**hbase数据导入与导出**

使用importTsv将csv文件导入HBase；

其格式如下：

hbase [类] [分隔符] [行键，列族] [表] [导入文件]

hbase org. apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator=","

-Dimporttsv.columns=HBASE\_ROW\_KEY,f1 test /user/f.csv

使用import功能，将数据导入HBase；

使用import功能进行数据导入，导入的文件必须是sequence文件。与import相对的还有export，export导出的文件为sequence文件。

首先使用export将HBases表中的数据导出：

hbase org.apache.hadoop.hbase.mapreduce.Export test /user/hbase/output

再用import将数据导入

hbase org.apache.hadoop.hbase.mapreduce.Import test2 /user/hbase/output

**简单使用kafka**

开启zookeeper服务：

zkServer.sh start

启动kafka服务：  
bin/kafka-server-start.sh config/server.properties &

再开一个终端：

调用/kafka/bin目录下kafka-topic.sh脚本创建一个topic名字为：。sayaword

bin/kafka-topics.sh \

--create \

--zookeeper localhost:2181 \

--replication-factor 1 \

--topic sayaword \

--partitions 1

调用/kafka/bin目录下kafka-console-producer.sh，来生产一些消息，producer也就是生产者。用来生产数据。

bin/kafka-console-producer.sh \

--broker-list localhost:9092 \

--topic sayaword

再开一个终端：

调用bin目录下kafka-console-consumer.sh，启动consumer，consumer作为消费者，用来消费数据。

bin/kafka-console-consumer.sh \

--zookeeper localhost:2181 \

--topic sayaword \

--from-beginning

**简单使用Storm**

开启服务：

以后台形式开启dev-zookeeper，并将标准输出和错误信息重定向到zk.out中。

./bin/storm dev-zookeeper >> logs/zk.out 2>&1 &

以后台形式开启nimbus，并将标准输出和错误信息重定向到nimbus.out中。

./bin/storm nimbus >> logs/nimbus.out 2>&1 &

以后台形式开启ui，并将标准输出和错误信息重定向到ui.out中。

./bin/storm ui >> logs/ui.out 2>&1 &

以后台形式开启supervisor，并将标准输出和错误信息重定向到supervisor.out中。

./bin/storm supervisor >> logs/supervisor.out 2>&1 &

以后台形式开启logviewer，并将标准输出和错误信息重定向到logviewer.out中。

./bin/storm logviewer >> logs/logviewer.out 2>&1 &

使用jar命令，提交一个名为wordcount的topology。

./bin/storm

jar examples/storm-starter/storm-starter-topologies-0.9.4.jar storm.starter.WordCountTopology wordcount

查看当前Topology的运行状态。

./bin/storm list

使用deactivate命令，将名为wordcount的Topology状态改为INACTIVE(非活跃)。

./bin/storm deactivate wordcount

使用rebalance命令，该命令用于将指定Topology的Worker均匀分散到集群各节点

./bin/storm rebalance wordcount

**简单使用Sqoop**

**使用Sqoop将Mysql中的数据导入到HDFS:**

sqoop import \

--connect jdbc:mysql://localhost:3306/mydb \

--username root \

--password strongs \

--table record -m 1 \

--target-dir /mysqoop2

**将HDFS中数据存入到Mysql数据库中：**

sqoop export \

--connect jdbc:mysql://localhost:3306/mydb?characterEncoding=UTF-8

--username root \

--password strongs \

--table recordfromhdfs \

--export-dir hdfs://localhost:9000/mysqoop2/part-m-00000

**将Mysql中数据导入到HBase中：**

sqoop import \

--connect jdbc:mysql://localhost:3306/mydb?characterEncoding=UTF-8

--username root \

--password strongs \

--table record \

--hbase-create-table \

--hbase-table hbaserecord \

--column-family mycf \

--hbase-row-key dt -m 1

**使用Sqoop将Mysql中record表中的数据，导入到Hive中的hiverecord表中。**

**使用hive必须在环境变量里添加：**

export HADOOP\_CLASSPATH=$HADOOP\_CLASSPATH:/apps/hive/lib/\*

**开始导入：**

sqoop import \

--connect jdbc:mysql://localhost:3306/mydb?characterEncoding=UTF-8

--username root \

--password strongs \

--table record \

--hive-import \

--hive-table hiverecord \

--fields-terminated-by ',' -m 1

**使用Sqoop将Hive表hiverecord表中的数据，导出到Mysql中的recordfromhive表中。**

sqoop export \

--connect jdbc:mysql://localhost:3306/mydb?characterEncoding=UTF-8

--username root \

--password strongs \

--table recordfromhive \

--export-dir /user/hive/warehouse/hiverecord/part-m-00000 \

--input-fields-terminated-by ','

**Sqoop增量导入:**

sqoop import

--connect jdbc:mysql://localhost:3306/sqoop

--username root

--password strongs

--table buyer -m 1

--incremental append

--last-value 10005

--check-column buyer\_id

--target-dir /mysqoop/111

**Flume监听端口**

Sources配置：

agent.sources.src.type=syslogtcp

agent.sources.src.port=6868

agent.sources.src.host=localhost

向端口发送数据：

echo “hello world” | nc localhost 6868