北风网 IBEIFENG.COM

大数据Hadoop高薪直通车课程

数据转换工具Sqoop

讲师：轩宇（北风网版权所有)

课程大纲

1. Sqoop 概述架构
2. Sqoop 使用要点
3. 导入数据HDFS
4. 导出数据RDBMS
5. Hive数据导入导出

课程大纲

1. Sqoop 概述架构
2. Sqoop 使用要点
3. 导入数据HDFS
4. 导出数据RDBMS
5. Hive数据导入导出

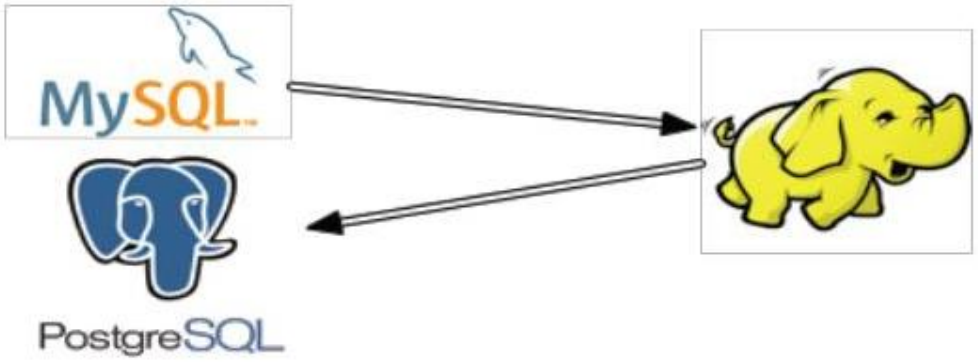
Apache Sqoop

Apache Sqoop(TM) is a tool designed for efficiently **transferring bulk data between Apache Hadoop and structured datastores** such as relational databases.

Apache Sqoop

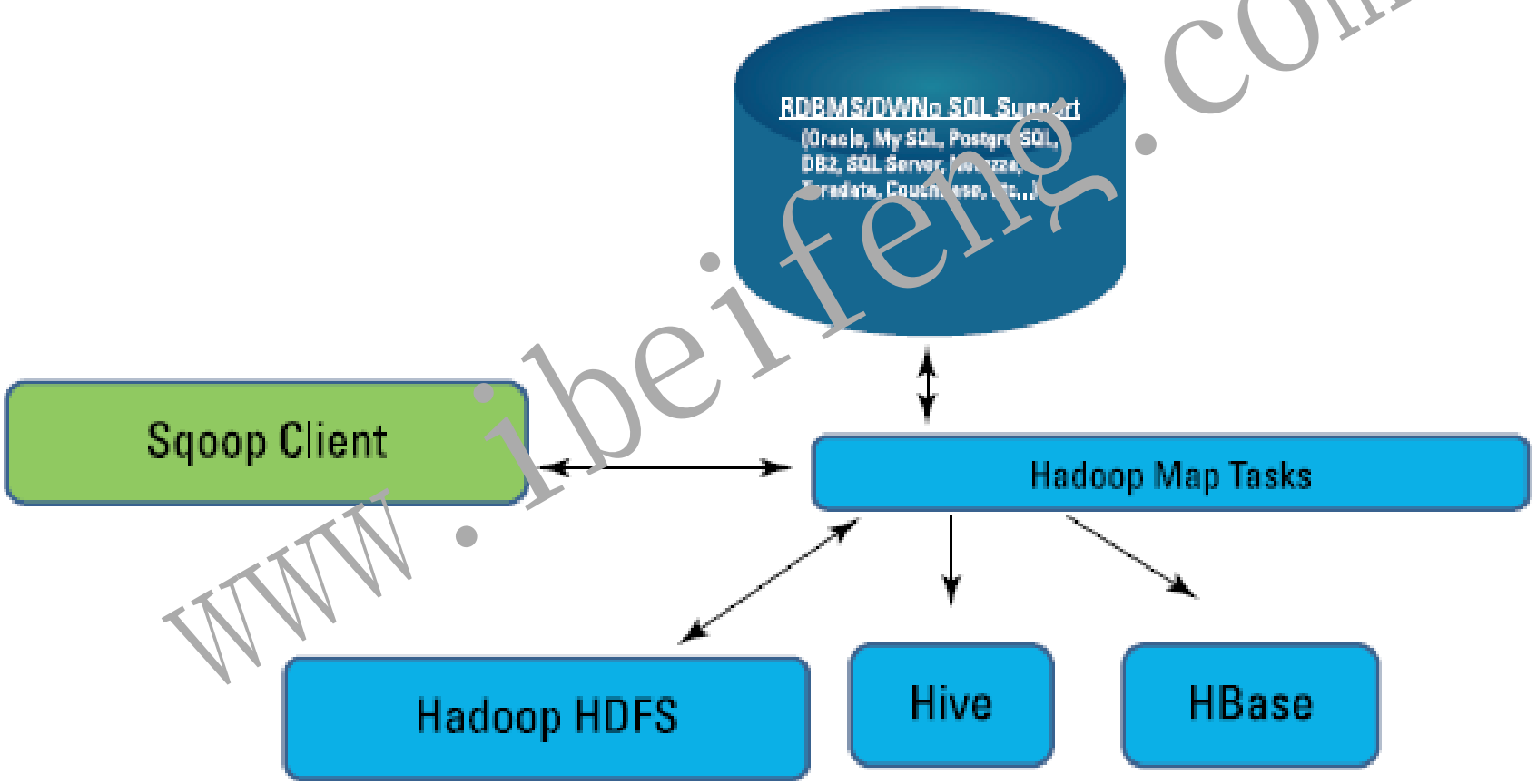
- Sqoop：SQL-to-Hadoop
- 连接传统关系型数据库和Hadoop的桥梁
  - 把关系型数据库的数据导入到Hadoop与其相关的系统(如HBase和Hive)中
  - 把数据从Hadoop系统里抽取并导出到关系型数据库里
- 利用MapReduce加快数据传输速度

