消息队列 CKafka 版

产品简介

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【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。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。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默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

产品简介
- 产品概述
- 产品优势
- 应用场景
- 技术原理
  - 系统架构
  - Consumer Group 状态说明
  - 限流机制说明
- 开源对比
- 使用限制
- 地域和可用区

产品简介
产品概述

最近更新时间：2025-03-17 14:34:22

观看视频

什么是消息队列 CKafka 版

消息队列 CKafka 版（TDMQ for CKafka）是一个分布式、高吞吐量、高可扩展性的消息系统，100%兼容开源 Kafka API 2.4、2.8、3.2 版本。CKafka 基于发布/订阅模式，通过消息解耦，使生产者和消费者异步交互，无需彼此等待。CKafka 具有高可用、数据压缩、同时支持离线和实时数据处理等优点，适用于日志压缩收集、监控数据聚合、流式数据集成等场景。

产品功能

- **收发解耦**
  有效解耦生产者、消费者之间的关系。在确保同样的接口约束的前提下，允许独立扩展或修改生产者/消费者间的处理过程。
- **削峰填谷**
  消息队列 CKafka 版能够抵挡突增的访问压力，不会因为突发的超负荷的请求而完全崩溃，有效提升系统健壮性。
- **顺序读写**
  消息队列 CKafka 版能够保证一个 Partition 内消息的有序性。和大部分的消息队列一致，消息队列 CKafka 版可以保证数据按照顺序进行处理，极大提升磁盘效率。
- **异步通信**
  在业务无需立即处理消息的场景下，消息队列 CKafka 版提供了消息的异步处理机制，访问量高时仅将消息放入队列中，在访问量降低后再对消息进行处理，缓解系统压力。

> **说明：**
> 消息队列 CKafka 版支持私有化部署，您可以 联系我们 咨询。

产品优势

最近更新时间：2025-03-17 14:34:22

本文主要介绍消息队列 CKafka 版相比于自建开源 Apache Kafka 所具备的优势。

与周边服务打通的优势

连通云上服务

与对象存储、弹性 MapReduce等云上服务一键打通。

Kafka Connector

基于开源Kafka Connector 的数据传输服务。

自身优势特性

100%兼容开源，轻松迁移

- 消息队列 CKafka 版兼容开源 Kafka 2.4、2.8、3.2 版本。
- 消息队列 CKafka 版业务系统基于现有的开源 Apache Kafka 生态的代码，无需任何改造，即可迁移上云，享受到腾讯云提供的高性能消息队列 CKafka 版服务。

高性能

- 腾讯云消息队列专业团队对服务性能进一步调优，免除复杂的参数配置，提供更高性能。
- 界面化升配能力，高性能 IaaS 层支撑，生产性能超过开源 Kafka 10% - 20%。

高可用性

- 依托腾讯技术工程多年监控平台的技术积累，对集群全方位多角度监控，更有专业运维团队7 × 24小时处理告警保障消息队列 CKafka 版服务的高可用性。
- 专业版支持同地域自定义多可用区部署，提升容灾能力。

高可靠性

- 磁盘高可靠，即使服务器坏盘50%也不影响业务。
- 默认2副本，支持3副本，副本越多可靠性越高。

平行扩展

- 当集群的流量和磁盘容量超过告警阈值，后端会及时扩容设备，对客户端无感知。
- 解决开源 Kafka 长期以来迁移数据的痛点，配置升级无感知。

数据安全

- 消息队列 CKafka 提供鉴权与授权机制、主子账号等功能，提供企业级的安全防护。
- 腾讯云私有网络（VPC）：支持腾讯云 VPC 访问，网络环境安全。
- 支持 SASL 鉴权方式，公网访问更安全。
- 主子账号：全面支持腾讯云 CAM 主子账号、协作者等功能，实现主子账号之间以及企业间跨账号的授权服务。

与周边服务打通的优势

连通云上服务

消息队列 CKafka 支持与对象存储（COS）、弹性 MapReduce（EMR）等云上服务一键打通。

Kafka Connector

支持基于开源 Kafka Connector 的数据传递服务，两个 Kafka 集群间可互相传递数据。

版权所有：腾讯云计算（北京）有限责任公司

应用场景

最近更新时间：2025-03-17 14:34:22

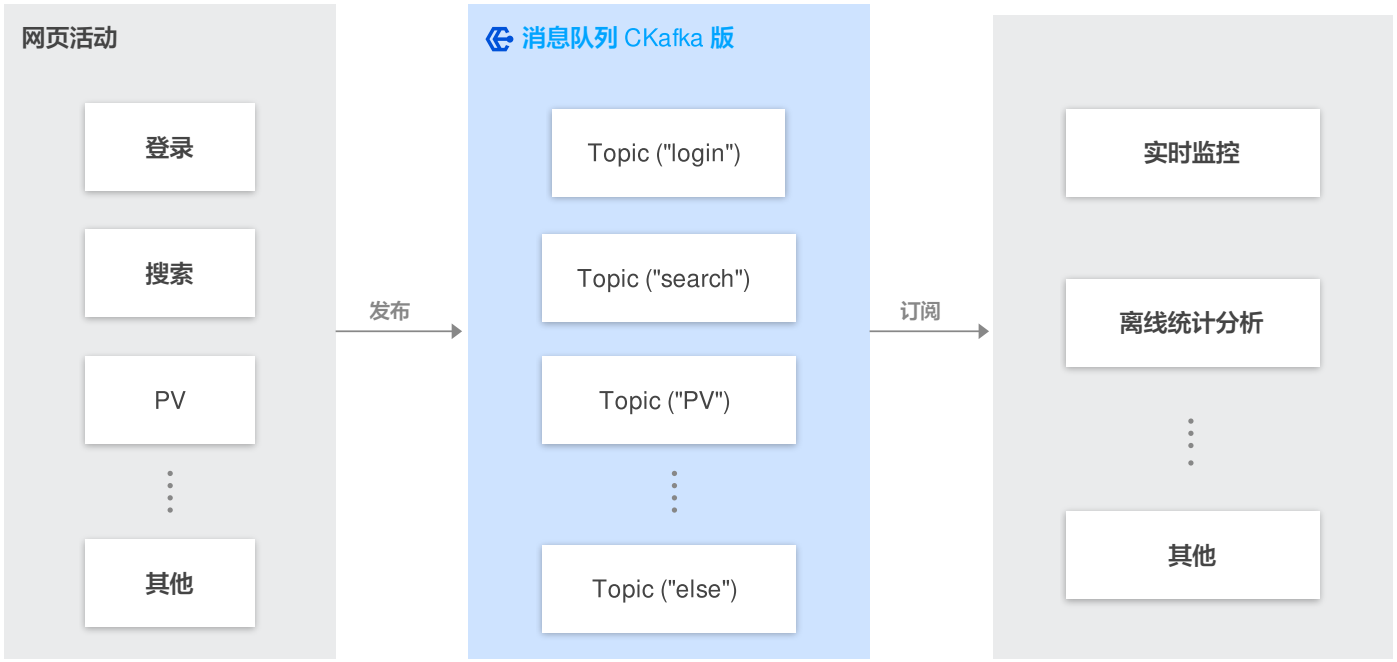
概述

消息队列 CKafka 版广泛应用于大数据领域，例如网页追踪行为分析、日志聚合、监控、流式数据处理、在线和离线分析等。
您可以通过以下方式让数据集成变得简单：

- 将消息队列 CKafka 版中的消息导入到腾讯云平台的 COS、流计算等数据仓库。
- 通过 SCF 触发器的方式连接云上其他产品。

网页追踪

消息队列 CKafka 版通过实时处理网站活动（PV、搜索、用户其他活动等），并根据类型发布到 Topic 中，这些信息流可以被用于实时监控或离线统计分析等。
由于每个用户的 page view 中会生成许多活动信息，因此网站活动跟踪需要很高的吞吐量，消息队列 CKafka 版可以满足高吞吐、离线处理等要求。



