**sqoop导入hive数据实战**

# sqoop版本介绍

sqoop的版本到目前为止，主要分为sqoop1 和sqoop2 ， sqoop1.4.4之前的所有版本称为sqoop1，之后的版本

sqoop1.991，sqoop1.99.2,sqoop1.99.3 称为sqoop2。

## 1.1sqoop1

sqoop 架构非常简单，其整合了 Hive 、 Hbase 和 Oozie ，通过 map-reduce 任务来传输数据，从而提供并发特性和容错。

sqoop 主要通过 JDBC 和关系数据库进行交互。理论上支持 JDBC 的 database 都可以使用 sqoop 和 hdfs 进行数据交互

## 1.2sqoop2

在架构上， sqoop2 引入了 sqoop server （具体服务器为 tomcat ），对 connector 实现了集中的管理。其访问方式也变得多样化了，其可以通过 REST API 、 JAVA API 、 WEB UI 以及 CLI 控制台方式进行访问。另外，其在安全性能方面也有一定的改善，在 sqoop1 中我们经常用脚本的方式将 HDFS 中的数据导入到 mysql 中，或者反过来将 mysql 数据导入到 HDFS 中，其中在脚本里边都要显示指定 mysql 数据库的用户名和密码的，安全性做的不是太完善。在 sqoop2 中，如果是通过 CLI 方式访问的话，会有一个交互过程界面，你输入的密码信息不被看到。

## 1.3 sqoop1和sqoop2的区别

### 1.3.1 工作模式

sqoop1 基于客户端模式，用户使用客户端模式，需要在客户端节点安装sqoop

和连接器/驱动器 sqoop2 基于服务的模式，是sqoop1的下一代版本，服务模式主要分为 sqoop2 server 和 client，用户使用服务的模式，需要在sqoop2 server安装连接器/驱动器，所有配置信

息都在sqoop2 server进行配置。

### 1.3.2 MR角度透视其区别

sqoop 只提交一个map作业，数据的传输和转换都由Mappers来完成。

sqoop2 提交一个MapReduce

作业，Mappers负责从数据源传输（transporting）数据，Reducers负责根据指定的源来转换（transforming）数据。

### 1.3.3 从安全的角度

sqoop2从安全的角度进行设计，管理员来配置数据的源和目标，然而，操作者直接使用已经建好的链接，不需要了解该链接的具体细节信息，按需获取即可。

### 1.3.4 其他方面

Web UI：sqoop 只提供了CLI方式，sqoop2同时也支持Web UI的方式。Sqoop Server以REST服务方式提供CLI和 Web UI

# 2. sqoop、hive常用命令

## 2.1 拷贝SQL jar包放到指定目录下

mysql-connector-java.jar上传到 /usr/share/java目录

上传mysql-connector-java-5.1.37-bin.jar包 到namenode 节点的/opt/cloudera/parcels/CDH-5.4.10-1.cdh5.4.10.p0.16/lib/hive/lib目录

上传ojdbc6.jar包 到namenode 节点的sqoop lib目录：

/opt/cloudera/parcels/CDH-5.4.10-1.cdh5.4.10.p0.16/lib/sqoop/lib

## 2.2 列出SQL中的所有库

sqoop list-databases --connect jdbc:mysql://192.168.100.100:3306/ -username root -password 123456

## 2.3 连接SQL 并列出库中的表

sqoop list-tables --connect jdbc:mysql://192.168.100.100:3306/oozie --username root --password 123456

sqoop list-tables --connect jdbc:mysql://192.168.100.100:3306/test --username root --password 123456

# 3. 数据导入方法举例

## 3.1 从关系数据库导入数据到hdfs

sqoop import --connect jdbc:mysql://192.168.100.100:3306/test --username root --password 123456 --table users -m 1

hadoop dfs -find / -name users

## 3.2 从关系数据库导入数据到hive

sqoop import -hive-import --connect jdbc:mysql://192.168.100.100:3306/test --username root --password 123456 --table users -m 1

## 3.3 hdfs数据导出到mysql

sqoop export -m 1 --connect jdbc:mysql://192.168.100.100:3306/test --username root --password 123456 --table users --export-dir '/user/hive/warehouse/users/part-m-00000' --fields-terminated-by '\001' -input-null-string '\\N' -input-null-non-string '\\N'