批处理方式进行数据传输



Apache Sqoop

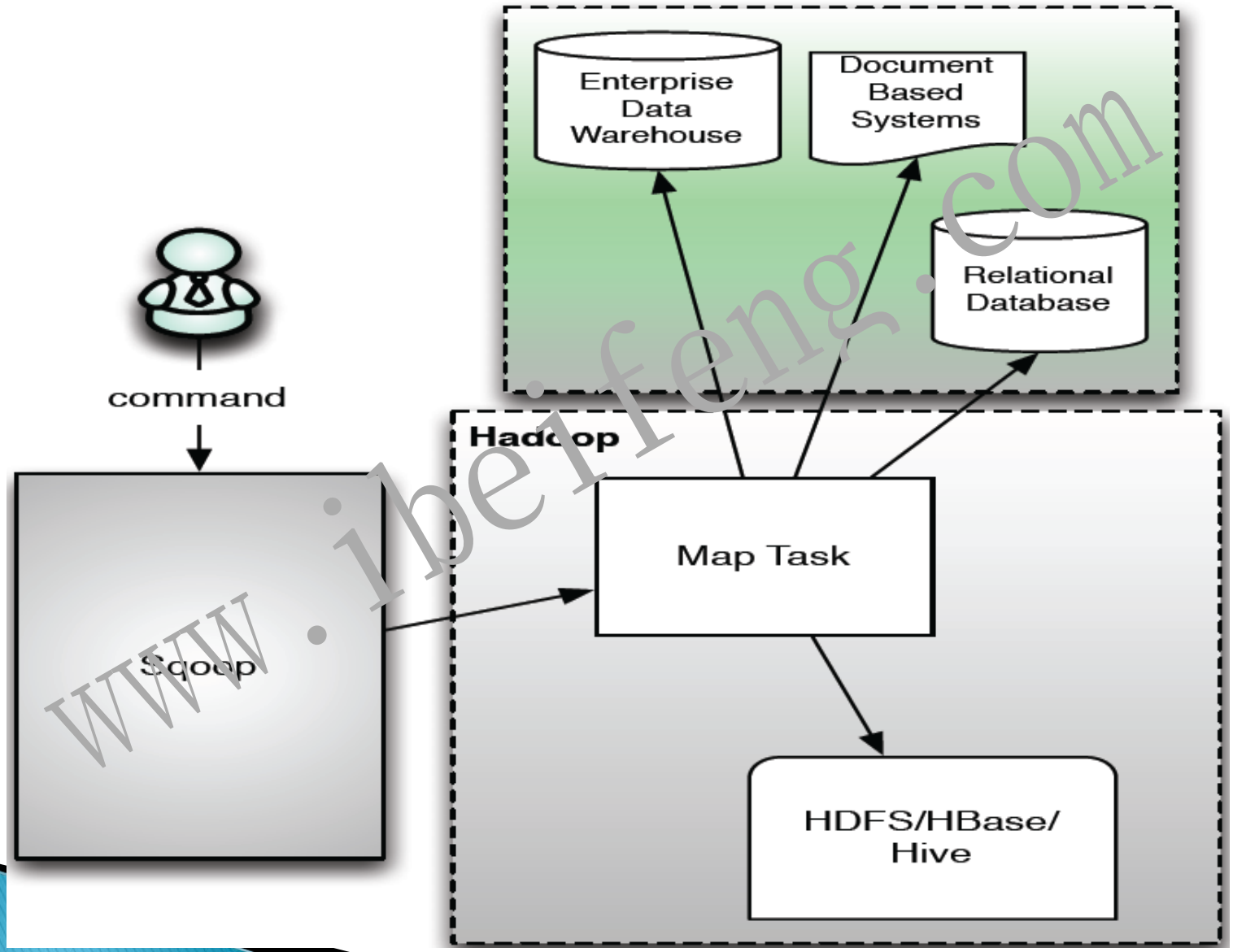
Sqoop Design



Sqoop1 & Sqoop2

- 两个不同版本，完全不兼容
- 版本号划分方式
  - Apache：1.4.x~ ，1.99.x~
- Sqoop2比Sqoop1的改进
  - 引入sqoop server，集中化管理Connector等
  - 多种访问方式：CLI，Web UI，REST API
  - 引入基于角色的安全机制

Sqoop1 架构



Sqoop1 架构

- You can use Sqoop to import data from a relational database management system (RDBMS) such as MySQL or Oracle into the Hadoop Distributed File System (HDFS), transform the data in Hadoop MapReduce, and then export the data back into an RDBMS.
- Sqoop automates most of this process, relying on the database to describe the schema for the data to be imported. Sqoop uses MapReduce to import and export the data, which provides **parallel operation as well as fault tolerance**.

