



Sqoop实践



星环科技
www.transwarp.io

版权声明



培训上所刊载的所有内容，包括但不限于文字报道、图片、声音、录像、图表、标志、标识、广告、商标、商号、域名、软件、程序、版面设计、专栏目录与名称、内容分类标准以及为注册用户提供的任何或所有信息，均受《中华人民共和国著作权法》、《中华人民共和国商标法》、《中华人民共和国专利法》及适用之国际公约中有关着作权、商标权、专利权及其他财产权所有法律的保护，为星环信息科技（上海）有限公司及/或相关权利人专属所有或持有。

使用者将星环信息科技（上海）有限公司提供的内容与服务用于非商业用途、非营利、非广告目的而纯作个人消费时，应遵守着作权法以及其他相关法律的规定，不得侵犯星环信息科技（上海）有限公司及/或相关权利人的权利。

使用者将星环信息科技（上海）有限公司提供的内容与服务用于商业、营利、广告性目的时，需征得星环信息科技（上海）有限公司及/或相关权利人的书面特别授权，注明作者及文章出处“星环信息科技（上海）有限公司”，并按有关国际公约和中华人民共和国法律的有关规定向相关权利人支付版税。

未经星环信息科技（上海）有限公司的明确书面特别授权，任何人不得变更、发行、播送、转载、复制、重制、改动、散布、表演、展示或利用星环信息科技（上海）有限公司的局部或全部的培训内容或服务或在非星环信息科技（上海）有限公司所属的服务器上作镜像，否则以侵权论，依法追究法律责任。

星环信息科技（上海）有限公司负责管理星环信息科技所有培训及其相关事务，未经星环信息科技（上海）有限公司的明确书面特别授权，任何人不得以任何形式在国际互联网上变更、发行、播送、转载、复制、重制、改动、散布、表演、展示星环信息科技（上海）有限公司的培训材料，录音，视频和捕捉视频资源而形成的图像，否则将视作侵权，依法追究法律责任。

© 2015, 星环信息科技（上海）有限公司版权所有。

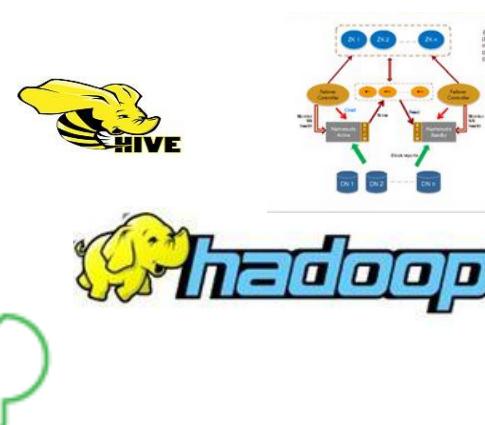
Transwarp和Transwarp Data Hub是星环信息科技（上海）有限公司在中国的商标或注册的商标。

