

# **TDMQ for Apache Pulsar**

## **Product Introduction**

## **Product Documentation**



## Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

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

## Trademark Notice



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

## Service Statement

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

# Contents

## Product Introduction

Overview

Strengths

Basic Concepts

Use Cases

Product Selection

Use Limits

Comparison with Apache Pulsar

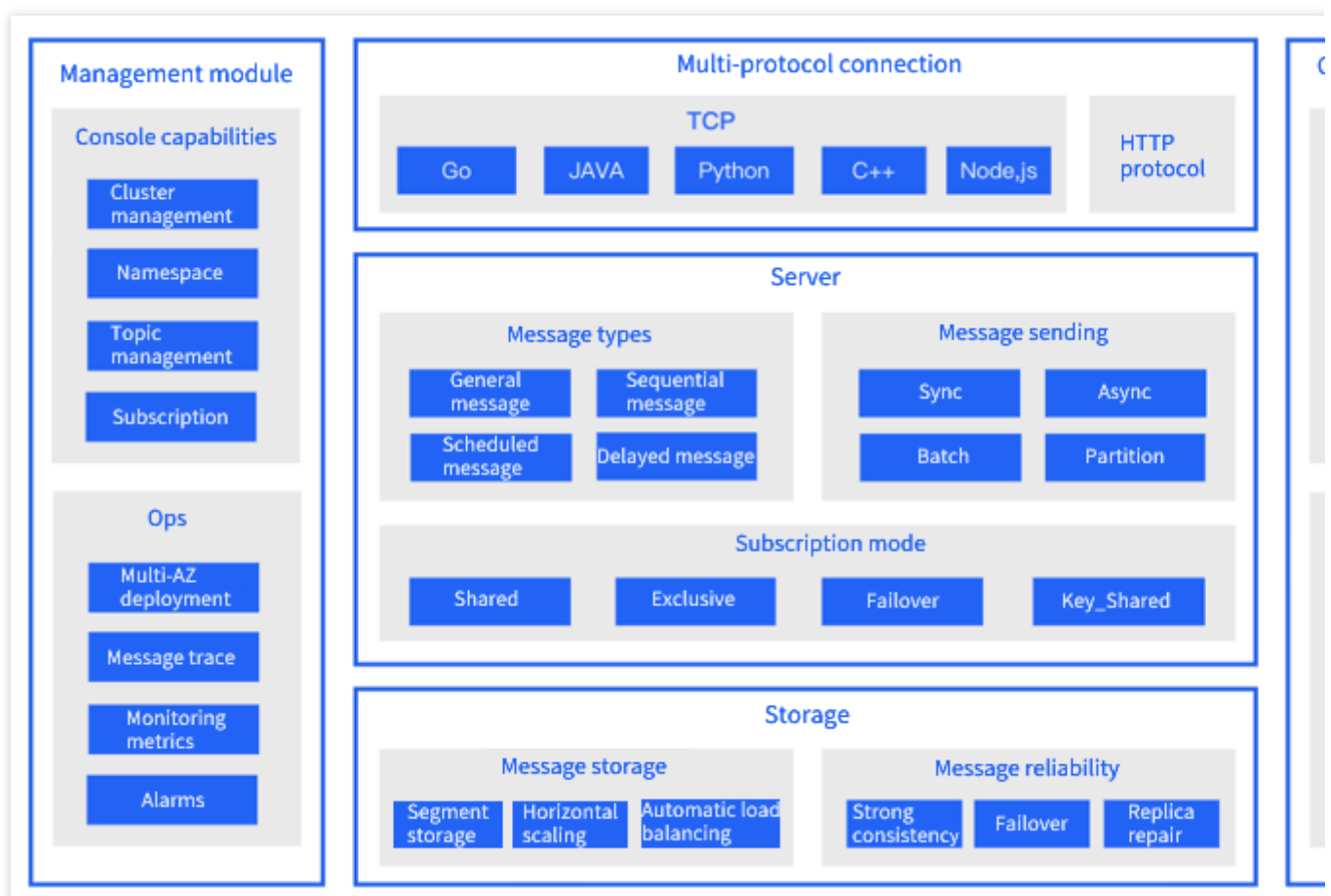
# Product Introduction

## Overview

Last updated : 2024-06-28 11:29:49

TDMQ for Apache 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 compute-storage separation and flexible scaling.