课程大纲

1. Sqoop 概述架构
2. Sqoop 使用要点
3. 导入数据HDFS
4. 导出数据RDBMS
5. Hive数据导入导出

Sqoop Installation

- Download

  http://archive.apache.org/dist/sqoop

- SetUp

```
$ HADOOP_COMMON_HOME=/path/to/some/hadoop \
  HADOOP_MAPRED_HOME=/path/to/some/hadoop-mapreduce \
  sqoop import --arguments...
```

or:

```
$ export HADOOP_COMMON_HOME=/some/path/to/hadoop
$ export HADOOP_MAPRED_HOME=/some/path/to/hadoop-mapreduce
$ sqoop import --arguments...
```

SQOOP HELP

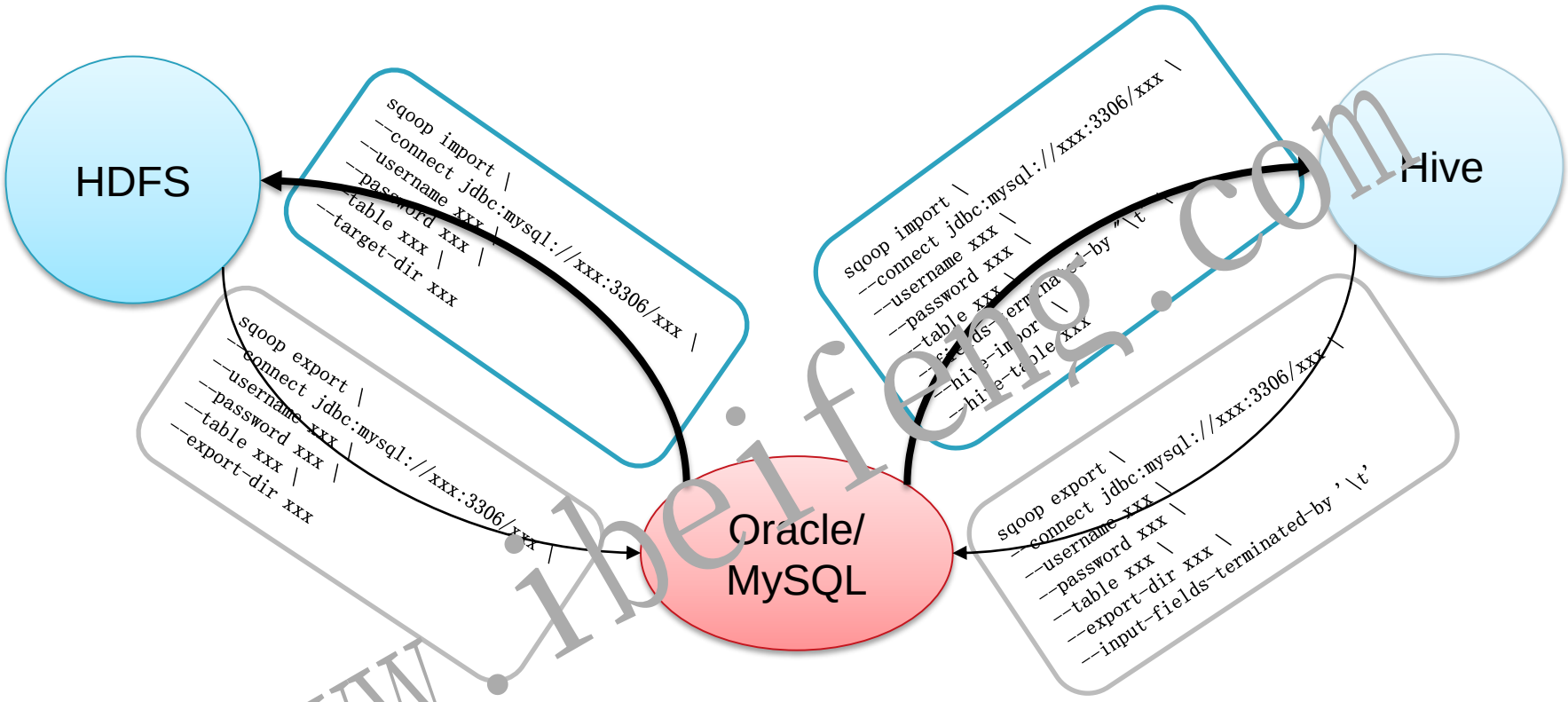
```
[hadoop@master ~]$ sqoop help
find: paths must precede expression
Usage: find [-H] [-L] [-P] [path...] [expression]
Warning: $HADOOP_HOME is deprecated.

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.
```

