

Sqoop

简介

Hadoop非常强大的一点，就是它可以处理可靠地存储来自众多来源的多种形式的数据。但是为了与HDFS之外的存储库中的数据进行交互，MapReduce程序需要使用外部API。而Sqoop是一款开源工具，可以从结构数据存储中获取数据来交给Hive、HBase或是MapReduce进行进一步处理，并且将结果返回到数据存储以供客户端使用。

Sqoop附带连接器，用于处理各种流行的数据库，包括MySQL, PostgreSQL, Oracle, SQL Server, DB2和Netezza。还有一个通用JDBC连接器，用于连接任何支持Java JDBC协议的数据库。Sqoop提供优化的MySQL, PostgreSQL, Oracle和Netezza连接器，使用特定于数据库的API更有效地执行批量传输

sqoop版本

- sqoop现有两种版本，sqoop1和sqoop2。两者在结构上有很大区别。
- Sqoop 2是Sqoop的重写，它解决了Sqoop 1的体系结构限制。例如，**Sqoop 1是一个命令行工具，不提供Java API**，因此很难将其嵌入到其他程序中。此外，在Sqoop 1中，每个连接器都必须知道每个输出格式，因此编写新连接器需要做很多工作。Sqoop 2有一个运行作业的服务器组件，以及一系列客户端：**命令行界面（CLI）**，**Web UI**，**REST API**和**Java API**。Sqoop 2也将能够使用其他执行引擎，例如Spark。
- Sqoop 1发布系列是当前的稳定版本系列，Sqoop 2正在积极开发中，功能尚未开发完成，大多资料与书籍都是关于sqoop1。另外与sqoop连接最方便的是HBase。

Sqoop安装

- 1、从官网下载Sqoop1:目前最新稳定版本1.4.7
 - <http://www.apache.org/dyn/closer.lua/sqoop/1.4.7> 下载sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz
- 2、将文件解压
 - tar xf sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz
 - mv mv sqoop-1.4.7.bin__hadoop-2.6.0 /usr/hadoop/sqoop
- 3、修改环境配置
 - vim /etc/profile
 - 添加

```
#sqoop
export SQOOP_HOME=/usr/hadoop/sqoop
export PATH=$PATH:$SQOOP_HOME/bin
export CLASSPATH=.:$SQOOP_HOME/lib:$CLASSPATH
```
 - ```
export SQOOP_HOME=/usr/hadoop/sqoop
export PATH=$PATH:$SQOOP_HOME/bin
export CLASSPATH=.:$SQOOP_HOME/lib:$CLASSPATH
```

# Sqoop安装

- 修改配置文件

```
cd $SQOOP_HOME/conf
mv sqoop-env-template.sh sqoop-env.sh
```

```
vim swoop-env.sh 并修改
export HADOOP_COMMON_HOME=/usr/hadoop/hadoop3.0.3
export HADOOP_MAPRED_HOME=/usr/hadoop/hadoop3.0.3
export HIVE_HOME=/usr/hadoop/hive
对应你本机的home地址，这里因为还没有装hbase，后期再修改它的路径
```

- 下载mysql驱动包

<http://ftp.ntu.edu.tw/MySQL/Downloads/Connector-J/>

我选择下载了mysql-connector-java-5.1.46.tar.gz

```
解压 tar xf mysql-connector-java-5.1.46.tar.gz
mv mysql-connector-java-5.1.46/mysql-connector-java-5.1.30-bin.jar $SQOOP_HOME/lib
```