TDMQ for Apache 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

Compute-storage separation and support for dynamic horizontal scaling

Seamless scalability out to over a million topics

Very low publishing 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 : 2024-06-28 11:29:49

## Strong Data Consistency

By writing message data backups to various physical machines with sync flush, TDMQ for Apache 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 Apache 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 Apache 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 Apache Pulsar cluster will not suffer much as the number of topics increases.

## Rich Diversity of Message Types

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

# Basic Concepts

Last updated : 2024-08-19 15:59:14

## Message Classification

In the message queue, based on the characteristics and use cases of messages, messages can be classified as follows:

Message Type	Consumption Order	Performance	Applicable Cases
Normal Messages	No Order	Best	Huge throughput with no requirements for production and consumption orders
Local Ordered Messages	All messages in the same partition follow the First In, First Out (FIFO) rule.	Better	High throughput, ordered within the same partition, and unordered across different partitions
Global Ordered Messages	All messages in the same Topic follow the First In, First Out (FIFO) rule.	General	General throughput, globally ordered, and single partition
Dead Letter Messages	-	-	Messages that cannot be consumed normally.
Scheduled Messages	Messages are consumed after being sent to the server and delayed until a certain time point.	There is an error of about 1 second.	Consume the message at a specified time point.
Delayed Messages	Messages are consumed after being sent to the server and delayed for a period of time.	There is an error of about 1 second.	Consume the message after a specified time interval.

### Normal Messages

Normal message is a basic message type, where a message is delivered to the specified Topic by the producer and then consumed by the consumer subscribed to the Topic. There is no concept of order in the Topic of a normal message, and multiple partitions can be used to improve the efficiency of message production and consumption, and its performance is best when the throughput is huge.

### Local Ordered Messages



Compared with normal messages, local ordered messages have an additional characteristic of local order. Within the same partition, consumers consume messages strictly in the order they are delivered to the partition by the producer. While ensuring a certain order, local ordered messages retain the partition mechanism to improve performance. However, they cannot guarantee the order across different partitions.

### **Global Ordered Messages**

The most distinctive characteristic of globally sequential messages is that they ensure messages are consumed in the strict order they are delivered by the producer. Therefore, it uses a single partition to process messages, and users cannot customize the number of partition. Compared with the other two message types, this type has lower performance.

### **Dead Letter Messages**

A dead letter message is a message that cannot be consumed normally. The TDMQ for Apache Pulsar will automatically create a dead letter queue to process such messages when creating a new subscription (when a consumer subscribes to a Topic).

### **Scheduled Messages**

After a message is sent to the server, the actual business does not want the consumer to receive the message immediately, but rather delay it to a certain point in time. These messages are collectively referred to as scheduled messages, which can be seen as a special usage of delayed messages, achieving the same final effect as delayed messages.

### **Delayed Messages**

After a message is sent to the server, the actual business does not want the consumer to receive the message immediately but rather delay it for a period of time. These messages are collectively referred to as delayed messages.

## **Retry Queue**

A retry queue is designed to ensure that messages are consumed normally. If no normal response is received after a message is consumed by the consumer for the first time, it will enter the retry queue, and when the retries reach a certain number of times, the retries are stopped and delivered to the dead letter queue.

In actual cases, messages may not be processed promptly due to temporary issues such as network jitter and service restart, and the retry mechanism of the retry letter topic can be a good solution in this case.

## **Dead Letter Queue**

A dead letter queue is a special type of message queue used to centrally process messages that cannot be consumed normally. If a message cannot be consumed after a specified number of retries in the retry queue, the TDMQ for Apache Pulsar will determine that the message cannot be consumed under the current situation and deliver it to the dead letter queue.