Sqoop 使用要点



RDBMS：
1）jdbcurl
2）username
3）password
4）tablename

WAYS：
1）import
2）export

HADOOP：
1）hdfs：
path
2）hive
tablename

Sqoop Installation

◆ 进行测试连接

在【SQOOP_HOME】目录下执行如下命令来显示192.168.191.4服务器上的所有数据库：

```
bin/sqoop list-databases \
--connect jdbc:mysql://192.168.191.4:3306 \
--username root \
--password pass123
```

```
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
15/04/10 04:45:16 INFO sqoop.Sqoop: Running Sqoop version: 1.4.5-cdh5.2.0
15/04/10 04:45:16 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
15/04/10 04:45:17 INFO manager.SqlManager: Using default fetchSize of 1000
information_schema
dimensoft
mysql
performance_schema
test
[hadoop@hadoop-main sqoop-1.4.5]$
```

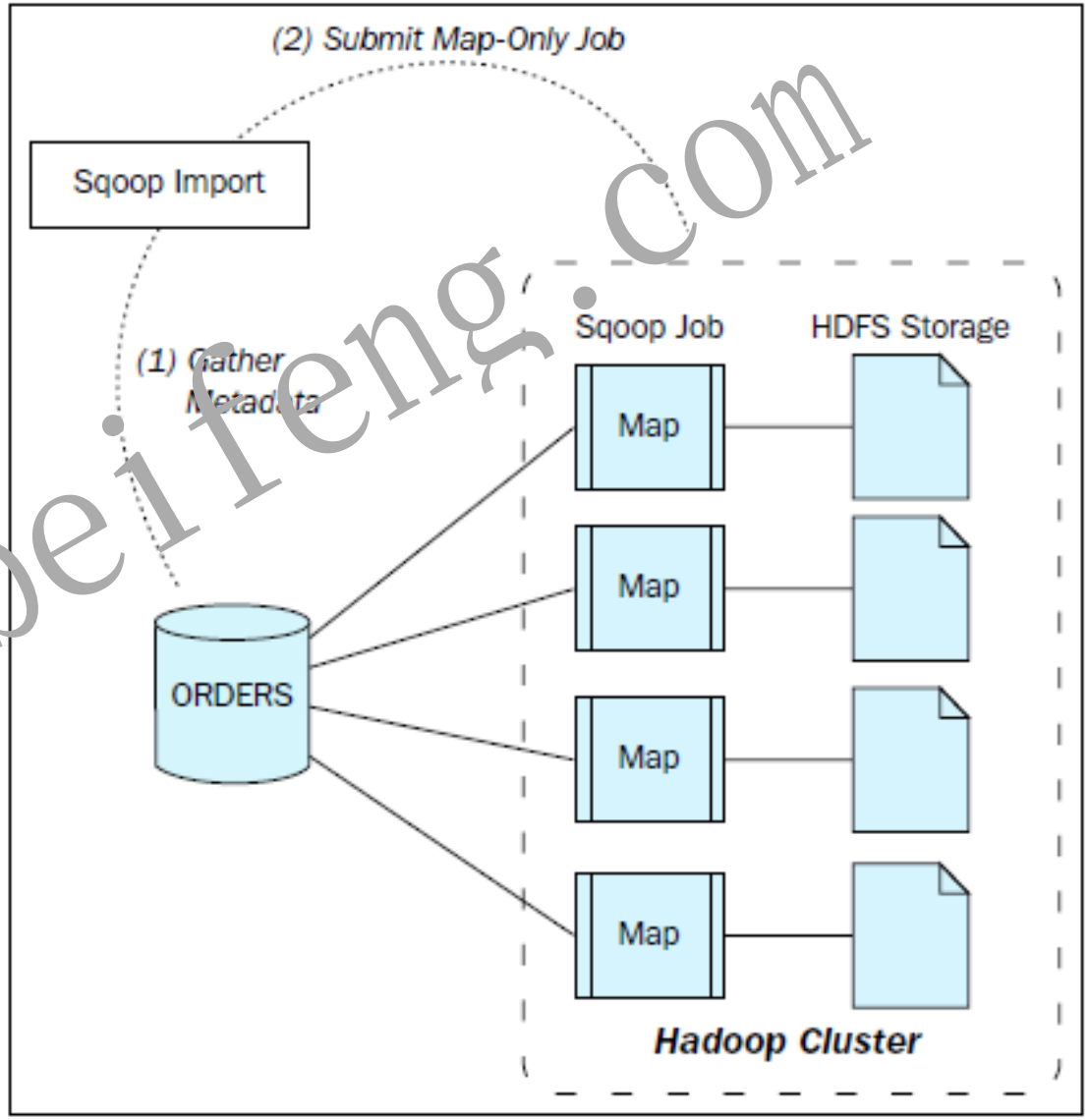
课程大纲

1. Sqoop 概述架构
2. Sqoop 使用要点
3. 导入数据HDFS
4. 导出数据RDBMS
5. Hive数据导入导出

Imports

Sqoop import is executed in two steps:

1. Gather metadata
2. Submit map only job



Sqoop import

- Import an entire table:

```
sqoop import \
--connect jdbc:mysql://mysql.example.com/sqoop \
--username sqoop \
--password sqoop \
--table cities
```

- Import a subset of data:

```
sqoop import \
--connect jdbc:mysql://mysql.example.com/sqoop \
--username sqoop \
--password sqoop \
--table cities \
--where "country = 'USA'"
```

