TDSQL-A for PostgreSQL

Product Introduction

Product Documentation

Contents

Product Introduction
Overview

Last updated : 2021-07-02 16:16:07

TDSQL-A for PostgreSQL is Tencent's proprietary distributed analytic database system based on a shared-nothing architecture. It is compatible with SQL:2011 standards, PostgreSQL syntax, and Oracle syntax. Its proprietary column storage engine supports row storage and column storage as well as hybrid storage with a high compression ratio. Its new-generation vectorized execution engine can provide high-performance and efficient real-time query and analysis capabilities for massive amounts of data.
In addition, as a competitive enterprise-grade data warehouse service, TDSQL-A for PostgreSQL supports complete distributed transaction processing, multi-level disaster recovery, and multidimensional resource isolation and provides a powerful multi-level security system, auto scaling capabilities, and comprehensive enterprise-grade management capabilities. It offers a complete set of solutions covering disaster recovery, backup and restoration, monitoring, security, and audit, making it ideal for online analytical processing (OLAP) scenarios containing gigabytes to petabytes of data.

Features

Hybrid row/column storage

To better provide OLAP capabilities, on the basis of compatibility with row storage of the PostgreSQL ecosystem, TDSQL-A for PostgreSQL uses a proprietary column storage engine to provide complete column storage capabilities. You can select an appropriate storage format for the data to be written to the database based on your actual business needs so as to enjoy efficient hybrid row/column query capabilities.
Column storage provides powerful compression capabilities, including transparent compression and lightweight compression. The former supports compression algorithms such as zlib and zstd, and the latter supports delta, RLE, and bitpack algorithms. The algorithms can be automatically adjusted and optimized according to the data characteristics for more efficient compression with a compression ratio of up to 400:1.

Efficient processing of complex queries

The proprietary new-generation vectorized execution engine of TDSQL-A for PostgreSQL has efficient processing capabilities for complex queries. It can return results for trillions of correlated subqueries within seconds, with a performance several times to even hundreds of times higher than that of open-source and traditional data warehouses.

Smooth business migration

TDSQL-A for PostgreSQL is compatible with SQL:2011 syntax specifications, PostgreSQL syntax, and Oracle syntax and is equipped with the Tencent DBbridge migration tool, helping you migrate your business system to TDSQL-A for PostgreSQL as smoothly as possible.

Enterprise-Grade data security

TDSQL-A for PostgreSQL has a permission division mechanism of three roles (security admin, audit admin, and data admin) and provides multi-level policies such as data storage encryption, data masking, forced access control, and data audit to ensure data security comprehensively.

Complete distributed transactions

TDSQL-A for PostgreSQL has complete transaction ACID capabilities to guarantee the global transaction consistency. It uses GTM to manage distributed transactions and uses the proprietary patented distributed transaction consistency technology to ensure the data consistency and efficiency in the distributed architecture.

Rich ecosystem support

TDSQL-A for PostgreSQL has a comprehensive peripheral ecosystem:

- It supports powerful geographic information system (GIS). The clusterized PostGIS extension makes it a spatial database that can store spatial and geographic data and support efficient spatial data management, quantitative measurement, and geometric topology analysis through SQL statements.
- It not only is a distributed relational database but also supports JSON, a non-relational data type.
- It supports the foreign data wrapper (FDW) feature, which implements a part of the SQL/MED specifications and allows you to access data that resides outside PostgreSQL by using regular SQL queries.
  The FDW feature provides a set of programming APIs, with which you can perform extension-based secondary development and establish a data channel between external data sources and databases. In most cases, you can use `oracle_fdw` , `mysql_fdw` , `postgres_fdw` , `redis_fdw` and `mongodb_fdw` of non-relational databases, and `hive_fdw` and `hdfs_fdw` of big data databases. Based on FDW and existing extensions, TDSQL-A for PostgreSQL provides powerful database federation capabilities, enabling you to access data from multiple existing data sources.
- It offers data migration and sync services and tools to help you easily sync source data in different deployment modes to it, such as Tencent Cloud, self-built database, and other cloud databases. The high stability and performance of this sync feature ensure an integrated data experience.