日志聚合

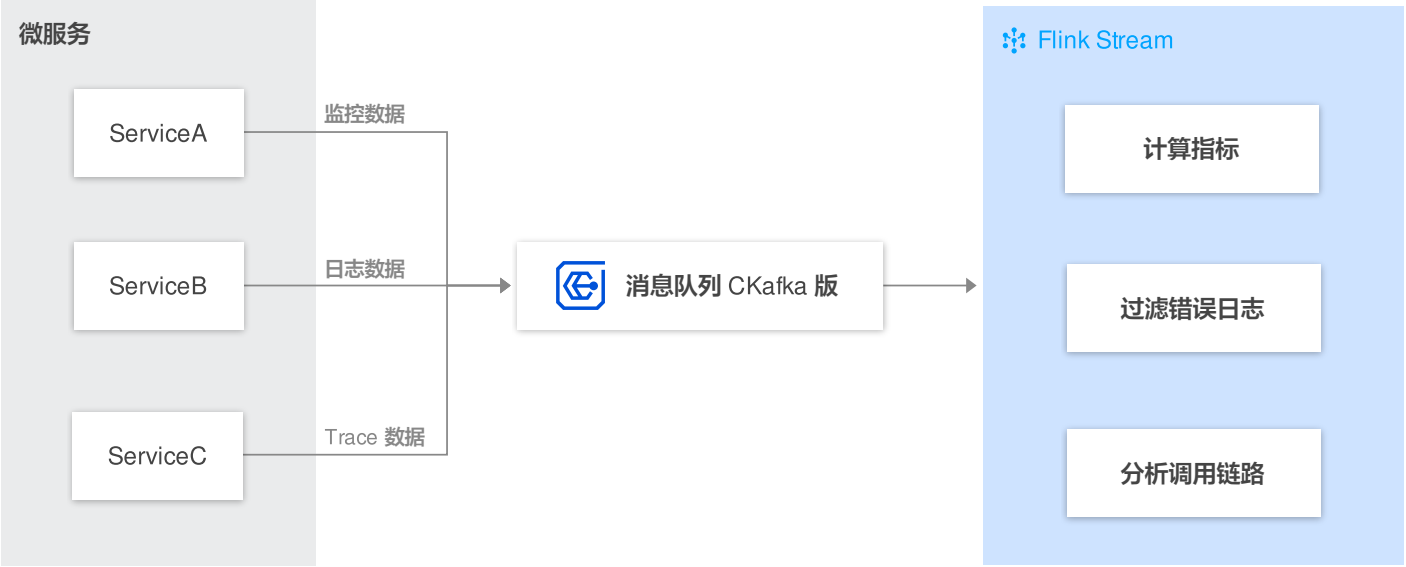
消息队列 CKafka 版的低延迟处理特性，易于支持多个数据源和分布式的数据处理（消费）。相比于中心化的日志聚合系统，消息队列 CKafka 版可以在提供同样性能的条件下，实现更强的持久化保证以及更低的端到端延迟。消息队列 CKafka 版的特性决定它非常适合作为“日志收集中心”。多台主机/应用可以将操作日志批量、异步地发送到消息队列 CKafka 版集群，而无需保存在本地或者 DB 中；消息队列 CKafka 版可以批量提交消息/压缩消息，对于生产者而言，几乎感觉不到性能的开支。此时消费者可以使用 Hadoop 等其他系统化的存储和分析系统对拉取日志进行统计分析。

大数据场景

在一些大数据相关的业务场景中，需要对大量并发数据进行处理和汇总，此时对集群的处理性能和扩展性都有很高的要求。消息队列 CKafka 版在数据分发机制上的实现，磁盘存储空间的分配、消息格式的处理、服务器的选择以及数据压缩等方面，也决定其适合处理海量的实时消息，并能汇总分布式应用的数据，方便系统运维。
在具体的大数据场景中，消息队列 CKafka 版能够很好地支持离线数据、流式数据的处理，并能够方便地进行数据聚合、分析等操作。

用户链路观测

在一个典型的微服务架构中，系统由多个独立服务组成，每个服务都会产生大量的监控数据（例如 CPU、内存使用率）、日志数据（例如请求日志、错误日志）和 Trace 数据（例如服务调用链路）。为了实现对系统的全面可观测性，可以将这些数据统一采集并发送到 Kafka 中。下游通过 Flink Stream 实时消费这些数据，进行聚合、分析和异常检测，帮助运维团队快速发现和解决问题，提升系统的稳定性和可维护性。

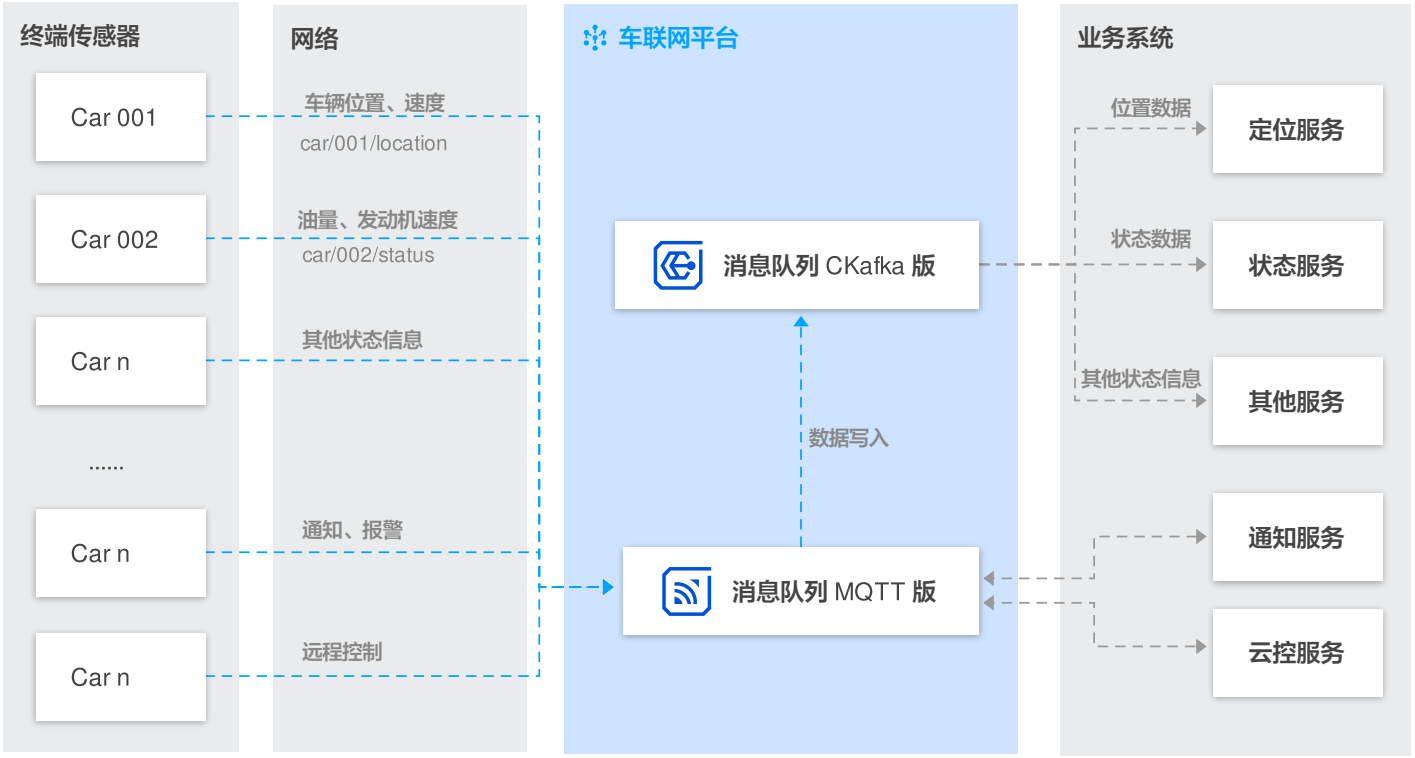


物联网数据采集分发

在物联网场景中，设备通过 MQTT 协议，将设备数据投递到 CKafka，经过规则引分发到不同的系统做进一步的处理。例如车辆通过搭载的传感器和控制器收集各种信息，例如车辆位置、速度、油量、发动机状态等，这些信息需要实时或定期传输到车厂的服务器，以便进行数据分析、故障预警、远程控制等操作。

- 终端设备通过 MQTT 协议接入消息队列 MQTT 版，通过规则引擎对接 CKafka 集群实现数据的转发。
- 车联网服务平台、高精地图服务、定位服务和其他车辆网相关应用可以直接通过订阅 CKafka 数据进行消费，同时通过消息队列 MQTT 版实现对车控（远程控制）消息的双向通信。

