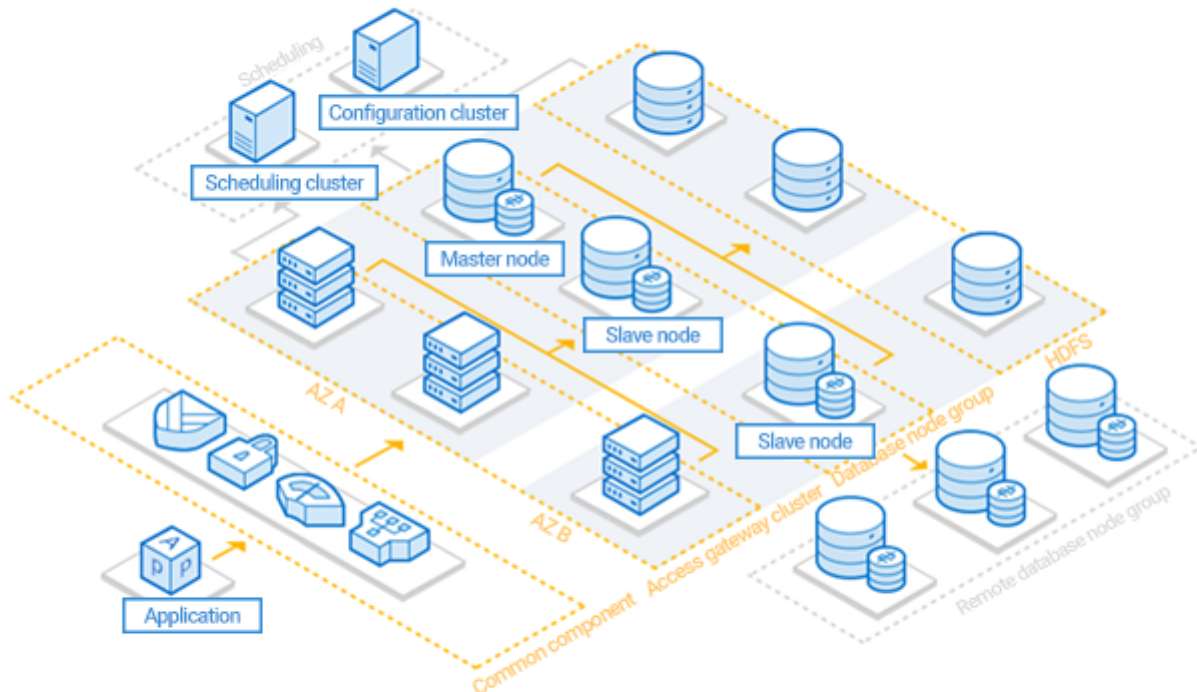
Cluster architecture

TencentDB for MariaDB adopts a cluster architecture. A set of independent TencentDB for MariaDB system consists of at least 10 systems or components as shown below:



The three most important modules in TencentDB for MariaDB are scheduling cluster (Tschedule), database node group (SET), and access gateway cluster (TProxy), and they communicate with one another through the configuration

cluster (ZooKeeper).



- **Database node group (SET):** it consists of the engine compatible with MySQL and monitoring and information collection tool (Tagent). It uses an architecture of "one primary node (Primary, several replica nodes (Replica_n), and several remote backup nodes (Watcher_n)". Generally:
 - 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.
- **Scheduling cluster (Tschedule):** this acts as the cluster management scheduling center and is mainly used to manage the normal operations of SET and record and distribute global database configurations. It includes the following components:
 - 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):** this manages SQL parsing and assigns routes at the network layer (it is not 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 : 2021-09-09 17:29:04

A database instance is a standalone database environment running in Tencent Cloud. It can contain multiple user-created databases and can be accessed using the same tools and applications as those for a standalone database instance.

