

TDMQ for Pulsar

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



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

- Use Limits

- Comparison with Open-Source Edition

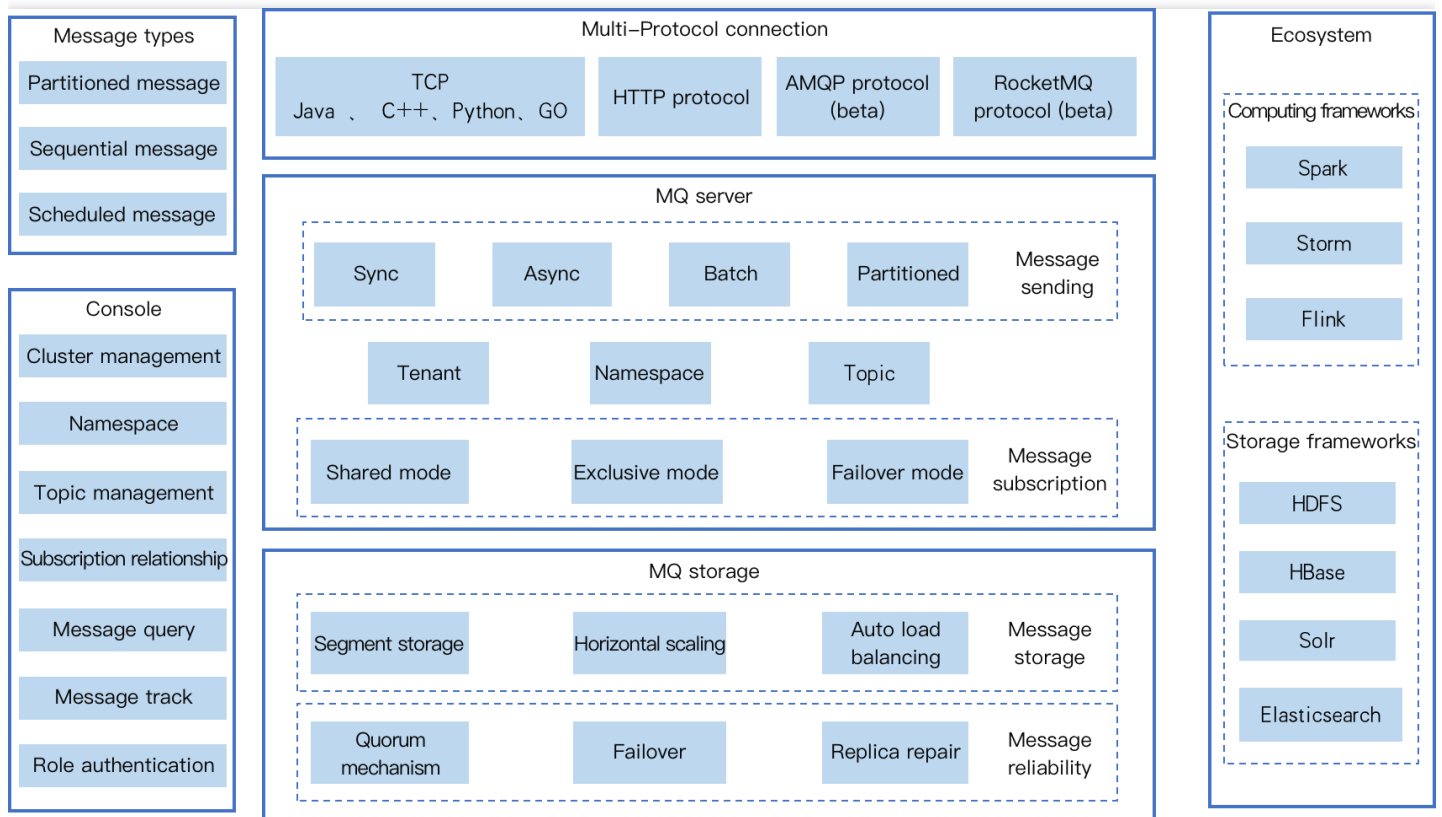
Product Introduction

Overview

Last updated : 2022-02-11 10:25:34

TDMQ for Pulsar is Tencent's proprietary messaging middleware based on Apache Pulsar. It comes with excellent cloud native and serverless features and is compatible with all components and principles of Pulsar. It also has the underlying benefits of computing-storage separation and flexible scaling.