Features

Last updated : 2021-07-06 14:48:35

Support for Column Storage and Multiple Compression Algorithms

TDSQL-A for PostgreSQL supports column storage. You can define data tables as columnar tables based on your business needs. Generally, we recommend you set tables with a large width and requirements for a high compression ratio as columnar tables.
Columnar tables support a wide variety of compression algorithms, such as delta, zlib, zstd, RLE, and bitpack, which have different compression levels. For more information, please see the corresponding section in the development guide. TDSQL-A for PostgreSQL supports the new-generation columnar vectorized execution engine, which offers high query performance for hybrid row/column storage and query.

Efficient Distributed JOIN Computing

Business analysis scenarios generally have logical JOIN operations between two or more tables. JOIN is a simple operation in standalone mode, but in cluster mode, as data is distributed among one or multiple physical nodes, the processing will be more complicated. In many distributed solutions, JOIN pulls data to one node for correlation and computing, which not only wastes many network resources but also dramatically prolongs statement executions.

Based on the efficient global query plan and data redistribution technology, TDSQL-A for PostgreSQL can take full advantage of parallel computing and complete the joins efficiently. It uses the following methods to efficiently compute distributed JOIN operations:

- In terms of the execution method, the CN receives your SQL requests, generates the optimal cluster-level distributed query plan based on the collected cluster statistics, and distributes it to the DNs participating in the computation for execution, that is, the CN sends the execution plan, and the DNs are responsible for executing the plan.
- In terms of data interaction, efficient data exchange tunnels are established between DNs, enabling efficient data exchange. The process of data exchange is called data redistribution in TDSQL-A for PostgreSQL.

Multi-Core Parallel Computing

TDSQL-A for PostgreSQL uses parallel computing inside nodes, where multiple processes are started to collaboratively execute the same query. This makes full use of the processing capabilities of multiple server cores to execute queries more quickly and efficiently. Generally, TDSQL-A for PostgreSQL starts multiple processes for one query to shorten the query time, and the more the resources, the linearly shorter the query time.
TDSQL-A for PostgreSQL decides whether to perform parallel computing based on the table size. Only when the table size exceeds the threshold will parallel computing be used, and in this case, the concurrency (i.e., the number of required processes) will be calculated based on the table size.

Data Security Protection

Data encryption

TDSQL-A for PostgreSQL provides two data encryption methods:

- Encryption on the business side: the business calls the encryption function built in TDSQL-A for PostgreSQL and writes encrypted results to the database. Normally, the encrypted data will be read and decrypted in the application.
- Encryption built in TDSQL-A for PostgreSQL: the encryption process is imperceptible to the business, which has the following advantages:
  - Encryption operations (function calls) are decoupled from the business. The business is only responsible for writing the original data to the database kernel, and the subsequent encryption calculation will be conducted inside the database, which is imperceptible to the business.
  - Encryption algorithms are maintained by the database, and operations such as encryption algorithm selection and key management are all performed by the security admin independently.
    Kernel-based encryption calculation supports async encryption to implement data encryption while delivering a stable system throughput. The supported encryption algorithms include AES-128, AES-192, AES-256.

Data masking

TDSQL-A for PostgreSQL supports transparent data masking that can return masked data to unauthorized users in an imperceptible manner.
Finer-Grained data access control can be implemented in the aforementioned two dimensions to enhance the control over existing access requests in a way that is imperceptible to the business system.

Comprehensive audit

TDSQL-A for PostgreSQL supports all-round auditing in multiple dimensions. Bypass detection is used for auditing, which has little impact on database operations. The following types of audits are available:

- Statement audit: audits a certain type of statements.
- Object audit: audits the operations of a certain database object.
- User audit: audits the operations of a certain database user.
- Fine-Grained audit (FGA): uses expressions as audit conditions and allows you to set actions when audit is triggered, such as sending emails or making phone calls.

Hot/Cold data separation

The kernel natively supports hot/cold data separation, so that the business can provide a unified database view with no need to perceive the differences between underlying storage media.

- Hot data and cold data are stored in different node groups with different physical server configurations, so as to implement hot/cold data separation and reduce the costs.
- Scheduled backend tasks can automatically migrate data according to the configured hot/cold data rules. In this way, the system can implement automated hot/cold data separation, and the business doesn't need to care about the storage of hot/cold data in the cluster.
  This feature is currently available on the Private Cloud Edition but unavailable on the Public Cloud Edition.

Multi-Level Disaster Recovery

TDSQL-A for PostgreSQL ensures the cluster disaster recovery capabilities in multiple dimensions:

Strong sync replication

TDSQL-A for PostgreSQL supports strong sync replication. Ensuring that the primary and standby nodes have the identical data is the basis of the entire disaster recovery system. If the primary node fails, the database service can be switched to the standby node without any data loss. The strong sync replication mechanism requires that a success be returned only after the user request is executed and the log is successfully written to the standby node, guaranteeing that the data on the primary and standby nodes are always consistent.

