**canal & kafka 数据同步说明文档**

**一：项目背景**

由于目前Goldoffice 后台系统（BO）采用的是postgresql数据库，IX交易服务采用的是mysql数据库，bo需要兼容维护和管理ix的数据，所以需要把IX trade的mysql数据，同步到对应的postgresql数据库（trade和bo表结构大致相同），便于后台统一管理和维护。同步步骤如下：

1. mysql数据库开启binlog日志，将数据的变化写入日志文件
2. 开启canal监听，将mysql binlog的日志文件获取，
3. Canal client分析日志文件，把数据表里每行记录的变化组装成json格式的字符串发送消息到kafka
4. Kafka client 消费消息，将接收到的消息转化成数据源写入postgresql数据库，完成数据的同步

**二：canal&kafka工作原理**

1. **canal**
   1. 意译（通道/管道/沟渠 ），基于数据库增量日志解析，提供增量数据订阅&消费，目前主要支持了mysql

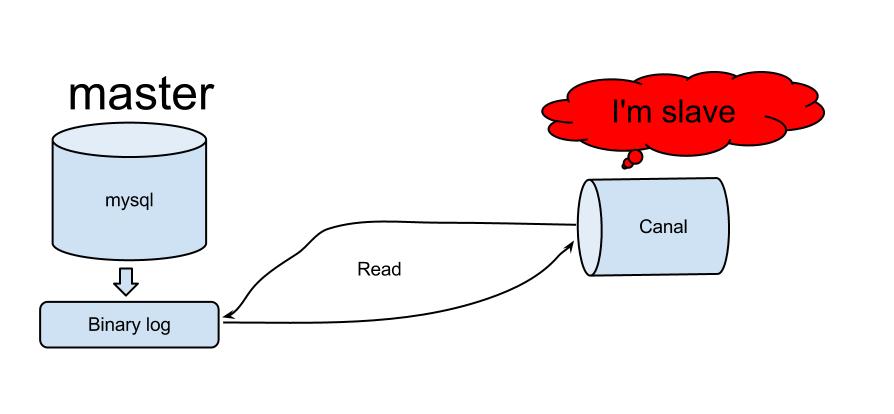
2）mysql master将改变记录到二进制日志(binary log)中（这些记录叫做二进制日志事件，binary log events

3）canal模拟mysql slave的交互协议，伪装自己为mysql slave，向mysql master发送dump协议

4）mysql master收到dump请求，开始推送binary log给slave(也就是canal)

5）canal解析binary log对象(原始为byte流)

如下图：



1. **kafka**

kafka是分布式发布-订阅消息系统, 同时为发布和订阅提供高吞吐量。

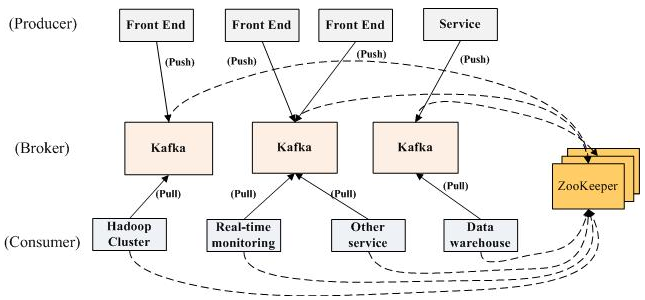
1. Topic：特指Kafka处理的消息源（feeds of messages）的不同分类。

2) Partition：Topic物理上的分组，一个topic可以分为多个partition，每个partition是一个有序的队列。partition中的每条消息都会被分配一个有序的id（offset）。

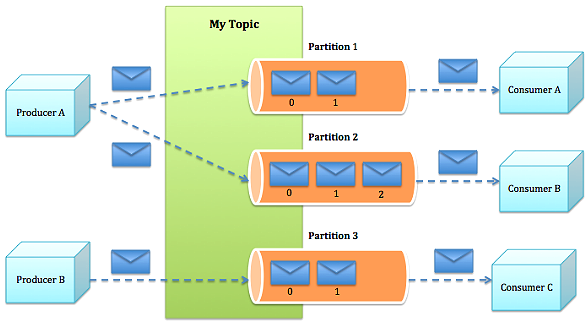
1. Message：消息，是通信的基本单位，每个producer可以向一个topic（主题）发布一些消息。
2. Producers：消息和数据生产者，向Kafka的一个topic发布消息的过程叫做producers。
3. Consumers：消息和数据消费者，订阅topics并处理其发布的消息的过程叫做consumers。