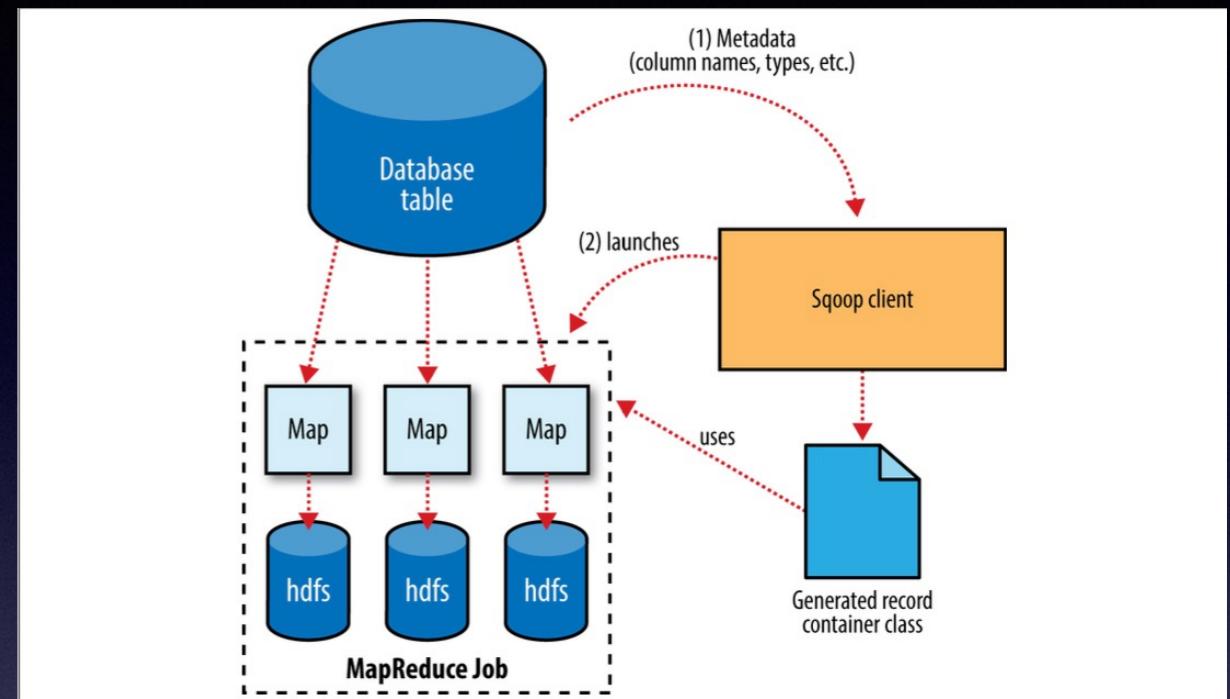
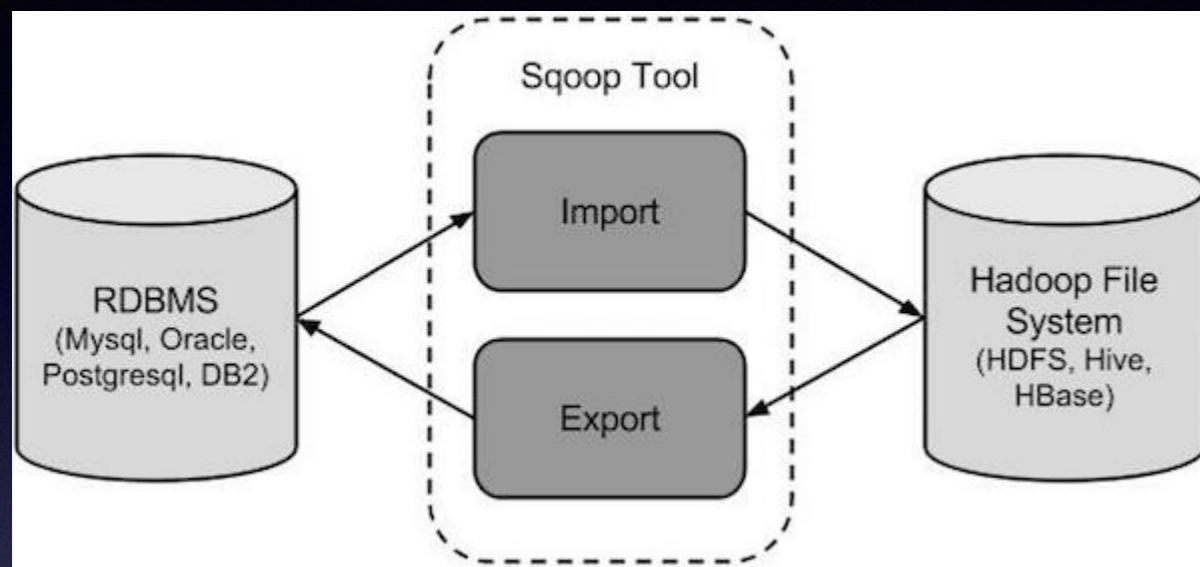
- 完成测试连接，确保mysql服务已开启： `service mysql start`

