

 beautiful_huang
码龄1年 暂无认证

137
原创

12万+
周排名

5万+
总排名

5万+
访问

 等级

2033
积分

70
粉丝

171
获赞

19
评论

173
收藏

私信

关注

搜博主文章



热门文章

【Java】牛客网华为机试108题汇总 

3797

【Scala】groupBy、grouped 方法详解 

3029

【Kafka】Kafka消费者组subscribe和assign的正确使用 

2620

【Saprk】Spark DataFrame 列的类型转换 

2560

Oracle VM VirtualBox 安装CentOS7操作系统 

2170

分类专栏

 机器学习

1篇

 Java

14篇

 Spark

31篇

 maxcompute

3篇

 BI

1篇

 Mysql

6篇



最新评论

Oracle VM VirtualBox 安装CentOS7操作...
Ein playboy: 大赞!! 感谢博主, 装好了

Java 处理Get/Post请求参数中文乱码问题
雾进: 都是博主辛苦创作, 我来支持一下, 奥利给!

idea 常用快捷键
ctotalk: 学习

Linux 用户/用户组权限设置与变更
ctotalk: 好好学习, 天天向上.

【Spark】Spark Dataframe实现列转行
beautiful_huang 回复 檐前潜心学种瓜: 不会的

最新文章

【Java】Mybatis Druid连接池配置详细

Java 处理Get/Post请求参数中文乱码问题



【MaxCompute】MaxCompute 处理复杂json、json数组

2021年 2篇

2020年 78篇

2019年 57篇

【Kafka】Kafka自定义分区器

原创 beautiful_huang 2020-02-25 15:35:54  468  收藏 2 版权

分类专栏: [kafka](#) 文章标签: [kafka](#)

1. 默认的分区策略

(1) 如果键值为 null, 并且使用了默认的分区器, 那么记录将被随机地发送到主题内各个可用的分区上。分区器使用轮询 (Round Robin) 算法将消息均衡地分布到各个分区上。

(2) 如果键不为空, 并且使用了默认的分区器, 那么 Kafka 会对键取 hash 值然后根据散列值把消息映射到特定的分区上。这里的关键之处在于, 同一个键总是被映射到同一个分区上, 所以在进行映射时, 我们会使用主题所有的分区, 而不仅仅是可用的分区。这也意味着, 如果写入数据的分区是不可用的, 那么就会发生错误。但这种情况很少发生。

2. 自定义分区器

为了满足业务需求, 你可能需要自定义分区器, 例如, 通话记录中, 给客服打电话的记录要存到一个分区中, 其余的记录均分的分布到剩余的分区中。我们就这个案例来进行演示。

(1) 自定义分区器

```
1 package com.bonc.rdpe.kafka110.partitioner;
2
3 import java.util.List;
4 import java.util.Map;
5
6 import org.apache.kafka.clients.producer.Partitioner;
7 import org.apache.kafka.common.Cluster;
8 import org.apache.kafka.common.PartitionInfo;
9
10 /**
11  * @Author: Stephen
12  * @Date: 2020/2/25 15:18
13  * @Content: 自定义分区器
14  */
15 public class PhonenumPartitioner implements Partitioner{
16
17     @Override
18     public void configure(Map<String, ?> configs) {
19         // TODO nothing
20     }
21
22     /**
23      * 自定义kafka分区主要解决用户分区数据倾斜问题 提高并发效率 (假设 3 分区)
24      * @param topic 消息队列名
25      * @param key 用户传入key
26      * @param keyBytes key字节数组
27      * @param value 用户传入value
28      * @param valueBytes value字节数据
29      * @param cluster 当前kafka节点数
30      * @return 如果3个节点数 返回 0 1 2 如果5个 返回 0 1 2 3 4 5
31      */
32     @Override
33     public int partition(String topic, Object key, byte[] keyBytes, Object value, b
34         // 得到 topic 的 partitions 信息
35         List<PartitionInfo> partitions = cluster.partitionsForTopic(topic);
36         int numPartitions = partitions.size();
37         // 模拟某客服
38         if(key.toString().equals("10000") || key.toString().equals("11111")) {
39             // 放到最后一个分区中
40             return numPartitions - 1;
41         }
42         String phoneNum = key.toString();
43         return phoneNum.substring(0, 3).hashCode() % (numPartitions - 1);
44     }
45
46     @Override
47     public void close() {
48         // TODO nothing
49     }
50 }
```

