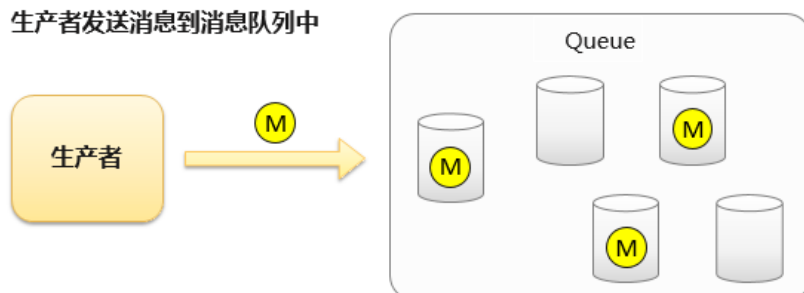


华为云 DMS 概述

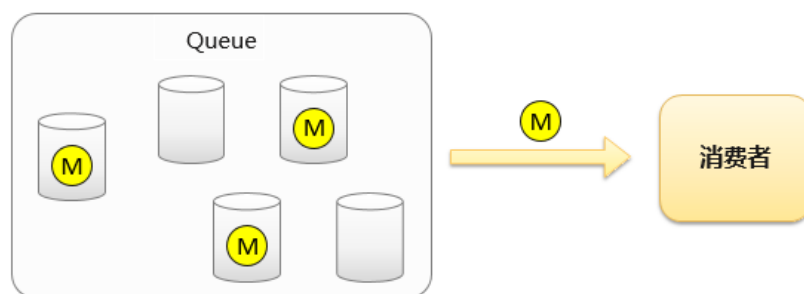
分布式消息服务（Distributed Message Service，简称 DMS）是一项基于高可用分布式集群技术的消息中间件服务，提供了可靠且可扩展的托管消息队列，用于收发消息和存储消息。

使用 DMS，您可以创建消息队列，将消息队列作为一个传输消息的中转站，存储应用程序不同组件间传递的消息，从而做到在应用程序的不同组件之间传输消息时，不要求各个组件同时处于可用状态。