TDMQ for Pulsar can add async decoupling and peak shifting capabilities to distributed application systems. It features massive message retention, high throughput, and reliable retry mechanism required by internet applications. Currently, it is widely used in Tencent's most billing scenarios, including primary payment process as well as real-time reconciliation, monitoring, and big data analysis.



Features

- High consistency, reliability, and concurrency
- Storage-Service separation and support for dynamic horizontal scaling

- Seamless scalability out to over a million topics
- Very low publish and end-to-end latency
- Multiple subscription modes for topics (exclusive, shared, and failover)
- A serverless lightweight computing framework Pulsar Functions offers the capability for stream-native data processing
- Native support for multiple clusters in a Pulsar instance, with seamless geo-replication of messages across clusters

Strengths

Last updated : 2022-02-11 10:25:51

Strong Data Consistency

By writing message data backups to various physical machines with sync flush, TDMQ for Pulsar achieves strong data consistency (like with the Raft algorithm) by using [BookKeeper consistency protocol](#). When one of the physical machines fails, the backend data replication mechanism can quickly migrate the data to guarantee data backups are available.

High Performance and Low Latency

With over 100,000 QPS per cluster, TDMQ for Pulsar can easily maintain the production and consumption of millions of messages, as well as retain an unlimited number of messages. It well sustains Tencent's all billing scenarios. It also offers a duration protection mechanism to ensure minimal latency and help you easily meet business performance requirements.

Millions of Topics

TDMQ for Pulsar's computing and storage structures are designed to be independent of one another, allowing it to support millions of message topics with ease. When compared to other message queue products on the market, the performance of a TDMQ for Pulsar cluster will not suffer much as the number of topics increases.

Rich Diversity of Message Types

TDMQ for Pulsar offers a rich diversity of message types, such as general, sequential (global and partitioned), distributed transaction, and scheduled messages, meeting the requirements for advanced features in various demanding scenarios.

Unlimited Consumers

Different from Kafka's message consumption pattern, the number of consumers is not limited by the number of topics in TDMQ for Pulsar, and the quantity of messages per consumer is balanced using algorithms. Businesses can start with the appropriate number of consumers as needed.

Multi-Protocol Connection

TDMQ for Pulsar provides a client API with language bindings for Java, Go, and C++. It also supports HTTP protocol for extended accessibility. It can be connected from open-source RocketMQ and RabbitMQ clients. If you only use its basic features to produce and consume messages, you can swiftly migrate to it with no code modifications required.

Isolation Control

TDMQ for Pulsar offers a mechanism of topic isolation by tenant. It accurately controls the production and consumption speeds of each tenant, prevents the tenants from affecting each other, and ensures that message processing won't cause resource competition.