(2) 使用自定义分区器

```
1 package com.bonc.rdpe.kafka110.producer;
2
3 import java.util.Properties;
4 import java.util.Random;
5
6 import org.apache.kafka.clients.producer.KafkaProducer;
7 import org.apache.kafka.clients.producer.Producer;
8 import org.apache.kafka.clients.producer.ProducerRecord;
9 import org.apache.kafka.clients.producer.RecordMetadata;
10
11 /**
12  * @Title PartitionerProducer.java
13  * @Description 测试自定义分区器
14  * @Author YangYunhe
15  * @Date 2018-06-25 15:10:04
16  */
17 public class PartitionerProducer {
18
19     private static final String[] PHONE_NUMS = new String[]{
20         "10000", "10000", "11111", "137000000003", "137000000004",
21         "10000", "155000000006", "11111", "155000000008",
22         "176000000009", "10000", "176000000011"
23     };
24
25     public static void main(String[] args) throws Exception {
26
27         Properties props = new Properties();
28         props.put("bootstrap.servers", "192.168.42.89:9092,192.168.42.89:9093,192.1
29         // 设置分区器
30         props.put("partitioner.class", "com.bonc.rdpe.kafka110.partitioner.Phonenum
31         props.put("key.serializer", "org.apache.kafka.common.serialization.StringSe
32         props.put("value.serializer", "org.apache.kafka.common.serialization.String
33
34         Producer<String, String> producer = new KafkaProducer<>(props);
35
36         int count = 0;
37         int length = PHONE_NUMS.length;
38
39         while(count < 10) {
40             Random rand = new Random();
41             String phoneNum = PHONE_NUMS[rand.nextInt(length)];
42             ProducerRecord<String, String> record = new ProducerRecord<>("dev3-yang
43             RecordMetadata metadata = producer.send(record).get();
44             String result = "phonenum [" + record.value() + "] has been sent to par
45             System.out.println(result);
46             Thread.sleep(500);
47             count++;
48         }
49         producer.close();
50     }
51 }
```

(3) 测试结果

```
1 phonenum [11111] has been sent to partition 2
2 phonenum [11111] has been sent to partition 2
-
```



举报

(1条消息) 【Kafka】Kafka自定义分区器_beaufu...CSDN博客_kafka自定义分区

5

phonenum [13700000003] has been sent to partition 1

6

phonenum [10000] has been sent to partition 2

7

phonenum [10000] has been sent to partition 2

8

phonenum [15500000008] has been sent to partition 1

9

phonenum [10000] has been sent to partition 2


10

phonenum [17600000009] has been sent to partition 0


有程序员接外包吗?靠谱


Java kafka如何实现自定义分区类和拦截器08-19


主要介绍了Java kafka如何实现自定义分区类和拦截器,文中通过示例代码介绍的非常详细,对大家的学习或者工作具有一定...

- 

优质评论可以帮助作者获得更高权重



评论
- 

丶以梦ya: The configuration 'partitioner.class' was supplied but isn't a known config, 求解, 这个配置无法识别是为什么, 分区器的包路径名称没问题, 类重写的内容也与您写的一致 10 月前 回复 ...
- 

多吃核桃会补脑: 那么如何同时使用多个自定义分区器呢? 比如我有5个分区逻辑, 我想写到5个不同的分区器里面, 又想他们同时起作用 1 年前 回复 ...

