**RocketMQ-kafka**

**设计手册**

|  |  |  |
| --- | --- | --- |
| 文件状态：  [] 草稿  [√] 正式发布  [ ]正在修改 | 文件标识： |  |
| 当前版本： | 0.5 |
| 作者： |  |
| 完成日期： |  |

**文档修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修订人 | 起止日期 | 修订说明 |
| 0.1 | Zyc | 15/4/21 | 初始版本 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 功能概述 - 1 -](#_Toc417396708)

[2 运行环境 - 1 -](#_Toc417396709)

[3 参数配置说明 - 1 -](#_Toc417396710)

[3.1 rocekt.properties配置项说明 - 1 -](#_Toc417396711)

[3.2 log4j.properties配置项说明 - 2 -](#_Toc417396712)

# 功能概述

将RocketMQ中的消息推送到kafka中，扮演了承上启下的角色或者是中转站的角色。

程序中涉及两个功能角色：RocketMQ Consume和kafka Producer。

运行此系统前提是已配置好RocketMQ和kafka运行环境。

# 版本说明

RocketMQ：3.2.5

Kafka：0.8.1.1

# 系统需求及运行环境

Rocfka将rocketmq中的队列镜像到kafka中，为kakfa用户提供访问rocketmq中数据的路径。

Kafka

RocketMQ

Rocfka

Kafka

Kafka

RocketMQ

RocketMQ

系统功能结构

# 设计

## 队列

一个rocketmq中队列对应一个kakfa中队列。

## 分区与顺序

分区与kakfa无一一对应关系，当前版本中不提供分区内有序特定。

Rocketmq partition

Rocketmq partition

Rocketmq partition

Rocketmq partition

Rocketmq partition

后续会考虑提供可选的分区的一一对应特性，以提供分区内消息顺序和rocketmq保持一致。

## 分布式与性能

每个节点运行一个复制实例，可多个节点同时运行。复制实例包括一个rocketmq的consumer和kafka的producer。

Kafka

Producer

Consumer

Producer

Consumer

Producer

Consumer

RocketMQ

## 稳定性

多个节点上的复制实例实现功能互备，提高稳定性。

## 复制实例

## 灵活性

可以选择从上次位置或最新位置开始。

# 程序接口与逻辑

iie.udps.rockfka.RocketPushToKafka:系统功能实现类

iie.udps.rockfka.util.PartitionerUtil:RocketMQ 消息转换为kafka消息

iie.udps.rockfka.test开发过程中用到的测试类

iie.udps.rockfka.kafka：kafka的Producer和consume，用于测试

iie.udps.rockfka.rocketmq:RocketMQ的Producer和consume，用于测试

rocket.properties:系统的主配置文件

log4j.properties:log4j的配置文件

## 统计数据格式

mysql字段说明

* topic： 当前topic名称
* count: interval间隔内处理message数量
* time： 系统当前时间
* gap： consume和producer时间间隔
* host： 当前统计消息所在主机，IP
* bytes： 当前批次接收到的数据字节数

# 安装

1. 创建mysql数据库

在配置ntp服务的集群中，创建mysql database,创建的database名称要和rocket.properties中url中的名称对应。

mysql> create database rocfkadb;

1. 赋予足够的权限给rocfka的运行节点

mysql> grant all on rocfkadb.\* to ‘username’@’ROCFKA\_HOST\_OR\_IP’ identified by ‘password’;

mysql> flush privileges;

