数据集成作业2路线2

> mjh
>
> 注：文档中部分命令省略了sudo，需要时自行添加；文档中存在部分错误，请使用者根据需要调整。

A.数据库表部分技术文档

1 所需工具下载

- Linux系统
  - 这里采用的方法是在虚拟机上安装CentOS7
- MySQL 8.0.23
  - 这里采用的方法是yum下载，需要更新yum库
  - 下载链接：https://box.nju.edu.cn/f/a67e86b0623f43d9895a/?dl=1
- Java 1.8.281
  - 下载链接：https://box.nju.edu.cn/f/7d4057cc1c1b47f2820f/?dl=1
- Hadoop 3.1.3
  - 下载链接：https://box.nju.edu.cn/f/86548120773043959979/?dl=1
- Hive 3.1.2
  - 下载链接：https://box.nju.edu.cn/f/ac0170fa2e73485e812d/?dl=1
- Sqoop 1.4.7
  - 下载链接：https://box.nju.edu.cn/f/9c0fef4268b9423594db/?dl=1
- MySQL connector 8.0.23
  - 下载链接：https://box.nju.edu.cn/f/67bedd97e39941ecb01f/?dl=1

2 安装mysql

本地安装yum源

```
yum localinstall mysql80-community-release-el7-3.noarch.rpm
```

查看yum库

```
yum repolist enabled | grep "mysql.*-community.*"
```

安装mysql

```
yum install -y mysql-community-server
```

启动mysql

```
systemctl start mysqld
```

查看mysql

```
systemctl status mysqld
```

/var/log/mysqld.log下找到root默认密码

```
grep 'temporary password' /var/log/mysqld.log
```

修改或不修改密码后，进入mysql

```
mysql -u root -p
```

3 安装java

解压安装包到/usr/local，改名为java并修改用户权限

```
tar -zxvf jdk-8u281-linux-x64.tar.gz -C /usr/local
mv jdk-8u281-linux-x64 java
cd /usr/local
chown -R mjh:mjh java
```

修改~/.bashrc文件，并验证java是否安装成功

```
export JAVA_HOME=/usr/local/java
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JRE_HOME}/lib/rt.jar:${JAVA_HOME}/lib/dt.jar:${JAVA_HOME}/lib/tools.jar
export HADOOP_HOME=/usr/local/hadoop
export HIVE_HOME=/usr/local/hive
export SQOOP_HOME=/usr/local/sqoop
export PATH=${JAVA_HOME}/bin:${PATH}
```

```
source ~/.bashrc
```

```
java -version
javac -version
```

4 安装hadoop

解压安装包到/usr/local，改名为hadoop并修改用户权限

```
tar -zxvf hadoop-3.1.3.tar.gz -C /usr/local
mv hadoop-3.1.3 hadoop
cd /usr/local
chown -R mjh:mjh hadoop
```

配置$HADOOP_HOME/etc/hadoop/core-site.xml

```
<configuration>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>file:/usr/local/hadoop/tmp</value>
        <description>Abase for other temporary directories.</description>
    </property>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

配置$HADOOP_HOME/etc/hadoop/hdfs-site.xml

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/usr/local/hadoop/tmp/dfs/name</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:/usr/local/hadoop/tmp/dfs/data</value>
    </property>
</configuration>
```

$HADOOP_HOME/etc/hadoop/hadoop-env.sh

$HADOOP_HOME/etc/hadoop/mapred-env.sh

$HADOOP_HOME/etc/hadoop/yarn-env.sh

```
export JAVA_HOME=/usr/local/java
```

配置ssh公钥

```
ssh-keygen -t rsa
cat .ssh/id_rsa.pub >> .ssh/authorized_keys
chmod 700 .ssh
chmod 600 .ssh/authorized_keys
```

执行NamNode的格式化（只需执行一次）

启动hadoop

查看启动后

> datanode
>
> namenode
>
> secondarynamenode
>
> nodemanager
>
> resoucemanager

```
cd $HADOOP_HOME
bin/hdfs namenode -format
sbin/start-all.sh
jps
```

5 安装hive

解压安装包到/usr/local，改名为hive并修改用户权限

```
tar -zxvf apache-hive-3.1.2-bin.tar.gz -C /usr/local
mv apache-hive-3.1.2-bin hive
cd /usr/local
chown -R mjh:mjh hive
```

配置$HIVE_HOME/bin/hive-config.sh

```
export JAVA_HOME=/usr/local/java
export HADOOP_HOME=/usr/local/hadoop
export HIVE_HOME=/usr/local/hive
```

配置$HIVE_HOME/conf/hive-env.sh

```
cd $HIVE_HOME
cp hive-env.sh.template hive-env.sh
```

```
export HADOOP_HOME=/usr/local/hadoop
```

