

# 消息队列 CMQ

产品简介

产品文档



腾讯云

---

**【版权声明】**

©2013-2019 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

**【商标声明】**

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

**【服务声明】**

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

## 文档目录

产品简介

产品概述

产品功能

产品优势

# 产品简介

## 产品概述

最近更新时间：2020-12-18 13:04:14

## CMQ 消息队列介绍

消息队列 CMQ (Cloud Message Queue, 以下简称 CMQ) 是基于腾讯自研消息引擎的分布式消息队列系统, CMQ 通过腾讯自研分布式 Raft 算法保证消息强一致, 消息同步3副本落盘保障消息高可靠, 提供消息队列、发布订阅、消息回溯、延时消息、顺序消息、消息轨迹等服务。具有高可靠、高可用、高性能、动态伸缩等优势。CMQ 的开发迭代历程超过7年, 为腾讯内部包括微信、webank、QQ 秀、手机 QQ 等大型业务提供异步的消息服务支持。

CMQ 已经正式商用, 目前在腾讯云全球多个地域提供了高可用消息云服务, 机房硬件设施按照腾讯自建 IDC 的高标准来实施。单个地域内采用多机房部署, 即使整个机房都不可用, 仍然可以为应用提供消息发布服务。同时在深圳金融专区、上海金融专区均有部署, 提供金融级数据高可靠消息队列服务。

CMQ 目前提供 HTTP(S)、TCP 协议的接入。提供 PHP、Java、Python 等丰富语言的 SDK 接入。

接入方式	HTTP(S) 接入	TCP 接入
面向场景	提供基于 HTTP(S) 协议的同步接入方式, 支持 Restful API 和多语言的 SDK 简单方便接入。	提供 TCP 同步/异步接入方式, 支持多语言 SDK, 提高生产端和消费端的效率, 提供更高性能的消息队列服务。
亮点能力	消息无限堆积、水平扩展金融级消息高可靠, 消息实时落盘。	TCP 异步非阻塞方式收发消息, 提升效率。消息无限堆积、水平扩展金融级消息高可靠, 消息实时落盘。