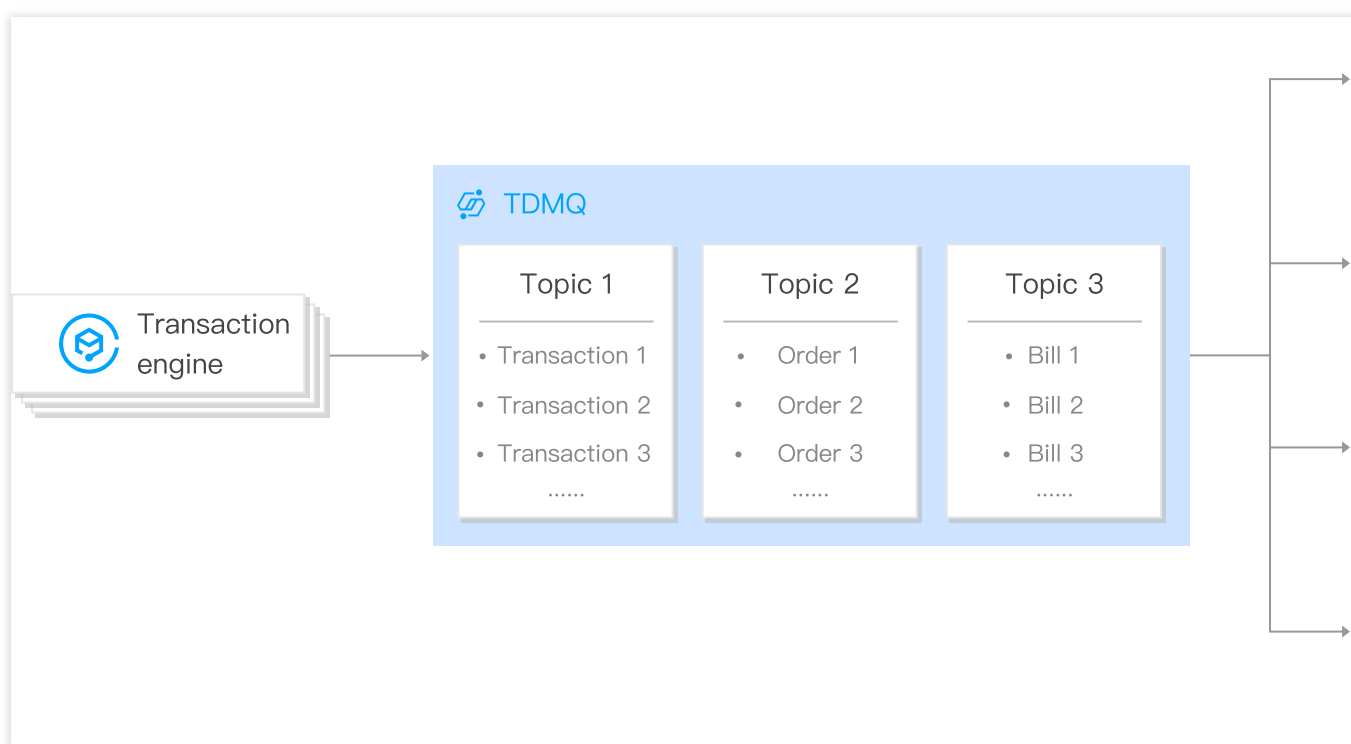
In actual cases, messages may not be consumed due to prolonged service downtime or network disconnection. In such cases, messages will not be discarded immediately; instead, they will be persisted for a longer period in the dead letter queue. Once a solution is found, users can create a consumer subscription to the dead letter queue.

# Use Cases

Last updated : 2024-06-28 11:29:49

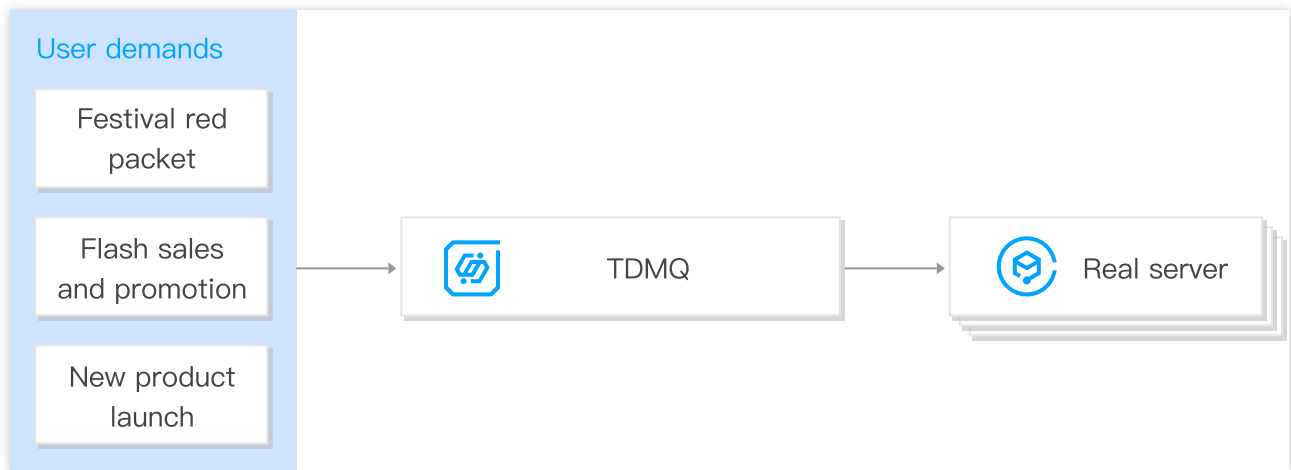
## Async Decoupling

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 Apache Pulsar can implement efficient async communication and application decoupling to ensure the business continuity of the primary site.



