使用Python掌握Apache Kafka应当了解的3个库

作者：闻数起舞 2020-05-29 09:48:54

开发 后端 Kafka

Apache Kafka是一个分布式流平台，可以实时发布，订阅，存储和处理消息。 其基于拉式的体系结构减轻了重负载对服务的压力，使其易于扩展。 它以们从源移动到目的地。

关于推与拉架构的思考

我最近与人们讨论了不同服务架构的优缺点...

Kafka是基于JVM的平台，因此客户端的主流编程语言是Java。 但是，随着社区的蓬勃发展，高质量的开源Python客户端也已面世，并已用于生产中。

在本文中，我将介绍最著名的Python Kafka客户端：kafka-python，pykafka和confluent-kafka并进行比较。 最后，我将对每个图书馆的利弊发表自己的看

我们为什么要Kafka?

首先是第一件事。 为什么选择Kafka? Kafka旨在增强事件驱动的体系结构。 它通过提供高吞吐量，低延迟，高耐用性和高可用性解决方案来增强体系结构着您可以同时拥有所有它们，总会有一个权衡。阅读此白皮书以了解更多信息。）

如何为高性能和低延迟部署和优化Kafka

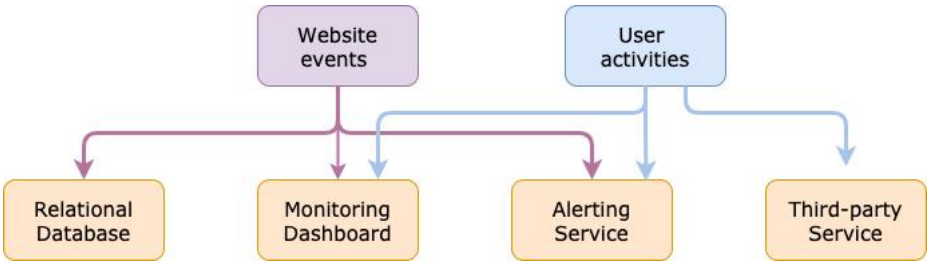
Apache Kafka®是一个功率流处理平台，他的白皮书讨论了如何针对以下情况优化Kafka部署：

除了其高性能外，另一个吸引人的功能是发布/订阅模型，其中发件人没有专门向收件人发送邮件。 而是根据主题将邮件传递到收件人可以订阅的集中位置

这样，我们可以轻松地将应用程序解耦并摆脱整体设计。 让我们看一个例子，了解为什么去耦效果更好。

您创建的网站需要将用户活动发送到某个地方，因此您可以编写从网站到实时监控仪表板的直接连接。 这是一个简单的解决方案，效果很好。 有一天，您存储在数据库中以备将来分析。 因此，您将另一个直接数据库连接写入到您的网站。 同时，您的网站越来越多的流量，并且您想通过添加警报服务，实时强它的功能。

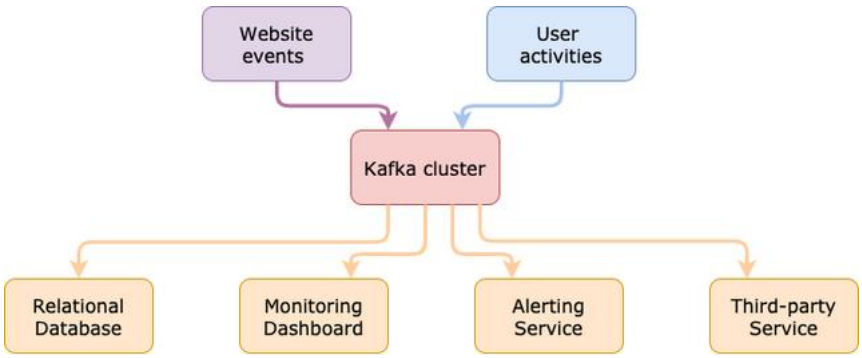
您的架构最终将像这样。 诸如大量代码仓库，安全性问题，可伸缩性问题和可维护性问题之类的问题将伤害您。



> Architecture without decoupling (Created by Xiaoxu Gao)

您需要一个中心来分隔具有不同角色的应用程序。 对于创建事件的应用程序，我们称它们为生产者。 他们将事件发布到集中式中心。 每个事件(即消息)都消费者位于枢纽的另一侧。 他们从中心订阅了他们需要的主题，而无需直接与制作人交谈。