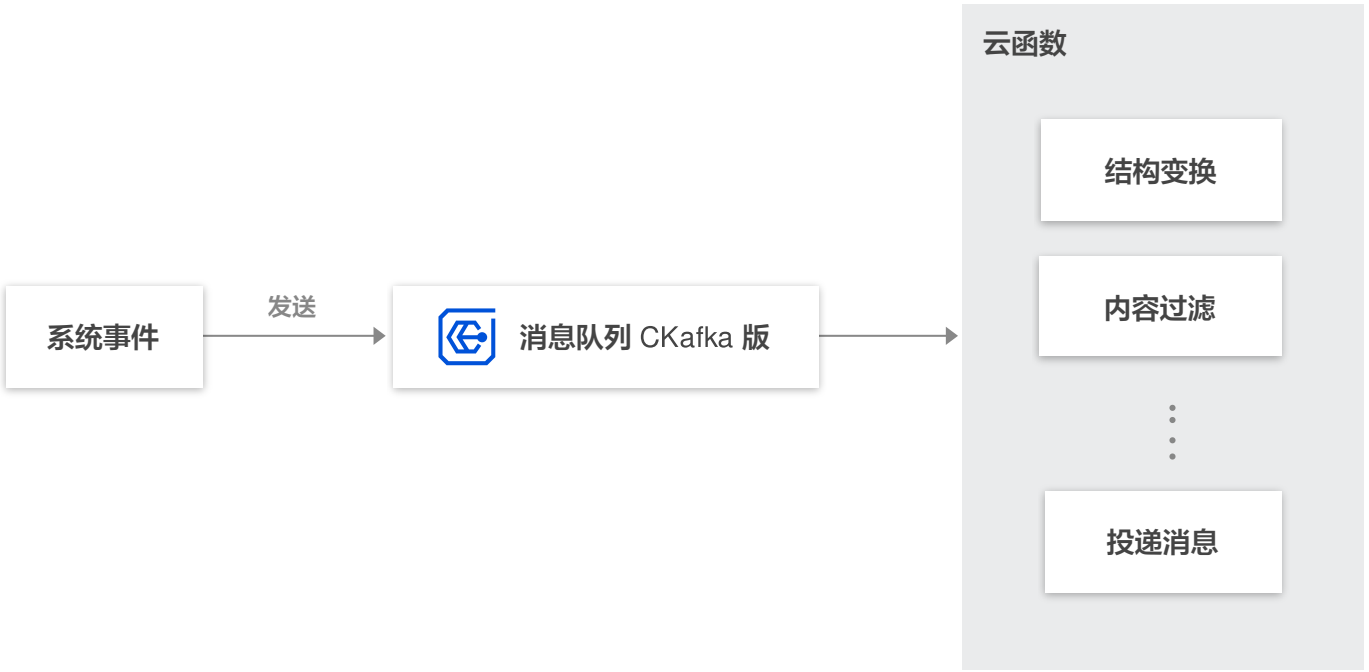
版权所有：腾讯云计算（北京）有限责任公司



云函数触发器

消息队列 CKafka 版可以作为云函数触发器，在消息队列中接收到消息时将触发云函数的运行，并会将消息作为事件内容传递给云函数。例如，CKafka 触发云函数时，云函数可以对消息进行结构变换、内容过滤等处理或者将消息投递到 Elasticsearch Service（ES）中。

> **说明：**
> 云函数的可用性详细参见《云函数服务等级协议》。

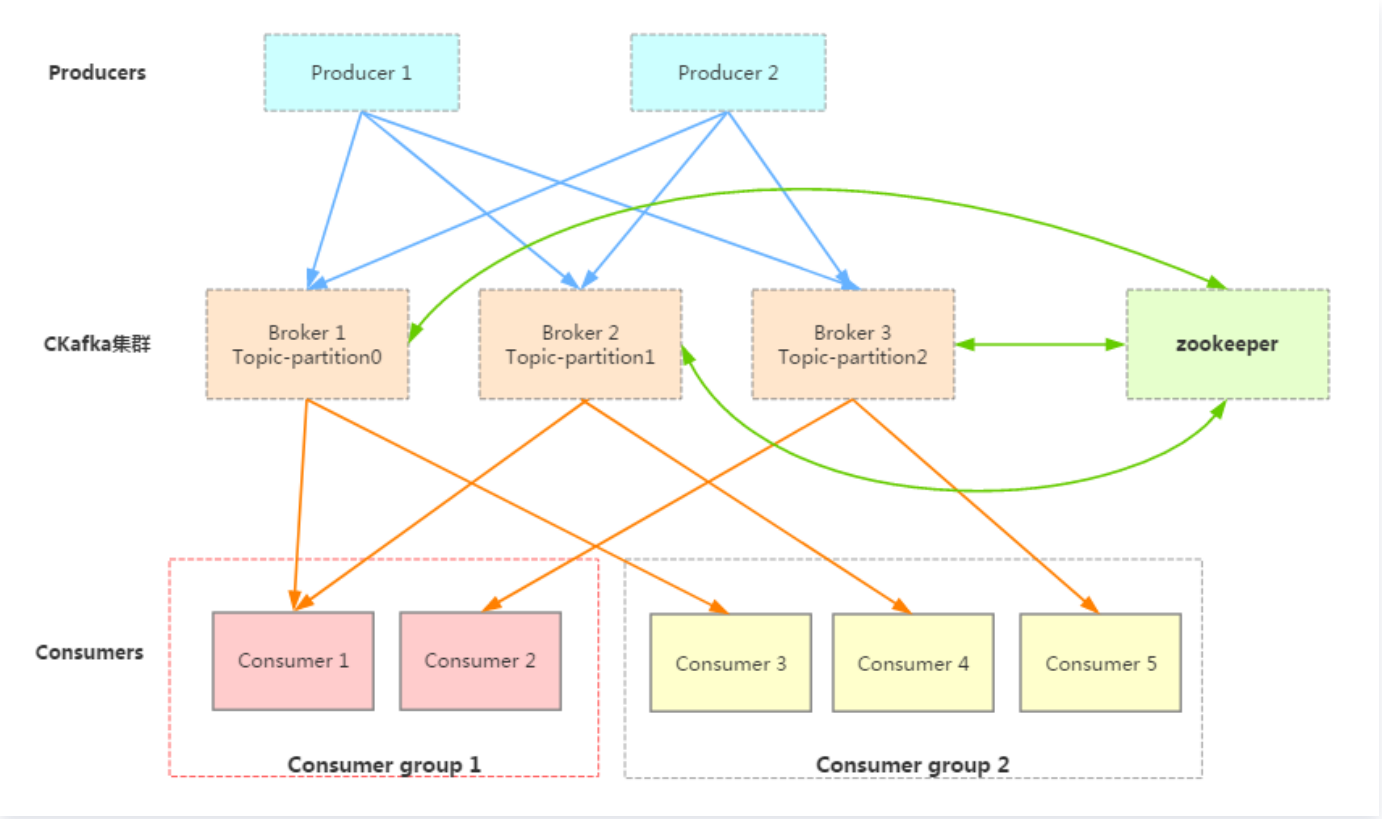

云函数
结构变换
内容过滤
投递消息
系统事件
发送
消息队列 CKafka 版

技术原理
系统架构

最近更新时间：2025-05-14 11:21:01

架构说明

消息队列 CKafka 版的架构图如下所示：



Producer

消息的生产者，生产的消息可能是网页活动产生的消息、服务日志等信息。生产者通过 push 模式将消息发布到 CKafka 的 Broker 集群。

CKafka 集群

- **Broker：**存储消息的服务器，Broker支持水平扩展，节点数量越多，集群吞吐率越高。
- **Partition：**分区，一个 Topic 可以有多个 Partition，每个 Partition 在物理上对应一个文件夹，用于存储该 Partition 中的消息和索引文件。Partition 可以有多个副本，提升可用性，但也会增加存储和网络开销。
- **Zookeeper：**负责集群的配置管理、leader选举、故障容错等。