- `sqoop list-databases --connect jdbc:mysql://localhost/ --username root -P`
- 输入密码后可以看到mysql的库就成功了
- `sqoop version`

# Mysql配置

- 1、`vim /etc/mysql/mysql.conf.d/mysqld.cnf` , 注释掉 `bind-address = 127.0.0.1` 。 (以实现mysql使用其他IP登陆)
- 2、进入mysql：
  - `insert into mysql.user(Host,User,Password) values("%","root",password("123"));`
  - `GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' WITH GRANT OPTION;`
  - `flush privileges ;`
- 3、退出Mysql, `service mysql restart`

# 数据导入



`sqoop import --connect jdbc:mysql://localhost/BaseName --table TABLENAME`

详见 `sqoop help import`

Sqoop 将会启动一个 MapReduce 任务，然后连接 MySQL，并读取表。值得注意的是，分配了 n 个 mapper 进行处理，就会在同一文件夹生成 n 个文件。

另外，在 MySQL 里，也需要对该库修改权限：

```
GRANT ALL PRIVILEGES ON basename.* TO '*'@'localhost';
```

# 数据导入的一些例子

```
sqoop import \
--connect jdbc:mysql://localhost/userdb \
--username root \
--table emp_add \
--m 1 \
--where "city ='sec-bad'" \
--target-dir /wherequery
```

## 增量导入

```
sqoop import \
--connect jdbc:mysql://localhost/userdb \
--username root \
--table emp \
--incremental append \
--check-column id \
--last-value 1205
```

因为在实际的应用中，经常需要定期的导入数据库内的表，这时候Sqoop的增量导入特性就会经常用到。我们可以通过check-column来规定列，last value来选择导入比这个value值大的所有行

# 直接模式

某些数据库提供了专门用于快速提取数据的工具。例如，MySQL的mysqlimport应用程序可以从具有比JDBC通道更大吞吐量的表中读取。在Sqoop的文档中使用这些外部工具称为直接模式。必须由用户专门启用直接模式（通过—direct参数），因为它不像JDBC方法那样通用。

Sqoop还可以从PostgreSQL，Oracle和Netezza执行直接模式导入。

即使使用直接模式访问数据库的内容，仍然可以通过JDBC查询元数据。

# Sqoop与Hive

前提：将Hive / conf下的hive-site.xml  
拷贝到Sqoop / con下

```
sqoop create-hive-table --connect jdbc:mysql://localhost/BASENAME --table
TABLENAME --fields-terminated-by ','
hive> LOAD DATA INPATH "sqoop导入文件" INTO TABLE tablename;
```