There are four types of instances available in TencentDB for MariaDB.

| Instance Type | Definition | Feature |
|-----------------------------|---|---|
| Master instance | An instance that can be read from and written to | A master instance can mount disaster recovery instances for remote disaster recovery |
| Master instance (dedicated) | A master instance in dedicated cluster database mode | It is a database instance applied for in a dedicated cluster and has all features of a TencentDB master instance |
| Disaster recovery instance | An instance that enables 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 region different from the master instance |
| Temp instance | A temporary instance generated by rollback | It is used to verify and recovery data after rollback. |

Instance Architecture

Last updated : 2021-02-07 10:15:20

| Instance Architecture | Definition | Node | Features |
|--|--|---|--|
| Standard Edition (one primary and one replica) | Each shard provides a high-availability architecture based on primary/replica active-active deployment. | Two nodes: one primary node and one replica node | <ul style="list-style-type: none"> • Supports read-only replicas. • Automatic node failover. • Default sampling granularity for monitoring: once every 5 minutes. • Maximum backup time: 30 days. • Operation log backup time: 60 days. |
| Standard Edition (one primary and two replicas) | Each shard provides a high-availability architecture based on primary/replica multi-site active-active deployment. | Three nodes: one primary node and two replica nodes | <ul style="list-style-type: none"> • Supports database audit where audit logs can be retained for 15 days and an unlimited number of rules which can be configured. |
| Finance Edition (one primary and one replica) | Each shard provides a high-availability architecture based on primary/replica active-active deployment. | Two nodes: one primary node and one replica node | <ul style="list-style-type: none"> • Supports other deployment schemes. To use other schemes, please contact your Tencent Cloud sale rep. • Supports read-only replicas and implements intelligent load scheduling for read-only replicas. • Automatic node failover. • Default sampling granularity for monitoring: once every 1 minute. • Maximum backup time: 3,650 days. You need to submit a ticket for application. |
| Finance Edition (one primary and two replicas) | Each shard provides a high-availability architecture based on primary/replica multi-site active-active deployment. | Three nodes: one primary node and two replica nodes | <ul style="list-style-type: none"> • Operation log backup time: 60 days by default (1 year for archive storage). • Supports database audit where audit logs can be retained for 15 days. • Provides assistance for regulatory compliance. |

TDSQL Edition

Last updated : 2021-05-21 16:39:15

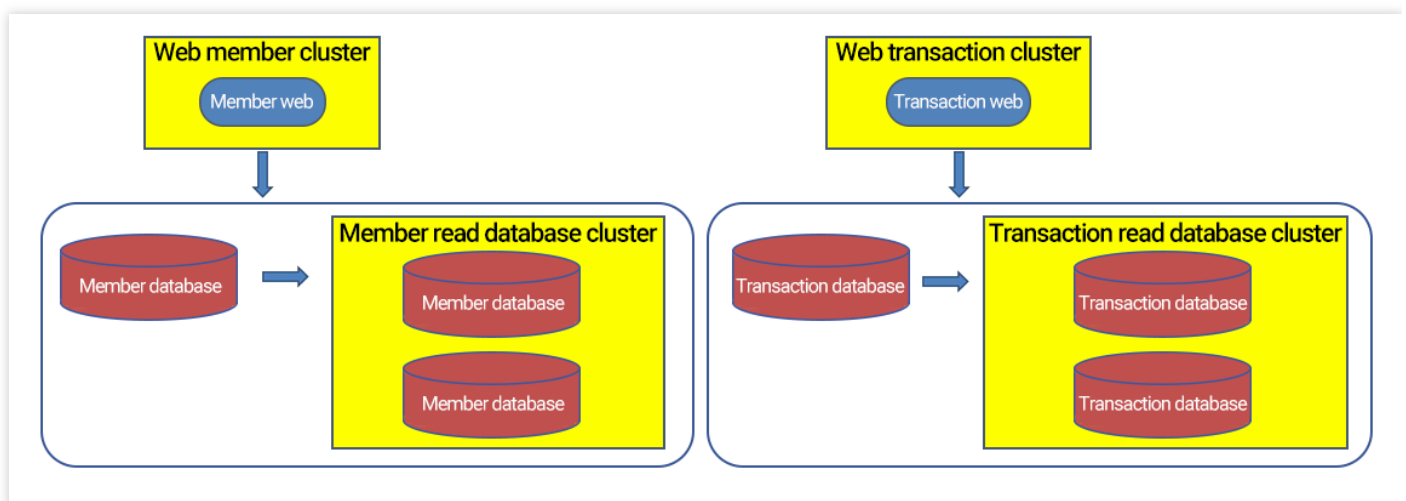
TDSQL for MariaDB is a distributed edition of TencentDB for MySQL. You can use TDSQL for MySQL if you anticipate that your business will grow so rapidly that the stand-alone edition, i.e. MariaDB, will no longer be able to support it.