Consumer

消息消费者，Consumer 被划分为若干个 Consumer Group，消费者通过 pull 模式从 Broker 中消费消息。

架构原理说明

高吞吐

消息队列 CKafka 版中存在大量的网络数据持久化到磁盘和磁盘文件通过网络发送的过程。这一过程的性能直接影响 Kafka 的整体吞吐量，主要通过以下几点实现：

- **高效使用磁盘**：磁盘中顺序读写数据，提高磁盘利用率。
- 写 message：消息写到 page cache，由异步线程刷盘。
- 读 message：消息直接从 page cache 转入 socket 发送出去。
- 当从 page cache 没有找到相应数据时，此时会产生磁盘 IO，从磁盘加载消息到 page cache，然后直接从 socket 发出去。
- **Broker 的零拷贝（Zero Copy）机制**：使用 sendfile 系统调用，将数据直接从页缓存发送到网络上。
- **减少网络开销**
- 数据压缩降低网络负载。
- 批处理机制：Producer 批量向 Broker 写数据、Consumer 批量从 Broker 拉数据。

数据持久化

消息队列 CKafka 版的数据持久化主要通过如下原理实现：

- **Topic 中 Partition 存储分布**
- 在消息队列 CKafka 版文件存储中，同一 Topic 有多个不同 Partition，每个 Partition 在物理上对应一个文件夹，用于存储该 Partition 中的消息和索引文件。例如，创建两个 Topic，Topic1 中存在5个 Partition，Topic2 中存在10个 Partition，则整个集群上会相应生成5 + 10 = 15个文件夹。
- **Partition 中文件存储方式**
  Partition 物理上由多个 segment 组成，每个 segment 大小相等，顺序读写，快速删除过期 segment, 提高磁盘利用率。

水平扩展（Scale Out）

- 一个 Topic 可包含多个 Partition，分布在一个或多个 Broker 上。
- 一个消费者可订阅其中一个或者多个 Partition。
- Producer 负责将消息均衡分配到对应的 Partition。
- Partition 内消息是有序的。

消费者分组（Consumer Group）

- 消息队列 CKafka 版不删除已消费的消息。
- 任何 Consumer 必须属于一个 Group。
- 同一 Consumer Group 中的多个 Consumer 不同时消费同一个 Partition。

版权所有：腾讯云计算（北京）有限责任公司

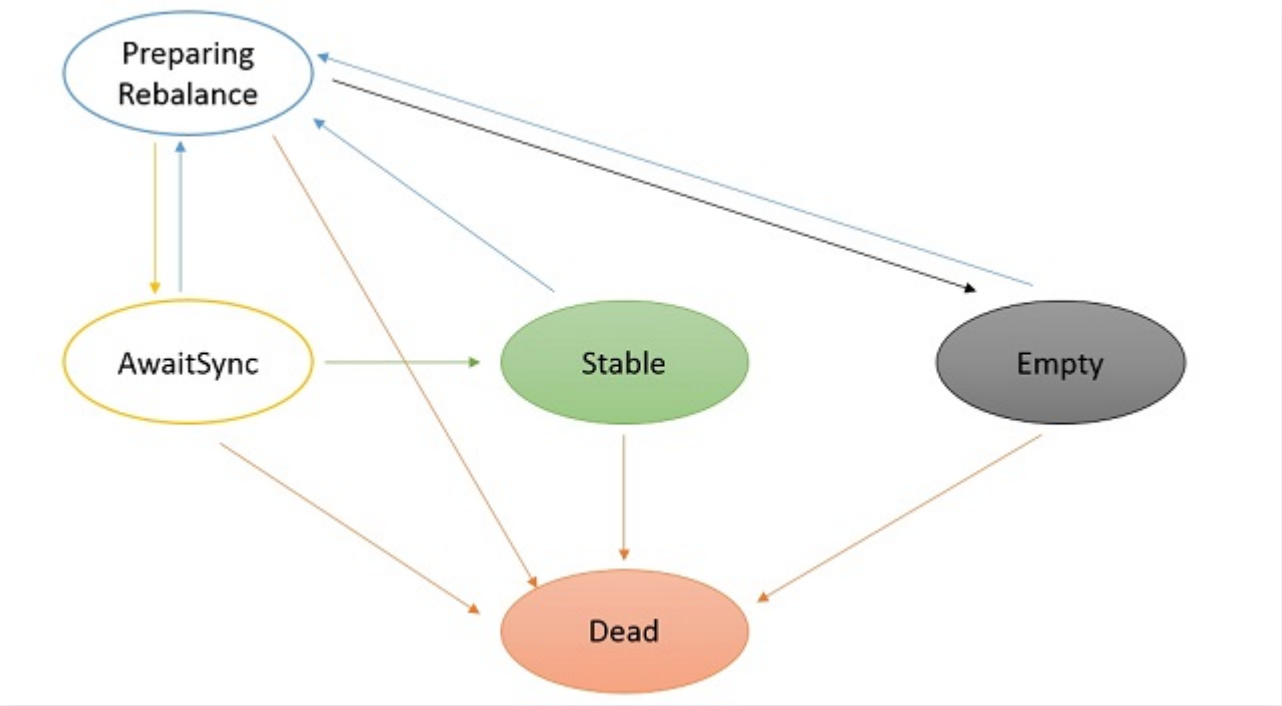
- 不同 Group 同时消费同一条消息，多元化（队列模式、发布订阅模式）。

Consumer Group 状态说明

最近更新时间：2025-03-17 14:34:24

Consumer Group 状态说明

消费者组列表页中 Consumer Group 的状态主要有 Dead、Empty、PreparingRebalance、AwaitingSync、Stable 几种，其中最常见的是 Empty、Stable 和 Dead 三种状态。Consumer Group 中的状态机转换如下图所示：



- Dead：消费者组内无成员并且 Metadata 已经被移除。
- Empty：消费分组内当前没有任何成员。如果组内所有 offset 都已过期，则会变为 Dead 状态。一般新创建的 Group 默认为 Empty 状态。
  开源 Kafka 0.10.x 版本规定，**当消费分组内没有任何成员且状态持续超过7天，此消费分组将会被自动删除。**
- Stable：消费分组中各个消费者已经加入，处于稳定状态。

Rebalance 状态详解

Rebalance 发生原因

根据 Consumer Group 的状态机可知，当 Consumer Group 为 Empty、AwaitSync 或 Stable 状态时，Group 可能会进行 Rebalance。以下情况可能会发生 Rebalance：

- 一个消费者订阅了 Topic。
- 消费者被关闭。
- 某个 Consumer 被 Group Coordinator（协调器）认为是 Dead 状态时。
  如果某个Consumer 在 `session.timeout.ms` 时间内没有给 Group Coordinator 发心跳，则该

Consumer 将被认为是 Dead 状态，并且发起 Rebalance。详请参见 CKafka 常用参数配置指南 。

- 分区数增加。
- 订阅了不存在的 Topic。
  如果您订阅了一个还未创建的 Topic，那么当这个 Topic 创建后会发生 Rebalance；同理，如果一个已经被订阅的 Topic 被删除，也会发生 Rebalance。
- 应用崩溃。

Rebalance 过程分析

以0.10版本Kafka 的机制为例，Rebalance 过程分析如下：

1. 任何一个 Consumer 想要加入到一个 Consumer Group 中时，会发送一个 JoinGroup 的请求给 Group Coordinator。第一个加入 Group 的 Consumer 会变成 Group Leader。
2. Leader 会从 Group Coordinator 处收到这个 Group 中所有 Consumer 列表，并且负责给 Group 中的 Consumer 分配 partition。分区的分配可以通过 PartitionAssignor 接口来实现。
3. 分配完成后，Leader 会把分配结果发给 Group Coordinator，Coordinator 会把结果发送给所有的 Consumer。
   因此，每个 Consumer 只能查看到自己被分配的 partition，Leader 是唯一能够拿到Consumer Group 中的 Consumer 以及其分区情况的节点的 Consumer。