Primary-Standby high availability

The primary-standby high availability scheme of TDSQL-A for PostgreSQL mainly uses multi-replica redundancy in each node group to ensure that there are no or only momentary service interruptions.

If the primary node in a group fails and cannot be recovered, a new primary node will be automatically selected from the corresponding standby nodes to continue service provision. Based on primary/standby high availability, TDSQL-A for PostgreSQL supports the following features:

- Automated failover: if the primary node in the cluster fails, the system will automatically select a new primary node from the corresponding standby nodes, and the failed node will be automatically isolated. The strong sync replication policy ensures complete primary/standby data consistency in case of primary/standby failover, fully meeting the finance-grade requirements for data consistency.
- Failure recovery: if a standby node loses data due to disk failure, the database admin (DBA) can recover the standby node by building it again or add a standby server to a new physical node to recover the primary/standby relationship and thus ensure the system reliability.
- Replica switch: each node in the primary/standby architecture (which can contain one primary node and multiple standby nodes) has a complete data replica that can be switched to by the DBA as needed.
- Do-Not-Switch configuration: it can be set that failover will not be performed during the specified period of time.
- Cross-AZ deployment: even if the primary and standby nodes are in different data centers, the data can be replicated through Direct Connect in real time. If the local node is the primary and the remote node is the standby, the local node will be accessed first, and if it fails or becomes unreachable, the remote standby node will be promoted to the primary node to provide service.

TDSQL-A for PostgreSQL supports the high availability scheme based on strong sync replication. If the primary node fails, the system will automatically select the optimal standby node immediately to take over the tasks. The switch process is imperceptible to users, and the access IP remains unchanged. TDSQL-A for PostgreSQL offers 24/7 continuous monitoring for system components, and if there is a failure, it will automatically restart or isolate the failed node and select a new primary node from the standby nodes to continue service provision.

Support for full and incremental backup modes

TDSQL-A for PostgreSQL supports backup-based data restoration at the time points when transaction consistency is guaranteed so as to avoid data loss due to misoperations. Backup modes include full backup (cold backup) and incremental backup (xlog backup).

- Full backup: all database data excluding operations logs and xlogs is backed up. This mode is usually periodic, such as daily, weekly, or once every N days.
- Incremental backup: incremental data is backed up, which is generally implemented through xlog files. After generating an xlog file, the database system backs up the file onto the backup server.

This mode is generally in real time.

If an incident or disaster occurs, you can use backup data to recover the system.

Strengths

Last updated : 2021-07-02 16:16:07

Global Consistency of Distributed Transactions

TDSQL-A for PostgreSQL adopts the global transaction manager (GTM) to dedicatedly ensure the consistency of distributed transactions, that is, it uses two-phase commit and global transaction management policies to guarantee transaction consistency in fully distributed environments. In addition, it provides a distributed transaction reliability guarantee mechanism to avoid problems such as resource blockage, data inconsistency, and CN failures.

High Compatibility with SQL

TDSQL-A for PostgreSQL is compatible with SQL:2011 specifications and has a great advantage in SQL compatibility. It is compatible with most PostgreSQL syntaxes, such as complex query, foreign key, trigger, view, and stored procedure, meeting the needs of most enterprise users. It is also compatible with most Oracle data types and functions, facilitating business migration from Oracle data warehouses to TDSQL-A for PostgreSQL databases.

Hybrid Row/Column Storage

On the basis of compatibility with row storage of the PostgreSQL ecosystem, TDSQL-A for PostgreSQL uses a proprietary column storage engine to provide complete column storage capabilities. You can select an appropriate storage format for the data to be written to the database based on your actual business needs.

The column storage of TDSQL-A for PostgreSQL provides powerful compression capabilities, including transparent compression and lightweight compression. The former supports compression algorithms such as zlib and zstd, and the latter supports delta, RLE, and bitpack algorithms. The algorithms can perform efficient compression according to the data characteristics with a compression ratio of up to 400+.

Efficient Processing of Complex Queries

The proprietary new-generation vectorized execution engine of TDSQL-A for PostgreSQL has efficient processing capabilities for complex queries. It can return results for trillions of correlated subqueries within seconds, with a performance several times to even hundreds of times higher than that of open-source and traditional data warehouses.