- Change file format, by default the data will be saved in tab separated csv format but Sqoop provides option for saving the data in Hadoop SequenceFile, Avro binary format and Parquet file:

```
sqoop import \
--connect jdbc:mysql://mysql.example.com/sqoop \
--username sqoop \
--password sqoop \
--table cities \
--as-sequencefile
```

```
sqoop import \
--connect jdbc:mysql://mysql.example.com/sqoop \
--username sqoop \
--password sqoop \
--table cities \
--as-avrodatafile
```

Sqoop import

- Compressing imported data:

```
sqoop import \
--connect jdbc:mysql://mysql.example.com/sqoop
--username sqoop \
--table cities \
--compress \
--compression-codec org.apache.hadoop.io.compress.BZip2Codec
```

- Bulk import:

```
sqoop import \
--connect jdbc:mysql://mysql.example.com/sqoop \
--username sqoop \
--table cities \
--direct
```

Sqoop import

- Incremental import:

```
sqoop import \
--connect jdbc:mysql://mysql.example.com/sqoop \
--username sqoop \
--password sqoop \
--table visits \
--incremental append \
--check-column id \
--last-value 1
```

- Free form query import:

```
sqoop import \
--connect jdbc:mysql://mysql.example.com/sqoop \
--username sqoop \
--password sqoop \
--query 'SELECT normcities.id, \
    countries.country, \
    normcities.city \
    FROM normcities \
    JOIN countries USING(country_id) \
    WHERE $CONDITIONS' \
--split-by id \
--target-dir cities
```