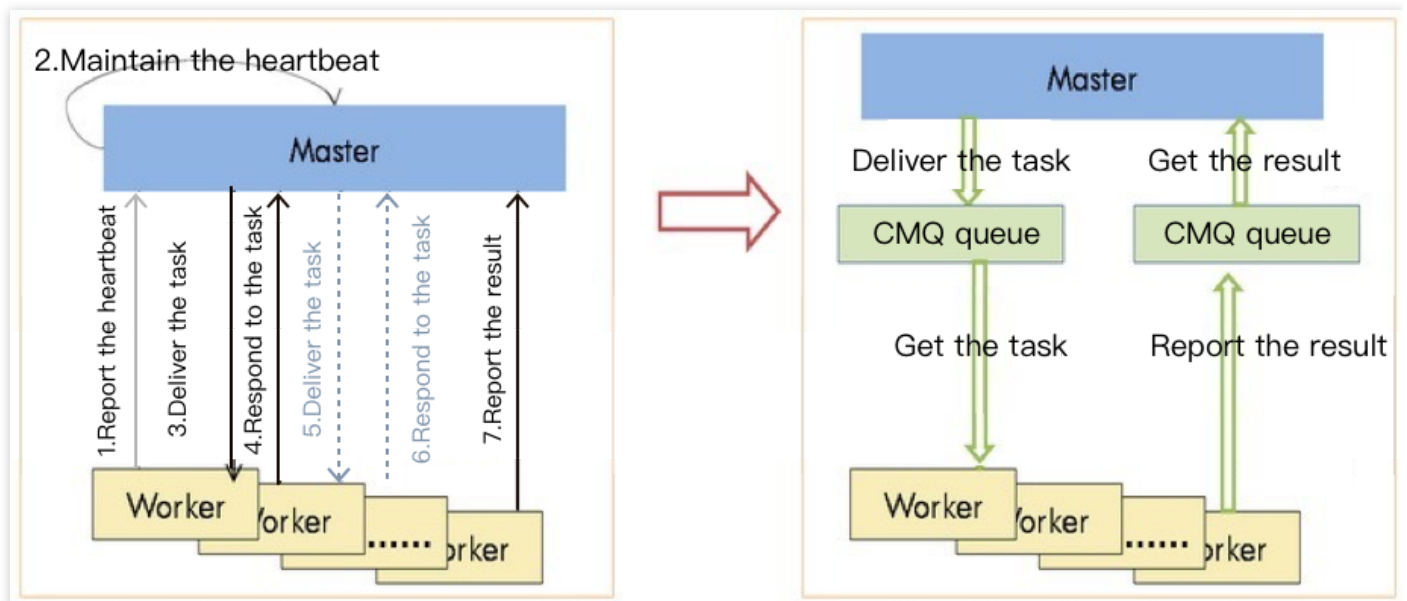
### ① 说明：

- CMQ TCP 接入方式正在内测中, 欢迎 [提交工单](#) 联系售后工程师, 将有专人为您开通内测服务。
- CMQ 支持私有化部署, 欢迎 [提交工单](#) 联系售后工程师, 将有专人为您开通内测服务。

## 应用场景概述

在需要进行异步通信的应用情景中推荐使用 CMQ。例如：

- 应用需要确保消息的可靠传递，即使发送消息时接收者由于断电、宕机或 CPU 负载过高等原因不可用，消息也可以在接收者可用时被送达。传统的消息队列把消息储存在内存中，故而不具备这一功能。腾讯云 CMQ 分布式消息队列中的消息会被持久化保存，直到接收者成功获取它。
- 需要在访问量与日俱增、囤积在队列中的消息数日益增长的情况下也能正常运转。传统的消息队列把消息存在本地内存中，单机的处理能力和内存容量都是有限的，不具备可扩展性。腾讯云 CMQ 的分布式架构保证了其扩容的简易性，更重要的是扩容对 CMQ 的使用者是完全透明的。
- 两个服务在网络不能互通或者应用的路由信息（例如 IP 和端口）不确定的情况下需要通信。例如，两个腾讯云上的服务在不知道对方地址的情况下需要进行通信，则可以通过约定队列名，一个向队列发送消息，一个从队列中收取消息而实现。
- 系统组件之间或者应用之间通信较多，需要组件或者应用自身维护彼此的网络连接，而且通信的内容不仅一种。这时，使用传统的架构会使得系统设计复杂。例如：当有一个中央处理服务需要向多个任务处理服务分派任务时（类似于 master-worker 模式），master 需要维护与所有 worker 的连接，并判断 worker 是否开始处理任务从而决策是否需要重新分派任务。同时，worker 的任务结果也需要汇报给 master。要在一个层面维护这样的系统会导致设计复杂，实现难度和维护成本大。如下图所示，使用腾讯云 CMQ 减轻两方之间的耦合性会使系统简洁高效许多。



- 系统组件之间或者应用之间耦合较紧，尤其对依赖的组件可控性较弱的情况下，希望降低耦合度。例如公司业务 CGI 收到用户提交的内容，将部分数据存储在自身的系统中，并将处理后的数据转发给其他业务应用（如数据分析系统、数据存储系统等）。传统的解决方案是服务间通过 socket 建立连接，此时如果接收方的 IP 或端口改变，或换了另一个接收方，则需要数据发送者进行修改。使用腾讯云 CMQ，发送者和接收者对彼此信息无感知，耦合度降低。

# 产品功能

最近更新时间：2020-08-04 14:30:19

## 异步通信协议

消息的发送者将消息发送到消息队列后可以立即返回，不用等待接收者的响应。消息会被保存在队列中，直到被接收者取出。消息的发送与处理是完全异步的。

## 提高可靠性

传统模式下消息可能因为长时间等待而导致请求失败。消息队列模式下，如果发送消息时接收者不可用，消息队列会保留消息直到成功传递它。

## 进程解耦

消息队列帮助减少两个进程间的耦合度。只要消息格式不变，即使接收者的接口、位置或者配置改变，也不会给发送者带来任何改变。并且，消息发送者无需知道消息接收者是谁，使得系统设计更清晰；相反的，进程间使用远程过程调用（RPC）或者 socket 连接，当一方接口、IP 或端口改变了，另一方则必须修改请求配置。

## 消息路由

发送者无需与接收者建立直接连接，双方通过消息队列保证消息能够从发送者路由到接收者，甚至对于本来网络不易互通的两个服务，也可以提供消息路由。

## 多终端

用户系统的多个部分可以同时发送或接收消息，腾讯云 CMQ 通过消息状态来进行消息可用性的控制。

## 多样性

每个队列均可独立配置，并非所有队列都要完全相同。在不同业务场景下的队列可以进行个性化的配置，例如一个队列中消息处理时间较长，可以针对队列属性进行优化。

## 云函数触发

CMQ 主题模型可将消息传递给云函数 SCF，并将消息内容和相关信息作为参数来调用该函数。

[云函数产品文档 >>](#)