Multi-Level Security Policies

The superuser privileges in traditional database system are too powerful and difficult to be controlled, which goes against the construction of a database security system. TDSQL-A for PostgreSQL uses a permission separation system where the traditional DBA role is divided into three independent roles: security admin, audit admin, and data admin. The security admin can configure data encryption rules based on the actual business needs to prevent data leakage. In addition, security features such as transparent data encryption and data masking are available.

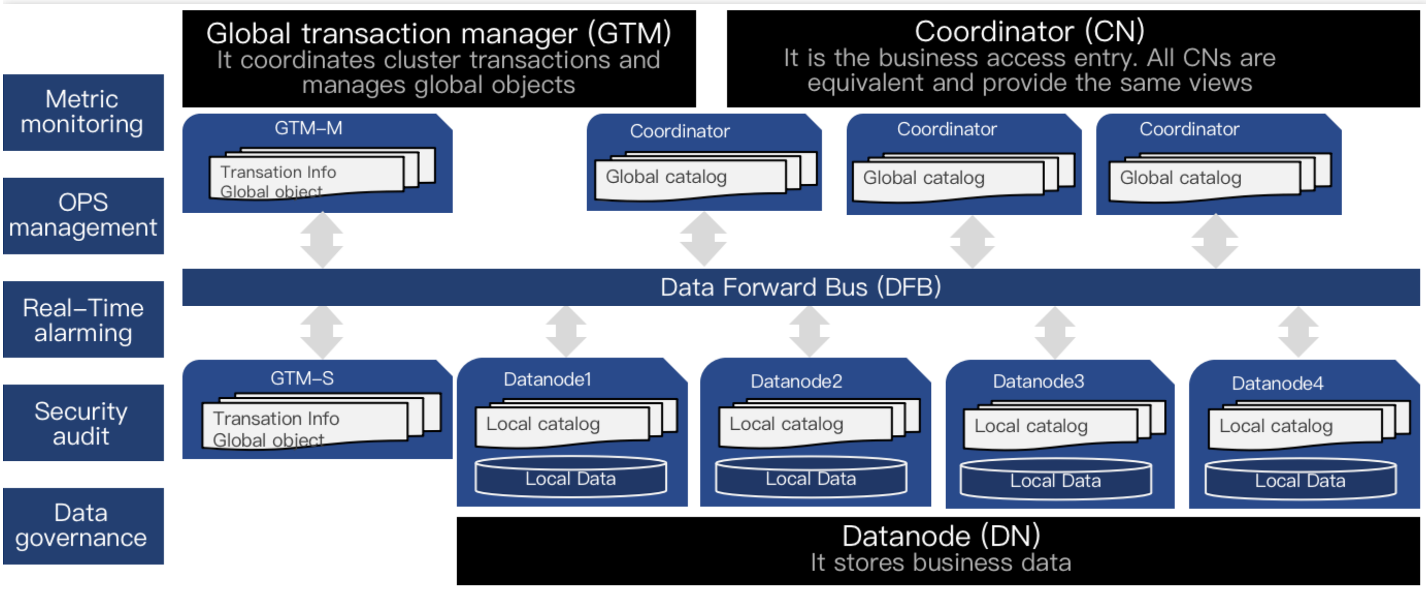
Comprehensive Peripheral Ecosystem

- TDSQL-A for PostgreSQL embraces the PostgreSQL ecosystem and keeps up with the community development. It supports a rich variety of ecosystem tools such as the PostGIS extension, JSON unstructured data type, and interconnection with other data sources through foreign data wrapper (FDW).
- Other data sources can be synced to TDSQL-A for PostgreSQL through a data migration service or product.

Architecture

Last updated : 2021-07-02 16:16:07

TDSQL-A for PostgreSQL uses a shared-nothing architecture where the nodes are independent from each other and process their own data separately. The processing results can be aggregated at the upper layer or transferred among the nodes, and the processing units communicate with each other over network protocols, offering better parallel processing and scaling capabilities. This means that you can deploy a TDSQL-A for PostgreSQL database cluster simply on x86 or ARM servers. The architecture is as follows:



Each module is as detailed below:

- Coordinator (CN): it provides APIs and is responsible for data distribution, query planning, pairing of multiple node locations, etc. Each node provides the same database views. In terms of functionality, it stores only the global metadata of the system rather than actual business data.
- Datanode (DN): it processes and stores the metadata related to itself and shards of the business data. In terms of functionality, it executes the execution requests distributed by the CN.
- Global Transaction Manager (GTM): it manages cluster transaction information and global cluster objects such as sequences.
- Data Forward Bus: it consists of FNs on servers in the cluster. The main purpose of using FNs is to reduce the number of connections created during data exchange between DNs and between CNs and DNs, thus ensuring that connection will not become a bottleneck for large-scale clusters.