Hadoop*, SPARK*是Apache 软件基金会在美国和其他国家的商标或注册的商标。

Java* 是 Oracle 和/或其子公司的注册的商标。

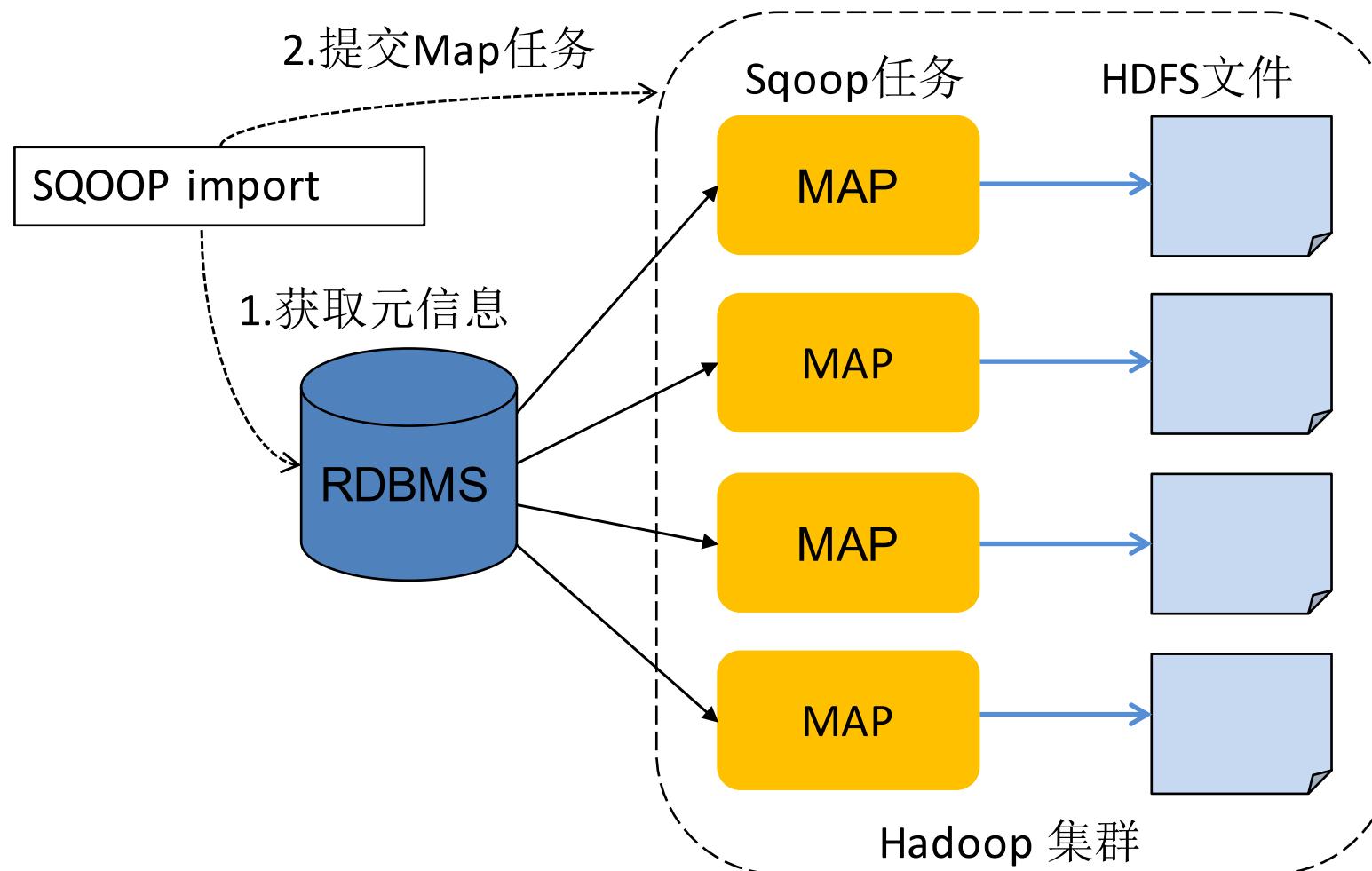
其他名称可能是商标各自所有者所有。

- Sqoop简介
 - Apache Sqoop (SQL-to-Hadoop) 项目旨在协助 RDBMS 与 Hadoop 之间进行高效的大数据交流。
 - 将Hadoop和关系型数据库中的数据相互转移的工具
 - 可以将一个关系型数据库（例如： MySQL , Oracle, DB2 Postgres等）中的数据导进到 Hadoop的HDFS\Hive\HBase中，也可以将HDFS的数据导进到关系型数据库中。
 - 通过hadoop的mapreduce把数据从关系型数据库中导入数据到HDFS。
- sqoop支持的数据库