Sqoop import

- Custom boundary query import:

```
sqoop import \
--connect jdbc:mysql://mysql.example.com/sqoop \
--username sqoop \
--password sqoop \
--query 'SELECT normcities.id, \
countries.country, \
normcities.city \
FROM normcities \
JOIN countries USING(country_id) \
WHERE $CONDITIONS' \
--split-by id \
--target-dir cities \
--boundary-query "select min(id), max(id) from normcities"
```

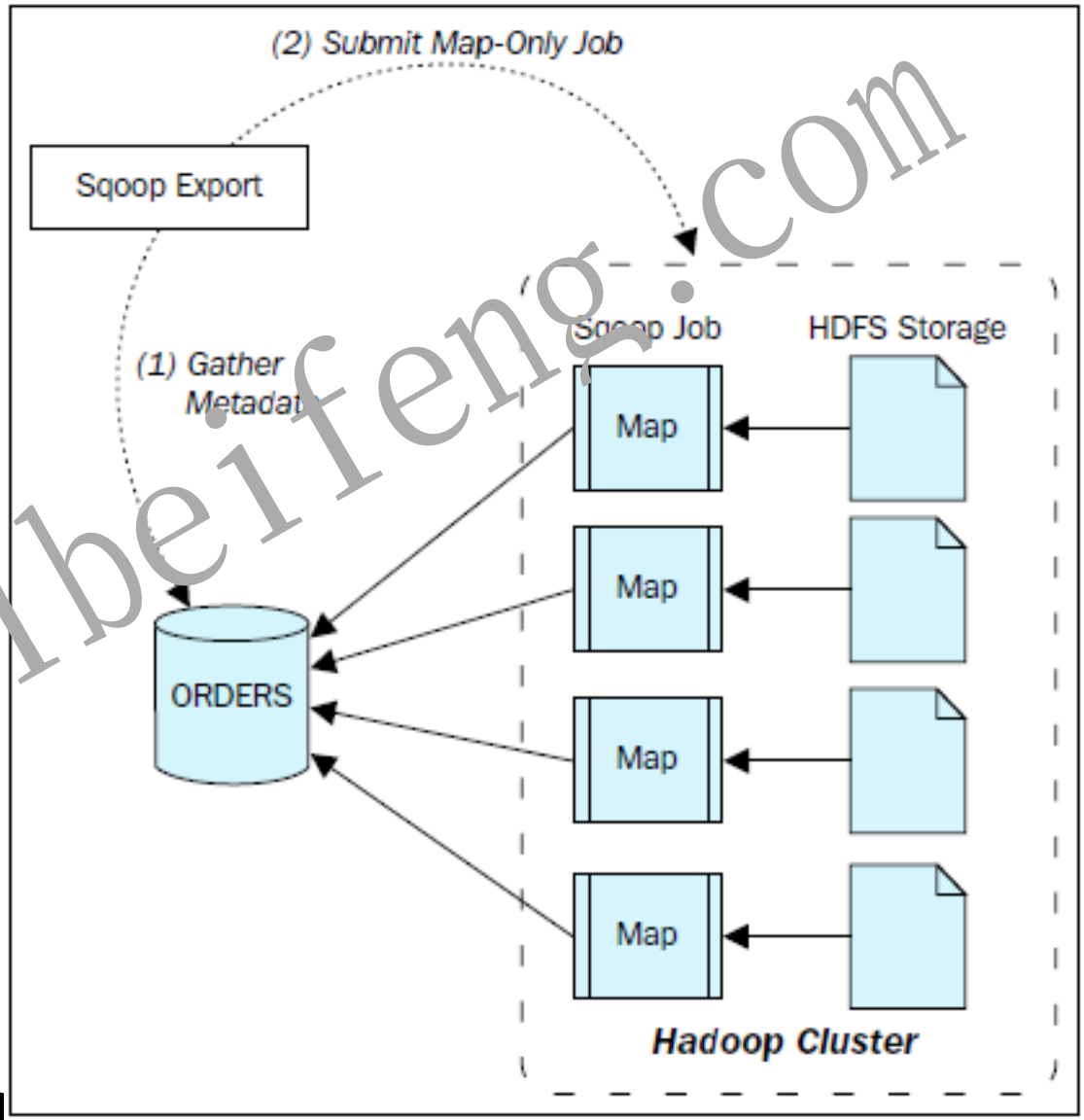
课程大纲

1. Sqoop 概述架构
2. Sqoop 使用要点
3. 导入数据HDFS
4. 导出数据RDBMS
5. Hive数据导入导出

Exports

Sqoop Export is also in a similar process, only the source will be HDFS. Export is performed in two steps;