Global Deployment

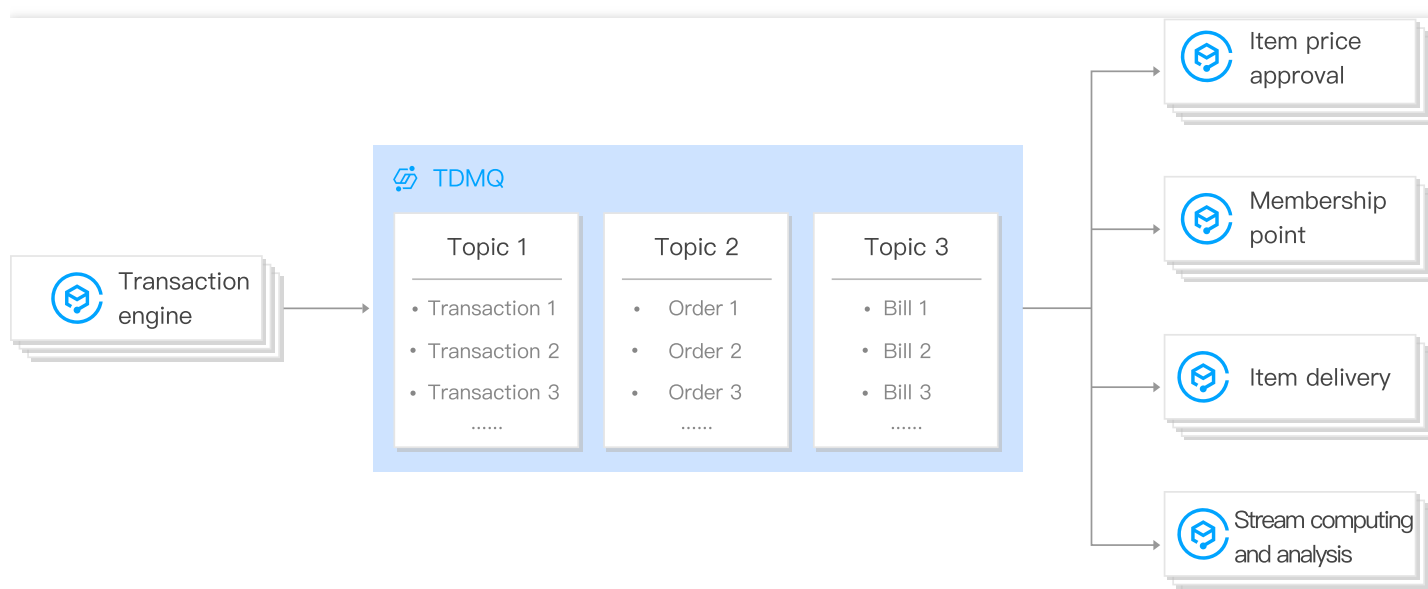
TDMQ for Pulsar furnishes global deployment capabilities, so you can choose a region close to your business presence for nearby access.

Use Cases

Last updated : 2022-02-11 10:26:02

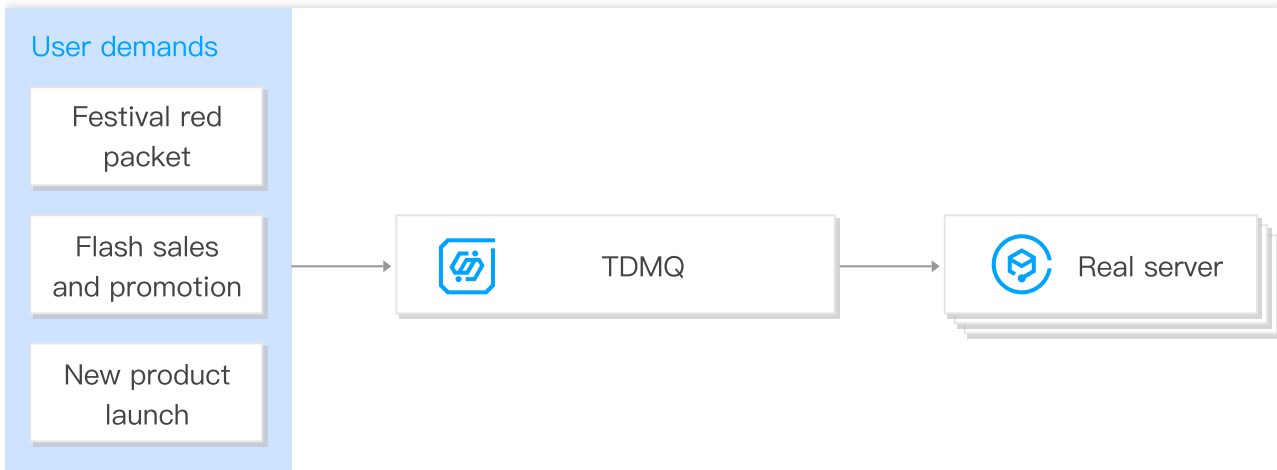
Async Decoupling

The transaction engine is the core system of Tencent billing. The data of each transaction order needs to be monitored by dozens of downstream business systems, including item price approval, delivery, reward point, and stream computing analysis. Such systems use different message processing logic, making it impossible for a single system to adapt to all associated business. In this case, TDMQ for Pulsar can implement efficient async communication and application decoupling to ensure the business continuity of the primary site.