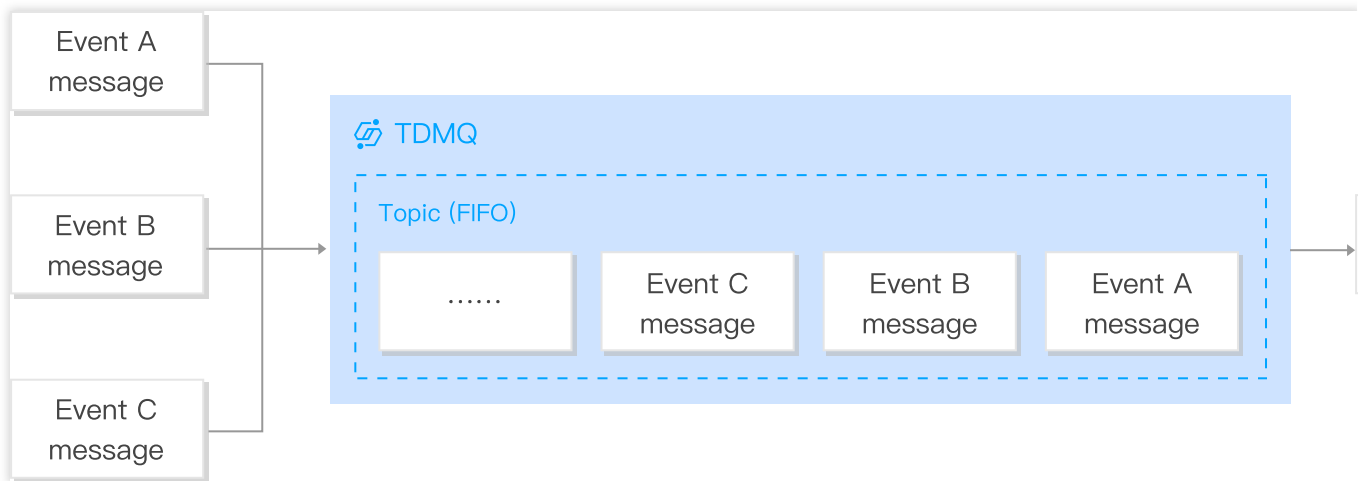
## Peak Shifting

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 Apache 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

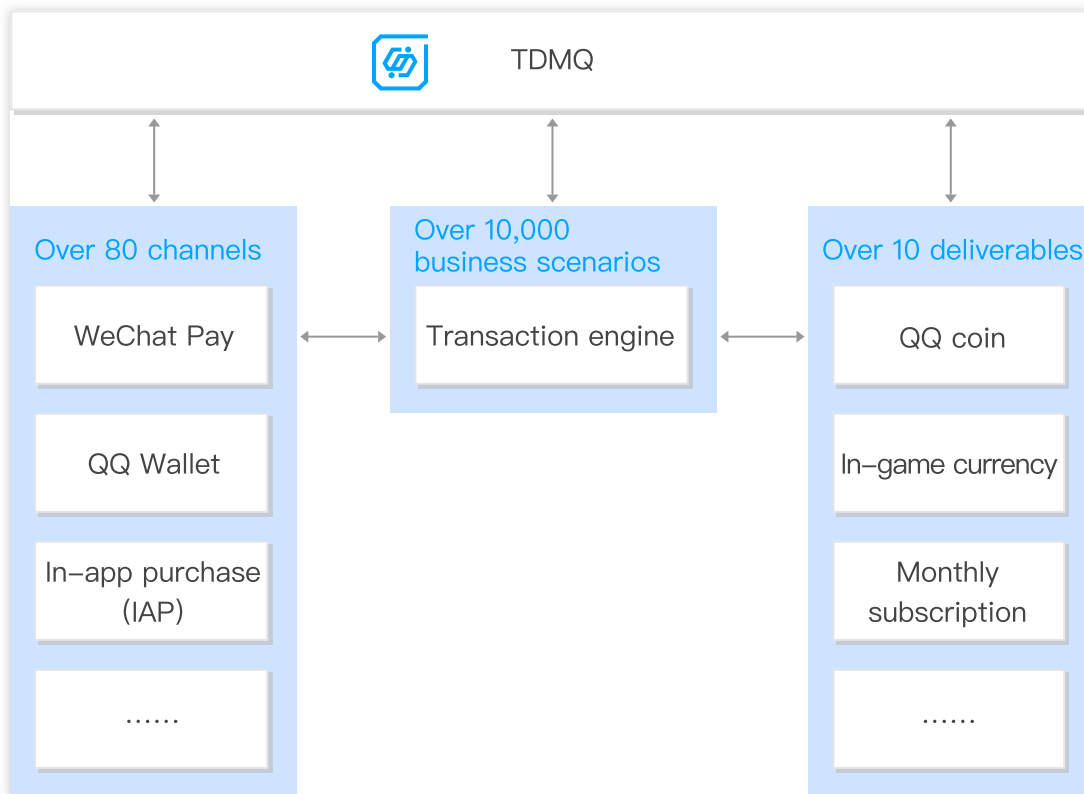
Sequential messages are used in some business scenarios, such as order creation, payment, delivery, and refund of in-app/game items, which are all strictly executed in sequence. Similar to the First In, First Out (FIFO) principle, TDMQ for Apache Pulsar offers a sequential message feature dedicated to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