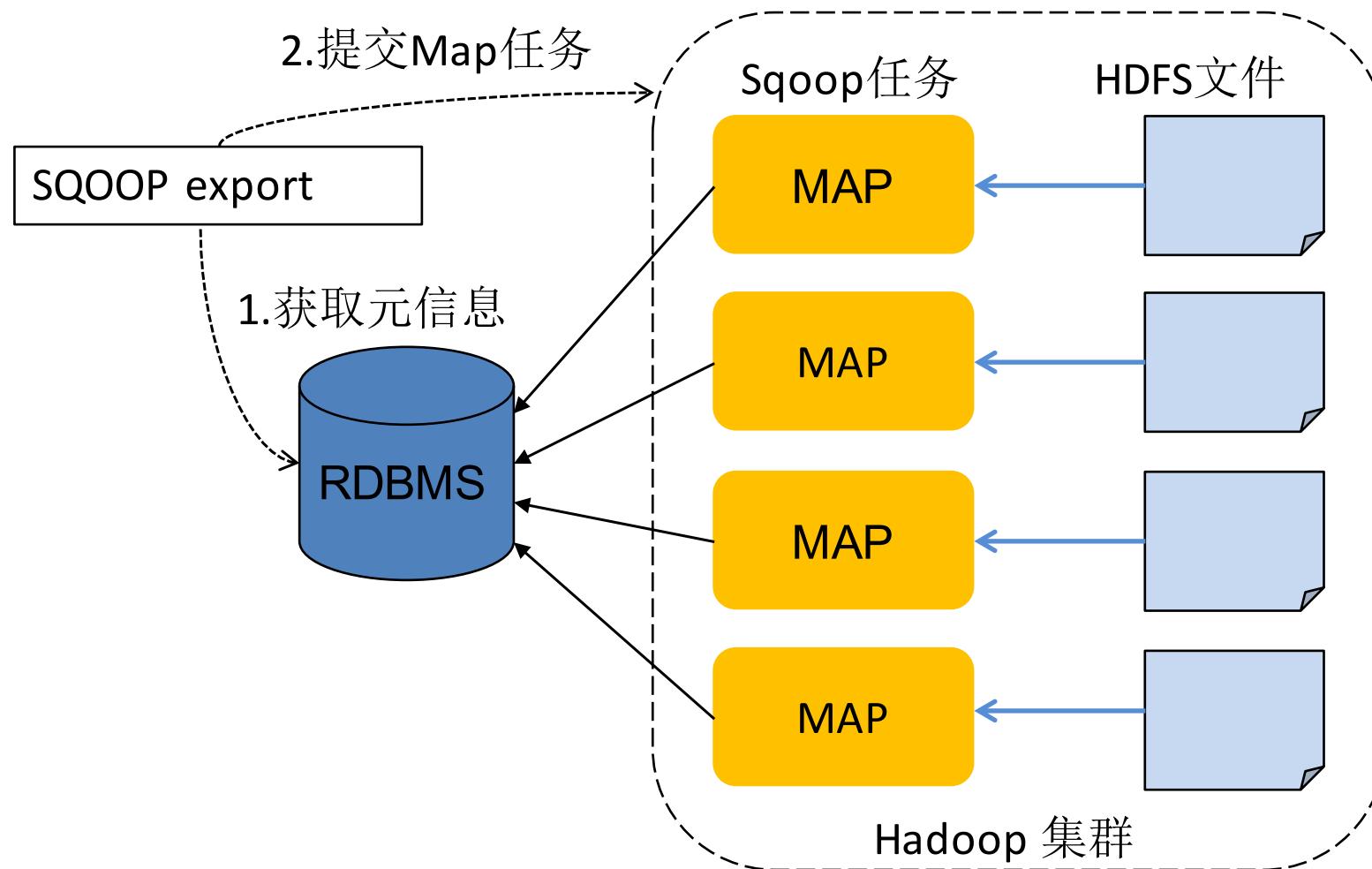


^[1] Sqoop(发音: skup), 项目开始于2009年, 最早是作为Hadoop的一个第三方模块存在, 后来独立成为一个Apache项目

基本原理：从RDB到Hadoop



基本原理：从Hadoop到RDB



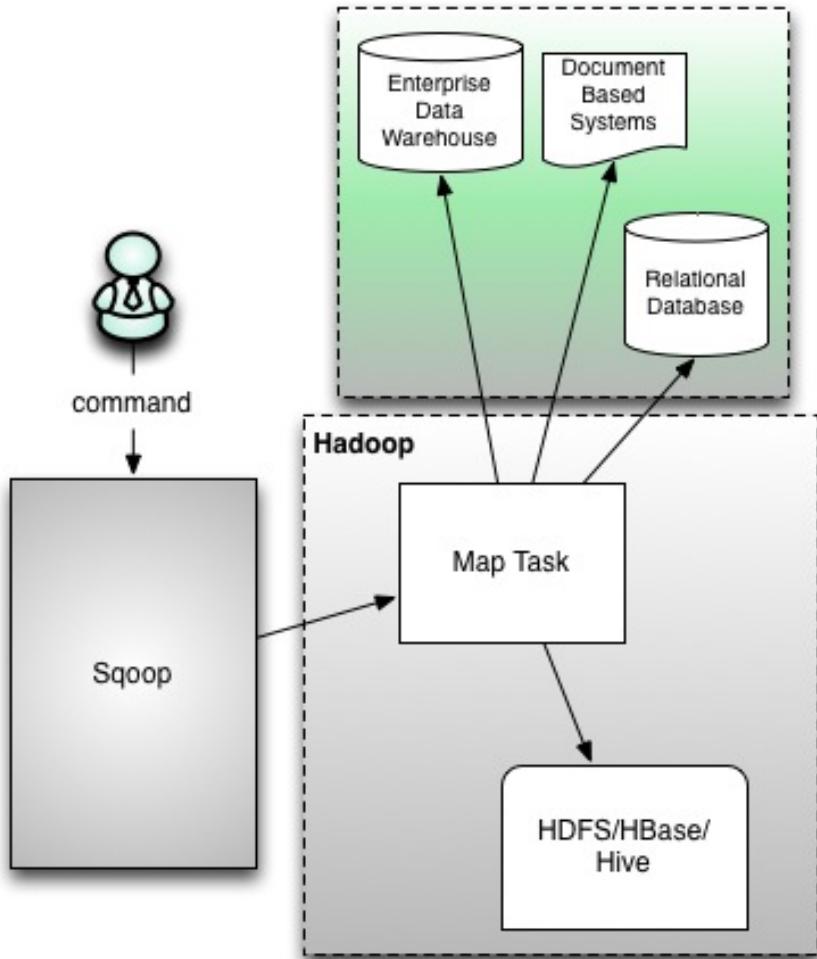
Sqoop使用概览



- 用户的Sqoop脚本最终会变成提交到YARN上的一个M/R任务。
- 通过Sqoop用户可以将数据从RDB抽取至HDFS。输入端是关系型数据库中的某张表，
Sqoop会去一行一行的把数据从这张表里读出来写入到HDFS；
- 输出端是HDFS上关于这个表的文件的集合。由于整个抽取ETL过程是并行化的，因此
输出端会有多个文件。
- 输出的文件可以指定列分隔符、换行符等分界符
- 通过Sqoop把HDFS上的文件回传到关系型数据库的过程和第二条类似。

- 关于版本
 - 目前为止，已经演化出了2个版本：sqoop1和sqoop2。
 - sqoop1的最新版本是1.4.6，sqoop2的最新版本是1.99.6；
- 两个版本是完全不兼容的
 - 版本号区别为1.4.x为sqoop 1，1.99x为sqoop 2。
- sqoop1和sqoop2在架构和用法上已经完全不同
 - 架构
 - sqoop1仅仅使用一个sqoop客户端，
 - sqoop2引入了sqoop server，对connector实现了集中的管理。
 - 访问方式
 - sqoop2可以通过REST API、JAVA API、WEB UI以及CLI控制台方式进行访问。
 - 安全性方面
 - sqoop1用脚本的方式实现HDFS与RDB的数据导入与导出，在脚本里边都要显示指定RDB的用户名和密码
 - sqoop2中，如果通过CLI方式访问的话，会有一个交互过程界面，输入的密码信息不会被看到
 - 同时Sqoop2引入基于角色的安全机制。