Overview

Data sharding

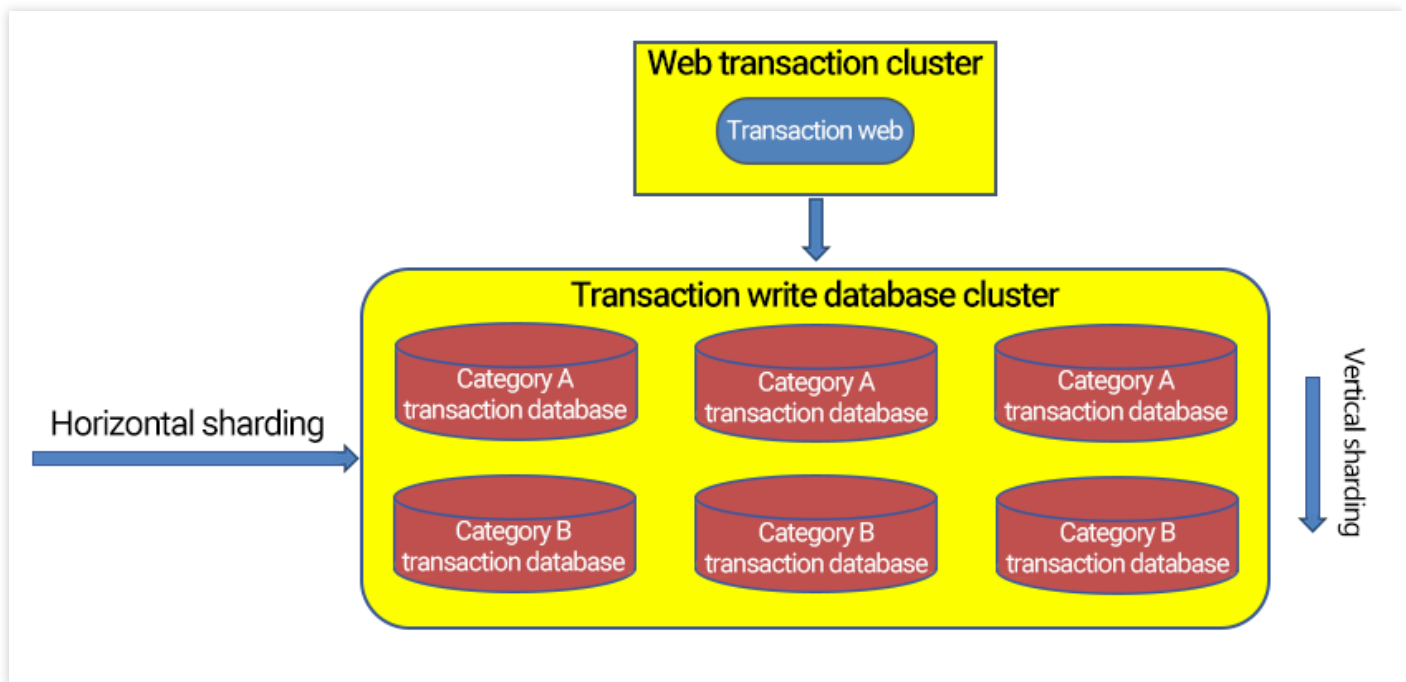
In a high-performance concurrent internet architecture, the performance bottleneck usually occurs in the database server, especially when the user base of the business reaches more than one million. In this case, you can cope with problems such as database performance and scalability by performing reasonable data sharding at the data layer. Database sharding can be performed in two dimensions: vertical sharding (by feature) and horizontal sharding.

- **Vertical sharding** is performed by functionality. It is closely associated with business type. For example, an ecommerce platform shards data by functionality into multiple databases for membership, commodity, transaction, logistics, etc. However, vertical sharding cannot completely handle the pressure as the load and capacity of a single database server is limited, which will become a bottleneck as the business grows. To deal with these problems, horizontal sharding is commonly used.