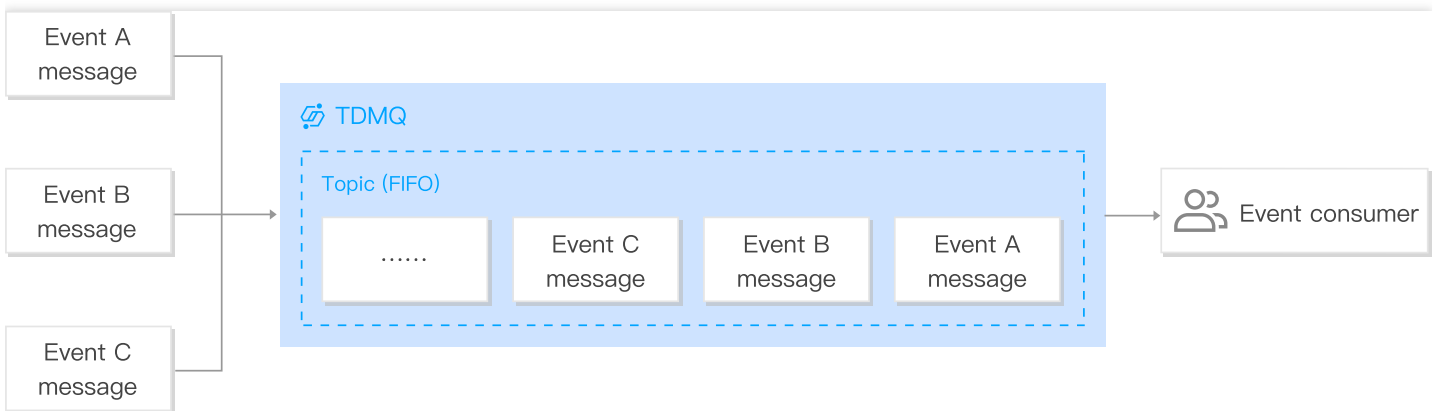
Peak Shifting

Companies hold promotional campaigns such as new product launch and festival red packet grabbing from time to time, which often cause temporary traffic spikes and pose huge challenges to each backend application system. In this case, TDMQ for Pulsar can act as a buffer to centrally collect the suddenly increased requests in the upstream, allowing downstream businesses to consume the request messages based on their actual processing capacities.



