

优秀

qq_41534566

码龄3年

暂无认证

123

30万+

3万+

14万+

原创

周排名

总排名

访问

等级

2604

74

38

29

164

积分

粉丝

获赞

评论

收藏

私信

关注

搜博主文章

Q

- 热门文章
- 基于redis实现的扣减库存

19359
- 那些百万年薪的算法工程师，都是经历了哪些九死一生？

7468
- 大牛教你使用dlib中的深度残差网络(ResNet)实现实时人脸识别

5267
- 为何选择Dubbo它的优势是什么

4804
- Java面试中常见的Spring方面的问题（56道含答案）

3828

- 分类专栏
-  Java

55篇
-  架构

107篇
-  Java进阶

80篇
-  springMVC

7篇
-  分布式

9篇
-  微服务

5篇

- 最新评论
- 基于redis实现的扣减库存

zllllllllll: mysql在RR的隔离级别下是解决了mysql的幻读问题的，在快照读的情况 ...
- 基于redis实现的扣减库存

隶属感: repeatable read 的A会话修改值 为4 没commit 会导致另一个B会话 阻塞 A ...
- 《人工智能（第二版）》入门新人必读的...

m0_50492156: 希望为读者方便着想!
- 《人工智能（第二版）》入门新人必读的...

m0_50492156: 付款以后阅读不方便，昨天看人工智能史蒂芬卢奇几章后，今日找? ...
- 大牛教你使用dlib中的深度残差网络(Res...

qq_43146609: 好的

- 最新文章
- 《人工智能（第二版）》入门新人必读的一本神书（文末送书）
- 干货分享！价值62的代码到底长什么样子？
- 那些百万年薪的算法工程师，都是经历了哪些九死一生？
- 2019年 10篇

2018年 152篇

2017年 2篇





Expose Yo

Competit

in One Clic

Learn More

大牛实战归纳——Kafka架构原理

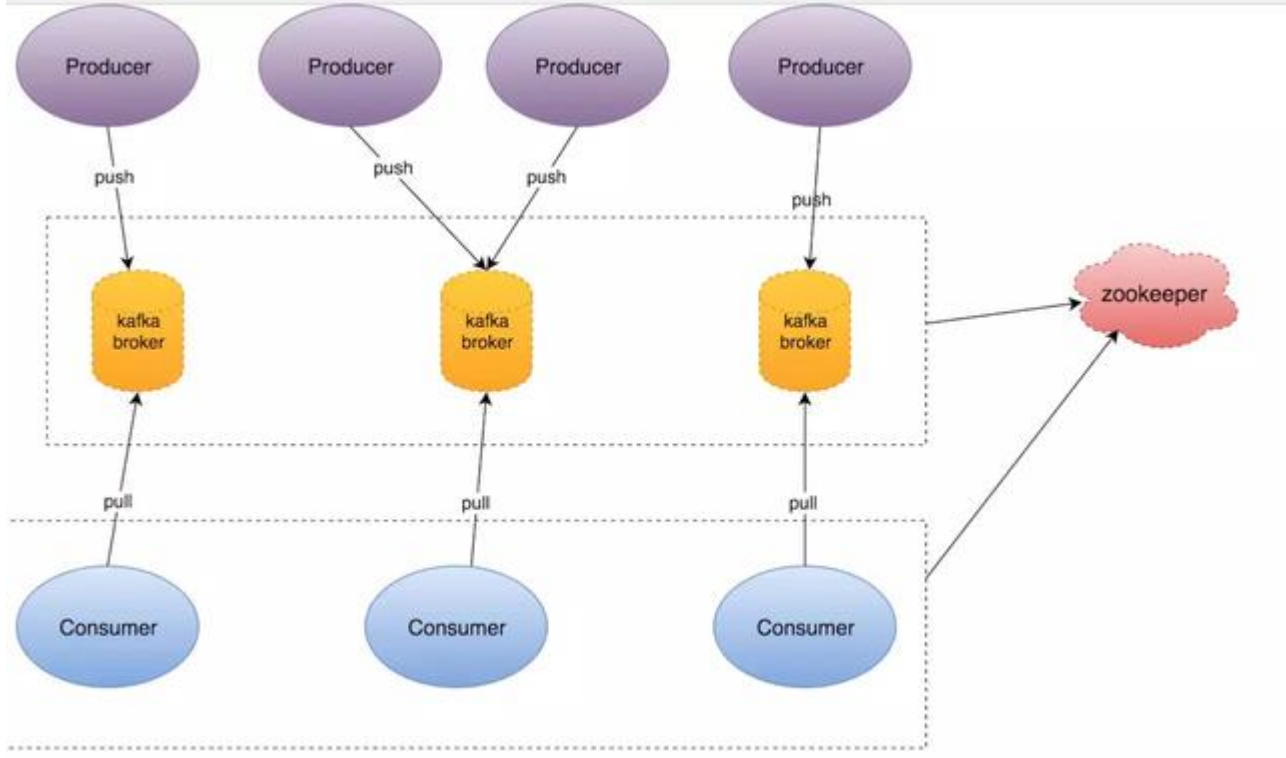
原创 qq_41534566 2018-07-25 21:23:46 1889 收藏 5

