

MaxWell

第一章 MaxWell概述

定义

Maxwell 是由美国 Zendesk 开源，用 Java 编写的 MySQL 实时抓取软件。实时读取 MySQL 二进制日志 Binlog，并生成 JSON 格式的消息，作为生产者发送给 Kafka，Kinesis、RabbitMQ、Redis、Google Cloud Pub/Sub、文件或其它平台的应用程序。

工作原理

MySQL主从复制过程

- 1.Master 主库将改变记录，写到二进制日志(binary log)中
- 2.Slave 从库向 mysql master 发送 dump 协议，将 master 主库的 binary log events 拷贝 到它的中继日志(relay log);
- 3.Slave 从库读取并重做中继日志中的事件，将改变的数据同步到自己的数据库。

MaxWell的工作原理

Maxwell 的工作原理很简单，就是把自己伪装成 MySQL 的一个 slave，然后以 slave 的身份假装从 MySQL(master)复制数据。

MySQL的binlog

什么是 binlog

MySQL 的二进制日志可以说 MySQL 最重要的日志了，它记录了所有的 DDL 和 DML(除了数据查询语句)语句，以事件形式记录，还包含语句所执行的消耗的时间，MySQL 的二进制日志是事务安全型的。

一般来说开启二进制日志大概会有 1%的性能损耗。二进制有两个最重要的使用场景：

其一：MySQL Replication 在 Master 端开启 binlog，Master 把它的二进制日志传递给 slaves 来达到 master-slave 数据一致的目的。

其二：自然就是数据恢复了，通过使用 mysqlbinlog 工具来使恢复数据。

二进制日志包括两类文件：二进制日志索引文件（文件名后缀为.index）用于记录所有的二进制文件，二进制日志文件（文件名后缀为.00000*）记录数据库所有的 DDL 和 DML(除了数据查询语句)语句事件。

开启binlog

找到 MySQL 配置文件的位置Linux: /etc/my.cnf 如果/etc 目录下没有，可以通过 locate my.cnf 查找位置。

在[mysqld] 区块，设置/添加 log-bin=mysql-bin 这个表示 binlog 日志的前缀是 mysql-bin，以后生成的日志文件就是 mysql-bin.000001 的文件后面的数字按顺序生成，每次 mysql 重启或者到达单个文件大小的阈值时，新生一个文件，按顺序编号。

binlog分类设置

mysql binlog 的格式有三种，分别是STATEMENT,MIXED,ROW。在配置文件中可以选择配置
binlog_format= statement|mixed|row

三种格式的区别： ■ statement 语句级，binlog 会记录每次一执行写操作的语句。相对 row 模式节省空间，但是可能产生不一致性，比如 update test set create_date=now(); 如果用 binlog 日志进行恢复，由于执行时间不同可能产生的数据就不同。

优点：节省空间

缺点：有可能造成数据不一致。

■ row 行级，binlog 会记录每次操作后每行记录的变化。

优点：保持数据的绝对一致性。因为不管 sql 是什么，引用了什么函数，他只记录 执行后的效果。

缺点：占用较大空间。

■ mixed 混合级别，statement 的升级版，一定程度上解决了 statement 模式因为一些情况 而造成的数据不一致问题。

默认还是 statement，在某些情况下，譬如：当函数中包含 UUID() 时；包含 AUTO_INCREMENT 字段的表被更新时；执行 INSERT DELAYED 语句时；用 UDF 时；会按照 ROW 的方式进行处理 优点：节省空间，同时兼顾了一定的一致性。缺点：还有些极个别情况依旧会造成不一致，另外 statement 和 mixed 对于需要对 binlog 监控的情况都不方便。

综合上面对比，Maxwell 想做监控分析，选择 row 格式比较合适

第二章 Maxwell使用

MySQL环境准备

```
1  sudo vim /etc/my.cnf
2  在[mysqld]模块下添加一下内容
3  [mysqld]
4  server_id=1
5  log-bin=lpigking
6  binlog_format=row
7  #binlog-do-db=test_maxwell
8  并重启 Mysql 服务
9  sudo systemctl restart mysqld
10 登录 mysql 并查看是否修改完成
11 mysql -uroot -p123456
12 mysql> show variables like '%binlog%';
13 查看下列属性
14 binlog_format | ROW
```

初始化MaxWell元数据库

```
1 在MySQL 中建立一个 maxwell 库用于存储 Maxwell 的元数据
2 mysql -uroot -p123456
3 mysql> CREATE DATABASE maxwell;
4
5 设置 mysql 用户密码安全级别
6 mysql> set global validate_password_length=4;
```

```

7  mysql> set global validate_password_policy=0;
8
9  分配一个账号可以操作该数据库
10 mysql> GRANT ALL ON maxwell.* TO 'maxwell'@'%' IDENTIFIED BY '123456';
11
12 分配这个账号可以监控其他数据库的权限
13 mysql> GRANT SELECT , REPLICATION SLAVE , REPLICATION CLIENT ON *.* TO maxwell@'%' ;
14
15 刷新 mysql 表权限
16 mysql> flush privileges;

```

MaxWell进程启动

Maxwell 进程启动方式有如下两种：

(1) 使用命令行参数启动 Maxwell 进程

```
1 bin/maxwell --user='maxwell' --password='123456' --host='master' --producer=stdout
```

(2) 修改配置文件，定制化启动 Maxwell 进程

```

1 cp config.properties.example config.properties
2 vim config.properties
3 bin/maxwell --config ./config.properties

```

第三章 两个小案例

1 监控MySQL数据

运行maxwell监控mysql

```
1 bin/maxwell --user='maxwell' --password='123456' --host='hadoop102' --producer=stdout
```

想mysql的text_maxwell库的test表插入一条数据，查看maxwell的控制台输出

```
1 mysql> insert into test values(1,'aaa');
```

2 输出到Kafka

一定要先启动zookeeper和kafka。

启动maxwell监控binlog。

```

1 bin/maxwell --user='maxwell' --password='123456' --host='master' --producer=kafka --
  kafka.bootstrap.servers=master:9092 --kafka_topic=maxwell

```

开启kafka消费者

```
1 kafka-console-consumer.sh --bootstrap-server master:9092 --topic maxwell
```

插入mysql数据

```
1 mysql> insert into test values (5,'eee');
```