1. 安装程序包

将所需文件资料拷贝到CDH\_HOME/lib/rocfka目录下

mv rocfka $CDH\_HOME/lib/

rocfka目录结构如下：

rocfka

|

|----lib

| |-----mylib3.25

| |-----kafka-rocketmq.jar

|

|----bin

| |----rocfka

|

|----log

备注：此项目用log4j作为日志输出，所以需要log4j-1.2.17环境

1. 修改配置文件rocket.properties

根据mysql地址，消息大小等信息修改相应的配置项，详见“配置说明”。

1. 部署配置文件

将修改完的rocket.properties文件移动到/etc/rocfka路径下。

mv rocket.properties /etc/rockfa/

# 启动和停止

* 启动

cd $CDH\_HOME/lib/rocfka/bin

sh rocfka.sh start

* 停止

cd $CDH\_HOME/lib/rocfka/bin

sh rocfka.sh stop

* 重启

cd $CDH\_HOME/lib/rocfka/bin

sh rocfka.sh restart

# 外部接口

## 配置说明

### 配置项说明

|  |  |  |
| --- | --- | --- |
| **配置项** | **值** | **说明** |
| consumeThreadMax | 25 | 消费线程池数量最大值 |
| consumeThreadMin | 9 | 消费线程池数量最小值 |
| ConsumeMessageBatchMaxSize | 1000 | 批量消费，一次消费多少条消息 |
| pullBatchSize | 32 | 批量拉消息，一次最多拉多少条 |
| pullInterval | 0 | 拉消息间隔 |
| adjustThreadPoolNumsThreshold | 1000000 | 当消息堆积达到此数量开始调整线程数量 |
| messages |  | 用于测试 |
| rocketNameServerDddr |  | RocketMQ的NameServer地址 |
| topics |  | 消费topic名称，如果消费多个topic，中间以逗号隔开 |
| groups |  | 每个topic对应的消费组名，如果消费多个topic，中间以逗号隔开 |
| brokers | Ip:9092 | Kafka broker的地址 |
| zkConnector | Ip:2181/kafka | Zookeeper的地址 |
| batchNumMessage | 200 | 消息在kafka Producer端buffer的数量 |
| partitionNums | 3 | 手动置顶kafka topic分区数量 |
| interval | 60000 | 日志统计信息输出的间隔时间，毫秒为单位 |
| bufferMaxMs | 500 | kafka Producer参数。当消息堆积时间达到此值，批量发送，单位毫秒。 |
| bufferMaxMessages | 100 | 当消息buffer数量达到此值，批量发送 |
| url |  | Mysql的url地址 |
| table |  | Mysql自定义表名 |
| username |  | Mysql用户名，默认为rocfka |
| password |  | Mysql密码，默认为’rocfka’ |

### 配置实例

配置说明

参数配置需要根据集群节点的硬件环境和Message大小设置，本实例程序运行环境如下：

|  |  |
| --- | --- |
| Cpu | 28 |
| Mem | 64 |
| Message大小 | cdrSize = 87661  ybrzSize = 1984081 |

需要动态调整的参数可以根据Message比例相应调整配置参数

#以下两个值由节点的CPU决定，最大不超过主机CPU核数，推荐CPUS\*80%

consumeThreadMax=25

consumeThreadMin=9

#根据消息大小成反比，消息越大，此值越小

ConsumeMessageBatchMaxSize=10

#如果消息1m以上，为了流控可以适当降低此值

pullBatchSize=32

#建议采用默认值

pullThresholdForQueue=1000

#kafka producer速度<consume速度，可以适当增大此值

pullInterval=0

#流控有关，采用默认值100000

adjustThreadPoolNumsThreshold=100000

#测试数据，无需设置

messages=100

#rocketMQ nameSrver 多个的话中间用分号隔开

rocketNameServerDddr=192.168.8.103:9876

#topic name to consume，多个topic中间以逗号隔开

topics=TopicTest\_1,TopicTest\_2

groups=group\_TopicTest\_1,group\_TopicTest\_2

#kafka brokers

brokers=192.168.8.105:9092,192.168.8.103:9092

#kafka zookeeper

zkConnector=192.168.8.105:2181/kafka

#==========kafka prop=========

#

batch.num.messages=200

#batchNumMessage=200

#分区数量，如果是两个topic，则分区数量中间用逗号隔开：3,3

partitionNums=3

#建议采用默认值

queue.buffering.max.ms=5000

#bufferMaxMs=5000

queue.buffering.max.messages=100

#bufferMaxMessages=100

#统计日志输出的时间间隔，如果测试建议3000，单位毫秒

interval=10000

#mysql

url=jdbc:myqsl://localhost:3306/test?