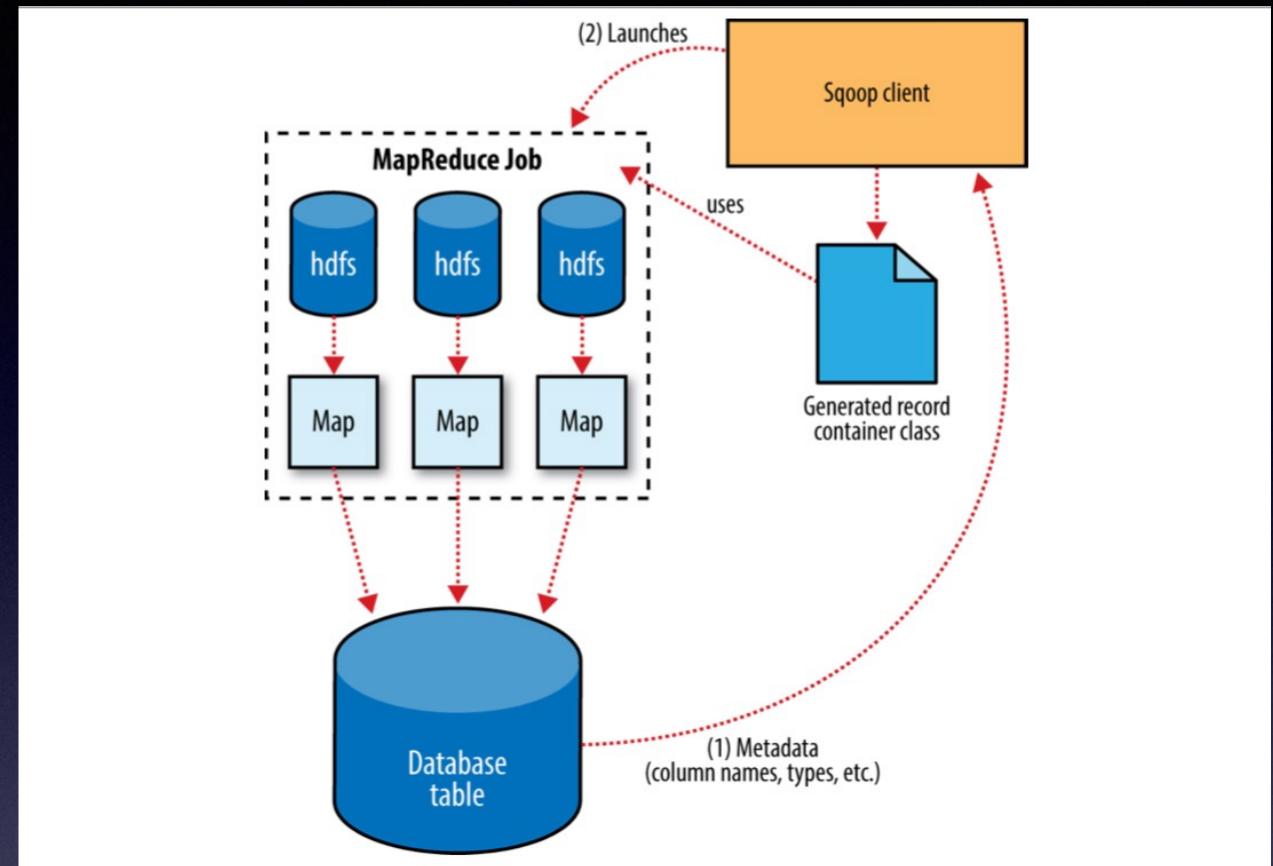
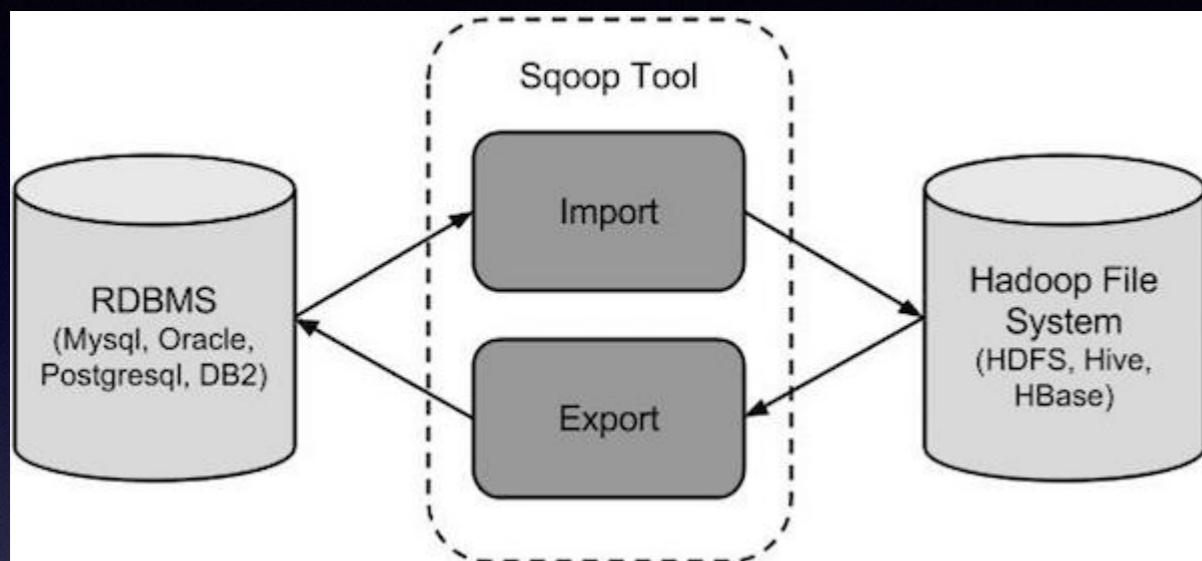
或者

```
sqoop import --connect jdbc:mysql://localhost/BASENAME --table tablename -m 1 --hive-import
```

