

Please visit website: <http://cxyroad.com>

腾讯面试：如何提升Kafka吞吐量？

=====

![公众号首图__2024-05-23+16_34_14.jpg](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/8a0a0e533e9c40f3893f3b02034a2033~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=900&h=383&s=20834&e=jpg&b=000000)

Kafka 是一个分布式流处理平台和消息系统，用于构建实时数据管道和流应用。它最初由 LinkedIn 开发，后来成为 Apache 软件基金会的顶级项目。

Kafka 特点是****高吞吐量、分布式架构、支持持久化、集群水平扩展和消费组消息消费****，具体来说：

1. ****高吞吐量****：Kafka 具有高性能和低延迟的特性，能够处理大规模数据，并支持每秒数百万条消息的高吞吐量。
2. ****分布式架构****：Kafka 采用分布式架构，可以水平扩展，多个节点之间能够实现负载均衡和高可用性。
3. ****可持久化****：Kafka 将消息持久化到磁盘中，保证消息的可靠性，即使消费者下线或出现故障，消息也不会丢失。
4. ****集群水平扩展****：Kafka 支持集群模式，可以方便地通过增加节点和分区来水平扩展、提高容量。
5. ****消息组支持****：Kafka 可以支持多个消费者订阅同一个主题 (Topic)，每个消费者组独立消费消息，方便构建多样化的数据处理架构。

并且与其他两个主流的中间件 RabbitMQ 和 RocketMQ 相比，Kafka 最大的优势就是高吞吐量。

既然高吞吐量是 Kafka 的优势，那么怎么才能让 Kafka 的优势发挥到极致，怎么才能更大程度的提升 Kafka 的吞吐量呢？

典型回答

提升 Kafka 的吞吐量涉及优化生产者、消费者、服务器配置以及整体架构设计等多个方面，以下是 Kafka 优化的一些关键策略和具体实现。

1. 生产者优化

生产者提升吞吐量的优化手段有以下几个：

1. ****消息批量发送****：增加 `batch.size`（批量消息数量设置）和适当调整 `linger.ms`（批次逗留时间），以允许生产者累积更多消息后再发送，减少网络请求次数。
2. ****消息压缩****：设置 `compression.type`（默认值为 `none`，不压缩），该参数为生产者发送数据的压缩方式，包括 `gzip`、`snappy`、`lz4`、`zstd` 等。启用消息压缩（如 `Snappy`、`LZ4`），减少网络传输的数据量，尽管这会增加 CPU 负担。
3. ****增大缓冲区大小****：通过增加 `buffer.memory` 配置（生产者内存缓冲区大小），允许生产者在等待发送时缓存更多消息。
4. ****优化 acks 配置****：适当降低 `acks` 级别以减少等待确认的时间，但需权衡数据的持久性。`acks` 级别含义如下：
 - * ****acks=0****：生产者不会等待来自 Broker 的消息发送成功与否的确认，如果 Broker 没有收到消息，那生产者是不知道的。该配置吞吐量高，但可能会丢失数据。
 - * ****acks=1****：默认值，生产者将消息写入 leader 副本后，就会收到 Broker 的确认消息。如果 leader 副本同步成功了，但还没有来得及同步给 follower 副本，此时就发生宕机了，那就会丢失数据。
 - * ****acks=-1****：生产者将消息写入 leader 副本和所有 follower 副本后，才会收到 Broker 的确认消息。该配置可以保证不丢数据，但是吞吐量低。
5. ****并行生产****：利用多线程或多生产者实例并行发送消息。

2. 消费者优化

生产者提升吞吐量的优化手段有以下几个：

1. ****增加消费者实例****：确保每个分区至少有一个消费者，以充分利用并行处理能力。
2. ****增加每次拉取的消息数量****：通过调整 `fetch.min.bytes`（消息拉取最小容量）和 `fetch.max.bytes`（消息拉取最大容量）增加每次拉取的消息数量。
3. ****并行处理****：在消费者内部使用多线程处理消息。

3. Kafka Broker配置优化

每个 broker 就是一个 Kafka 实例，它的优化手段有以下几个：

1. ****增加分区数量****：适当增加主题的分区数量，可以提高并行处理能力，但需避免过多分区导致的管理和协调开销。
2. ****优化节点配置****：包括但不限于 `num.network.threads`（网络线程数）、`num.io.threads`（I/O 线程数）、`socket.send.buffer.bytes/socket.receive.buffer.bytes`（套接字缓冲区大小）等，根据硬件资源和负载情况调整。
3. ****磁盘优化****：使用快速磁盘（如 SSD），并优化文件存储目录的布局以减少 I/O 竞争。
4. ****JVM调优****：Kafka 是运行在 JVM 上的，针对 Kafka 服务端的 JVM 进行适当的内存和 GC 优化，也可以提升有效的提升吞吐量。

4. 网络与硬件优化

网络和 Kafka 运行的硬件，也会影响 Kafka 的吞吐量，所以我们可以进行以下优化：

1. ****网络优化****：确保网络连接质量良好，减少网络延迟和丢包。
2. ****硬件升级****：增加服务器的 CPU、内存和磁盘性能。

5. 集群副本策略优化

合理配置副本放置，确保高可用的同时，减少跨数据中心的复制延迟，也可以有效的提升 Kafka 的吞吐量。

6. 监控与压测

1. ****持续监控****：使用 Kafka 自带的监控工具或集成第三方监控系统（如 Prometheus+Grafana），持续监控性能指标。
2. ****压测于调试****：基于监控数据和性能测试结果，不断调整上述参数以找到最优配置。

课后思考

除了以上策略外，还有没有其他提升 Kafka 吞吐量的手段？

> 本文已收录到我的面试小站 [www.javacn.site](http://cxyroad.com/"https://www.javacn.site"), 其中包含的内容有：Redis、JVM、并发、并发、MySQL、Spring、Spring MVC、Spring Boot、Spring Cloud、MyBatis、设计模式、消息队列等模块。

原文链接: <https://juejin.cn/post/7371753324154290176>