分类专栏: [Java进阶](#) [架构](#)

对于kafka的架构原理我们先提出几个问题？

- 1.Kafka的topic和分区内部是如何存储的，有什么特点？
- 2.与传统的消息系统相比,Kafka的消费模型有什么优点？
- 3.Kafka如何实现分布式的数据存储与数据读取？

一、Kafka架构图



1.kafka名词解释

在一套kafka架构中有多个Producer，多个Broker,多个Consumer，每个Producer可以对应多个Topic，每个Consumer只能对应一个ConsumerGroup。

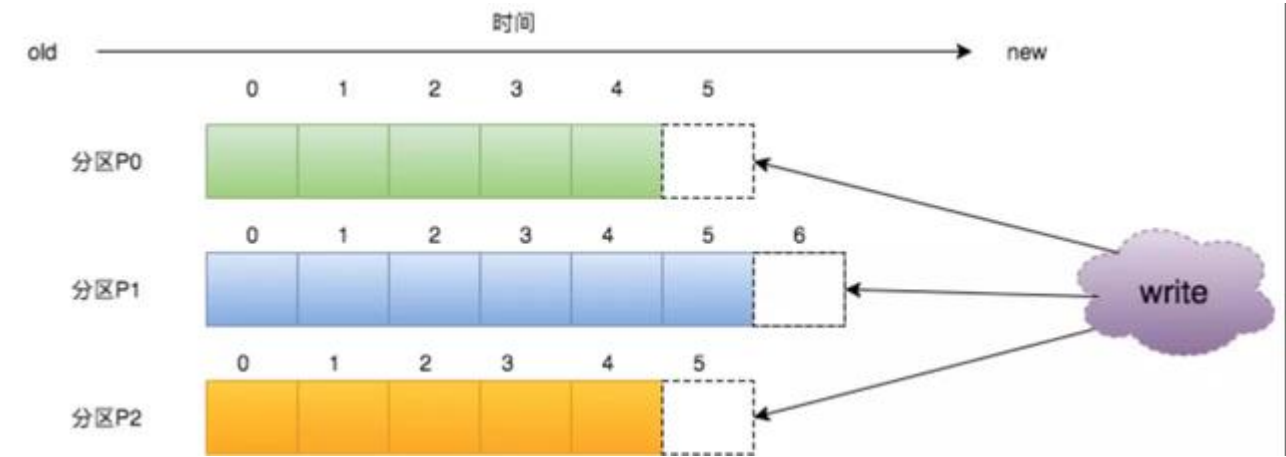
整个Kafka架构对应一个ZK集群，通过ZK管理集群配置，选举Leader，以及在consumer group发生变化时进行rebalance。

- 名称
- 解释
- Broker
- 消息中间件处理节点，一个Kafka节点就是一个broker，一个或者多个Broker可以组成一个Kafka集群
- Topic
- 主题，Kafka根据topic对消息进行归类，发布到Kafka集群的每条消息都需要指定一个topic
- Producer
- 消息生产者，向Broker发送消息的客户端
- Consumer
- 消息消费者，从Broken读取消息的客户端
- ConsumerGroup
- 每个Consumer属于一个特定的Consumer Group，一条消息可以发送到多个不同的Consumer Group，但是一个Consumer Group中只能有一个Consumer能够消费该消息
- Partition
- 物理上的概念，一个topic可以分为多个partition，每个partition内部是有序的