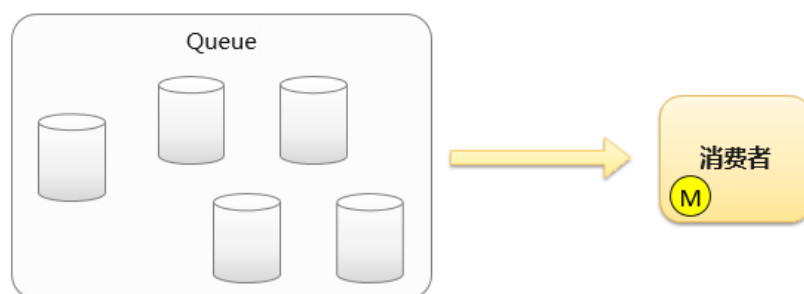
1 生产者发送消息到消息队列中



2 消费者从消息队列中消费消息



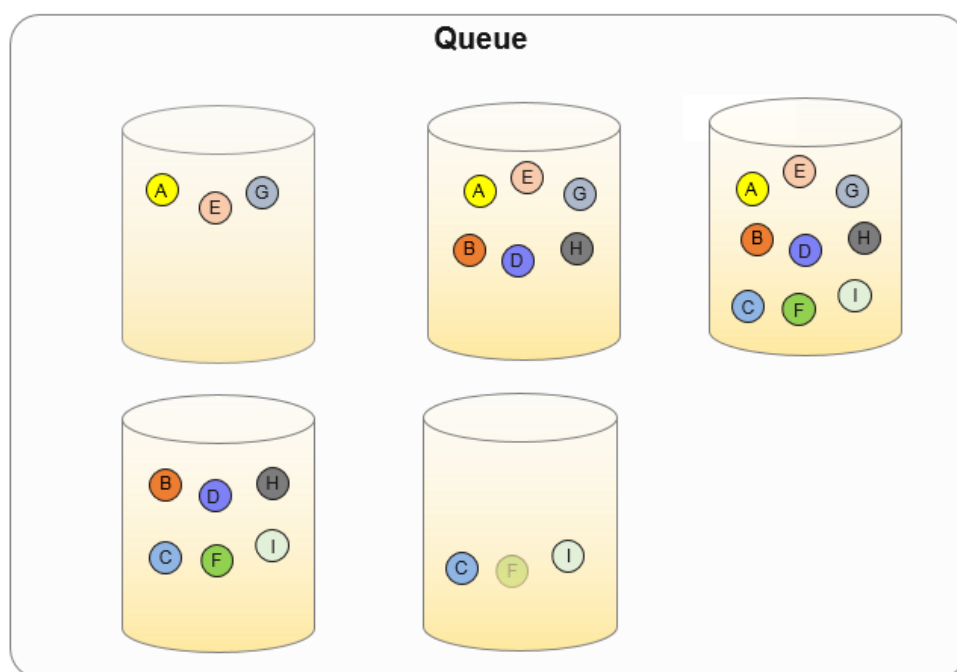
3 消费者确认消息已经消费



步骤说明：

1. 生产者将消息 M 发送到队列中，消息 M 在队列中冗余分布，存在多个副本。
2. 消费者从队列中消费消息，获取到消息 M。在消费者消费消息 M 期间，消息 M 仍然停留在队列中，但消息 M 从被消费开始的 30 秒内不能被该消费组再次消费，若在这 30 秒内没有被消费者确认消费完成，则 DMS 认为消息 M 未消费成功，将可以被继续消费。
3. 消费者确认消息 M 消费完成，消息 M 将不能被该消费者所在消费组消费。消息 M 仍然保持在队列中，并且可以被其它消费组消费，消息在队列中的保留时间为至少 72 小时（除非队列被删除），72 小时后将会被删除。

消息队列



消息队列具有如下属性：

1. 消息顺序

普通队列支持“分区有序”和“全局有序”两种模式，ActiveMQ 队列和 Kafka 队列均为分区有序。

分区有序的队列通过分布式处理，支持更高的并发，但由于队列的分布式特性，DMS 无

法保证能够以接收消息的精确顺序进行消费。如果用户要求保持顺序，建议在每条消息中放置排序信息，以便在收到消息时对消息重新排序。

全局有序的队列对消息消费遵循先入先出规则（FIFO），适用于对消费顺序要求较高的场景。

2. 至少一次传递

在极少数情况下，当用户接收或删除消息时，存储消息副本的服务器之一可能不可用。如果出现这种情况，则该不可用服务器上的消息副本将不会被删除，并且在接收消息时可能会再次获得该消息副本。

这被称为“至少一次传递”，因此，用户的应用程序应该设计为幂等的应用程序（即，如果应用程序多次处理同一条消息，与单次成功处理该条消息的效果等价）。

DMS 产品功能

分布式消息服务（Distributed Message Service）是完全托管的高性能消息队列服务，提供普通队列、有序队列、Kafka 队列、ActiveMQ 队列，兼容 HTTP、TCP、AMQP 协议，为分布式应用提供灵活可靠的异步通信机制。使用 DMS，您可以创建消息队列，将消息队列作为一个传输消息的中转站，存储应用程序不同组件间传递的消息，从而做到在应用程序的不同组件之间传输消息时，不要求各个组件同时处于可用状态。