6）Broker：缓存代理，Kafa集群中的一台或多台服务器统称为broker。

架构图：



消息发送流程图：



**三：软件安装配置**

**1. Canal**

1) 开启mysql binlog ,详见”六:附录”

2) 安装canal server,位置: 172.30.10.107 /usr/local/canal

3) 修改配置文件

/usr/local/canal/conf/canal.properties

/usr/local/canal/conf/example/instance.properties

4)启动canal server服务

1. **Kafka**
2. 安装kafka 到 /usr/local/kafka\_2.11
3. 修改文件：../config/server.properties

增加下面监听：

listeners=PLAINTEXT://192.168.35.108:9092

zookeeper.connect=192.168.35.108:2181

delete.topic.enable=true

将localhost改成对应ip地址

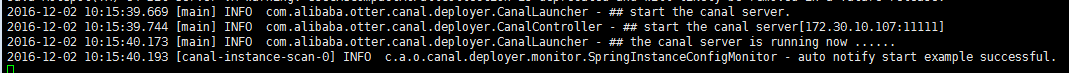
1. 启动zookeeper
2. 启动kafka服务端

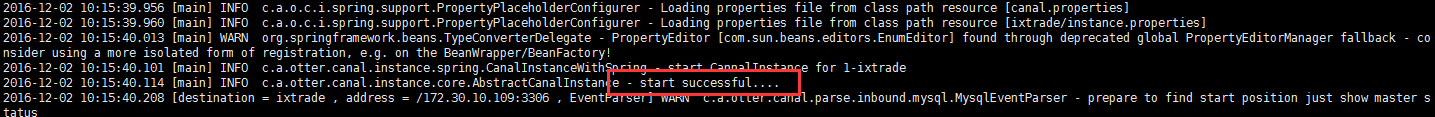
**五：日常管理维护**

1. **canal日常维护**
2. 查看canal和instance日志:查看运行情况:

tail -fn 200 /usr/local/canal/logs/canal/canal.log

tail -fn 200 /usr/local/canal/logs/example/example.log





1. 启动和停止:

bin/start.sh bin/stop.sh

1. 如果监控失效,可删除 meta.dat文件,重启canal
2. 查看binlog日志： mysqlbinlog /var/lib/mysql/mysql-bin.000001
3. 清空mysqlbinlog (可清空)

reset master;

flush privileges;

1. **kafka日常维护**
2. 启动zookeeper,

cd /usr/local/zookeeper-3.4.9/bin/

./ zkServer.sh start

关闭zookeeper, ./zkServer.sh stop

1. 启动和关闭kafka:

start：

nohup /usr/local/kafka\_2.11/bin/./kafka-server-start.sh /usr/local/kafka\_2.11/config/server.properties &

stop：

/usr/local/kafka\_2.11/bin/./kafka-server-stop.sh

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

cd /usr/local/kafka\_2.11/bin/

创建topic:

./kafka-topics.sh --create --zookeeper 192.168.35.110:2181 --replication-factor 1 --partitions 1 --topic KAFKA-TOPIC

查看topic list:

./kafka-topics.sh --list --zookeeper 192.168.35.110:2181

查看topic:

./kafka-topics.sh --describe --zookeeper 192.168.35.110:2181 --topic KAFKA-TOPIC

删除topic:

./kafka-topics.sh --delete --zookeeper 192.168.35.110:2181 --topic KAFKA-TOPIC

生产消息

./kafka-console-producer.sh --broker-list 192.168.35.110:9092 --topic KAFKA-TOPIC

消费消息

./kafka-console-consumer.sh --zookeeper 192.168.35.110:2181 --from-beginning --topic KAFKA-TOPIC （去掉from beginning就会只消费最新消息）

新增分区

./kafka-topics.sh --zookeeper 172.30.10.107:2182 -alter --partitions 20 --topic TABLE\_IX\_TRADE\_ALL\_BASE