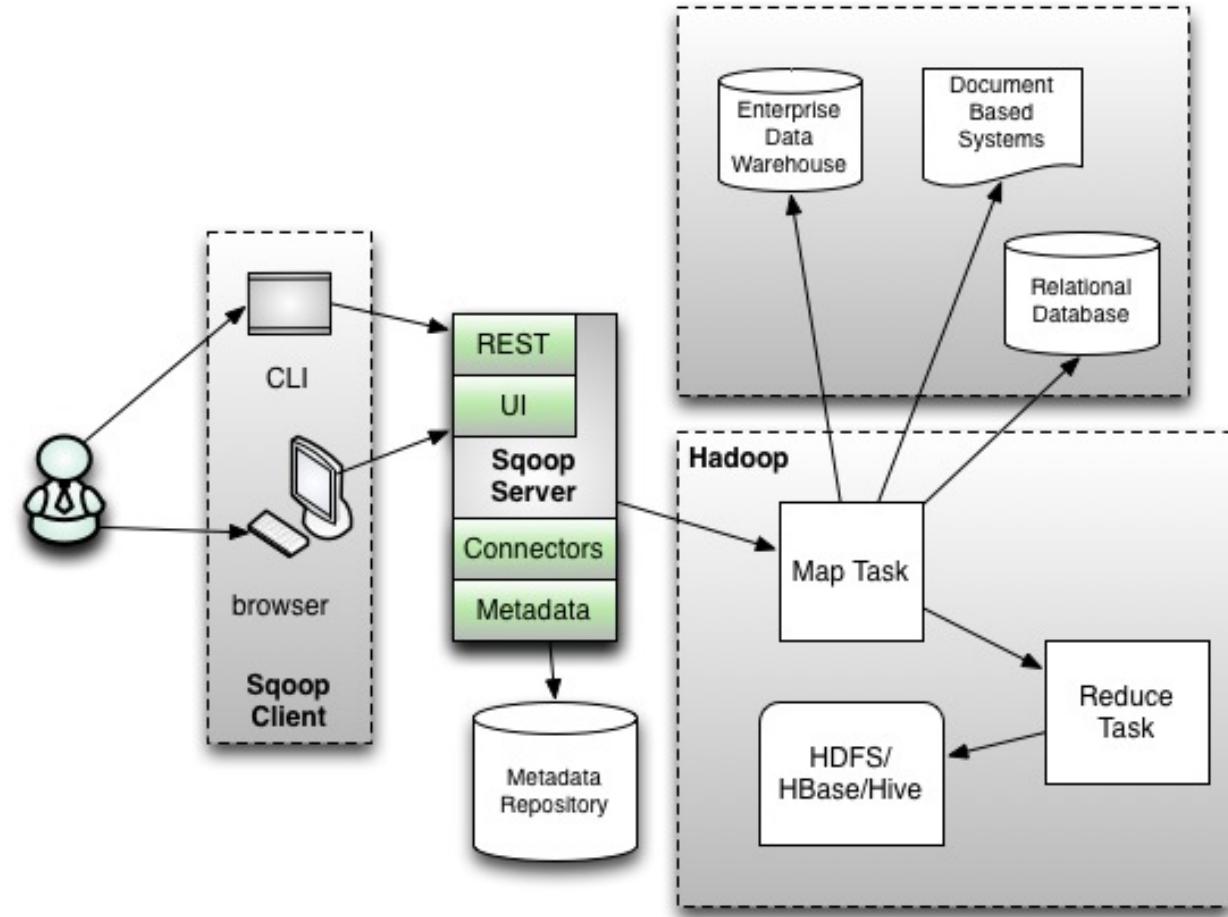
sqoop1架构图



- sqoop1优点
 - 架构部署简单
- sqoop1缺点
 - 命令行方式容易出错，格式紧耦合，无法支持所有数据类型，安全机制不够完善，例如密码暴漏，安装需要root权限，connector必须符合JDBC模型

sqoop2架构图

- sqoop2优点
 - 多种交互方式，命令行，web UI，rest API，connector集中化管理，所有的链接安装在sqoop server上，完善权限管理机制，connector规范化，仅仅负责数据的读写
- sqoop2缺点
 - 架构稍复杂，配置部署更繁琐



准备实验数据

- 修改MySQL root用户的密码
 - mysqladmin -u root password "transwarp"
- 创建测试数据库及赋权

```
mysql -u root -p
create database stock;
grant all privileges on *.* to root@localhost identified by 'transwarp';
flush privileges;
```

- 准备Mysql的JDBC驱动（只可在sqqoop1的实验中使用）
 - 将JDBC驱动（mysql-connector-java-5.1.31.jar）拷贝到/usr/lib/sqoop/lib目录下

sqoop1的实验

TDH中的sqoop



- TDH 4.1默认使用的是sqoop1，如果使用sqoop2，需要自己进行配置和安装
- TDH 4.2默认使用的是sqoop2，同时也支持sqoop1，不过需要自己进行下载

安装SQOOP1服务 (TDH4.1)



查看帮助

```
[root@tw-node2125 lib]# sqoop help
usage: sqoop COMMAND [ARGS]
```

Available commands:

- codegen Generate code to interact with database records
- create-hive-table** Import a table **definition** into Hive
- eval Evaluate a SQL statement and display the results
- export** Export an HDFS directory to a database table
- help List available commands
- import** Import a table from a database to HDFS
- import-all-tables Import tables from a database to HDFS
- job Work with saved jobs
- list-databases** List available databases on a server
- list-tables** List available tables in a database
- merge Merge results of incremental imports
- metastore Run a standalone Sqoop metastore
- version Display version information

See '**sqoop help COMMAND**' for information on a specific command.