- Gather metadata
- Submit map-only job



Sqoop Export

- Exporting files from under the HDFS directory to a table:

```
sqoop export \
--connect jdbc:mysql://mysql.example.com/sqoop \
--username sqoop \
--password sqoop \
--table cities \
--export-dir cities
```

- Batch inserts export:

```
sqoop export \
--connect jdbc:mysql://mysql.example.com/sqoop \
--username sqoop \
--password sqoop \
--table cities \
--export-dir cities \
--batch
```

- Updating existing dataset:

```
sqoop export \
--connect jdbc:mysql://mysql.example.com/sqoop \
--username sqoop \
--password sqoop \
--table cities \
--update-key id
```

Sqoop Export

- Upsert export:

```
sqoop export \
--connect jdbc:mysql://mysql.example.com/sqoop \
--username sqoop \
--password sqoop \
--table cities \
--update-key id \
--update-mode allowinsert
```

- Column export:

```
sqoop export
--connect jdbc:mysql://mysql.example.com/sqoop \
--username sqoop \
--password sqoop \
--table cities \
--columns country,city
```

课程大纲

1. Sqoop 概述架构
2. Sqoop 使用要点
3. 导入数据HDFS
4. 导出数据RDBMS
5. Hive数据导入导出

Import Hive

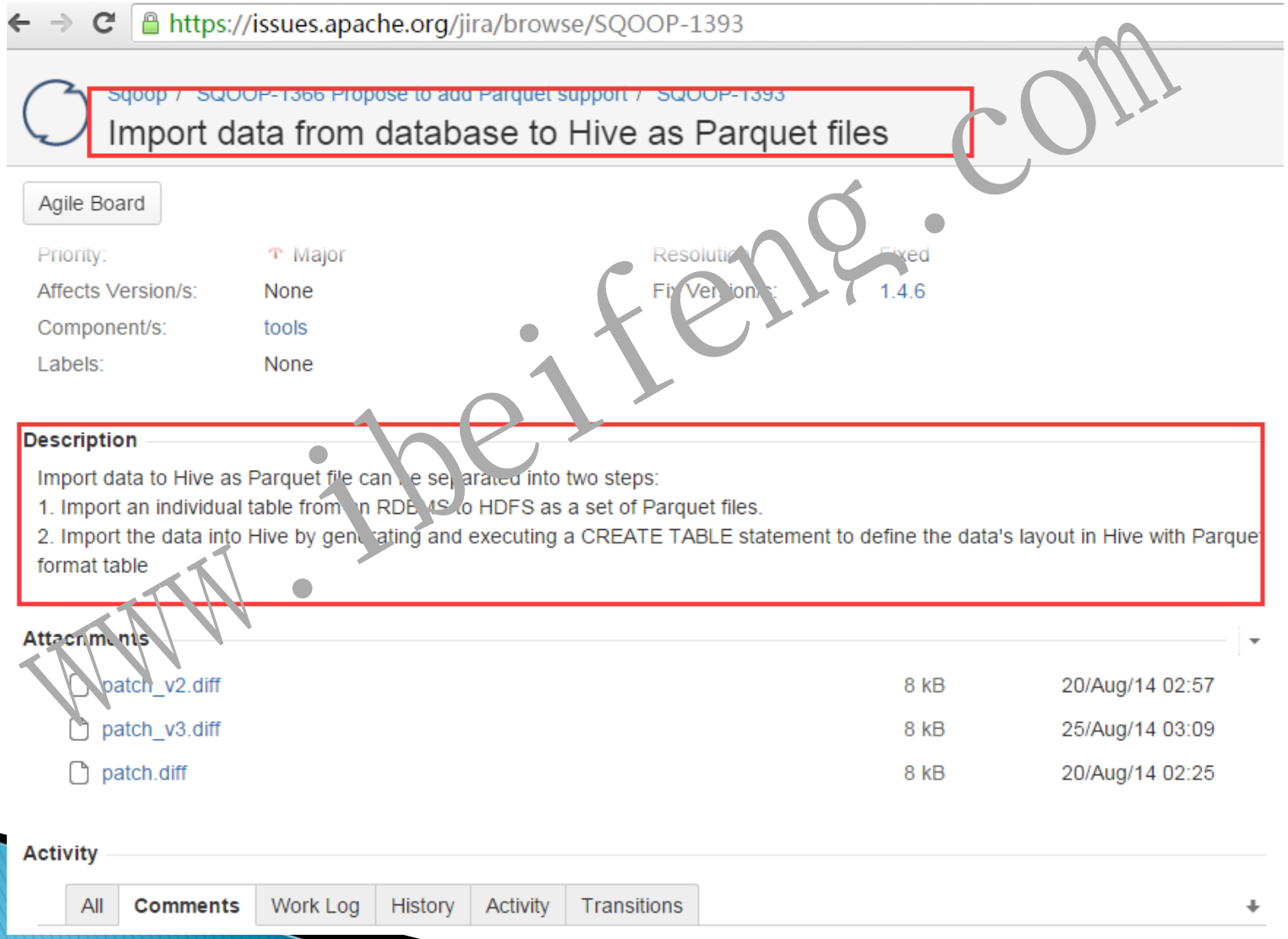
--create-hive-table	Fail if the target hive table exists
--hive-database <database-name>	Sets the database name to use when importing to hive
--hive-delims-replacement <arg>	Replace Hive record \0x01 and row delimiters (\n\r) from imported string fields with user-defined string
--hive-drop-import-delims	Drop Hive record \0x01 and row delimiters (\n\r) from imported string fields
--hive-home <dir>	Override $HIVE_HOME
--hive-import	Import tables into Hive (Uses Hive's default delimiters if none are set.)