## Consistency of Distributed Transactions

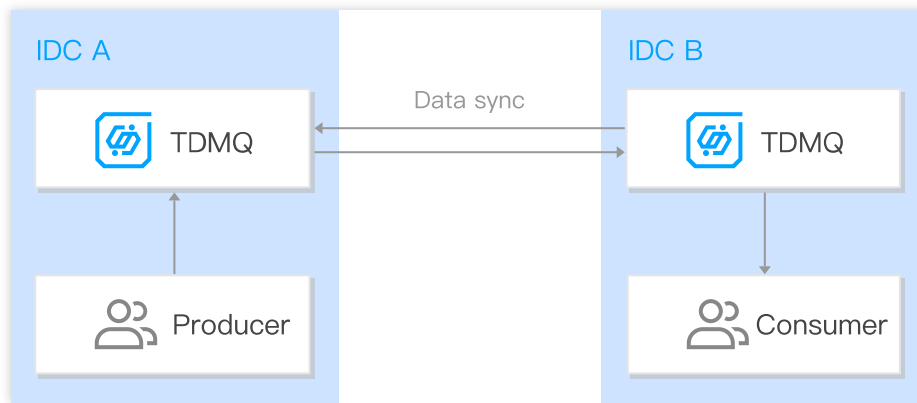
Tencent Billing (Midas) is an internet billing platform that incubates and sustains Tencent businesses' revenue of hundreds of billions of CNY and handles amounts up to hundreds of millions of CNY per day. It solves the core

problem of money-item consistency and uses TDMQ for Apache 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 Apache 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 Apache Pulsar.



## Data Sync

TDMQ for Apache 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 Apache Pulsar can easily implement real-time analysis of business data when combined with a stream computing engine.