有了此模型，就可以轻松扩展和维护体系结构。 工程师可以将更多精力放在核心业务上。



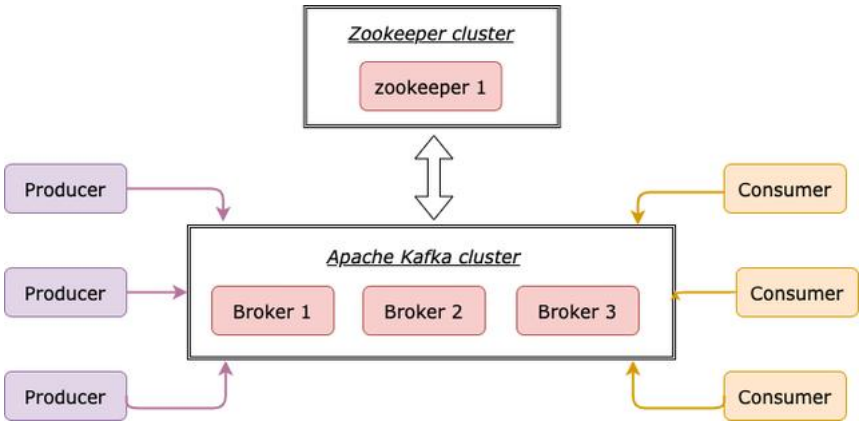
> Architecture with decoupling (Created by Xiaoxu Gao)

简而言之，Kafka设置

您可以从官方网站下载Apache Kafka。 快速入门可帮助您在10秒钟内启动服务器。

您也可以从Confluent平台下载Apache Kafka。 它是迄今为止最大的面向Kafka的流数据平台。 它为个人和企业提供了一系列围绕Kafka的基础架构服务，实时流提供。 创始人是最初创建Apache Kafka的团队的成员。

每台Kafka服务器都称为代理，您可以以独立模式运行它或形成集群。 除了Kafka，我们还需要Zookeeper存储有关Kafka的元数据。 Zookeeper的行为就像责管理分布式系统中每个代理的状态。



> Kafka setup (Created by Xiaoxu Gao)

假设我们已经与1位Zookeeper和1位Kafka经纪人建立了基础架构。 现在该连接了! 原始的Java客户端提供5个API：

- 生产者API：将消息发布到Kafka集群中的主题。
- 使用者API：使用来自Kafka集群中主题的消息。
- Streams API：使用主题中的消息，并将其转换为Kafka集群中的其他主题。 这些操作可以是过滤，联接，映射，分组等。
- 连接API：无需编码即可直接将Kafka群集连接到源系统或接收器系统。 该系统可以是文件，关系数据库，Elasticsearch等。
- 管理员API：管理和检查Kafka集群中的主题和代理。

Kafka的Python库

在Python世界中，已经实现了5个API中的3个，分别是Producer API，Consumer API和Admin API。 Python中还没有这样的Kafka Stream API，但是很好Faust。

本节中的测试是基于本地安装的1个Zookeeper和1个Kafka代理执行的。 这与性能调整无关，所以我主要使用该库提供的默认配置。

Kafka-Python

kafka-python的设计功能非常类似于官方的Java客户端，并带有大量pythonic接口。 最好与Kafka 0.9+版本一起使用。 第一版发布于2014年3月。正在积极

安装

pip install kafka-python

每个消息都是通过send()异步发送的。 调用时，它将记录添加到缓冲区并立即返回。 这使生产者可以以批处理方式将记录发送到Kafka经纪人以提高效率。地提高速度，但是我们还应该了解以下几点：

- 在异步模式下，不能保证排序。 您无法控制Kafka经纪人何时确认(确认)每封邮件。
- 为生产者提供成功回调和失败回调是一个好习惯。 例如，您可以在成功回调中编写信息日志消息，而在失败回调中编写异常日志消息。
- 由于无法保证顺序，因此在回调中收到异常之前，可能会发送额外的消息。

如果要避免这些问题，可以选择同步发送消息。 send()的返回是FutureRecordMetadata。 通过执行future.get(timeout = 60)，生产者将被阻止最多60秒钟确认消息为止。 缺点是速度，与异步模式相比，它相对较慢。

消费者

使用者实例是一个Python迭代器。 消费者类的核心是poll()方法。 它允许使用者继续从主题中提取消息。 它的输入参数timeout_ms之一默认为0，这意味着回所有在缓冲区中拉出并可用的记录。 您可以增加timeout_ms以返回更大的批次。

默认情况下，每个使用者都是一个无限的侦听器，因此它不会停止运行，直到程序中断。 但另一方面，您可以根据收到的消息停止使用者。 例如，您可以到某个偏移量时关闭使用者。