Import Hive

--hive-overwrite	Overwrite existing data in the Hive table
--hive-partition-key <partition-key>	Sets the partition key to use when importing to hive
--hive-partition-value <partition-value>	Sets the partition value to use when importing to hive
--hive-table <table-name>	Sets the table name to use when importing to hive
--map-column-hive <arg>	Override mapping for specific column to hive types.

SQOOP-1393

https://issues.apache.org/jira/browse/SQOOP-1393



Export Hive

- 在Mysql中准备如下表

```
CREATE DATABASE tag_db;
CREATE TABLE tag_db.users (
  id INT(11) NOT NULL,
  name VARCHAR(100) NOT NULL,
  PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE tag_db.tags (
  id INT(11) NOT NULL,
  user_id INT NOT NULL,
  tag VARCHAR(100) NOT NULL,
  PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

- 同时对应Hive中如下表

```
CREATE TABLE users (
  id INT,
  name STRING
) row format delimited fields terminated by '\t';

CREATE TABLE tags (
  id INT,
  user_id INT,
  tag STRING
) row format delimited fields terminated by '\t';
```

Export Hive

- 在Hive中准备数据

```
load data local inpath '/home/hadoop/dataset/users.txt' overwrite into table users;
--users.txt
1       jeffery
2       shirdrn
3       sulee

load data local inpath '/home/hadoop/dataset/tags.txt' overwrite into table tags;
--tags.txt
1 1 Music
2 1 Programming
3 2 Travel
4 3 Sport
```

- 执行导出

```
sqoop export --connect jdbc:mysql://192.168.136.103:3306/tag_db \
--username mysql --P --table users --export-dir /user/hive/warehouse/users \
--input-fields-terminated-by '\t' -- --default-character-set=utf-8

sqoop export --connect jdbc:mysql://192.168.136.103:3306/tag_db \
--username mysql --P --table tags --export-dir /user/hive/warehouse/tags \
--input-fields-terminated-by '\t' -m 1
```

Using Options Files to Pass Arguments

To specify an options file, simply create an options file in a convenient location and pass it to the command line via `--options-file` argument.

Whenever an options file is specified, it is expanded on the command line before the tool is invoked. You can specify more than one option files within the same invocation if needed.

For example, the following Sqoop invocation for import can be specified alternatively as shown below:

```
$ sqoop import --connect jdbc:mysql://localhost/db --username foo --table TEST

$ sqoop --options-file /users/homer/work/import.txt --table TEST
```

where the options file `/users/homer/work/import.txt` contains the following:

```
import
--connect
jdbc:mysql://localhost/db
--username
foo
```

Using Options Files to Pass Arguments

The options file can have empty lines and comments for readability purposes. So the above example would work exactly the same if the options file `/users/homer/work/import.txt` contained the following:

```
#
# Options file for Sqoop import
#

# Specifies the tool being invoked
import

# Connect parameter and value
--connect
jdbc:mysql://localhost/db

# Username parameter and value
--username
foo

#
# Remaining options should be specified in the command line.
#
```

本课程版权归北风网所有

欢迎访问我们的官方网站

www.ibeifeng.com