## 与原Hive建表方法的比较

```
hive> CREATE TABLE sales(widget_id INT, qty INT,
> street STRING, city STRING, state STRING,
> zip INT, sale_date STRING)
> ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
hive> LOAD DATA LOCAL INPATH "ch15-sqoop/sales.log" INTO TABLE sales;
```

# 数据导出



sqoop数据导出做不到非常“自动”，它  
需要在mysql里创建相同架构的表

## (可选) 临时表

- Sqoop每隔几千行提交一次结果，以确保它不会耗尽内存。导出继续时，这些中间结果可见。在导出过程完成之前，不应启动将使用导出结果的应用程序，否则可能会只得到部分结果。
- 解决：--staging 导出到临时表
- 临时表必须已存在且与目标具有相同的架构。它也必须为空。或者使用 --clear-staging-table 选项。

# 其他用法

- sqoop作业：sqoop作业指定参数来识别并调用以保存的工作。主要在增量导入时使用。
- eval工具：sqoop直接查询数据库内库、表信息，不需要进入例如mysql环境。

# sqoop作业

```
创建作业 sqoop job --create myjob --import \
--connect jdbc:mysql://localhost/db \
--username root --table employee --m 1
```

```
查看作业 sqoop job —list
sqoop job --show myjob
```

```
执行作业 sqoop job --exec myjob
```

# Eval工具

```
sqoop eval \
--connect jdbc:mysql://localhost/db \
 --username root \
--query "SELECT * FROM employee LIMIT 3"
```

```
sqoop eval \
--connect jdbc:mysql://localhost/db \
 --username root \
-e "INSERT INTO employee VALUES(1207,'Raju','UI dev',15000,'TP')"
```

```
sqoop list-databases \
--connect jdbc:mysql://localhost/ \
 --username root
```

```
sqoop list-tables \
--connect jdbc:mysql://localhost/userdb \
 --username root
```

# sqoop使用实例

## 在mysql建立测试数据

```
create table illegals (
 id VARCHAR(40),
 number VARCHAR(40),
 plateNumber VARCHAR(40),
 engineNumber VARCHAR(40),
 action VARCHAR(100),
 place VARCHAR(100),
 date VARCHAR(40),
 dockPoint VARCHAR(20),
 penaltyAmount VARCHAR(40),
 status VARCHAR(20),
 awardNumber VARCHAR(40),
 certificateNumber VARCHAR(40),
 litigant VARCHAR(40),
 leeFee VARCHAR(20)
);
```

```
alter table illegals CONVERT TO CHARACTER SET utf8;

load data local infile '/Users/pingguo/Desktop/winstar/data/illegals/illegal_10000.csv'
 into table illegals CHARACTER SET 'utf8'
 fields terminated by ','
 lines terminated by '\n';
```

# sqoop使用实例

## 查看数据库

```
sqoop list-databases \
--connect jdbc:mysql://localhost/ \
--username root --password 123
```

```
rification.
information_schema
hive
mysql
performance_schema
sqoopTest
sys
root@master:/home/cgy# □
```

## 查看表

```
sqoop list-tables \
--connect jdbc:mysql://localhost/sqoopTest \
--username root --password 123
```

```
rification.
illegal
root@master:/home/cgy# □
```