driver=com.mysql.jdbc.Driver

username=root

password=

#database=rocfka

table=rocfka

------------------------------------------------

### log4j.properties配置说明

log4j.logger.Tlog=info,appender1,FILE

log4j.logger.Mlog=debug,appender1,appender2

log4j.appender.appender1=org.apache.log4j.ConsoleAppender

log4j.appender.appender1.layout=org.apache.log4j.PatternLayout

log4j.appender.FILE=org.apache.log4j.FileAppender

log4j.appender.FILE.File=/tmp/rklog/run.log

log4j.appender.FILE.Append=true

log4j.appender.FILE.layout=org.apache.log4j.PatternLayout

log4j.appender.appender2=org.apache.log4j.DailyRollingFileAppender

log4j.appender.appender2.File=/tmp/rklog/count.log

log4j.appender.appender2.DatePattern='.'yyyy-MM-dd

log4j.appender.appender2.Append=false

log4j.appender.appender2.layout=org.apache.log4j.PatternLayout

Tlog：用于程序调试日志输出

Mlog:用于统计信息输出

appender1:控制台输出

FILE:文件输出

Applender2:文件输出（每天产生新的文件）

## 流量统计接口

统计视图层

MySQL

Rocfka

统计视图层

## 安装

1. 赋予集群节点查询权限

在**rocfka使用的mysql所在的节点**执行grant.sh脚本，以给所有主机赋予查询数据库的权限。

cd $CDH\_HOME/lib/rocfka/bin

sh grant.sh

1. 导出环境变量

在所有节点的/root/.bashrc中添加如下内容

CDH\_HOME=/opt/cloudera/parcels/CDH

export PATH=$CDH\_HOME/lib/rocfka/bin:$PATH

**注：可以使用如下命令来快速添加**

clush –g all “echo \” CDH\_HOME=/opt/cloudera/parcels/CDH \” >> /root/.bashrc”

clush –g all “echo \”export PATH=$CDH\_HOME/lib/rocfka/bin:$PATH\” >> /root/.bashrc”

## 用户接口

* 查询格式：kafka-stats.sh 参数1 参数2 参数3

比如：dataflow 2015-01-01 2015-11-11 hour

* 参数说明：
  + 参数1表示查询开始日期
  + 参数2表示查询结束日期
  + 参数3表示查询格式（day或者hour）
* 参数格式说明：
  + 日期格式必须为yyyy-MM-dd格式
  + 参数3必须是hour或者day，两者中的一个

使用说明：需要手动创建database，在rocket.properties的url中指定，同时配置文件中指定table名称以及username和password。只有正确安装配置好mysql，此功能才能使用。

附录：rocfka脚本内容如下

#!/bin/bash

echo the path of rocket.properteis is /etc/rocketmq-kakfa/

echo the path of count.log is /opt/cloudera/parcels/CDH/lib/rocketmq-kafka/log/

export ROCFKA=/opt/cloudera/parcels/CDH/lib/rocketmq-kafka

#export PATH=$ROCFKA/bin:$PATH

export CLASSPATH=$ROCFKA/lib/kafka-rocketmq.jar:$ROCFKA/lib/mqlib3.2.5/\*:/opt/cloudera/parcels/CDH/jars/\*

#java -jar /opt/cloudera/parcels/CDH/lib/rocketmq-kafka/lib/kafka-rocketmq.jar /etc/rocketmq-kafka/rocket.properties

java iie.udps.rockfka.RocketPushToKafka /etc/rocketmq-kafka/rocket.properties