

SOFAStack

消息队列 SOFAKafka 产品简介

产品版本：AntStack Plus 1.11.0

文档版本：20221019



法律声明

蚂蚁集团版权所有©2022，并保留一切权利。

未经蚂蚁集团事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

商标声明

 蚂蚁集团 ANT GROUP 及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

免责声明

由于产品版本升级、调整或其他原因，本文档内容有可能变更。蚂蚁集团保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在蚂蚁集团授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置>网络>设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击 确定 。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.概述	05
2.产品优势	06
3.产品架构	07
4.功能特性	08
5.应用场景	10
6.基础术语	11

1.概述

SOFAShark Kafka 消息队列（SOFAShark Kafka，简称 SOFAShark Kafka）是基于 Apache Kafka 构建的分布式消息中间件，并与金融分布式架构 SOFAShark 深度集成，广泛用于日志收集、监控数据聚合、流式数据处理、在线和离线分析等领域。

SOFAShark Kafka 主要特点是基于 Pull 的模式来处理消息消费，追求高吞吐量。最初的目的就是用于日志收集和传输，适合产生大量数据的互联网服务的数据收集业务。而 SOFAMQ 具有高吞吐量、高可用性、适合大规模分布式系统应用的特点。相对 SOFAShark Kafka，它对消息的可靠传输及事务性做了优化，目前被广泛应用于交易、充值、流计算、消息推送、日志流式处理、Binglog 分发等场景。

2. 产品优势

本文主要介绍 SOFAKafka 版相比于自建开源 ApacheKafka 所具备的优势。

开箱即用

- SOFAKafka 兼容社区版 Kafka 的 API，具备原生 Kafka 的所有消息处理特性。
- SOFAKafka 100% 兼容开源 Apache Kafka，您可以直接使用开源 Apache Kafka 客户端与 SOFAKafka 通讯。

重要

SOFAKafka 支持的客户端版本为 0.10.x~2.6.0，推荐客户端与服务端使用一致的版本。

高可用性

- 数据持久化：专业团队保障更高可用性，消息持久化落盘到消息队列，数据高可靠、服务高可用。
- 高吞吐能力：在海量消息堆积的情况下，始终能保持 SOFAKafka 集群的高吞吐能力。