# 产品优势

最近更新时间：2021-08-17 10:44:23

## 对比 RabbitMQ 的优势

- **CMQ-QPS 优势**：在保证高可靠前提下，同等物理设备，CMQ 的吞吐量优于 RabbitMQ 四倍以上。单集群 QPS 超过10万。
- **RabbitMQ 不支持消息回溯**：RabbitMQ 不支持消息回溯，CMQ 支持按照时间回溯消息。例如从一天之前的某时某分某秒开始重新消费消息。典型业务场景如 Consumer 做订单分析，但是由于程序逻辑或者依赖的系统发生故障等原因，导致今天消费的消息全部无效，需要重新从昨天零点开始消费，那么以时间为起点的消息重放功能对于业务非常有帮助。
- **一致性算法对比**：CMQ 和 RabbitMQ 都能够使用多台机器进行热备份，提高可用性。CMQ 基于 Raft 算法实现，简单易维护。RabbitMQ 使用自创的 GM 算法（Guaranteed Multicast），学习难度较高。
- **RabbitMQ 运维难度大**：RabbitMQ 的开发语言用的是 Erlang，较小众、学习成本高。

## 对比 RocketMQ 的优势

- **RocketMQ 在极端情况下，会丢失数据**：RocketMQ 允许未刷盘就向客户端返回确认，在机器异常宕机时，会丢失消息。
- **RocketMQ 需搭建多 Master、Slave 才能保证业务高可用**：RocketMQ 只有在 ISR 中有存活节点时，才能保证可用性和可靠性，ISR 中无存活节点时，可用性和可靠性无法保证，开销较大。

因此，相比传统开源 MQ 应用，腾讯云 CMQ 具有以下优势：

对比项	腾讯云消息队列	开源消息中间件软件
高性能	兼顾性能与可靠性，单 CMQ 实例 QPS 达到5000	数据可靠性与性能无法兼顾
高扩展性	<ul style="list-style-type: none"> <li>• 队列数量及队列存储容量可扩展性强</li> <li>• 底层系统根据业务规模，自动弹性伸缩，上层业务无感知</li> <li>• 高效支持亿级消息收发、推送、堆积，容量不设上限</li> <li>• 提供北京、上海、广州地域的多地域服务</li> </ul>	<ul style="list-style-type: none"> <li>• 队列数量和消息堆积数量有限</li> <li>• 每个 IDC 机房必须重新部署购买设备、部署，非常繁琐</li> </ul>

对比项	腾讯云消息队列	开源消息中间件软件
高可靠性	<ul style="list-style-type: none"> <li>• 基于腾讯自研 CRMQ (Cloud Reliable Message Queue) 分布式框架, 已在腾讯内部 QQ 微信红包、彩票等业务上得到广泛使用</li> <li>• 消息服务每条消息在返回给用户写成功之时就确保数据已被复制3份写到不同物理机上, 并且后台数据复制机制能够保证任何一台物理机故障时其上的数据能够快速的做迁移, 时刻保证用户数据3份 copy 可用, 可靠性达99.999999%</li> <li>• 引入改良后的 Raft 一致性算法, 保证数据强一致性</li> <li>• 业务可用性承诺: 99.95%</li> </ul>	<ul style="list-style-type: none"> <li>• 数据单机或简单主从结构, 存在数据单点问题, 一旦丢失不可回溯</li> <li>• 开源的 replica 算法, 在集群新增、删除服务器节点时, 会引发全局的数据重新均衡, 引起可用性急剧下降</li> <li>• 如 Kafka 使用异步刷盘方式, 异步 Replication, 无法保证数据强一致性</li> </ul>
业务安全	<ul style="list-style-type: none"> <li>• 多维度的安全防护和防 DDoS 攻击服务</li> <li>• 每个消息服务提供单独命名空间, 客户间数据严格隔离</li> <li>• 支持 HTTPS 访问</li> <li>• 支持跨地域的安全消息服务</li> </ul>	<ul style="list-style-type: none"> <li>• 安全防护功能有限</li> <li>• 考虑到公网的网络威胁, 经常无法提供跨地域、跨 IDC 的公网服务</li> </ul>