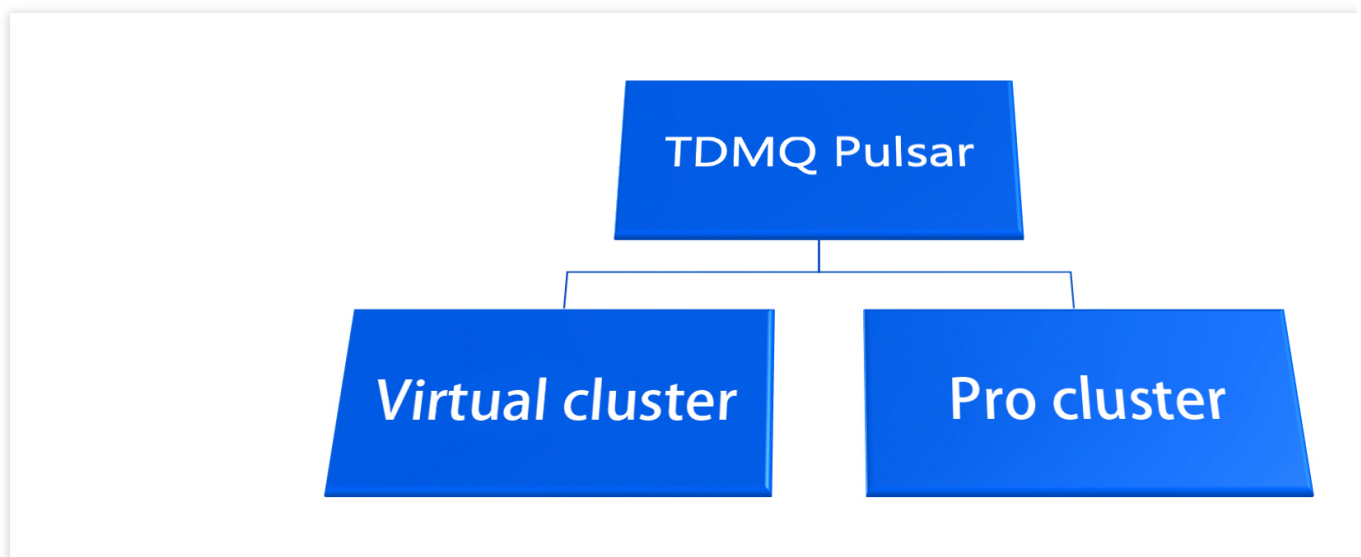
# Product Selection

Last updated : 2024-06-28 11:29:49

To meet diverse requirements in different scenarios, TDMQ for Apache Pulsar provides pro and virtual clusters. We recommend you consider the business scenario, product capabilities, and use costs when purchasing a cluster.

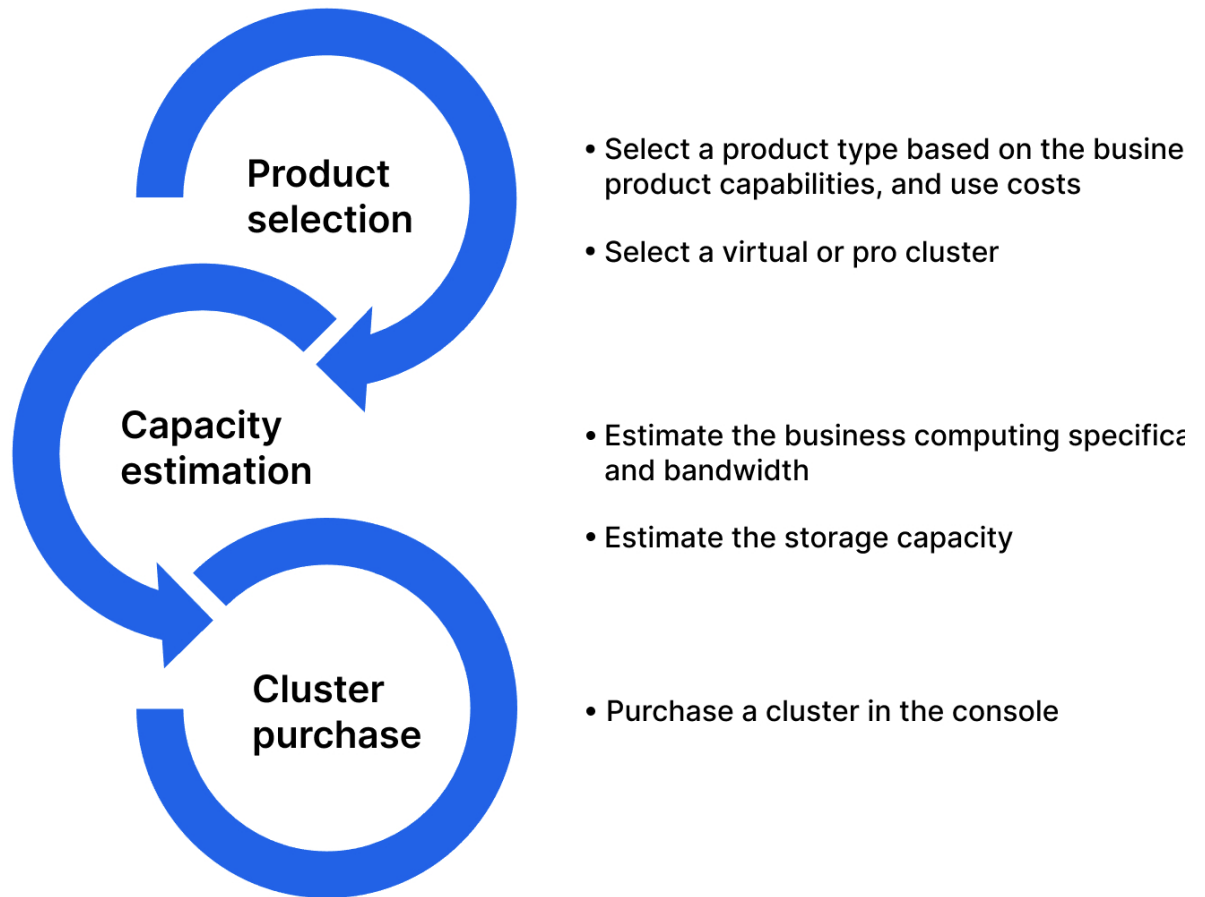
## Product Types

TDMQ for Apache Pulsar product portfolio is as shown below.