数据安全

SOFAKafka 利用 SASL 机制对用户身份进行认证，并利用 SSL 对通道进行加密传输，确保数据在传输过程中不被窃取或篡改，保证您的数据安全。

高扩展性

您可以根据自身业务规模按需扩容，上层业务无感知。SOFAKafka 支持集群扩容和分区扩容两种模式。

- 集群扩容：支持横向扩容节点。
- 分区扩容：支持快速扩容分区。

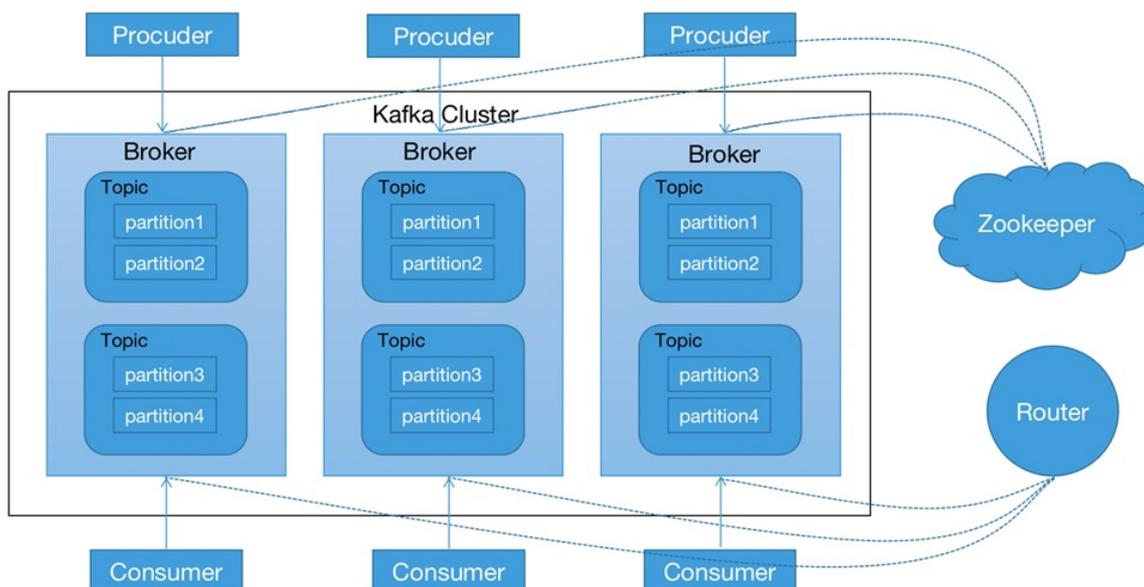
专业服务

SOFAKafka 拥有专业且经验丰富的运维团队，以及成熟的运维体系。监控报警：SOFAKafka 提供完整的监控图表和报警，帮助您及时发现问题。

3. 产品架构

本文介绍 SOFAKafka 的系统结构和发布/订阅模型。

系统部署架构如下图所示。



SOFAKafka 消息队列包括以下 6 个组件：

- **Zookeeper**：存储 SOFAKafka 集群的元数据信息，比如记录注册的 Broker 列表、Topic 元数据信息、partition 元数据信息等。
- **Broker**：消息中转角色，负责存储和转发消息。每个 Kafka 节点都是一个 Broker，Broker 存储了 Topic 的数据。一个 Topic 的数据可能存在多个 Broker 节点上，Topic 的数据会有多个副本，每个 Broker 节点既可能存储主副本数据也可能存储备份副本数据。
- **生产者 Producer**：消息发布者，与 Broker 集群建立长连接。主要作用是生产数据，并将产生的数据推送给 SOFAKafka 集群。
- **消费者 Consumer**：消息消费者，与 Broker 集群建立长连接。主要作用是消费 SOFAKafka 集群的数据。Consumer 既可以主动拉取 Broker 的数据也可以被动接收 Broker 推送数据。
- **Console**（未在图中展示）：SOFAKafka 的图形化管理控制台。
- **Router**：LDC 场景下，根据配置的 Router 规则，进行逻辑单元之间消息转发，使 SOFAKafka 支持 LDC 能力的组件。

发布/订阅模型

- **发布**：Consumer Group 和 Topic 的对应关系是 N : N，即一个 Consumer Group 可以同时订阅多个 Topic，一个 Topic 也可以被多个 Consumer Group 同时订阅。
- **订阅**：某个 Topic 的一条消息可以被多个 Consumer Group 同时订阅，但只能被同一个 Consumer Group 内的任意一个 Consumer 消费。

4. 功能特性



快速开发集成

资源管理

SOFAKafka 控制台支持 Topic 管理、Group 管理、消息查询，用户可以直接通过控制台发送消息。

开源兼容

100% 兼容开源 Apache Kafka，用户可以直接使用开源 Apache Kafka 客户端与 SOFAKafka 通讯。

多场景支持

消息类型

- 普通消息：消息队列中无特性的消息，区别于有特性的事务消息。
- 事务消息：实现类似 X/Open XA 的分布事务功能，以达到事务最终一致性状态。

消息模式

- 支持发布订阅模式。
- 支持集群消费和广播消费。
- 支持 Tag 过滤，消费者可以根据指定的 Tag 进行消息消费。

便捷运维管控

消息全生命周期管理

- 消息回溯：根据时间或位点重置消费进度，允许用户进行消息回溯或者丢弃堆积消息。
- 消息查询：支持按照位点或者时间查询消息。

精细化监控告警

提供完整的监控图表和报警，支持查看监控账户下创建的资源，包括实例、Topic、Consumer Group 等，帮助用户实时掌握资源状态，针对可能存在的问题及时处理，保障其稳定运行。

高性能低延迟

- 高吞吐：千万级高并发处理能力。

- 低延迟：毫秒级端到端延迟。
- 抗堆积：海量消息堆积能力。
- 可扩展：可根据业务需求弹性扩展。

高可用运维

- 支持多种容灾场景，支持单机房高可用、同城双活、两地三中心、异地多活部署架构。
- 高可用巡检，问题自动发现。

高安全

- 多租户资源隔离。
- 细粒度授权管理。

5. 应用场景

SOFAKafka 具有分布式、高吞吐、可扩展的特性，主要用于不同系统间的数据交流和传递，在企业解决方案、电商、物联网等众多领域都有广泛应用，其中主要包括以下应用场景。