列出源端RDB库名列表

- 列出RDB中的数据库列表（明文密码）

```
sqoop list-databases \
--username root \
--password transwarp \
--connect jdbc:mysql://192.168.128.234:3306/
```

语法范式解析：

sqoop list-databases: SQQOP 命令
--username: RDB用户名
--password: RDB明文密码
-P: 需要用户输入密码
--password-file: 用户密码文件（400权限）
--connect: 数据库JDBC连接串

- 列出RDB中的数据库列表（需要手工输入密码）

```
sqoop list-databases \
--username root \
-P \
--connect jdbc:mysql://192.168.128.234:3306/
```

- 列出RDB中的数据库列表（密码文件）

```
sqoop list-databases \
--username root \
--password-file file:/root/.pwd \
--connect jdbc:mysql://192.168.128.234:3306/
```

列出源RDB库内的表名

- 列出RDB中某数据库中的表（明文密码）

```
sqoop list-tables \
--username root \
--password transwarp \
--connect jdbc:mysql://192.168.128.234:3306/stock
```

- 列出RDB中某数据库中的表（需要手工输入密码）

```
sqoop list-tables \
--username root \
--P \
--connect jdbc:mysql://192.168.128.234:3306/stock
```

- 列出RDB中的数据库列表（密码文件）

```
sqoop list-tables \
--username root \
--password-file file:/root/.pwd \
--connect jdbc:mysql://192.168.128.234:3306/stock
```

语法范式解析：

sqoop list-databases: SQuoP 命令
--username: RDB用户名
--password: RDB明文密码
-P: 需要用户输入密码
--password-file: 用户密码文件（400权限）
--connect: 数据库JDBC连接串

从RDB的Table到HDFS

- 使用sqoop把数据从Mysql test数据库下的表t_daily_quote_day抽取到HDFS的目录 /user/root/person_all

```
sqoop import \
--connect jdbc:mysql://192.168.128.234:3306/stock \
--username root \
--password transwarp \
--table t_daily_quote_day \
--target-dir /hyperbase1/data/dailystock -m 4
```

语法范式解析：

sqoop import: SQOOP 命令，从关系型数据库导数到 Hadoop
--username: RDB用户名
--password: RDB明文密码
--connect: 数据库JDBC连接串
--table: 源数据库表名
--target-dir: HDFS目标文件夹位置

```
bash-4.1$ hdfs dfs -ls /hyperbase1/data/dailystock
Found 5 items
-rw-r--r-- 3 hdfs hbase 0 2015-12-08 09:43 /hyperbase1/data/dailystock/_SUCCESS
-rw-r--r-- 3 hdfs hbase 311478657 2015-12-08 09:41 /hyperbase1/data/dailystock/part-m-00000
-rw-r--r-- 3 hdfs hbase 43223324 2015-12-08 09:41 /hyperbase1/data/dailystock/part-m-00001
-rw-r--r-- 3 hdfs hbase 332969897 2015-12-08 09:43 /hyperbase1/data/dailystock/part-m-00002
-rw-r--r-- 3 hdfs hbase 287347 2015-12-08 09:41 /hyperbase1/data/dailystock/part-m-00003
```

其他选项	举例	说明
--split-by	--split-by id	数据划分列，默认为主键
-m	-m4	mapper个数，抽数线程数，默认4
--columns	--columns id,name	源表中选取的列名
--fetch-size	--fetch-size 1000	单次取多少条记录
--null-string	--null-string '\\N'	string类型的空值
--null-non-string	--null-non-string '\\N'	非string类型的空值
--delete-target-dir	--delete-target-dir	在导入数据前，先把HDFS目录清空
--append	--append	导入数据是以追加增量的方式导入
-z	-z,--compress	是否设置压缩，默认gzip格式
--fields-terminated-by	--fields-terminated-by "\01"	列分割符
--hive-drop-import-delims	--hive-drop-import-delims	去除^A, \n,\r 等特殊字符