上述过程会在每次 Rebalance 发生时执行一次。

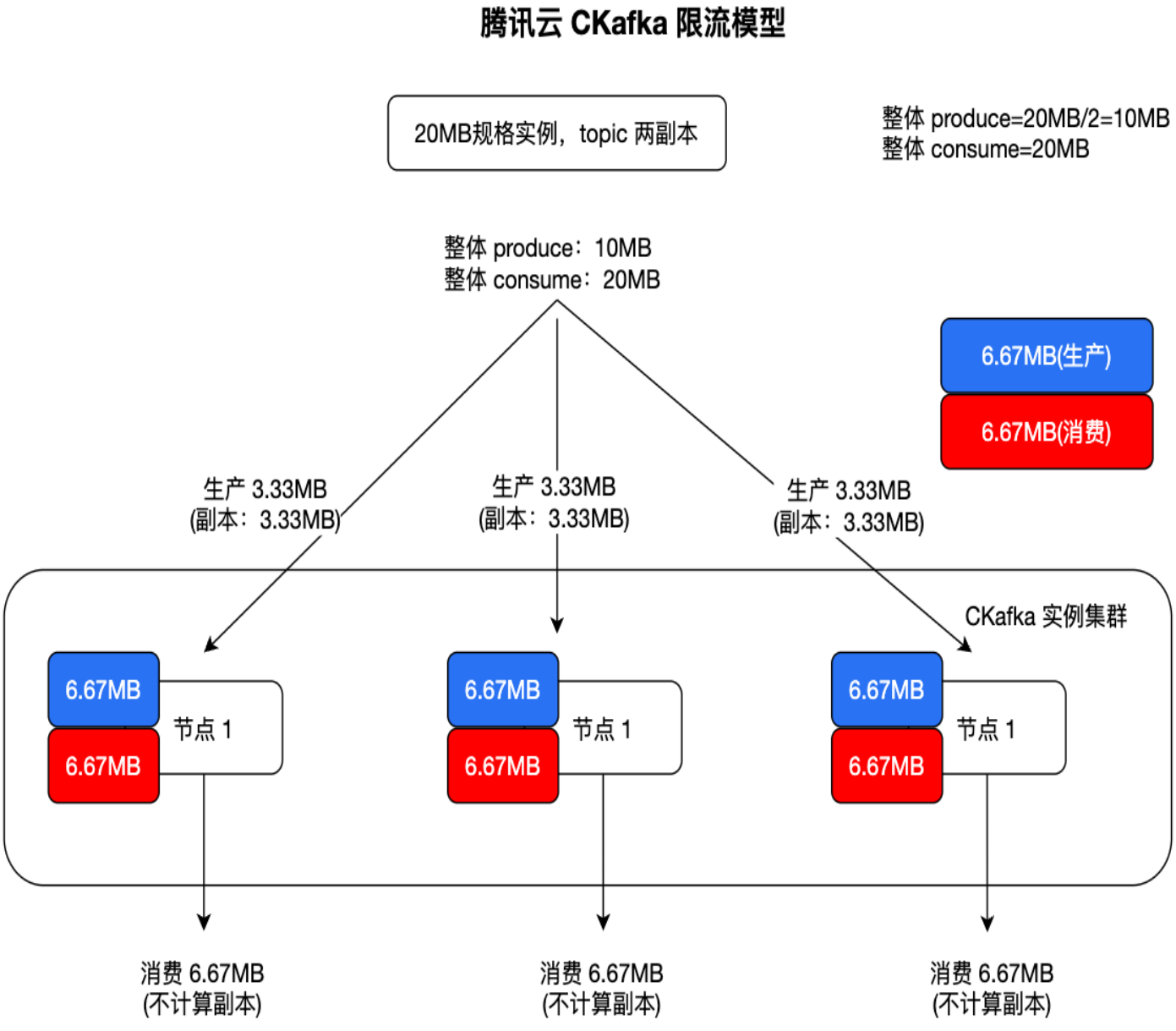
限流机制说明

最近更新时间：2025-06-17 16:09:32

概述

CKafka 生产者和消费者以极高的速度生产/消费大量数据或产生请求，从而消耗 Broker 上的资源，造成网络 IO 饱和。因此 CKafka 设计限流方案，对节点进行自我保护，避免因为资源消耗过高而影响全量业务。
以下以售卖规格为 20MB 的实例为例，其中需要说明几点：

1. CKafka 针对客户使用场景和需求，至少为 3 节点部署，因此 20MB 的流量，设计为每个节点承受 6.67MB 的读和写流量，因此在使用上建议您设置的分区数尽量为节点的 2-3 倍，使请求流量均衡，方能发挥最大效能。



2. 目前提供两种层级的限流能力，限流能力分别如下：

- **集群级限流**
  - 写限流整体限制为 20MB/s，意味着客户最多压测到的写流量(计算副本)为 20MB/s 附近，但是 3 节点，单节点配置 6.67MB/s，因此在单分区情况下写入最大 6.67MB/s，如果单节点双分区，计算副本流量，最大写入为 3.33MB/s。
  - 读限流整体限制为 20MB/s，意味着客户最多压测到的消费流量(不计算副本)为 20MB/s 附近。
- **Topic 级限流**
  - 客户可以配置 Topic 的限流，例如配置 Topic:Test，2 副本，写入限流 7MB/s（已计算副本），消费限流 20MB/s。
  - 写入限流为：7MB/s
  - 消费限流为：20MB/s

限流机制说明

CKafka 的限流机制是软限流，即当用户流量超过配额后，采用延时回包的方式进行处理，而不是给客户端返回报错。

以 API 限流为例，举例如下：

硬限流：假设调用频率为100次/s，当每秒内客户端调用超过100次时，服务端就会返回错误，客户端就需要根据业务逻辑进行处理。

软限流：假设调用频率为100次/s，正常耗时是10ms。当每秒内客户端调用超过100次时：

- 如为110次，则本次请求耗时20ms。
- 如为200次，则耗时为50ms。此时对客户端就是友好的，不会因为突增流量或者流量波动产生报错告警，业务可以正常进行。

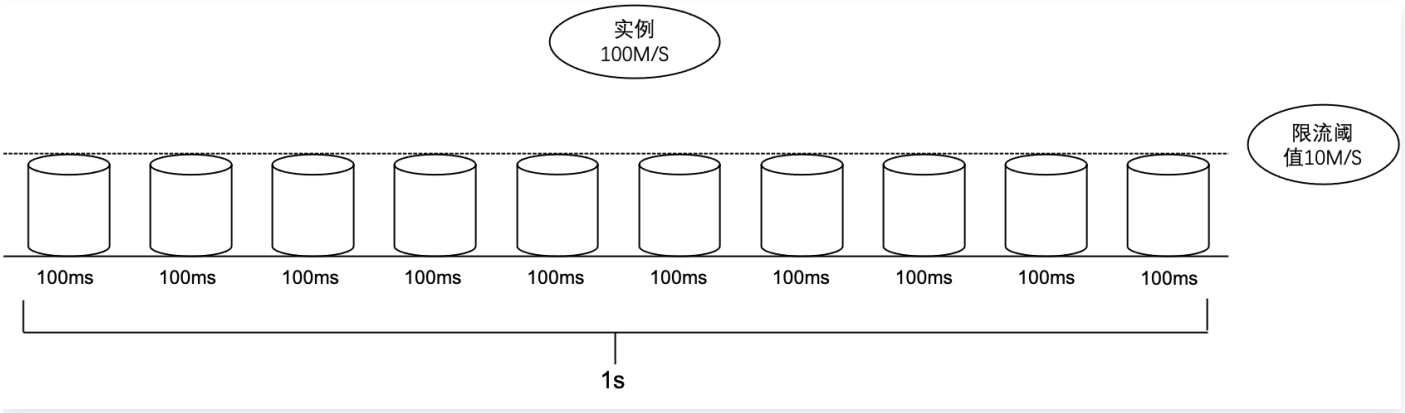
综上所述，在 Kafka 这种大流量的场景下，软限流是更符合用户体验的。

> **购买带宽和生产消费带宽的关系：**
> - 生产最大带宽（每秒）= 购买带宽 / 副本数

- 消费最大带宽（每秒）= 购买带宽

延时回包限流原理

CKafka 实例的底层限流机制是基于令牌桶原理实现的。将每秒分为多个桶，每个时间桶的单位为 ms。
限流策略会把每秒（1000ms）均分为若干个时间桶。例如分为10个时间桶，每个桶的时间则为100ms。每个时间桶的限流阈值就是总实例规格速度的1/10。如果某个时间桶内的 TCP 请求流量超过了该时间桶的限流阈值，会根据内部限流算法增加该请求的延时回包时间，使客户端无法快速收到 TCP 回包达到一段时间内的限流效果。