日志收集

SOFAKafka 可以收集各种服务的 log，支持通过 SOFAKafka 以统一接口服务的方式开放给各种 Consumer。

消息系统

解耦生产者和消费者、缓存消息等。

用户活动跟踪

SOFAKafka 看用于记录 Web 用户或者 APP 用户的各种活动，如浏览网页、搜索、点击等活动，这些活动信息被各个服务器发布到 SOFAKafka 的 Topic 中，然后消费者通过订阅这些 Topic 来做实时的监控分析，也可以保存到数据库。

运营指标

SOFAKafka 可用于记录运营监控数据，包括收集各种分布式应用的数据，生产各种操作的集中反馈，比如报警和报告。

流式处理

支持 Spark Streaming 和 Storm 接入 SOFAKafka。在大数据相关的业务场景中，使得 SOFAKafka 能够支持流式数据的处理，并能方便的进行数据聚合。

6. 基础术语

中文	英文	释义
消息主题	Topic	消息主题，一级消息类型，通过 Topic 对消息进行分类。
主题分区	Partition	一个主题对应多个分区，生产者发送给 Topic 的消息会进入其中一个分区。
消息	Message	消息队列中信息传递的载体。
偏移量	offset	主题分区内的消息会分配从 0 开始的 offset 作为唯一标识。消费者根据 offset 标识自己当前消费位置。
Message Key	Key	消息的附带内容，相同的 Key 会 hash 到同一个 partition 内。
消息生产者	Producer	消息生产者，也称为消息发布者，负责生产并发送消息。
Producer 实例	Producer instance	Producer 的一个对象实例，不同的 Producer 实例可以运行在不同进程内或者不同机器上。Producer 实例线程安全，可在同一进程内多线程之间共享。
消息消费者	Consumer	消息消费者，也称为消息订阅者，负责接收并消费消息。
Consumer 实例	Consumer instance	Consumer 的一个对象实例，不同的 Consumer 实例可以运行在不同进程内或者不同机器上。一个 Consumer 实例内配置线程池消费消息。
Group	Group	一类 Consumer，这类 Consumer 通常消费同一类消息，且消费的逻辑一致。
Group ID	Group ID	Group 的标识。
队列	Queue	每个 Topic 下会由一到多个队列来存储消息。

中文	英文	释义
集群消费	Clustering consumption	一个 Group ID 所标识的所有 Consumer 平均分摊消费消息。例如某个 Topic 有 9 条消息，一个 Group ID 有 3 个 Consumer 实例，那么在集群消费模式下每个实例平均分摊，只消费其中的 3 条消息。
广播消费	Broadcasting consumption	一个集群的所有消费者都分配唯一的 Group ID。例如某个 Topic 有 9 条消息，一个集群有 3 个 Consumer 实例分配 3 个 Group Id，那么在广播消费模式下每个实例都会各自消费 9 条消息。
事务消息	Transactional message	消息队列提供类似 X/Open XA 的分布事务功能，通过消息队列的事务消息能达到分布式事务的最终一致。
顺序消息	Ordered message	消息队列提供的一种按照顺序进行发布和消费的消息类型，分为全局顺序消息和分区顺序消息，当前仅支持分区顺序消息。
分区顺序消息	Partitionally ordered message	对于指定的一个 Topic，所有消息根据 Sharding Key 进行区块分区。同一个分区内的消息按照严格的 FIFO 顺序进行发布和消费。Sharding Key 是顺序消息中用来区分不同分区的关键字段，和普通消息的 Message Key 是完全不同的概念。
消息堆积	Message accumulation	Producer 已经将消息发送到消息队列的服务端，但由于 Consumer 消费能力有限，未能在短时间内将所有消息正确消费掉，此时在消息队列的服务端保存着未被消费的消息，该状态即消息堆积。
消息过滤	Message filtering	Consumer 可以根据消息 Key 对消息进行过滤，确保 Consumer 最终只接收被过滤后的消息类型。消息过滤在消息队列的服务端完成。
重置消费位点	Reset consumption offset	以时间轴为坐标，在消息持久化存储的时间范围内（默认 3 天），重新设置 Consumer 对已订阅的 Topic 的消费进度，设置完成后 Consumer 将接收设定时间点之后由 Producer 发送到消息队列服务端的消息。