## Product Selection Process

We recommend you select a product type based on the following process.



## Product Selection Analysis

The characteristics of TDMQ for Apache Pulsar cluster types are as compared below:

Item	Virtual Cluster	Pro Cluster
Instance type	Physical resource sharing among logical tenants	Physical isolation
Customer group and scenario	Entry-level customers with a moderate business traffic, short-term testing, and great traffic fluctuations.	Top customers whose production environment has high requirements for service stability and resource isolation and generates high amounts of business traffic.
Billing mode	Pay-as-you-go - postpaid	Monthly subscription - prepaid
Billable item	API call, message storage, and partition topic resource usage.	Cluster specification, mainly including TPS and bandwidth.



Messaging TPS limits	5,000 TPS for production and consumption each per cluster per topic	On-demand purchase based on different computing and storage specifications (starting from 2,000 TPS)
SLA	Data reliability: Eight 9s Service availability: 99.95%	Data reliability: Ten 9s Service availability: 99.99%
Pulsar engine version	2.7.2	2.9.2
Upper limit for expansion	Elastic expansion and use within a certain range	Up to one million TPS with greater elasticity
High availability	Multi-AZ deployment in the same region is not supported.	Custom multi-AZ deployment in the same region is supported to enhance disaster recovery capabilities.
Event support and expert service	The standard ticket service of Tencent Cloud is provided.	Event support is provided for major events such as product upgrade, business launch, and promotion campaign to ensure smooth business operations.

## Capacity Estimation

After selecting the cluster type, you need to estimate the computing and storage specifications actually needed by your business.

**Computing specification:** In a TDMQ for Apache Pulsar pro cluster, the computing specification indicates the upper limits on messaging TPS and bandwidth of the instance, which you can select as needed.

**Storage specification:** You can calculate the required storage space based on the estimated message volume and size of your business.

Note that the TDMQ for Apache Pulsar pro cluster adopts the three-copy mode for message storage.

### Purchasing a cluster

Select a region and create/purchase a cluster [here](#).

# Use Limits

Last updated : 2024-08-07 14:29:27

This document lists different cluster forms of TDMQ Pulsar, and limitations on some metrics and performance. Please be careful not to exceed the corresponding limit values to avoid exceptions.

**Note:**

For the difference comparison between virtual clusters and dedicated clusters, see [product selection](#).

## Cluster

Limit	Pro Cluster	Virtual Cluster
Maximum number of clusters in a single region	Unlimited	5
Cluster-level TPS (receiving and sending)	Depends on cluster specification.	10,000
Cluster-level peak bandwidth (receiving and sending)	Depends on cluster specification.	180 MB/s
Cluster name length	Up to 128 characters	Up to 128 characters
Maximum storage capacity	20 TB	100 GB
Number of connections for a single broker	It is 100,000 by default, adjustable based on specification.	100000
Single IP limit for broker	It is 200 by default, adjustable based on specification.	200

## Namespace

Limit	Virtual Cluster	Pro Cluster
Maximum number of namespaces in a single cluster	100	Depends on cluster specifications

## Topic

Limit	Pro Cluster	Virtual Cluster
Maximum number of Topics in a single cluster	Depends on cluster specification.	1000

Maximum number of partitions per Topic	32	32
Topic Name Length	Up to 128 characters	Up to 128 characters
Max Production TPS Per Topic Partition	It is 5,000 by default, adjustable based on cluster specification.	5000
Production Bandwidth Limit Per Topic Partition	It is 50 Mbps by default, adjustable based on cluster specification.	40 Mbps
Max TPS Per Topic Partition	It is 10,000 by default, adjustable based on cluster specification.	5000
Max Consumption Bandwidth Per Topic Partition	It is 50 Mbps by default, adjustable based on cluster specification.	40 Mbps
Maximum Number of Producers Per Topic Partition	1000	1000
Single IP limit of a producer	It is 200 by default, adjustable based on cluster specification.	200
Maximum number of subscriptions per topic partition	1000	1000
Maximum number of consumers per topic partition	2000	2000

## Message

Limit	Pro Cluster	Virtual Cluster
Max Message Retention Period	15 days	15 days
Max Message Delay	10 days	10 days
Max Message Size	5MB	5MB
Reset consumption offset	15 days	15 days
Maximum number of unacknowledged messages received	It is 5,000 by default, adjustable based on cluster specification.	5,000 entries