相关推荐

- kafka自定义分区API_依旧的博客4-21

props.put("key.serializer","org.apache.kafka.common.serialization.StringSerializer"); props.put("value.serializer","org.apac...
- Kafka分区策略浅谈08-14

Kafka分区策略浅谈, Kafka分区策略浅谈, Kafka分区策略浅谈。
- kafka 自定义分区器HH_KELE的专栏 110

kafka 自定义分区器 package cn.xiaojf.kafka.producer; import org.apache.kafka.clients.producer.Partitioner; import org.apa...
- Kafka-生产者-分区器详解Just Do IT 2439

注: 本文源码解析基于Kafka2.1.0版本 我们知道, Kafka中的每个Topic一般会分配N个Partition, 那么生产者(Producer)在将...
- Kafka分区策略及自定义李双喆 9797

默认分区策略默认分区策略是: 取正 (bytearray生成32位hash值) %numpartitions 这个公式的结果是得到0-(numpartitions...
- kafka中生产者自定义分区器无业游民的专栏 1134

kafka中生成者自定义分区器, 以及分区的分发策略, 先来直接看看如何实现, 直接附上代码案例 首先先实现 Partitioner接...
- kafka 自定义分区实例liuxiangke0210的专栏 4149

第一步: 使用 ./kafka-topics.sh 命令创建topic及partitions 分区数 bin/kafka-topics.sh --create --zookeeper 192.168.31.130:2...
- kafka消费者分区分配策略及自定义分配策略无业游民的专栏 3042

kafka消费者如何分配分区以及分配分区策略和源码解释 我们知道kafka的主题中数据数据是按照分区的概念来的, 一个主题...
- Kafka分区分配策略 (3) ——自定义分区分配策略朱小厮的博客 6139

接上文: 1. 【Kafka分区分配策略 (1) ——RangeAssignor】 2. 【Kafka分区分配策略 (2) ——RoundRobinAssignor和Sti...
- Kafka中自定义分区Java APIwhile false 679

自定义分区API只是在Producer API的基础上加入了一些内容 这里直接展示出代码 ProducerPartition.java package cn.jsjh; i...
- [Kafka基础]-自定义Kafka分区器余额不足 4614

在调用Kafka的Producer API时, 如果没有指定分区器, 那么数据将会根据默认分区器的算法均分到各个分区。然而实际的...
- kafka的自定义分区洛云凡 97

首先定义一个自定义分区类 package Partition; import org.apache.kafka.clients.producer.Partitioner; import org.apache.kafk...
- kafka-自定义分区器 (partitioner) _06yangyaming's blog 827

上一篇文章讲了kafka的默认的分区器 (DefaultPartitioner) 源码, 这里我们写一个自定义分区器的小例子 生产者代码如下...
- 自定义kafka分区器PartitionerQYHuiQ 801

研究分区器先从ProducerRecord看起, 因为分区是在每条record的基础上实现的。 ProducerRecord的字段: 在构造函数数中...
- Kafka分区分配策略 (4) ——分配的实施朱小厮的博客 2015

接上文: 1. 【Kafka分区分配策略 (1) ——RangeAssignor】 2. 【Kafka分区分配策略 (2) ——RoundRobinAssignor和Sti...
- kafka自定义分区weixin_43947279的博客 10

默认的分区策略 1.如果键值为 null, 并且使用了默认的分区器, 那么记录将被随机地发送到主题内各个可用的分区上。分区...
- Kafka 自定义分区的生产者u012430775的专栏 57

1, 实现 接口 Patitioner public class KafkaPartitioner implements Partitioner { @Override public int partition(String s, Objec...
- 【大数据开发】Kafka——Kafka必知必会②白色风车 48

一、封装工具类 下载consumer.properties、producer.properties文件到resources中 二、偏移量offset 三、自定义分区器
- kafka中的自定义分区器使用详解weixin_43230682的博客 206

综述 在Kafka中, topic是逻辑上的概念, 而partition是物理上的概念。不用担心, 这些对用户来说是透明的。生产者 (prod...

©2020 CSDN 皮肤主题: 深蓝海洋 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 出版物许可证 营业执照



举报