2.Topic和Partition

在Kafka中的每一条消息都有一个topic。一般来说在我们应用中产生不同类型的数据，都可以设置不同的主题。一个主题一般会有多个消息的订阅者，当生产者发布消息到某个主题时，订阅了这个主题的消费者都可以接收到生产者写入的新消息。

kafka为每个主题维护了分布式的分区(partition)日志文件，每个partition在kafka存储层面是append log。任何发布到此partition的消息都会被追加到log文件的尾部，在分区中的每条消息都会按照时间顺序分配到一个单调递增的顺序编号，也就是我们的offset,offset是一个long型的数字，我们通过这个offset可以确定一条在该partition下的唯一消息。在partition下面是保证了有序性，但是在topic下面没有保证有序性。



在上图中在我们的生产者会决定发送到哪个Partition。

- 1.如果没有Key值则进行轮询发送。
- 2.如果有Key值，对Key值进行Hash，然后对分区数量取余，保证了同一个Key值的会被路由到同一个分区，如果想队列的强顺序一致性，可以让所有的消息都设置为同一个Key。

3.消费模型

消息由生产者发送到kafka集群后，会被消费者消费。一般来说我们的消费模型有两种:推送模型(push)和拉取模型(pull)

基于推送模型的消息系统，由消息代理记录消费状态。消息代理将消息推送到消费者后，标记这条消息为已经被消费，但是这种方式无法很好地保证消费的处理语义。比如当我们已经把消息发送给消费者之后，由于消费进程挂掉或者由于网络原因没有收到这条消息，如果我们在消费代理将其标记为已消费，这个消息就永久丢失了。如果我们利用生产者收到消息后回复这种方法，消息代理需要记录消费状态，这种不可取。如果采用push，消息消费的速率就完全由消费代理控制，一旦消费者发生阻塞，就会出现问題。

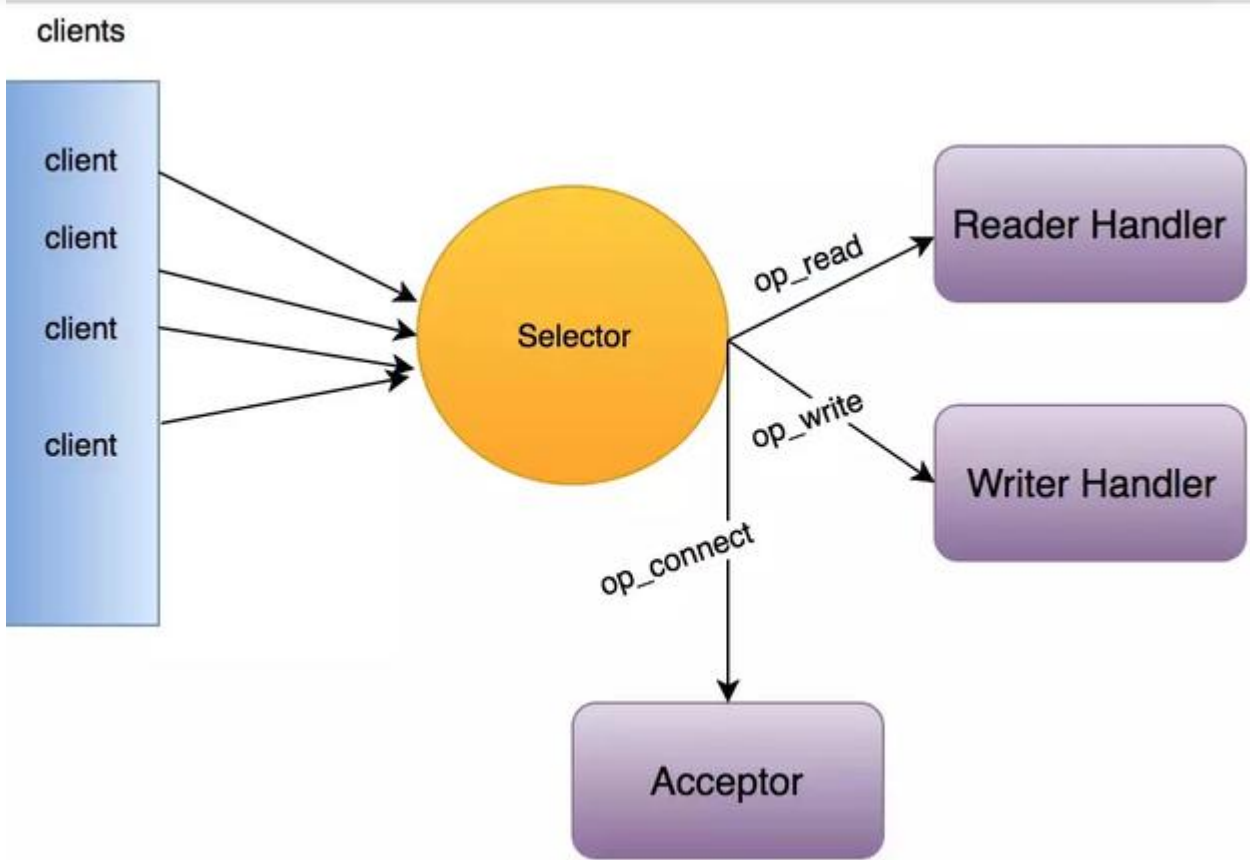
Kafka采取拉取模型(poll)，由自己控制消费速度，以及消费的进度，消费者可以按照任意的偏移量进行消费。比如消费者可以消费已经消费过的消息进行重新处理，或者消费最近的消息等等。