---

Maximum number of messages for a single query	65,536 entries	65,536 entries
--	----------------	----------------

# Comparison with Apache Pulsar

Last updated : 2024-06-28 11:29:49

The performance comparison between TDMQ for Apache Pulsar and open-source Apache Pulsar is detailed below:

Item	TDMQ for Apache Pulsar	Apache Pulsar
Strengths	<p>It supports on-demand usage and the pay-as-you-go billing mode, eliminating your need to care about configuration.</p> <p>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.</p> <p>It has a high SLA, and its parameters are fine-tuned in a targeted manner.</p>	<p>It stays up to date with new open-source features.</p> <p>It supports flexible parameter configuration</p>
Shortcomings	<p>It cannot stay up to date with all open-source features in time.</p>	<p>It depends on a large number of components and therefore has a high Ops workload.</p> <p>It doesn't provide an SLA.</p> <p>It has limited security protection capabilities.</p> <p>It cannot be configured precisely, which causes resource waste.</p>
Cost	<p>It supports on-demand usage and the elastic pay-as-you-go billing mode.</p> <p>It requires no manual Ops, making the cost controllable.</p>	<p>It cannot use resources elastically, resulting in a low resource utilization.</p> <p>It requires manual Ops, incurring high cost.</p>
Scalability	<p>It is very flexible and easy to scale.</p> <p>You don't need to pay attention to the scaling process and can fully utilize the scale effect to sustain sudden high loads.</p>	<p>It can add broker nodes flexibly. However, it is complex to manually scale out a BookKeeper cluster, during which maloperations can easily occur and affect the data.</p>
Availability	<p>It is deployed across multiple AZs, with messages stored in three replicas in different AZs. Tencent Cloud guarantees availability of above 99.95% and supports cluster traffic throttling and optimization to prevent the cluster from being crashed by high traffic.</p>	<p>It requires deployment in different regions to guarantee the availability. You need to ensure the cluster availability in case of a high traffic load on your own.</p>

Security protection	It natively provides security protection capabilities by using public cloud security products.	It requires the installation of open-source plugins for security protection.
Monitoring and alarming	It natively provides monitoring and alarming capabilities by using related public cloud products.	It requires the installation of open-source plugins for security protection.

## Key Features

1. It supports the message retry and dead letter mechanisms.
2. Tagged messages can be filtered by tag.
3. With the `listenerName` identifier added to the client, multi-network access is supported.
4. The long restart duration and restart jitters of clusters on the server have been fixed, reducing the impact of restart on your business.

## Exclusive Features

### All messages and their traces can be queried throughout their lifecycle

TDMQ for Apache Pulsar supports querying messages and their traces throughout the production-storage-consumption lifecycle, which allows you to quickly determine the status of abnormal messages.

### The server can actively push messages again

You can configure whether to allow the server to actively push long unacknowledged messages again, which can prevent messages from getting lost due to acknowledgement failures. You will be notified of the acknowledgement failure of messages to avoid the message heap in backlogs.

### Throttling can be implemented for a single VM in the tenant dimension

TDMQ for Apache Pulsar supports the throttling of the production/consumption speed and traffic in the tenant dimension.

### Refined monitoring metrics are made available for the memory utilization of core resources as well as the internal data pulling speed and traffic

TDMQ for Apache Pulsar provides more refined metrics to monitor the memory utilization of core resources and provides related statistics. It also supports the monitoring of the speed and traffic of reading messages from BookKeeper.

### BookKeeper data compression is monitored in a visual manner

TDMQ for Apache Pulsar supports the visual display of the complete BookKeeper data compression information, including the ledger being compressed and the compression duration of each ledger.

**Throttling is configured for reads/writes during BookKeeper data compression and can be adjusted dynamically**

TDMQ for Apache Pulsar supports read throttling during data compression to avoid excess use of disk and bandwidth. The throttling configuration can be adjusted dynamically.

**BookKeeper client is optimized to accelerate disaster recovery in an AZ**

TDMQ for Apache Pulsar supports the quick removal of faulty BookKeeper nodes to improve the cluster's overall disaster recovery speed.

## Other Features

**Backend maintenance**

As open-source Pulsar doesn't incorporate some features from previous versions into new releases, TDMQ for Apache Pulsar performs regular backend maintenance by selecting desirable features from the open-source community for further development and bug fixes for enhancements.

**Event support and expert service**

TDMQ for Apache Pulsar ensures smooth business operations with event support for major events such as product upgrade, new releases, and promotion campaigns.