Sequential Message Sending/Receiving

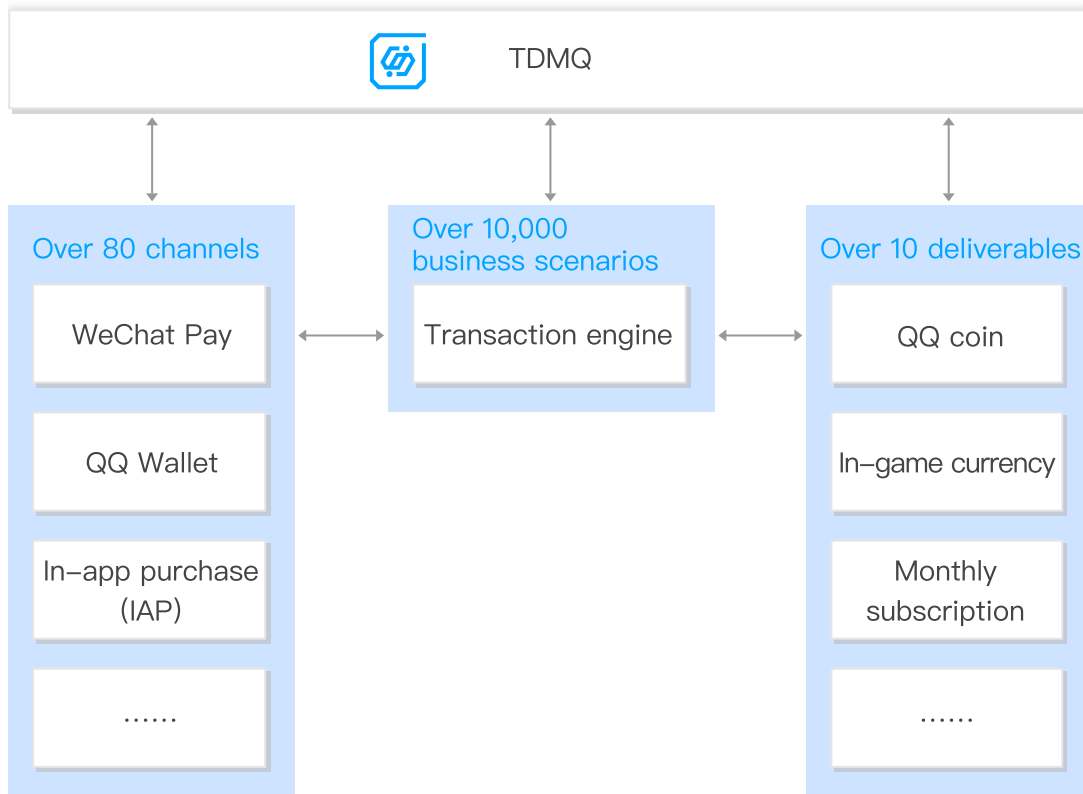
Sequential messages are used in some business scenarios, such as order creation, payment, delivery, and refund of in-app/game items, which are all strictly executed in sequence. Similar to the First In, First Out (FIFO) principle, TDMQ for Pulsar offers a sequential message feature dedicated to such scenarios to ensure message FIFO.



Consistency of Distributed Transactions

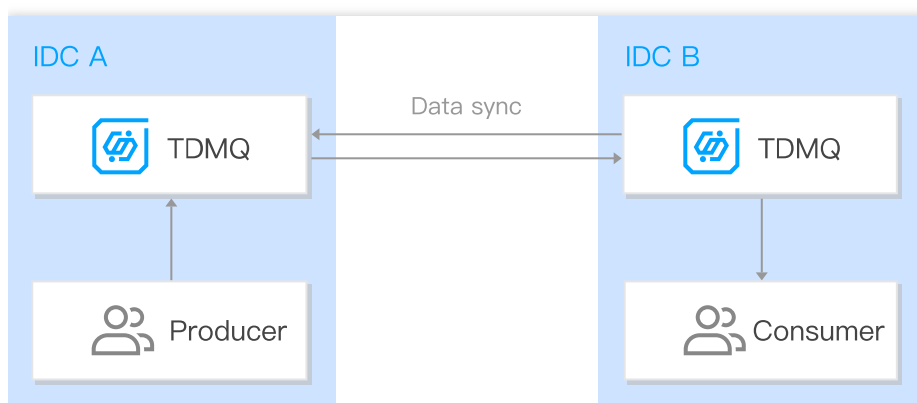
Tencent Billing (Midas) is an internet billing platform that incubates and sustains Tencent businesses' revenue of hundreds of billions of CNY and handles amounts up to hundreds of millions of CNY per day. It solves the core problem of money-item consistency and uses TDMQ for Pulsar and distributed transactions to process business transactions, which greatly improve the efficiency and performance. A billing system often has a long transaction linkage with a significant chance of error or timeout. TDMQ for Pulsar's automated repush and abundant message

retention features can be used to provide transaction compensation, and the eventual consistency of payment tips notifications and transaction pushes can also be achieved through TDMQ for Pulsar.



Data Sync

TDMQ for Pulsar can easily implement cross-IDC sync if messages need to be consumed across many IDCs.