# sqoop使用实例

## eval查看表内容

```
sqoop eval \
--connect jdbc:mysql://localhost/sqoopTest \
--username root --password 123 \
--query "SELECT number,place,litigant FROM illegals LIMIT 3"
```

| number        | place | litigant       |
|---------------|-------|----------------|
| "number":"610 | 1"    | "litigant":"王  |
| "number":"610 | 1"    | "litigant": "  |
| "number":"610 | 1"    | "litigant": "闫 |

root@master:/home/sqoop#

# sqoop使用实例

```
GRANT ALL PRIVILEGES ON sqoopTest.* TO 'root'@'localhost'
```

```
sqoop import \
--connect jdbc:mysql://localhost/sqoopTest \
--username root --password 123 \
--table illegals \
--target-dir /data/sqoopTest
```

```
File Input Format Counters
```

```
Bytes Read=0
```

```
File Output Format Counters
```

```
Bytes Written=3885188
```

```
2018-07-31 09:37:11,243 INFO mapreduce.ImportJobBase: Transferred 3.7052 MB in 27.8842 seconds (136.0675 KB/sec)
```

```
2018-07-31 09:37:11,248 INFO mapreduce.ImportJobBase: Retrieved 10000 records.
```

```
root supergroup 0 2018-07-31 09:37 /data/sqoopTest
hadoop fs -ls /data/sqoopTest
```

```
root supergroup 0 2018-07-31 09:37 /data/sqoopTest/_SUCCESS
root supergroup 3885188 2018-07-31 09:37 /data/sqoopTest/part-m-00000
```

# sqoop使用实例

## Sqoop到Hive

```
sqoop import --connect jdbc:mysql://master/sqoopTest --hive-database sqoopTest --table illegals --username root --password 123 -m 1 --hive-import
```

```
2018-07-31 10:33:01,553 INFO hive.HiveImport: Hive import complete.
2018-07-31 10:33:01,555 INFO hive.HiveImport: Export directory is contains the _SUCCESS file only, removing the director
```

```
hive>
 > select count(number) from illegals ;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a
 engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = root_20180731103410_93e8f0d8-f69a-4017-ba58-043128e5cff0
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
 set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
 set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
 set mapreduce.job.reduces=<number>
Starting Job = job_1533000929856_0009, Tracking URL = http://master:8088/proxy/application_1533000929856_0009/
Kill Command = /usr/hadoop/hadoop-3.0.3/bin/hadoop job -kill job_1533000929856_0009
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-07-31 10:34:23,777 Stage-1 map = 0%, reduce = 0%
2018-07-31 10:34:31,215 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.83 sec
2018-07-31 10:34:38,453 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.35 sec
MapReduce Total cumulative CPU time: 3 seconds 350 msec
Ended Job = job_1533000929856_0009
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.35 sec HDFS Read: 3894776 HDFS Write: 105 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 350 msec
OK
10000
Time taken: 29.793 seconds, Fetched: 1 row(s)
```

## sqoop使用实例

### Sqoop导出到Mysql

```
sqoop export --connect jdbc:mysql://master/sqoopTest --username root --password 123 -m 1
--table illegals2 --export-dir /data/sqoopTest/ -input-fields-terminated-by ","
```

```
sqoop export --connect jdbc:mysql://master/sqoopTest --username root --password 123 -m 1
--table illegals2 --export-dir /user/hive/warehouse/sqoopTest.db/illegals --input-fields-
terminated-by '\001'
```

```
2018-07-31 13:43:08,218 INFO mapreduce.ExportJobBase: Transferred 3.7053 MB in 20.9441 seconds (181.1607 KB/sec)
2018-07-31 13:43:08,222 INFO mapreduce.ExportJobBase: Exported 10000 records.
root@master:/usr/hadoop/hadoop-3.0.3/logs#
```

```
mysql> select count(id) from illegals2 ;
+-----+
| count(id) |
+-----+
| 10001 |
+-----+
1 row in set (0.00 sec)

mysql>
```