In this architecture, a cluster has the following capabilities:

- Multi-Site active-active/multi-primary: all CNs provide the same cluster views, so that you can write through any of them. The cluster topology is imperceptible to the business.
- Read/Write scaling: data is sharded onto different DNs, so the cluster read/write capabilities can be scaled as the cluster size increases.
- Cluster write consistency: the write transactions of the business that occur on one CN node will be consistently presented on other CN nodes, as if these transactions occurred on them.
- Transparent cluster architecture: the data is distributed on different DNs. During data query, the business does not need to care about on which nodes the data resides.

The shared-nothing architecture of TDSQL-A for PostgreSQL makes business access easier and lowers the threshold for business access.

Use Cases

Last updated : 2021-07-08 11:03:21

Data Warehouse

Based on the shared-nothing architecture, TDSQL-A for PostgreSQL can smoothly scale the cluster, sustain gigabytes to petabytes of data, and use its fully parallel architecture and vectorized execution engine to efficiently process JOIN queries involving tens of billions of rows in multiple tables. This makes it ideal for operational data stores (ODSs), enterprise data warehouses (EDWs), and data marts (DMs).

Storage of Massive Amounts of Data and Real-time Online Analysis

With the rapid increase in the number of internet users and long-term operations of the system, more and more data has been accumulated, incurring high storage costs in some industries (such as payment businesses, where data must be stored persistently for the purpose of regulatory compliance) and bringing urgent performance problems of complex correlated subqueries in scenarios with high data volumes.

The online linear scaling capabilities of TDSQL-A for PostgreSQL can scale the cluster as needed to ensure that the cluster can store petabytes of data and automatically move infrequently accessed historical business data to cheaper storage devices, striking a balance between performance and cost.

System Dependent on High Data Security

In government, civic service, and finance industries, a lot of personal information and financial transaction data are stored, and data security is the top priority, because once the data is lost or leaked, it will cause incalculable losses. Therefore, this type of businesses is highly dependent on the security of the database system that stores core data, including data query result encryption, data storage encryption, and post-transaction audit.

TDSQL-A for PostgreSQL can provide multi-level security policies to protect the security of such systems.

Multi-Point Aggregation Business System

Governments, banks, and large-sized state-owned enterprises usually use an HQ-branch-division organization architecture. As a result, some of their core IT systems are also built in this architecture, yet different divisions may use different databases. As the needs for business interconnection, personnel communication, and information exchange become increasingly stronger, businesses are gradually aggregated to the HQ. Therefore, whether data can be aggregated efficiently is a very important metric of the system.

TDSQL-A for PostgreSQL has efficient heterogeneous database replication capabilities to share data properly among multiple databases.

Oracle Elimination

In recent years, there has been a growing trend for eliminating IBM, Oracle, and EMC services from IT systems in various industries, among which Oracle elimination is relatively difficult.

As a highly scalable database cluster service, TDSQL-A for PostgreSQL is compatible with PostgreSQL syntaxes and most Oracle syntaxes, and further offers various enterprise-grade features such as stored procedure, window function, and unstructured data, making it an ideal substitute for Oracle databases. Therefore, using TDSQL-A for PostgreSQL to replace Oracle data warehouse applications can greatly reduce the migration costs.

Product Specifications

Last updated : 2021-07-02 16:16:07

Currently, the specifications of TDSQL-A for PostgreSQL trial instance are specified as follows:

Node	Specification	Primary-Standby Configuration
GTM	1 vCPU core, 4 GB memory, and 20 GB disk	One set with one primary node and one standby node
CN	1 vCPU core, 4 GB memory, and 20 GB disk	Two sets with one primary node and one standby node each
DN	1 vCPU core, 8 GB memory, and 300 GB disk	Four sets with one primary node and one standby node each
FN	1-core vCPU, 4 GB memory, and 20 GB disk	One on each server where an instance resides

If you want to scale the cluster during use, please submit a ticket for assistance.