注意事项：

- 当-m大于1时，需要指定--split-by字段
- target-dir 后面跟的HDFS目录需要确认用户有写权限

从RDB的Query到HDFS

- 使用sqoop查询数据Mysql test数据库下的表t_daily_quote_day，并抽取数据到HDFS的目录/user/root/

```
sqoop import \
--connect jdbc:mysql://192.168.164.25:3306/stock \
--username root \
--password transwarp \
--query "select F01,F02 from t_daily_quote_day where \$CONDITIONS"
\
--target-dir /tmp/tmp \
--split-by F01 \
-m 2
```

2个mapper，最终生成两个文件

固定范式

语法规式解析：

sqoop import: SQQOP 命令，从关系型数据库导数到Hadoop
--username: RDB用户名
--password: RDB明文密码
--connect: 数据库JDBC连接串
--query: SQL语句，其中 \\$CONDITIONS 是固定写法
--target-dir: HDFS目标文件夹位置

```
bash-4.1$ hdfs dfs -ls /hyperbase1/data/dailystock
Found 5 items
-rw-r--r-- 3 hdfs hbase 0 2015-12-08 09:43 /hyperbase1/data/dailystock/_SUCCESS
-rw-r--r-- 3 hdfs hbase 311478657 2015-12-08 09:41 /hyperbase1/data/dailystock/part-m-00000
-rw-r--r-- 3 hdfs hbase 43223324 2015-12-08 09:41 /hyperbase1/data/dailystock/part-m-00001
-rw-r--r-- 3 hdfs hbase 332969897 2015-12-08 09:43 /hyperbase1/data/dailystock/part-m-00002
-rw-r--r-- 3 hdfs hbase 287347 2015-12-08 09:41 /hyperbase1/data/dailystock/part-m-00003
```

注意事项：

- query 这种形式一般需要指定 --split-by 参数
- 当-m大于1时，需要指定--split-by 字段
- target-dir 后面跟的HDFS目录需要确认用户有写权限

控制并发度



- -m 和--split-by 参数，分别控制着Sqoop从RDB中抽取数据的并发度和数据分片列。例如 -m 4 --split-by id 代表按id列把数据分成4份，有4个map同时并行抽取。
- 整个抽取数据的瓶颈主要在下面两个方面
 - RDB和Hadoop集群的网络带宽
 - RDB的IO限制
- 抽取数据过程中的性能可以按单Map 5~10MB/s 做估算
- 注意：Map个数并非越多越好，过多的Map会导致RDB方面发生IO抢占，反而降低了整体抽数效率
- 实际应用时，map并发数应采用逐渐增加的方式

数据增量导入 : APPEND

- 增量导入append模式

```
sqoop import \
--connect jdbc:mysql://192.168.164.25:3306/stock \
--username root \
--password 111111 \
--query "select F01, F02, F03, F04 from t_daily_quote_day where
\$CONDITIONS" \
--target-dir /tmp/tmp/ \
--split-by F01 \
-m 2 \
--check-column F03 \
--incremental append \
--last-value 100.0
```

Id大于4的记
录都被导出

语法规式解析 :

sqoop import: SQOOP 命令, 从关系型数据库导数到Hadoop
--check-column: 用于检查增量数据的列
--incremental append: 设置为增量模式
--last-value :源数据中所有大于--last value的值都会被导入Hadoop

从HDFS到RDB

- 数据从HDFS导入到RDB

```
sqoop export \
--connect
jdbc:mysql://192.168.128.231:3306/stock \
--username root \
--password transwarp \
--table t_daily_quote_day \
--export-dir /user/root/dailystock \
--staging-table t_daily_quote_day \
--clear-staging-table
```

其他选项	举例	说明
-m	--m4	mapper个数，抽数线程数，默认4
--columns	--columns id,name	目标表的列名
--input-null-string	--input-null-string "\N"	string类型的空值
--input-null-non-string	--input-null-non-string "\N"	非string类型的空值
--input-fields-terminated-by	--input-fields-terminated-by "\01"	列分割符
--update-mode (allowinsert updateonly)	--update-mode updateonly	update 模式updateonly只允许根据updatekey更新，而allowinsert允许额外数据的插入
--update-key	--update-key id	update模式依据