4.网络模型

4.1 KafkaClient --单线程Selector



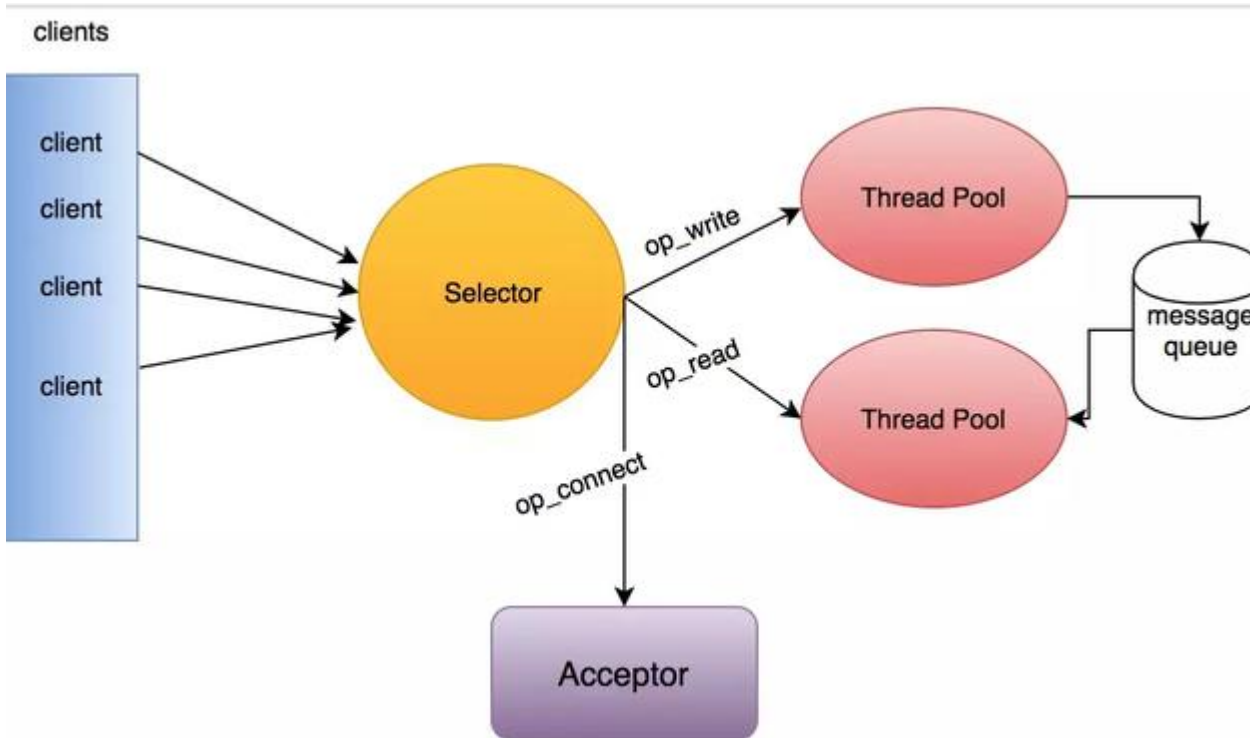
举报



单线程模式适用于并发链接数小，逻辑简单，数据量小。

在kafka中， consumer和producer都是使用的上面的单线程模式。这种模式不适合kafka的服务端，在服务端中请求处理过程比较复杂，会造成线程阻塞，一旦出现后续请求就会无法处理，会造成大量请求超时，引起雪崩。而在服务器中应该充分利用多线程来处理执行逻辑。