1. 队列类型和接入协议

兼容原生 Kafka 队列，支持 ActiveMQ，业务 0 改动迁移。

- 多队列类型

支持普通队列、FIFO 保序队列、Kafka 队列、ActiveMQ 队列。

- 多协议接入

支持 HTTP Restful API、TCP SDK、KAFKA SDK 接入。

2. 队列能力

支持按需弹性扩容，队列共享和授权。

- 自动弹性扩展

队列按需自动扩展，按实际使用量计费。

- 死信队列

将其无法成功处理的消息存放于死信队列，便于统一分析和处理。

- 队列共享和授权

支持租户间或租户内针对不同用户的队列共享和授权。

3. 消息能力

一键在线扩容，不影响上层业务。

- 消息过滤

根据消息标签对可消费的消息进行过滤。

- 消息回溯

通过指定时间或位置，对已经消费过的消息进行重新消费。

- 定时/延时消息

在消息生产时间（当前时间）之后的指定时间点投递给消费组消费。

- 广播消息

在同一个消费组内对所有消费者投递相同消息

- 消息重投

消费者可以决定将暂不消费的消息重新投递到队列中，延后再消费。

4. 高可靠

数据可靠性 99.999999999%，服务可用性 99.95%。

- 数据可靠性

同步落盘与多副本冗余，数据可靠性高达 99.99999999%。

- 服务可用性

集群化部署与跨 AZ 部署，确保服务可用性高达 99.95%。

5. 高性能

亿级消息堆积，可弹性扩展队列数，支持千万级并发。

- 海量堆积

支持亿级消息堆积，在海量堆积下不影响队列性能。

- 高并发

单队列并发最高可超过 10 万 TPS（每秒处理的消息数），扩展队列数可获得更高并发。

- 低时延

消息投递时间可至毫秒级，保证消息及时性。

6. 安全防护

业务操作可回溯，消息存储加密等有效安全措施。

- 可追溯

提供云审计进行租户管理操作的记录和审计。

- 消息加密

数据加密存储，防止消息被窃取。

DMS 应用场景

分布式消息服务可应用在多个领域，包括异步通信解耦、企业解决方案、金融支付、电信、电子商务、快递物流、广告营销、社交、即时通信、手游、视频、物联网、车联网等。

分布式消息服务可以应用但不局限于以下业务场景：

1. 业务解耦

将业务中依赖其他系统同时属于非核心或不重要的部分使用消息通知即可，无需同步等待其他系统的处理结果。

如电商网站在促销期间抢购订单，抢到的商品订单信息放入消息队列，出库、发货等后续会从队列里读取任务信息然后执行。

2. 最终一致性

在交易或支付系统中，不同的子系统/模块的状态需要最终保持一致，或都成功或都失败。子系统/模块之间传递的数据不能丢失，需要有可靠消息传递，能保证业务的连续性。

DMS 可以用于子系统/模块间的高可靠数据传递，实现两者之间的事务最终一致，降低实现难度和成本。

例如用户使用储蓄余额购买理财产品。由于理财系统一般对理财申购交易采用日终统一处理，允许用户的储蓄余额与理财系统资金余额存在一定时间段的不一致，或者说中间状态，因此，可以对理财产品的购买支付流程使用 DMS 服务的消息处理机制，保证储蓄余额与理财余额的最终一致，同时避免系统间对账不平。

3. 错峰流控

在电子商务系统或大型网站中，上下游系统处理能力存在差异，处理能力高的上游系统的突发流量可能会对处理能力低的某些下游系统造成冲击，需要提高系统的可用性的同时降低系统实现的复杂性。电商大促销等流量洪流突然来袭时，可以通过队列服务堆积缓存订单

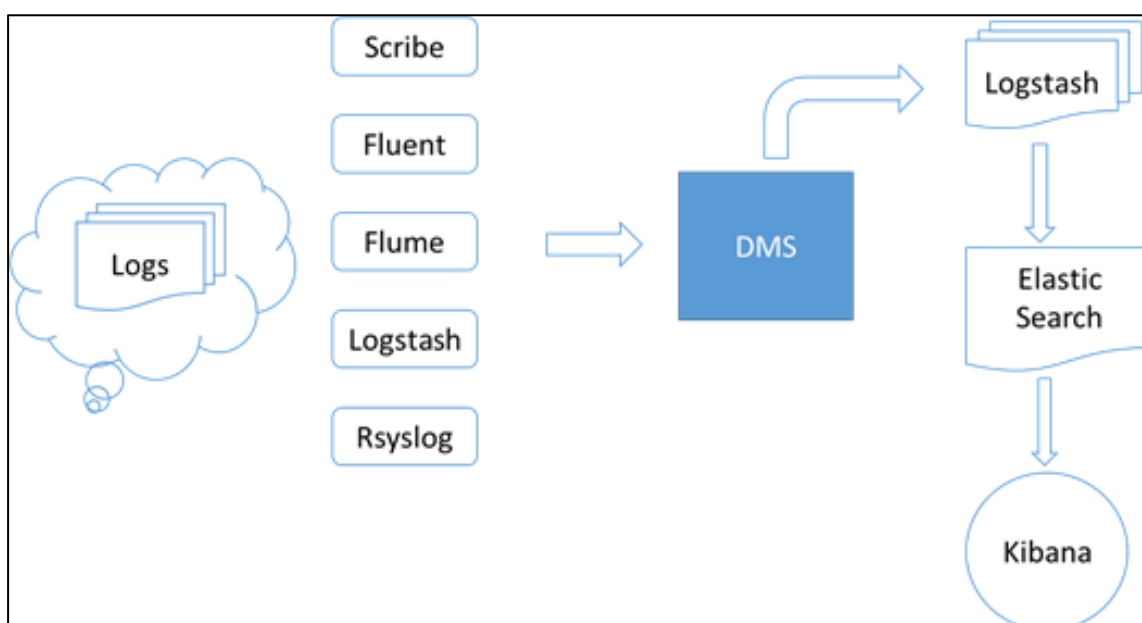
等信息，在下游系统有能力处理消息的时候再处理，避免下游订阅系统因突发流量崩溃。消息队列提供亿级消息堆积能力，3 天的保留时长，消息消费系统可以错峰进行消息处理。

4. 日志同步

应用通过可靠异步方式将日志消息同步到消息服务，再通过其他组件对日志做实时或离线分析，也可用于关键日志信息收集进行应用监控。

使用 DMS 实现日志同步一般按以下流程：

- 1) 日志采集客户端，负责用户应用服务的日志数据采集，以发送消息方式写入 DMS 消息队列。
- 2) DMS 消息队列，负责日志数据的接收、存储和转发管理。
- 3) 日志处理应用，订阅并消费 DMS 消息队列中的日志数据。