--export-dir '/user/root/part-m-00000' #hive中被导出的文件

--fields-terminated-by '\001' #hive中被导出的文件字段的分隔符

没有指定null字段分隔符，导致错位

# 4. blob数据导入处理

alex登陆

select \* from user\_tables;

sqoop list-tables --connect jdbc:oracle:thin:@192.168.100.200:1521:orcl --username ALEX --password 123456

select count(\*) from otdrtest1;

select count(\*) from otdrtest2;

select \* from otdrtest1 order by sdatetime desc;

select \* from otdrtest2 order by smdatetime desc;

## 4.1 测试导带blob字段oracle表到hive

sqoop import --hive-import --connect jdbc:oracle:thin:@192.168.100.200:1521:orcl --username ALEX --password 123456 --verbose --table OTDRTEST1 -m 1

导入报错

(NKIND,LNKID,SDATETIME,NSYSTEMID,STRACEFULL)=>(NUMBER,NUMBER,VARCHAR2,NUMBER,BLOB)

16/12/26 19:46:13 ERROR tool.ImportTool: Encountered IOException running import job: java.io.IOException: Hive does not support the SQL type for column strace

## 4.2 导入数据到hdfs

sqoop import --append --connect jdbc:oracle:thin:@192.168.100.200:1521:orcl --username ALEX --password 123456 --query "select strace,SDATETIME,NKIND,LNKID,STEMID from OTDRTEST1 where \$CONDITIONS and rownum<=100" --target-dir /otdrtest1 -m 1

sqoop import --append --connect jdbc:oracle:thin:@192.168.100.200:1521:orcl --username ALEX --password 123456 --query "select \* from OTDRTEST1 where \$CONDITIONS and rownum<=100" --target-dir /otdrtest1 -m 1

## 4.3 创建hive表

create table otdrtest1(STRACEFULL string,SDATETIME string,NKIND string,LNKID string,STEMID string ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

## 4.4 加载hdfs数据到hive表

load data inpath '/otdrtest1/part-m-00000' into table otdrtest1;

select count(\*) from otdrtest1;

hive -e 'select sdatetime from otdrtest1 order by sdatetime desc' >qq

# 5. sqoop定时增量导入

## 5.1 导入前一天数据到hdfs

增量数据导入

select strace,SDATETIME,NKIND,LNKID,STEMID from OTDRTEST1

where to\_date(sdatetime,'yyyy-mm-dd,hh24:mi:ss') >= trunc(sysdate)-1 and

to\_date(sdatetime,'yyyy-mm-dd,hh24:mi:ss') < trunc(sysdate);

sudo sqoop import --connect jdbc:oracle:thin:@192.168.100.200:1521:orcl --username ALEX --password 123456 --query "select strace,SDATETIME,NKIND,LNKID,STEMID from OTDRTEST1 where to\_date(sdatetime,'yyyy-mm-dd,hh24:mi:ss') >= trunc(sysdate)-1 and

to\_date(sdatetime,'yyyy-mm-dd,hh24:mi:ss') < trunc(sysdate) and lnkid=1 and \$CONDITIONS" -m 1 --target-dir /user/hive/warehouse/ecmk/otdrtest11 --fields-terminated-by ","

## 5.2 加载数据到hive表

sudo hive -e "load data inpath '/user/hive/warehouse/ecmk/otdrtest11/part-m-00000' into table otdrtest11"

## 5.3 删除数据存放目录

sudo hadoop dfs -rmr /user/hive/warehouse/ecmk/otdrtest11

## 5.4 清空回收站删除目录

sudo hadoop dfs -rmr /user/root/.Trash/Current/user/hive/warehouse/ecmk/otdrtest11

## 5.5定时任务

0 \*/2 \* \* \* /home/hadoop/sh/otdrtest2.sh

10 22 \* \* \* /home/hadoop/sh/otdrtest1.sh

\*/5 \* \* \* \* /home/hadoop/sh/lnkid.sh

# 6. 集群hive数据迁移

## 6.1 导出hive表数据到本地文件

导出hive表数据到本地文件(注意：加中间逗号分隔符)

insert overwrite local directory '/home/hadoop/exporthive'

select strace,sdatetime,nkind,lnkid,stemid from otdrtest1;

insert overwrite local directory '/home/hadoop/exporthive' ROW FORMAT DELIMITED FIELDS TERMINATED BY','

select strace,sdatetime,nkind,lnkid,stemid from otdrtest1;

## 6.2 本地文件压缩并远程发送

## 6.3 创建hive表

CREATE TABLE OTDRTEST1100(STRACEFULL string,SDATETIME string,NKIND string,LNKID string,STEMID string ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

## 6.4 加载本地文件到hive表

load data local inpath '/home/hadoop/exporthive/000000\_0' into table otdrtest1100;