4.2 Kafka--server -- 多线程Selector



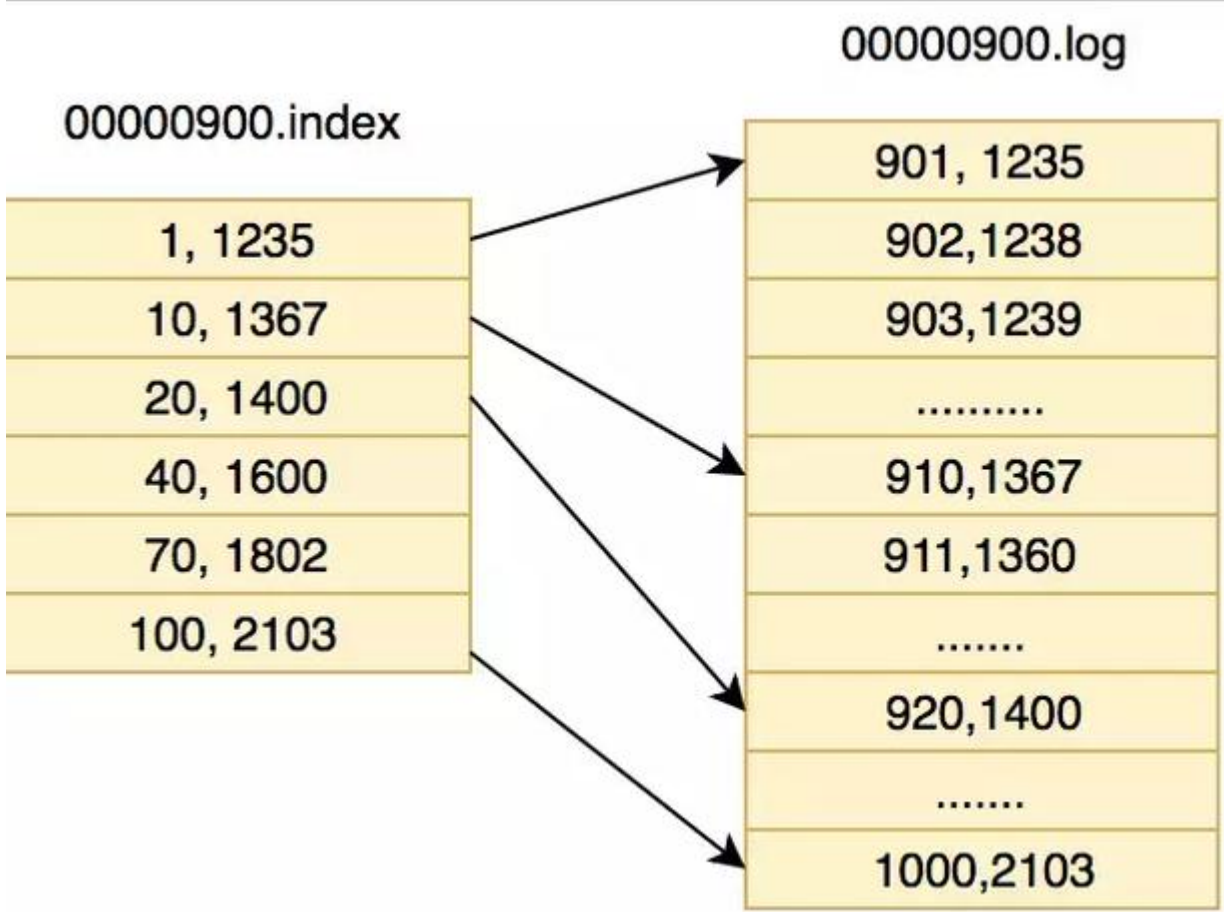
在kafka服务端采用的是多线程的Selector模型，Acceptor运行在一个单独的线程中，对于读取操作的线程池中的线程都会在selector注册read事件，负责服务端读取请求的逻辑。成功读取后，将请求放入message queue共享队列中。然后在写线程池中，取出这个请求，对其进行逻辑处理，即使某个请求线程阻塞了，还有后续的县城从消息队列中获取请求并进行处理，在写线程中处理完逻辑处理，由于注册了OP_WRITE事件，所以还需要对其发送响应。