Ckafka 限流实践教程

分区数的规划与局部限流

由于 Ckafka 实例采多节点分布式部署提供整体的写入和消费服务，因此每个节点分配固定的读写限流额度，为了更好的提高 Ckafka 流量的使用率，需要您保证分区数尽量维持节点数的倍数（在分区数是节点的倍数，Ckafka 会尽可能让每个节点存储同样的分区数），流量尽可能均衡（特殊场景例如指定消息的 key 会使写入流量不均衡，默认情况下 Ckafka 的客户端会尽可能按照分区均衡流量发送到服务端），此时可以有效避免局部热点问题带来的局部限流问题。

实例限流次数与延时回包监控说明

Ckafka 的实例限流次数统计的是各个节点限流次数之和，并不代表整个实例的写入和消费性能，也不代表所有节点都触发了限流，因此当出现限流次数，但是限流流量整体看低于规格值，客户可以在高级监控中，查看各个节点的限流次数的统计，用于判断是哪个节点出现了限流。
遇到这类问题，建议尽可能调整分区数到节点的倍数，提高整体实例的带宽利用率。目前 Ckafka 采用延迟回包策略，因此，当出现限流后，您需要密切关注的指标是**延迟回包的时间**，该监控可以在专业版的高级监控中查看。

写入和消费限流模型的区别

由于生产涉及副本同步，所以要除以副本数；消费只从 leader 上拉取数据，所以不需要除以副本数。

- 单节点的最大写入流量 = 规格值 / 节点数 / 副本数。
- 单节点的消费流量 = 规格值 / 节点数。

关于偶发限流次数的说明

因为副本会占用写入限流，因此更多的副本，意味着写入流量会更低。其中限流实际执行是节点级别的执行，限流观测的是秒级的监控，客户看到的监控是分钟粒度的监控，因此，限流更为敏感，在整体写入流量大于70%（除以副本）时候，在秒级个别节点会出现局部限流，但是并不会持续很久，针对这个问题，客户可以在高级监控节点限流查看具体超了多少流量，如果对响应时间有较高要求，尽量规格预留 30% 的 Buffer。

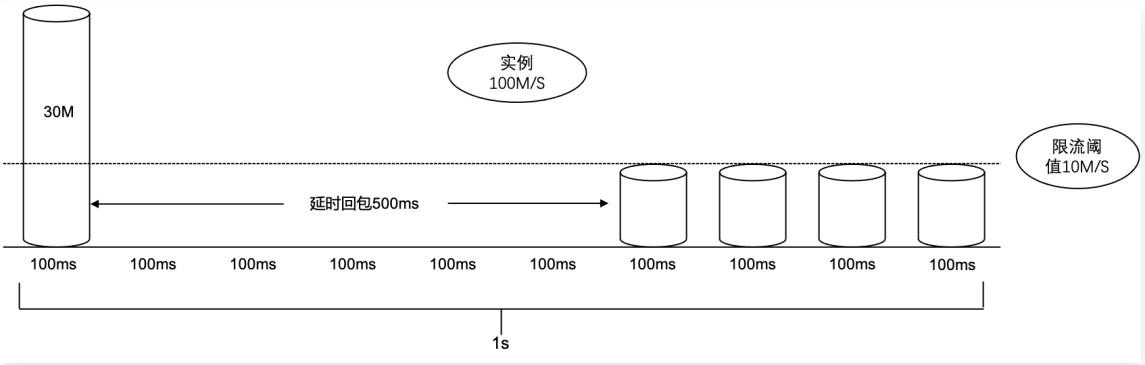
关于持续限流次数的说明

当实例出现限流次数，同时观察高级监控，每个节点都持续出现限流，实例的流量低于规格值 10% 以上，且排除 Topic 限流规则的影响，此种情况不符合预期，针对这类问题提工单支持。

常见现象

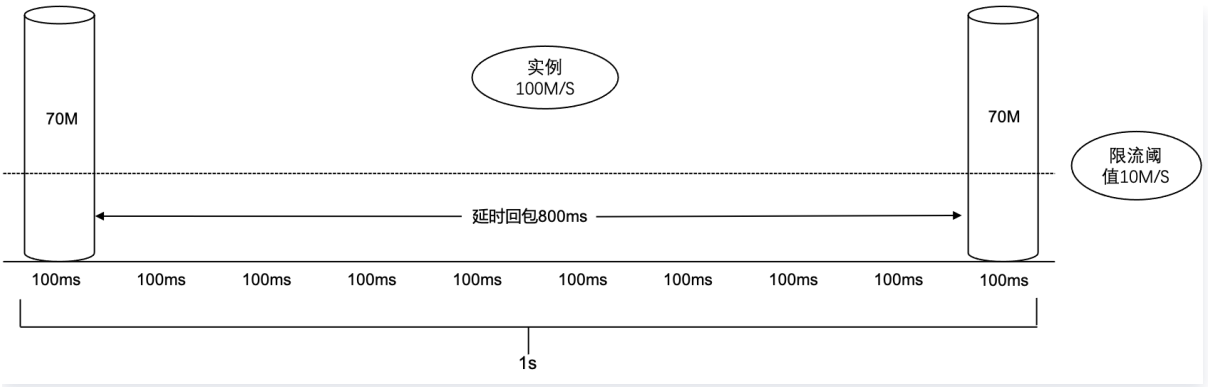
为什么监控生产/消费低于实例规格时会触发限流?

如上原理所述，因为限流是以 ms 为单位的，控制台监控平台数据是按每秒采集，分钟维度聚合（最大值或者平均值）。
按令牌桶原理可知，单个桶不会强制限制流量。如果实例 A 的带宽规格为 100MB/s，那么每个 100ms 的时间桶的限流阈值为 100MB/10 = 10MB/桶，假设实例A 的生产流量在某秒的第一个 100ms 时间桶达到了 30MB（时间桶限流阈值的3倍）。那么这时会触发 broker 限流策略增加延时回包时间，假设原先正常 TCP 返回时间是 100ms，超限后可能会增加 500ms 才返回。最终这秒的流量： 30MB × 1 + 0MB × 5 + 10MB × 4 = 70MB，即这秒内的流量速度为 70MB/s 小于实例规格 100MB/s。



为什么生产/消费峰值流量会高于实例规格?

再次假设实例 A 的带宽规格为 100MB/s，那么每个 100ms 的时间桶的限流阈值为 10MB，假设实例 A 的生产流量在某秒的第一个 100ms 时间桶达到了 70MB（时间桶限流阈值的7倍）。那么这时会触发 broker 限流策略增加延时回包时间，假设原先正常 TCP 返回时间是 100ms，超限后可能会增加 800ms 延时才返回，在第 900ms 回包后客户端立刻又在第10个时间桶打入了 70MB 流量。最终这秒的流量 (70MB × 1 + 0MB × 8 + 70MB × 1) = 140MB，即这秒内的流量速度为 140MB/s 大于实例规格 100MB/s。



限流次数为什么会暴增?

限流次数是以 TCP 请求统计的，如果实例A在某秒第一个时间桶流量打超了，那么超限后这个时间桶的剩余时间内所有的 TCP 请求都会被限制并统计限流次数。

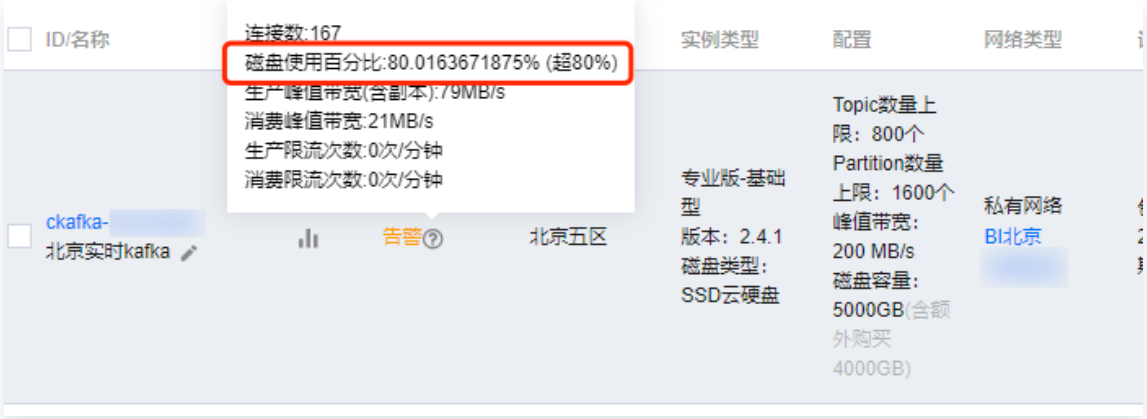
CKafka 如何进行限流?