- **Horizontal sharding** is to split the data of a table across multiple physically independent database servers according to certain rules to form "separate" database "shards". Multiple shards together form a logically complete

database instance.



Sharding rules

A relational database is a two-dimensional model. To shard data, it is usually necessary to find a shardkey field to determine the sharding dimension. Then, a rule needs to be defined to actually shard the database. You need to comprehensively evaluate your business needs to find a suitable sharding rule. Several common sharding rules are described below:

1. Based on date order, such as sharding by year (one shard for 2015 and another for 2016).
 - Advantages: simple and easy to query.
 - Disadvantages: the server performance for the current hot data (e.g., data in 2016) 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.

TDSQL for MySQL

Automatic horizontal sharding (sharding databases and splitting tables)

TDSQL for MySQL is a distributed database service deployed in Tencent Cloud that is compatible with MySQL protocol and syntax and supports automatic sharding (horizontal sharding). With a distributed database, your business obtains a complete logical database table which is split and distributed evenly across multiple physical shard nodes on the backend. Currently, TDSQL deploys the primary-secondary architecture by default and provides a full set of solutions for disaster recovery, backup, restoration, monitoring, and migration, making it ideal for storing terabytes to petabytes of data.

Development of TDSQL can be traced back to 2004, when the internet-based value-added services of Tencent started to burst, which brought huge scaling pressure to the MySQL database. Therefore, the mechanism of database sharding and table splitting was introduced to solve a challenging problem: a huge table is pre-split into multiple subtables by `ShardKey`, which are spread across different physical server nodes. Nowadays, the volume of data stored on the TDSQL backend is very huge. Taking "Midas" as an example, TDSQL sustains 10 billion accounts in all channels of Midas, where the number of users is close to 900 million, and daily transaction amount exceeds 1 billion CNY.

TDSQL has the following advantages to easily sustain massive amounts of business requests:

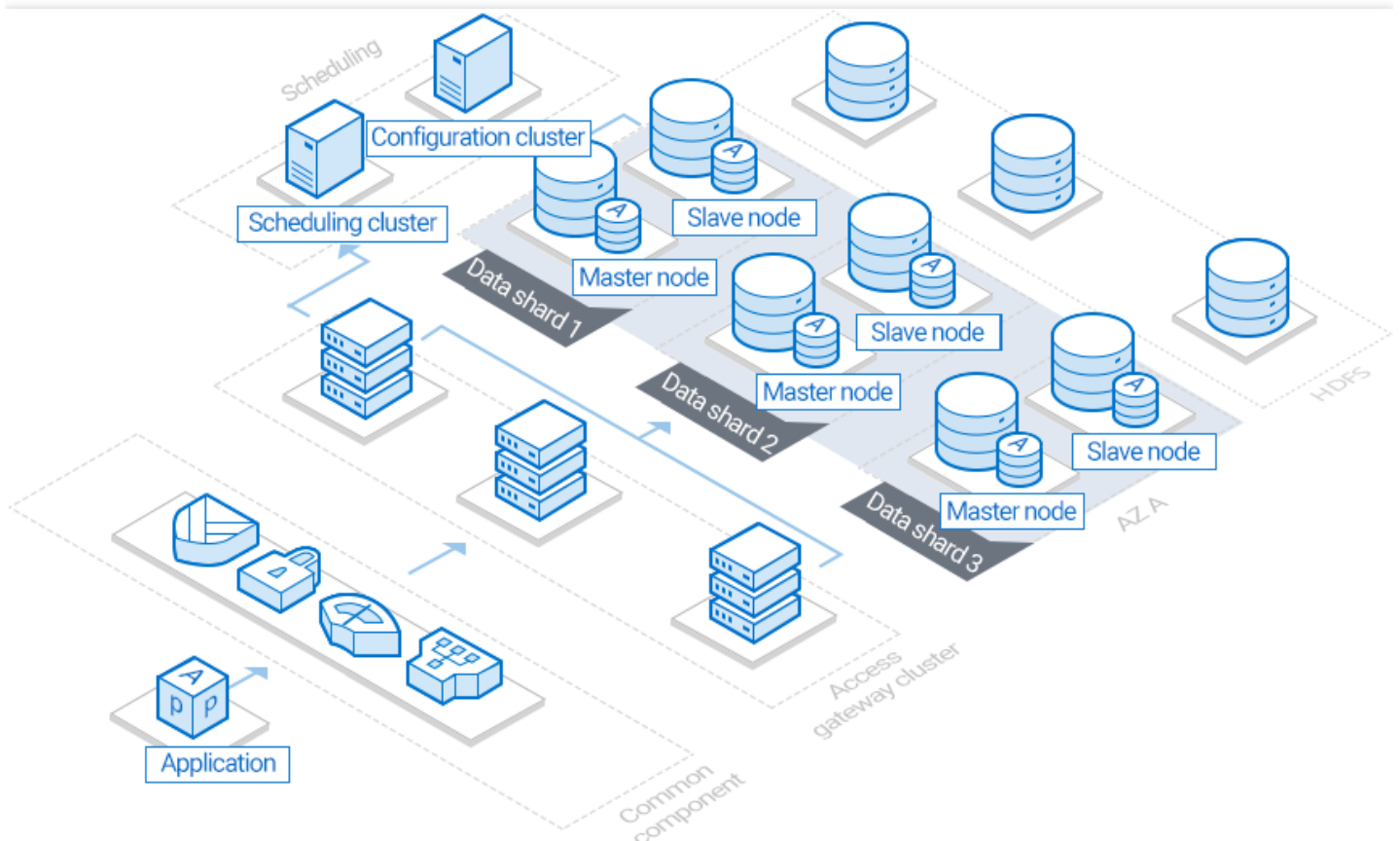
- **Automatic table splitting:** TDSQL supports automatic database sharding and table splitting and implements on-demand capacity scaling together with a unified data scheduling mechanism. As TDSQL blocks internal details of database sharding and table splitting by using a gateway, you do not need to worry about how to shard data and route request waits. You only need to initialize the sharding field (shardkey), develop programs directly for logical tables, and focus on implementation of business logics, which greatly reduces the program complexity.
- **Automatic failover:** for IoT, big data, or payment services, any business with massive amounts of data requires high availability of backend storage databases. As a common solution, failover needs inspection and cooperation of the business and is deeply coupled with the business, which results in a complex switch process that even may require human intervention. After the business is restored, faulty data that may appear during failover has to be repaired manually, which requires much OPS labor. In contrast, TDSQL data nodes and gateways implement multi-