5.高可靠分布式存储模型

在Kafka中保证高可靠模型的依靠的是副本机制，有了副本机制之后，就算机器宕机也不会发生数据丢失。

5.1高性能的日志存储

kafka一个topic下面的所有消息都是以partition的方式分布式的存储在多个节点上。同时在kafka的机器上，每个Partition其实都会对应一个日志目录，在目录下面会对应多个日志分段(LogSegment)。LogSegment文件由两部分组成，分别为“.index”文件和“.log”文件，分别表示为segment索引文件和数据文件。这两个文件的命令规则为： partition全局的第一个segment从0开始，后续每个segment文件名为上一个segment文件最后一条消息的offset值，数值大小为64位，20位数字字符长度，没有数字用0填充，如下，假设有1000条消息，每个LogSegment大小为100，下面展现了900-1000的索引和Log：



由于kafka消息数据太大，如果全部建立索引，即占了空间又增加了耗时，所以kafka选择了稀疏索引的方式，这样的话索引可以直接进入内存，加快偏查询速度。

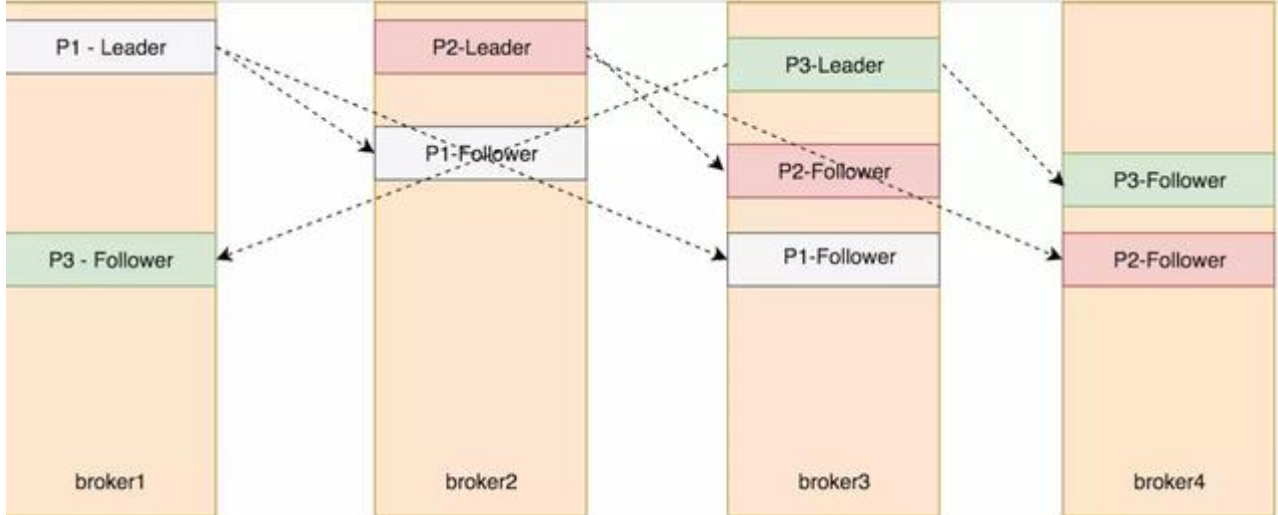
简单介绍一下如何读取数据，如果我们要读取第911条数据首先第一步，找到他是属于哪一段的，根据二分法查找到他属于的文件，找到0000900.index和00000900.log之后，然后去index中去查找 (911-900) =11这个索引或者小于11最近的索引,在这里通过二分法我们找到了索引是[10, 1367]然后我们通过这条索引的物理位置1367，开始往后找，直到找到911条数据。

上面讲的是如果要找某个offset的流程，但是我们大多数时候并不需要查找某个offset,只需要按照顺序读即可，而在顺序读中，操作系统会对内存和磁盘之间添加page cahe，也就是我们平常见到的预读操作，所以我们的顺序读操作时速度很快。但是kafka有个问题，如果分区过多，那么日志分段也会很多，写的时候由于是批量写，其实就会变成随机写了，随机I/O这个时候对性能影响很大。所以一般来说Kafka不能有太多的partition。针对这一点，RocketMQ把所有的日志都写在一个文件里面，就能变成顺序写，通过一定优化，读也能接近于顺序读。