也可以将使用者分配给一个分区或来自多个主题的多个分区。

这是kafka-python库的测试结果。 每个消息的大小为100字节。 生产者的平均吞吐量为1.4MB / s。 使用者的平均吞吐量为2.8MB / s。

Confluent-kafka

Confluent-kafka是Python的高性能Kafka客户端，它利用高性能C客户端librdkafka。 从1.0版开始，这些作为PyPi上的OS X和Linux的独立二进制轮分发。0.8+版本。 第一版发布于2016年5月。正在积极维护中。

安装

对于OS X和Linux，软件包中包括librdkafka，需要单独安装。

pip install confluent-kafka

对于Windows用户，在我撰写本文时，confluent-kafka尚未在Windows上支持Python3.8二进制轮子。 您将遇到librdkafka的问题。 请查看他们的发行说明极开发中。 另一种解决方案是降级到Python3.7。

Confluent-kafka在速度方面具有令人难以置信的性能。 API的设计有点类似于kafka-python。 您可以通过将flush()放入循环中来使其同步。

消费者

confluent-kafka中的Consumer API需要更多代码。 您无需自己处理高级循环方法(例如，消耗())，而需要自己处理while循环。 我建议您创建自己的consu是一个Python生成器。 只要有一条消息被拉出并且在缓冲区中可用，它就会产生该消息。

这样，主要功能将变得干净，您可以自由控制消费者的行为。 例如，您可以在consumpt()中定义一个"会话窗口"。 如果在X秒钟内未提取任何消息，则使者，您可以添加标志infinite = True作为输入参数，以控制使用者是否应为无限侦听器。

这是confluent-kafka库的测试结果。 每个消息的大小为100字节。 生产者的平均吞吐量为21.97MBps。 消费者的平均吞吐量为16.8～28.7MB / s。

PyKafka

PyKafka是Python的程序员友好的Kafka客户端。 它包括Kafka生产者和使用者的Python实现，可以选择由基于librdkafka的C扩展支持。 它支持Kafka 0.82发布于2012年8月，但自2018年11月以来未进行过更新。

安装

pip install pykafka

该软件包不附带librdkafka，您需要在所有操作系统中分别安装。

pykafka具有KafkaClient接口，该接口涵盖了ProducerAPI和Consumer API。

消息可以异步和同步模式发送。 我发现pykafka会修改某些生产者配置(例如linger_ms和min_queued_messages)的默认值，这会对发送少量数据产生影响

您可以将其与Apache Kafka网站上的默认配置进行比较。

如果要获取每个消息的回调，请确保将min_queued_messages更改为1，否则如果数据集小于70000，则不会收到任何报告。

```
class pykafka.producer.Producer(cluster, topic, partitioner=<function random_partitioner>,
compression=0, max_retries=3, retry_backoff_ms=100, required_acks=1, ack_timeout_ms=10000,
max_queued_messages=100000, min_queued_messages=70000, linger_ms=5000, block_on_queue_full=True,
sync=False)
```

> pykafka-producer-config

消费者

您可以从KafkaClinet界面获取SimpleConsumer。 这类似于kafka-python，其中民意调查被包装在SimpleConsumer类中。

这是pykafka库的测试结果。 每个消息的大小为100字节。 生产者的平均吞吐量为2.1MB / s。 使用者的平均吞吐量为1.57MB / s。

结论

到目前为止，我已经解释了每个库的Producer API和Consumer API。 就Admin API而言，kafka-python和confluent-kafka确实提供了显式的Admin API。 在
题的单元测试中使用它，然后在执行下一个测试之前将其删除。 此外，如果您想使用Python构建Kafka监控仪表板，则Admin API可以帮助您检索集群和主

Confluent-kafka：

毫无疑问，Confluent-kafka在这三个库中表现最佳。 该API的设计经过精心设计，参数与原始Apache Kafka相同的名称和默认值。 您可以轻松地将其链接
个人而言，我喜欢自定义消费者行为的灵活性。 Confluent也正在积极开发和支持它。

缺点是Windows用户可能需要花费一些时间才能使其工作。 并且由于C扩展，调试可能很棘手。

kafka-python：

kafka-python是没有C扩展的纯Python库。 该API经过精心设计，对于初学者来说很容易使用。 这也是一个积极开发的项目。

python-kafka的缺点是它的速度。 如果您确实关心性能，建议您改用confluent-kafka。

pykafka：

与kafka-python和conflunet-kafka相比，pykafka的开发活动较少。 该版本的历史记录表明，自2018年11月以来尚未进行过更新。此外，pykafka具有不同的
用了不同的默认参数，这可能不是第一次。

责任编辑：华轩　　来源： 今日头条

Python　开发　Kafka