注意：

- stage-table是RDB中的临时表，所有的HDFS数据会先入到临时表，等所有的任务完成后，再原子性把数据从临时表中转移到目标表
- 默认为插入模式，HDFS中的数据每条都会以增量的形式插入到RDB中。这样有可能违法主键唯一性而导致任务失败。这种情况下应该尝试update模式
- Update模式有两种，一种是更新模式，按照update key更新RDB中的数据，不会导致数据增加；另一种是allowinsert模式，当数据主键在RDB中不存在是转为insert操作

语法规式解析：

sqoop export: SQuoop 命令，从Hadoop到RDB
--table: RDB中的目标表
--export-dir: HDFS的源数据文件目录
--staging-table: 临时表
--clear-staging-table: 每次Sqoop前，清空临时表

sqoop_oracle范例

```
1 ORACLE_USERNAME="brj1jk"
2 ORACLE_PASSWORD='brj1jk2010'
3
4 ORACLE_CONNECTOR="jdbc:oracle:thin:@200.5.0.69:1521/xlxsj"
5
6 ORACLE_DATABASE="bjttms"
7
8 TABLENAME="tt_vehicle_t"
9 HDFS_URL="/data_oracle"
10 Partition="TT_VEH_20150101"
11 sudo -u hdfs hdfs dfs -mkdir $HDFS_URL/$TABLENAME
12
13 QUERY=
"vid,xtbh,sbcj,zddm,jcdbh,dmbh,sbbh,sjlx,cd,zxd,uploadtype,xsfx,jcsj,jcsj_1,jcsj_2,jcsj_3,jcsj_4,hphm,hpzl,sf,cpys,csys,sbcx,cb,xssd
dms,tpsl,tplj,xmlfile,ftpbh,wjfwqbh,jsbz,rksj,update_time,txtp"
14 sudo -u hdfs sqoop import --connect $ORACLE_CONNECTOR --username $ORACLE_USERNAME --password $ORACLE_PASSWORD --fetch-size 1000 \
15   -m 1 --target-dir $HDFS_URL/$TABLENAME/$Partition \
16   --query "select $QUERY from $ORACLE_DATABASE.$TABLENAME partition($Partition) where \$CONDITIONS" \
17   --fields-terminated-by "\\\01" \
18   --hive-drop-import-delims \
19   --null-string '\\N' \
20   --null-non-string '\\N'
21
22 sudo -u hdfs hdfs dfs -chmod 777 $HDFS_URL/$TABLENAME/$Partition
23
```

sqoop与InceptorSQL和Hyperbase



- 数据导入inceptorSQL分三步：
 - 1) 导入数据到HDFS
 - 2) 在inceptorSQL建表
 - 3) 使用 “LOAD DATA INPAHT” 将数据LOAD到表中
- 数据导入HpberBase分二步：
 - 1) 导入数据到HDFS
 - 2) 调用HBase put操作逐行将数据写入表

修改hdfs的权限

```
[root@host1 sqoop2]# hdfs dfs -ls /
Found 8 items
drwxr-xr-x - hbase hbase      0 2015-11-25 00:44 /hyperbase1
drwxr-xr-x - hbase hbase      0 2015-11-25 00:44 /hyperbase1_hregionindex
drwxr-xr-x - hdfs hbase      0 2015-11-25 00:48 /inceptorsql1
drwxr-xr-x - hdfs hbase      0 2015-12-04 01:12 /test
drwxrwxrwx - hdfs hadoop    0 2015-12-04 01:56 /tmp
drwxr-xr-x - hdfs hbase      0 2015-12-06 23:35 /transwarp-health-check-dir
drwxrwxrwx - hdfs hadoop    0 2015-11-26 01:43 /user
drwxr-xr-x - hdfs hbase      0 2015-11-25 00:41 /yarn1
[root@host1 sqoop2]# hdfs dfs -ls /test
Found 1 items
-rw-r--r-- 3 hdfs hbase     8 2015-12-04 01:12 /test/a
[root@host1 sqoop2]# hdfs dfs -chmod 777 /test
chmod: changing permissions of '/test': Permission denied
[root@host1 sqoop2]# su - hdfs
-bash-4.1$ hdfs dfs -chmod 777 /test
```