可以思考一下:1.为什么需要分区，也就是说主题只有一个分区，难道不行吗？ 2.日志为什么需要分段

5.2副本机制

Kafka的副本机制是多个服务端节点对其他节点的主题分区的日志进行复制。当集群中的某个节点出现故障，访问故障节点的请求会被转移到其他正常节点(这一过程通常叫Reblance),kafka每个主题的每个分区都有一个主副本以及0个或者多个副本，副本保持和主副本的数据同步，当主副本出故障时就会被替代。



在Kafka中并不是所有的副本都能被拿来替代主副本，所以在kafka的leader节点中维护着一个ISR(In sync Replicas)集合，翻译过来也叫正在同步中集合，在这个集合中的需要满足两个条件:

- 节点必须和ZK保持连接
- 在同步的过程中这个副本不能落后主副本太多

另外还有个AR(Assigned Replicas)用来标识副本的全集,OSR用来表示由于落后被剔除的副本集合，所以公式如下:ISR = leader + 没有落后太多的副本; AR = OSR+ ISR;

这里先要说下两个名词:HW(高水位)是consumer能够看到的此partition的位置，LEO是每个partition的log最后一条Message的位置。HW能保证leader所在的broker失效，该消息仍然可以从新选举的leader中获取，不会造成消息丢失。

当producer向leader发送数据时，可以通过request.required.acks参数来设置数据可靠性的级别：

- 1（默认）：这意味着producer在ISR中的leader已成功收到的数据并得到确认后发送下一条message。如果leader宕机了，则会丢失数据。
- 0：这意味着producer无需等待来自broker的确认而继续发送下一批消息。这种情况下数据传输效率最高，但是数据可靠性确是最低的。
- -1：producer需要等待ISR中的所有follower都确认接收到数据后才算一次发送完成，可靠性最高。但是这样也不能保证数据不丢失，比如当ISR中只有leader时(其他节点都和zk断开连接，或者都没追上)，这样就变成了acks=1的情况。

扫码关注不迷路


一个专注分享架构心得的公众号





此图片来自微信公众平台
未经允许不可引用

Kafka原理篇 huahua Xiaoshao 的博客 250
1 Kafka简介 Kafka是分布式发布订阅消息系统。他最初由Linkedin公司开发，之后成为Apache项目的一部分。Kafka是...

kafka原理及常见问题总结 hw120219 的博客 969
broker：（Kafka的节点，也就是服务器）1.接受来自生产者的消息，为消息设置偏移量，并提交消息到磁盘保存。2.为消费...

 优质评论可以帮助作者获得更高权重

 抢沙发

 评论

相关推荐

Kafka 设计与原理详解_Heaven Wang 的专栏_kafka原理 3-24
批量发送:Kafka支持以消息集合为单位进行批量发送,以提高push效率。 push-and-pull: Kafka中的Producer和consumer采...

深入剖析kafka架构内部原理_数据工匠记_kafka架构和原理 3-31
这里先从Kafka文件存储机制入手,从最底层了解Kafka的存储细节,进而对其的存储有个微观的认知。之后通过Kafka复制原...

kafka原理及Docker环境部署 weixin_34260991 的博客 136
技术原理 Kafka是由Apache软件基金会开发的一个开源流处理平台，由Scala和Java编写。Kafka为处理实时数据提供一个...

一、kafka简介 jeny Zhang 的专栏 1658
转自：http://news.watchstor.com/news-161695.htm 1 概述 Kafka起初是由LinkedIn公司开发的一个分布式的消息系统，后...

Kafka底层原理架构_qichangjian的博客_kafka底层原理 3-1
在kafka文件存储找中,同一个topic下有多个partition,每一个partition为一个目录,partition命名规则为:topic 名称+有序序号,第...

Kafka史上最详细原理总结 ---用于自己看的_微步的博客... 3-24
- 高吞吐量、低延迟_kafka每秒可以处理几十万条消息,它的延迟最低只有几毫秒,每个topic可以分多个partition, consumer gr...