为保证服务的稳定性，CKafka 在消息出入上都做了流量管控。

- 用户所有副本流量之和超过购买时的峰值流量时，会发生限流。
- 在生产端发生限流时，CKafka 会延长一个 TCP 链接的回应时间，延迟时间取决于用户瞬时流量超过限制的大小。和道路交通管制的原理有点类似，流量超得越多，延时算法得出来的延时值越高，最高5分钟。
- 在消费端发生限流时，CKafka 会缩小每次 `fetch.request.max.bytes` 的大小，控制消费端的流量。

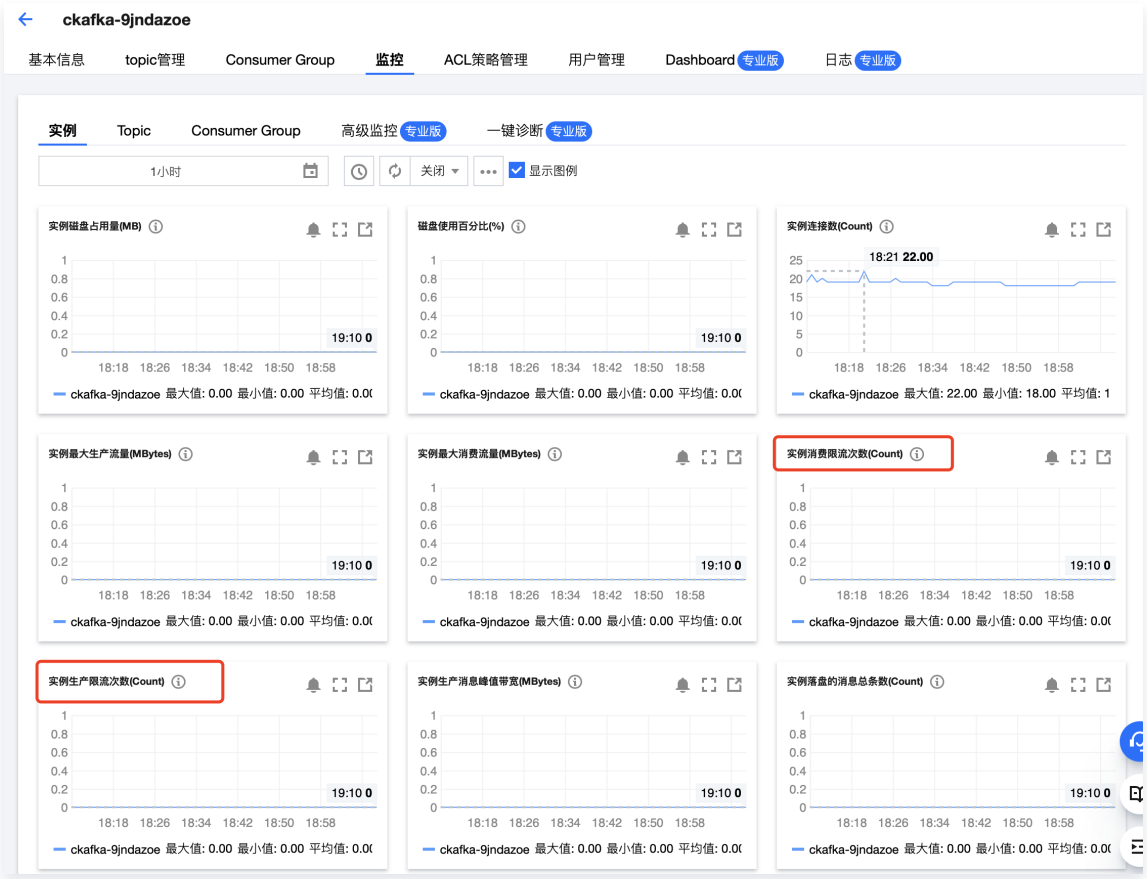
如何判断 CKafka 是否发生限流?

1. 在实例列表上，每个集群都有对应的健康度展示。当健康度显示为“警告”字样时，可以将鼠标移至其上查看弹出的详细数据。这个数据会展示当前用户的峰值流量以及发生限流的次数，用户可以根据这里的数据判断该实例是否发生过限流。



2. 用户可以打开监控数据查看流量的最大值，如果 **(流量的最大值 × 副本数) > 购买时的峰值带宽**，则表明至少发生过一次限流。可通过配置限流告警得知是否发生限流。

3. 在 CKafka 控制台的监控页面查看实例监控，当限流次数大于0，证明发生过限流。



开源对比

最近更新时间：2025-05-26 17:00:03

TDMQ CKafka 版和开源自建 Kafka 的详细如下：

项目	功能项	TDMQ CKafka 版	自建 Kafka
基础功能	Topic 参数配置	支持在控制台调整多项参数，包括分区数、副本数、消息清理策略、最小同步副本数、Segment 分片滚动的时长、消息保留时长等。	命令行配置，业务人员难以自定义操作。
	Topic 分区数变更	支持在控制台变更 Topic 分区数。	命令行配置，业务人员难以自定义操作。
	发送消息	支持在控制台发送消息。	不支持。
	消费进度重置	支持在控制台重置消费位点。	后端手动配置，繁琐容易出错。
	管理能力	• 支持对集群、Topic、消费组等进行可视化管理，通过标签可以提供更多精细化管理能力； • 支持对 Topic 生产、消费流量统计，流量明细一目了然。	需要搭配开源管理系统，易用性较弱。
智能化运维	**磁盘自动均衡**	支持对数据倾斜的 CKafka 集群实例进行 Broker 之间的磁盘数据均衡，减少数据倾斜对集群的压力，降低对磁盘容量的需求。	不支持。
	智能巡检	提供一键诊断能力，能够主动排查集群问题和隐患，并基于腾讯云专家经验沉淀给出问题解决方案，自动归纳健康检查结果生成报告，实现运维体验闭环。	不支持。
	磁盘智能配置	支持磁盘水位自动处理。当磁盘水位达到阈值时，支持动态调整消息保留或者自动扩容磁盘，避免影响业务。	不支持。
隔离性	多实例	支持基于一套物理集群创建多个实例。	不支持。
	带宽配额控制	每个实例支持按带宽限制使用量。	不支持。

版权所有：腾讯云计算（北京）有限责任公司

	Topic 限流	支持 Topic 维度的限流，最大化的保障业务可靠性。	不支持。
	磁盘配额控制	每个实例支持按磁盘限制使用量。	不支持。
	多接入点配置	支持配置实例路由策略，在指定 VPC 下设置接入点。	不支持，发送消息需指明 Broker 地址，接收消息需 ZooKeeper 协调调度，不够灵活。
	管控面与数据面隔离	架构上管控面与数据面隔离，管控操作通过腾讯云云 API SDK 调用管控组件实现。数据面操作通过开源 Kafka SDK 调用，完全兼容开源。	不支持。
	主账号隔离	基于腾讯云主子账号体系，实现主账号之间数据逻辑隔离。	不支持。
监控告警	开箱即用	成熟标准化的部署、监控方案，开箱即用	不支持，用户需要自行搭建并维护开源监控系统，且监控系统如果不能满足需求，难以定制化。
	监控告警	提供生产和消费峰值带宽、生产和消费平均带宽、生产和消费消息数、磁盘使用率、实例连接数、未消费消息数等丰富监控指标，支持实例、Topic 和 消费分组多维度监控告警。	不支持，需要额外对接开源第三方组件。
	指标排序	支持查看 Topic 生产消费流量 Top10 和占用磁盘容量 Top10，Consumer Group 消费速度 Top10。	不支持。
高可用	**跨 AZ 高可用部署**	支持跨 AZ 高可用部署，成熟的故障恢复方案。支持从单可用区部署升级到跨可用区部署。	支持但较为繁琐，需要自行设计部署方案和参数。
	跨地域容灾	支持实例级别同步数据，支持不同地域实例间的复制迁移功能，实例的所有数据和元数据都会同步。	支持但较为繁琐，且开源方案有一定缺陷。
安全合规性	Topic 维度 ACL 访问控制	支持按照用户 + IP 段控制访问权限，权限配置支持到实例+Topic 维度，而且支持黑名单和白名单两种模式。	通过命令行配置参数，繁琐且容易出错。

