

Kafka 學習筆記

一、概述

1、定义

1) 统定义

分布式 发布消息列

发布消息：分多种类型 生产者根据需求 选择性

2) 最新定义

流平台（存、算）

2、消息列用场景

1) 存消峰

2) 解耦

3) 异步通信

3、两种模式

1) 点对点

(1) 一个生产者 一个消费者 一个topic 会删除数据 不多

2) 发布消息

(1) 多个生产者 消费者多个 而且相互独立 多个topic 不会删除数据

4、架构

1) 生产者

100T数据

2) broker

(1) broker 服务器 hadoop102 103 104

(2) topic 主题 数据分类

(3) 分区

(4) 可靠性 副本

(5) leader follower

(6) 生产者和消费者 只读leader操作

3) 消费者

(1) 消费者和消费者相互独立

(2) 消费者 (某个分区 只能由一个消费者消费)

4) zookeeper

(1) broker.ids 0 1 2

(2) leader

二、入门

1、安装

1) broker.id 必须全局唯一

2) broker.id、log.dirs zk/kafka

3) 启动停止 先停止kafka 再停zk

4) 脚本

```
#!/bin/bash
```

```
case $1 in
```

```
"start")
```

```
for i in hadoop102 hadoop103 hadoop104
```

```
do
```

```
ssh $i "mkdir"
```

```
done
```

```
;;
```

```
"stop")
```

```
;;
```

```
esac
```

2、常用命令行

1) 主题 kafka-topic.sh

(1) --bootstrap-server hadoop102:9092,hadoop103:9092

(2) --topic first

(3) --create

(4) --delete

(5) --alter

(6) --list

(7) --describe

(8) --partitions

(9) --replication-factor

2) 生产者 kafka-console-producer.sh

(1) --bootstrap-server hadoop102:9092,hadoop103:9092

(2) --topic first

3) 消费者 kafka-console-consumer.sh

(1) --bootstrap-server hadoop102:9092,hadoop103:9092

(2) --topic first

三、生产者

1、原理

2、异步发送API

0) 配置

(1) 连接 bootstrap-server

(2) key value序列化

1) 创建生产者

```
KafkaProducer<String,String>()
```

2) 发送数据

```
send() send(,new Callback)
```

3) 源码

3、同步发送

。。。

send() send(,new Callback).get()

。。。

4、分片

1) 好处

存片

计算

2) 默认分片

(1) 指定分片 按分片走

(2) key key的hashCode值%分片

(3) 有指定key 有指定分片 粘性

第一随机

3) 自定义分片

定义类 实现partitioner接口

5、吞吐量提高

1) 批次大小 16k 32k

2) linger.ms 0 => 5-100ms

3) 内存

4) 内存大小 32m => 64m

6、可靠性

acks

0 丢失数据

1 也可能丢失 输出普通日志

-1 完全可靠 + 副本大于等于2 isr >=2 => 数据重复

7、数据重复

1) 幂等性

<pid, 分片号, 序列号>

默认打印

2) 事务

底层基于幂等性

(1) 初始化

(2) 启动

(3) 消费者offset

(4) 提交

(5) 中止

8、数据有序

分片有序 (有条件)

多分片有序怎么?

9、顺序

1) inflight =1

2) 有幂等性 inflight =1

3) 有幂等性

四、broker

1、zk存储了哪些信息

(1) broker.ids

(2) leader

(3) 辅助 controller

2、工作流程

3、服役

1) 准备一台干服务器 hadoop100

2) 哪个主操作

3) 形成计划

4) 执行计划

5) 计划

4、退役

1) 要退役的点不存数据

2) 退出点

5、副本

1) 副本的好处 提高可靠性

2) 生环境中通常2个 默认1个

3) 有leader follower leader

4) isr

5) controller isr[0 2 3] 存活 ar[0 1 2 3]

6) Leader 挂了

7) follower 挂了

8) 副本分配 默认

9) 手动副本分配 制定计划 执行计划 计划

10) leader partition的均衡 10%

11) 手动增加副本因子

6、存机制

broker topic partition log segment 1g 稀疏索引 4kb

7、删除数据

默认7天 3天 7小时

两种 删除

删除:

两种:

8、高效

1) 集群 采用分片

2) 稀疏索引

3) 顺序

4) 零拷贝 和页存

五、消费者

1、总体流程

2、消费者

3、按照主消

0) 配置信息

接

反序列化

id

1) 建消费者

2) 主

3) 消息据

4、按照分

5、消费者案例

id

6、分分配策略 再平衡

7个分 3个消费者

range

0 1 2 x

3 4

5 6

roundrobin

轮询

0 3 6

1 4

2 5

粘性 2 2 3

0 3 4

2 6

1 5

7、offset

1) 默认存在系统主

2) 自动提交 5s 默认

3) 手动提交 同步 异步

4) 指定offset消 seek ()

5) 按照间消

6) 漏消 重复消

8、事务

生端 =》 集群

集群 =》 消费者

消费者 =》 框架

9、据

1、增加分 增加消费者个

2、生 =》 集群 4个参

3、消端 两个参 50m 500条

六、生优 硬件

1、100万日志 * 人每天生日志100条 = 1亿条 (中型公司)

处理日志速度 1亿条 / (24 * 3600s) = 1150条/s

1条日志 (0.5k - 2k 1k)

1150条 * 1k/s = 1m/s

高峰值 (中午小高峰 8-12) : 1m/s * 20倍 = 20m/s -40m/s

2、多少台服务器

服务器台 = 2 * (生产者峰值速率 * 副本 / 100) + 1

= 2 * (20m/s * 2 / 100) + 1

= 3 台

3、磁

kafka 按照顺序 机械硬和固硬 顺序速度差不多

1亿条 * 1k = 100g

100g * 2个副本 * 3天 / 0.7 = 1t

建三台服务器总的磁大小 大于1t

4、存

kafka 存 = 堆存 (kafka 部配置) + 页存 (服务器存)

1) 堆存 10-15g

2) 页存 segment (1g) (分Leader (10) * 1g * 25%) / 3 = 1g

一台服务器 10g + 1g

5、CPU

32cpu

6、网

:

1、batch.size=16384 linger.ms=0 9.76 MB/sec

2、batch.size=32768 linger.ms=0 9.76 MB/sec

3、batch.size=4096 linger.ms=0 3.81 MB/sec

4、batch.size=4096 linger.ms=50 3.83 MB/sec

5、batch.size=4096 linger.ms=50 compression.type=snappy 3.77 MB/sec

6、batch.size=4096 linger.ms=50 compression.type=zstd 5.68 MB/sec

7、batch.size=4096 linger.ms=50 compression.type=gzip 5.90 MB/sec

8、batch.size=4096 linger.ms=50 compression.type=lz4 3.72 MB/sec

9、batch.size=4096 linger.ms=50 buffer.memory=67108864 3.76 MB/sec

消者 一次处理500条 81.2066m/s

消者 一次处理2000条 138.0992m/s

消息者 一次处理2000条 fetch.max.bytes=104857600 145.2033m/s

🔍 修訂版本 #2

★ 由 treeman 建立於 21 🍈Q🍈🍈 2023 18:07:38

✎ 由 treeman 更新於 26 🍈@🍈🍈 2024 18:15:31