Apache kafka原理与特性(0.8V) 深入一点,你会更加快乐 54
前言: kafka是一个轻量级的分布式的/具备replication能力的日志采集组件,通常被集成到应用系统中,收集"用户行为日志"等...

kafka之二：Kafka 设计与原理详解 weixin_34279246 的博客 1738
一、Kafka简介 本文综合了我之前写的kafka相关文章，可作为一个全面了解学习kafka的培训学习资料。 转载请注明出处：...

kafka实现原理_samson_kafka实现原理 3-31
kafka实现原理 一_kafka定义 Kafka是一款分布式消息发布和订阅系统,它的特点是高性能、高吞吐量。最早设计的目的是作...

Kafka原理你真的知道吗?_假的鱼的博客_kafka原理 3-31
Kafka原理你真的知道吗? 原文链接:https://www.jianshu.com/p/a7f7d9973fea 1.概述 Apache Kafka最早是由LinkedIn开源出...

kafka工作原理介绍 Saint 7万+
两张图读懂kafka应用： Kafka 中的术语 broker：中间的kafka cluster，存储消息，是由多个server组成的集群。 topic：kafk...

大白话 kafka 架构原理 Felix 590
大数据时代来临，如果你还不知道Kafka那就真的out了！据统计，有三分之一的世界财富500强企业正在使用Kafka，包括...

Kafka工作原理_weixin_30701521的博客_kafka原理 3-6
Kafka工作原理 Kafka 中的术语设计: 1、Broker 中间的kafka cluster,存储消息,是由多个server组成的集群。 2、topic与消息 ...

kafka——原理总结_MJ的博客 3-13
partition默认都在/tmp/kafka-logs/下面,每个partition会分为多个大小相等segment(逻辑概念),每个segment由索引文件00000...

Kafka 原理简介 王小明的专栏 322
Kafka 原理简介 Kafka 是一种高吞吐的分布式发布订阅的消息系统，可以处理消费者规模的网站中的动作流数据，具有高性...

kafka系列之原理简介 Nicky's blog 7885
kafka原理简介 Producer Broker Consumer Zookeeper Kafka优点 单机版安装教程 1关闭SELINUX 2配置防火墙 3安装JDK ...

kafka原理剖析 程研板的博客 251
一.Message Queue好处 解耦：允许你独立的扩展或修改两边的处理过程，只要确保它们遵守同样的接口约束。 峰值处理...

Kafka原理解析 爱宝贝丶 的博客 127
【推荐】2019 Java 开发者跳槽指南.pdf(吐血整理) >>> ...

再过半小时，你就能明白kafka的工作原理了 weixin_33711641 的博客 640
本文在个人技术博客不同步发布，详情可猛戳 亦可扫描屏幕右侧二维码关注个人公众号，公众号内有个人联系方式，等你...

Kafka史上最详细原理总结 我是Alvin家鸡鸭鱼的小米米 4万+
Kafka Kafka是最初由Linkedin公司开发，是一个分布式、支持分区的（partition）、多副本的（replica），基于zookeeper...

Kafka学习之：Kafka背景及架构介绍 不积跬步，无以至千里；不积小流，无以成江海！ 3044
转载自：http://www.infoq.com/cn/articles/kafka-analysis-part-1 Kafka是由LinkedIn开发的一个分布式的消息系统，使用Scal...

kafka原理和实践（一）原理：10分钟入门 博客园搬来测试账号 6454
目录一、kafka拓扑图二、名词解释三、生产模型四、消费模型 正文系列目录 kafka原理和实践（一）原理：10分钟入门 kaf...

kafka原理及面试套路 weixin_35720385 的博客 3752
kafka原理及面试套路一、 面试：1、列举kafka的使用场景2、Kafka消息是采用Pull模式，还是Push模式？ 3、Kafka 与传...

kafka原理解析 yuyuyuxiaolei 的博客 3177
Apache的Kafka™是一个分布式流平台(a distributed streaming platform)，这到底意味着什么？ 我们认为，一个流处理平台...

©2020 CSDN 皮肤主题: 大白 设计师:CSDN官方博客 返回首页

关于我们 招贤纳士 广告服务 开发助手 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 网络110报警服务 中国互联网举报中心 家长监护 Chrome商店下载 ©1999-2021北京创新乐知网络技术有限公司 版权与免责声明 版权申诉



举报