DMS 基本概念

1. 消息

消息队列中信息传递的载体，以 JSON 格式存在，可以发送或消费一条消息，也可以批量发送或消费，其中批量发送只能通过 API 接口调用完成。

目前消息负载最大限制为 512KB。

2. 消息属性

每条消息均可以定义消息属性，由属性名及属性值组成，并随消息一起被发送和消费，属于消息的一部分。

消息属性可以帮助消息的消费者决定如何处理消息，而不必先处理消息正文。

3. 消息队列

消息队列是在消息的传输过程中保存消息的容器，用于接收消息并以文件的方式存储，一个队列的消息可以同时被多个消息消费者消费，每个用户默认最多可以创建 30 个队列。

- 普通队列：包括分区有序和全局有序，分区有序和普通队列具有更高的并发性能，不保证先入先出（FIFO）的严格顺序，全局有序和普通队列保证消息先入先出（FIFO）的严格顺序。

- ActiveMQ 队列：使用 AMQP 协议的队列。Advanced Message Queuing Protocol，是统一消息服务的应用层标准的高级消息队列协议，是应用层协议的一个开放标准，为面向消息的中间件设计。

- Kafka 队列：支持使用开源 kafka 客户端进行生产消息和消费消息。支持高吞吐和高可靠两种模式，高吞吐模式的消息副本异步落盘，具有较高的性能，高可靠模式的消息多副本同步落盘，保证消息的可靠性。

- RabbitMQ 队列：采用物理隔离方式部署，支持用户自定义内存规格和自定义特性，您可以根据业务需要定制相应计算能力和存储空间的 RabbitMQ 实例。

4. 死信消息

死信消息是指无法被正常消费的消息。

DMS 支持对消息进行异常处理。当消息进行多次重复消费仍然失败后，DMS 会将该条消

息转存到死信队列中，有效期为 72 小时，用户可以根据需要对死信消息进行重新消费。

消费死信消息时，只能消费该消费组产生的死信消息。

全局有序的普通队列的死信消息依然按照先入先出（FIFO）的顺序存储在死信队列中。

ActiveMQ 队列和 Kafka 队列不支持死信功能。

5. 消息生产者

发送消息的一方，也叫消息发送者，发送消息到指定的消息队列中。

6. 消息消费者

消费消息的一方，也叫消息接收者，通过调用消息服务的消费接口从队列中读取消息，消费确认后消费者将无法再次消费该消息。

7. 消息消费组

消息消费组用于区分不同类型的消息消费者，一个消息队列最多可以创建三个消费组。

消息队列中的所有消息可以被不同消费组分别消费一次，一个消费组消费掉的消息，其它消费组仍然可以消费。

同一个消费组下可以有多个消息消费者并行消费，消费同一个队列里的不同消息。

8. 消息重置

以时间轴为坐标，在消息持久化存储的时间范围内（普通队列和 ActiveMQ 队列默认 72 小时），重新设置消费组对指定队列的消费进度，设置完成后消费者将接收设定时间点之后由消息生产者发送到 DMS 服务端的消息。

9. 消息过滤

是通过对消息增加标签来区分队列中的消息分类，DMS 允许消费者按照标签对消息进行过滤，确保消费者最终只消费到他关心的消息类型。

发送消息时，无论是否添加消息标签，都可以发送到指定队列中。

消费消息时，指定过滤标签只能消费到匹配上标签的消息，不指定过滤标签，可以消费队列中所有类型的消息。

10. 延时消息

延时消息是指消息发送到 DMS 服务后，并不期望这条消息立即被消费，而是延迟一段时间后才能被消费。

仅普通队列（包括分区有序和全局有序）可以开启延时消息，ActiveMQ 队列和 Kafka 队列不支持延时消息的功能。

11. 消息重投

普通队列在使用 SDK 管理消息时支持消息重投，当消费消息异常时，确认消费设置为“重投”，服务端会将消息标记为已消费，并且转存到重投 Topic 中，用户可以继续往后消费。重投消息延迟至少 30 秒后才能再次被消费出来，再次消费确认时，依然可以设置为“成功”、“失败”或“重投”。

12. 广播消息

普通队列在使用 SDK 管理消息时支持广播消息，同一个消费组的所有消费者都可以消费相同的消息。可以设置是否自动确认，设置为 true 时，不需要手动提交确认消费。

在消费消息时，可以设置如下消费模式：

- 组播：消费成功后，该消费组的其他消费者无法再消费。
- 广播，从最早一条开始：从队列中第一条消息开始，该消费组的每个消费者都可以消费队列中的所有消息。
- 广播，从已消费的位置开始：当设置固定 Consumer id 时，可以从当前的消费位置开始，该消费组的每个消费者都可以消费队列中所有未消费的消息。
- 广播，从最新一条开始：从队列中最新的一条消息开始，该消费组的每个消费者都可

以消费之后生产的消息。

了解更多信息，请访问 [分布式消息服务 DMS 主页](#)