Big Data Analysis

Data creates value in the "flow". Most traditional data analysis are based on batch computing models, which means they cannot analyze data in real time. In contrast, TDMQ for Pulsar can easily implement real-time analysis of business data when combined with a stream computing engine.

Use Limits

Last updated : 2022-07-29 18:50:16

This document lists the limits of certain metrics and performance in TDMQ for Pulsar. Be careful not to exceed the limits during use to avoid exceptions.

Cluster

Limit	Description
Maximum number of clusters per region	5
Maximum cluster name length	128 characters
Maximum storage capacity	100 GB

Namespace

Limit	Description
Maximum number of namespaces per cluster	100

Topic

Limit	Description
Maximum number of topics per cluster	1,000
Maximum number of topic partitions per cluster	6,000
Maximum number of partitions per topic	32
Maximum topic name length	128 characters
Maximum production TPS per topic partition	5,000
Maximum production bandwidth per topic partition	40 Mbps
Maximum consumption TPS per topic partition	5,000
Maximum consumption bandwidth per topic partition	40 Mbps
Maximum number of producers per topic	1,000
Maximum number of subscriptions per topic	2,000

Limit	Description
Maximum number of consumers per topic	2,000

Message

Limit	Description
Maximum message retention period	15 days
Maximum message delay	10 days
Maximum message size	5 MB
Consumption offset reset	15 days
Maximum number of received but unacknowledged messages	5,000
Maximum number of messages per query	65,536

Comparison with Open-Source Edition

Last updated : 2022-05-20 17:13:57

The performance comparison between TDMQ for Pulsar and open-source Apache Pulsar is as follows:

Item	TDMQ for Pulsar	Apache Pulsar
Strengths	<ul style="list-style-type: none"> • It supports on-demand usage and the pay-as-you-go billing mode, eliminating your need to care about configuration. • It is Ops-free, eliminating your need to care about the underlying components. • It can send/receive messages over the HTTP protocol via TencentCloud API, which is easy to use. • It is natively interconnected with basic capabilities such as Tencent Cloud CM and CAM. • It has a high SLA, and its parameters are fine-tuned in a targeted manner. 	<ul style="list-style-type: none"> • It offers new open-source features in time. • It allows configuring some parameters more flexibly.
Shortcomings	<p>It generally takes two to three months to add a new open-source feature.</p>	<ul style="list-style-type: none"> • It depends on a large number of components and therefore has a high Ops workload. • It doesn't provide an SLA. • Its security protection capabilities are limited. • It cannot be configured precisely, which causes resource waste.

Item	TDMQ for Pulsar	Apache Pulsar
General performance	Hundreds of thousands of QPS	Hundreds of thousands of QPS
Stress test on 2-core 4 GB MEM server	100,000 QPS for write to three replicas (average message size of 1 KB)	80,000 QPS for write to three replicas (average message size of 1 KB)
Costs	It supports the pay-as-you-go billing mode, which is elastic, has a reasonable resource usage, makes costs controllable, and requires no manual Ops.	Self-built message queues cannot be used elastically and have a low resource utilization but high costs. They require manual maintenance and incur high Ops costs.
Scalability	It is very flexible and easy to scale. You don't need to pay attention to the scaling process and can fully utilize the scale effect to sustain sudden high loads.	Broker nodes can be flexibly added, but it is complex to manually scale out a BookKeeper cluster, during which maloperations can easily occur and affect the data.
Availability	It is deployed across multiple AZs and stores messages in three replicas in different AZs. Tencent Cloud guarantees an availability of above 99.95% and provides cluster traffic throttling and optimization to prevent the cluster from being crashed by high traffic.	You need to deploy it in different regions to guarantee the availability and ensure the cluster availability in case of a high traffic load on your own.

Item	TDMQ for Pulsar	Apache Pulsar
Security protection	It natively supports security protection capabilities by using Tencent Cloud security products.	You need to install and configure open-source plugins.
Monitoring and alarming	It natively supports monitoring and alarming capabilities with the aid of CM.	You need to install and configure open-source plugins.