	SASL 密码认证	支持控制台配置 SASL 认证。	通过命令行配置参数，繁琐且容易出错。
	SSL认证	支持控制台配置 SSL 认证。	通过命令行配置参数，繁琐且容易出错。
	管控操作权限控制	控制台操作基于腾讯云 CAM 控制权限，安全合规。	管控操作权限难以精细化控制。
	操作可追溯	管控操作对接操作审计，可回溯。	不支持。
其他	**在线升级能力**	CKafka 提供版本升级能力，方便在社区版出现 bug 或者安全漏洞时，无缝升级到新版本。	不支持，需要手动操作，手动升级可能会影响业务。

版权所有：腾讯云计算（北京）有限责任公司

使用限制

最近更新时间：2025-03-17 14:34:22

本文列举了消息队列 CKafka 版中对一些指标和性能的限制，请您在使用中注意不要超出对应的限制值，避免出现异常。

限制项	说明
Topic 总数量	根据产品规格不同固定上限，详情可参见 计费概述 。
Partition 数量	• 单 Topic 支持的 Partition 数量限制为3000个。 • 实例级别 Partition 数量限制包含了副本数量，副本数量一般为2或者3。 • 不支持缩减 partition 数量。
Partition 吞吐	• 在 ack = 1的情况下，受 CKafka 的分区架构、业务数据大小、请求频率等因素的影响，**CKafka 单分区的吞吐在30MB/s - 60MB/s之间**。 • 在 ack = -1(强一致)的情况下，受 CKafka 的分区架构、业务数据大小、请求频率等因素的影响，为了保证请求的耗时的稳定，**建议 CKafka 单分区的吞吐在10MB/s - 20MB/s之间**。 • 在使用到幂等或者事务的场景下，CKafka 服务端的负荷会进一步提升，对单分区的最大吞吐也会产生一定影响，如果业务关注延时情况，建议您选择不同的集群规格进行压测。
耗时	CKafka 是大流量、高吞吐的消息队列，无法保证每条请求的耗时都是低延时。建议超时时间设置如下： • 生产端当 ack = 1 的时候，超时时间默认**设置为 30s**。 • 生产端在 ack = -1 的时候，超时时间默认**设置为 60s**。 • 消费端的超时时间**设置为 60s**。
Consumer group 数量	• 标准版实例级别 Consumer group 数量建议不超过50个。 • 高级版和专业版实例级别 Consumer group 数量建议不超过500个，支持 联系我们 申请升级。
客户端使用	• 建议使用 CKafka 管控台或者云 API 进行集群、Topic 等元数据管理。 • 不建议使用通过 AdminClient 直接操作创建 Topic 、修改分区等元数据，避免引起控制台数据不一致的情况，后续计划封禁。
实例	• 不支持变更实例地域属性。 • 标准版客户端实例最大连接数量为5000，高级版和专业版客户端实例最大连接数量为50000，实例连接超过该最大值会导致客户端无法创建新的连接，如果评估该最大值在实际业务中不合理可以 联系我们 申请变更。

版本	• 兼容开源2.4、2.8、3.2版本。
多路由	单个实例最多创建5条路由，只能创建1条公网路由。
公网带宽	默认提供 3Mbps 免费公网带宽，专业版实例可以额外升配公网带宽。
暴露 ZooKeeper	不支持。
暴露底层资源	不支持，避免用户自行操作所带来的风险。
消息大小	不超过 12MB，若超过 12MB 消息会发送失败。
标签	每个云资源允许的最大标签数量为50。
控制台并发操作限制	由于某些管控操作会涉及实例元数据信息的操作与修改，当并发很高时可能会出现一致性问题，影响底层分发，所以会对某些接口操作加锁来限制请求并发度。 为了提高控制台操作的稳定性与成功率，目前只允许同一时间对**单个实例**进行至多 **20** 次控制台并发请求（包含 SDK 直接调用云 API 的方式）。
数据可靠性	由于 CKafka 自身的副本同步和 ACK 机制，会有一些极端情况出现数据丢失。例如副本数为 2 、 ack = 1，当分区发生 leader 切换时，而此时副本并未同步完成，这就会导致数据丢失的情况发生。如果您的业务无法容忍数据丢失，请务必参见 CKafka 数据可靠性说明 进行客户端配置。

说明：

受 CKafka 的分区架构、业务数据大小、请求频率、物理层稳定性等因素的影响，CKafka 无法保证每条请求的耗时都是低延时。

- 我们会尽量保证如下两种情况：

  1.1 单实例每月生产低耗时占比，和该实例的 SLA 保持一致。

  1.2 单实例每月消费低耗时占比，和该实例的 SLA 保持一致。

- 比例计算公式如下：

  当月生产低耗时占比 =（该月所有每分钟生产 P999 耗时未超过 30s 的分钟数 / 月度服务总分钟数）× 100%。

  当月消费低耗时占比 =（该月所有每分钟消费 P999 耗时未超过 60s 的分钟数 / 月度服务总分钟数）× 100%。

地域和可用区

最近更新时间：2025-03-17 14:34:22

地域（Region）是指物理的数据中心的地理区域。可用区（Zone）是指腾讯云在同一地域内电力和网络互相独立的物理数据中心。详细介绍请参见 云服务器-地域和可用区 。

支持的地域

中国地区

地域		取值
华南地区	广州	ap-guangzhou
	深圳	ap-shenzhen
	深圳金融	ap-shenzhen-fsi（仅限金融机构和企业通过 在线咨询 申请开通。）
华东地区	上海	ap-shanghai
	上海金融	ap-shanghai-fsi（仅限金融机构和企业通过 在线咨询 申请开通。）
	南京	ap-nanjing
	杭州	ap-hangzhou
	上海自动驾驶云	ap-shanghai-adc
华北地区	北京	ap-beijing
	北京金融	ap-beijing-fsi（仅限金融机构和企业通过 在线咨询 申请开通。）
	天津	ap-tianjin
华中地区	长沙	ap-changsha
	武汉	ap-wuhan
西南地区	成都	ap-chengdu
	重庆	ap-chongqing
港澳台地区	中国台北	ap-taipei

	中国香港	ap-hongkong

其他国家和地区

地域		取值
东南亚地区	新加坡	ap-singapore
	泰国	ap-bangkok
	雅加达	ap-jakarta
亚太地区	首尔	ap-seoul
	东京	ap-tokyo
美国西部	硅谷	na-siliconvalley
美国东部	弗吉尼亚	na-ashburn
南美地区	圣保罗	sa-saopaulo
欧洲地区	法兰克福	eu-frankfurt

如何选择地域和可用区

关于选择地域和可用区时，您需要考虑以下几个因素：

- CKafka 实例所在的地域、您以及您的目标用户所在的地理位置。
  建议您在购买 CKafka 实例时，选择最靠近您客户的地域，以降低访问时延、提高访问速度。
- CKafka 实例和其他云产品的关系。
  建议您在选择其他云产品时，尽量都在同个地域同个可用区，以便各云产品间可通过内网进行通信，降低访问时延、提高访问速度。
- 业务高可用和容灾考虑。
  即使只有一个私有网络的场景下，建议您将业务至少部署在不同的可用区，以保证可用区间的故障隔离，实现跨可用区容灾。
- 不同可用区间可能会有网络的通信延迟，需要结合业务的实际需求进行评估，在高可用和低延迟之间找到最佳平衡点。
- 如果您需要访问其他国家和地区的主机，建议您选择其他国家和地区的 CKafka 实例进行访问。如果您在 中国 创建 CKafka 实例，访问 其他国家和地区的主机 会有较高的访问延迟，不建议您使用。

资源位置说明

这里说明腾讯云 CKafka 哪些资源是全球性的、哪些资源是区分地域不区分可用区的，以及哪些资源是基于可用区的。

资源	资源 ID 格式（<资源缩写>-8位数字及字符）	类型	说明
用户账号	不限	全球唯一	用户可以使用同一个账号访问腾讯云全球各地资源。
CKafka 实例	ckafka-xxxxxxxx	只能在单地域的单个可用区下使用	用户只能在特定可用区下创建 CKafka 实例。

相关操作

将实例迁移到其他可用区

CKafka 支持将实例在同一地域下进行跨可用区的迁移，如果用户有这方面的需求需要 提交工单 进行处理。