point disaster recovery and automatically check the running status of instances. As soon as the primary node fails, primary/secondary failover will be triggered to ensure high database availability in disastrous situations such as primary failure, network failure, and IDC failure. This failover process is totally imperceptible to the business and does not need human intervention, greatly simplifying OPS while ensuring a smooth user experience.

- **High data consistency:** this feature is very helpful if you have zero tolerance for data loss or disorder. TDSQL adopts an innovative multi-thread strong sync replication mechanism based on the original async and semi-sync replication of MySQL, which ensures that there are at least two replicas in a cluster for every and each transaction before a response is returned to user. Then, it uses a series of failover mechanisms to make sure that data will not get lost or disorganized upon failover in case of node failure.
- **Cluster-based management and auto scaling:** the peak number of business requests will surge by several or dozens of times because of launch of new features or promotion campaigns. In the past, you needed to know the business trends and have the DBA manually scale out the database in advance. Generally, the scaling process of most distributed databases is complex and prone to errors as it requires a lot of manual operations. In contrast, TDSQL features automatic deployment, auto scaling, automatic backup and restore, scheduled data rollback, and multi-dimensional monitoring at the cluster level. When scaling is required, the DBA only needs to click some buttons to initiate the scaling process on the frontend for auto scaling. The cluster-based operation system of TDSQL greatly improves the DBA work efficiency and reduces mistakes caused by manual operations.

TDSQL architecture

The architecture of a TDSQL instance is as follows:



Data shard: compatible with open-source MySQL engine and supports the monitoring feature and data collector (Tagent).

说明：

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: this acts as the cluster management scheduling center and is mainly used to manage the normal operations of SET and record and distribute global database configurations.

Access gateway cluster (TPProxy): this manages SQL parsing and assigns routes at the network layer, which can be seen as middleware in an open-source distributed database.

说明：

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

Backup cluster: this is a backup cluster of TencentDB data.

说明：

TDSQL backups are stored for 7 days by default.