配置$HIVE_HOME/conf/hive-site.xml

```
cd $HIVE_HOME
cp hive-default.xml.template hive-site.xml
```

```
<!--mysql相关配置 -->
<property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://localhost:3306/hive?createDatabaseIfNotExist=true</value>
    <description>JDBC connect string for a JDBC metastore</description>
</property>
<property>
```

```
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
    <description>Driver class name for a JDBC metastore</description>
</property>
<property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>root</value>
    <description>Username to use against metastore database</description>
</property>
<property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <!--这里是mysql的密码-->
    <value>password</value>
    <description>password to use against metastore database</description>
</property>
<!--指定资源目录和日志目录，下面需要依次创建这些目录 -->
<property>
    <name>hive.exec.local.scratchdir</name>
    <value>/usr/local/hive/scratchdir</value>
    <description>Local scratch space for Hive jobs</description>
</property>
<property>
    <name>hive.downloaded.resources.dir</name>
    <value>/usr/local/hive/resourcesdir</value>
    <description>Temporary local directory for added resources in the remote file system.
</description>
</property>
<property>
    <name>hive.querylog.location</name>
    <value>/usr/local/hive/querylog</value>
    <description>Location of Hive run time structured log file</description>
</property>
<property>
    <name>hive.server2.logging.operation.log.location</name>
    <value>/usr/local/hive/operation_logs</value>
    <description>Top level directory where operation logs are stored if logging functionality is
enabled</description>
</property>
```

创建上文中相关目录

...

复制mysql驱动包到$HIVE/lib下

```
tar -zxvf mysql-connector-java-8.0.23.tar.gz
cd mysql-connector-java-8.0.23
cp mysql-connector-java-8.0.23.jar $HIVE_HOME/lib
```

初始化mysql元数据

```
cd $HIVE_HOME
bin/schematool -dbType mysql -initSchema --verbose
```

启动

```
cd $HIVE_HOME
bin/hive
```

初始化或启动过程中出现异常参见文章https://zhuanlan.zhihu.com/p/98658806

6 hiveserver2连接并导出至hdfs文件系统

配置$HIVE_HOME/conf/hive-site.xml

```
<!-- 这是hiveserver2 -->
    <property>
             <name>hive.server2.thrift.port</name>
             <value>10000</value>
    </property>

        <property>
            <name>hive.server2.thrift.bind.host</name>
            <value>172.29.4.17</value>
        </property>
```

配置$HADOOP_HOME/etc/hadoop/cote-site.xml

```
<property>
    <name>hadoop.proxyuser.root.hosts</name>
    <value>*</value>
 </property>
<property>
    <name>hadoop.proxyuser.root.groups</name>
    <value>*</value>
</property>
<property>
    <name>hadoop.proxyuser.zhaoshb.hosts</name>
    <value>*</value>
</property>
<property>
    <name>hadoop.proxyuser.zhaoshb.groups</name>
    <value>*</value>
</property>
```

启动hiveserver2

```
cd $HIVE_HOME
bin/hive --service hiveserver2
```

新开一个终端，启动beeline，输入用户密码，连接成功

```
cd $HIVE_HOME
bin/beeline
```

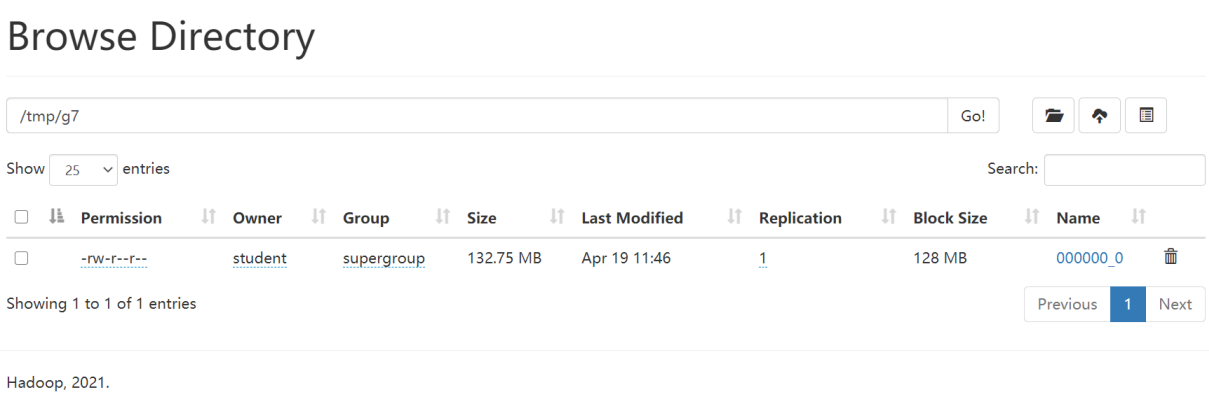
```
beeline> !connect jdbc:hive2://172.29.4.17:10000
Enter username for jdbc:hive2://172.29.4.17:10000: student
Enter password for jdbc:hive2://172.29.4.17:10000: nju2021
```

```
0: jdbc:hive2://172.29.4.17:10000>
```

导入hdfs

```
0: jdbc:hive2://172.29.4.17:10000> insert overwrite directory "/tmp/g7"
0: jdbc:hive2://172.29.4.17:10000> select * from buy_data;
```

查看内容



7 安装sqoop

解压安装包到/usr/local，改名为sqoop并修改用户权限

```
tar -zxvf sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz -C /usr/local
mv sqoop-1.4.7.bin__hadoop-2.6.0 sqoop
cd /usr/local
chown -R mjh:mjh sqoop
```

配置$SQOOP_HOME/conf/sqoop-env.sh

```
cd $SQOOP_HOME
cp sqoop-env-template.sh sqoop-env.sh
```

```
export HADOOP_COMMON_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=/usr/local/hadoop/share/hadoop/mapreduce
export HIVE_HOME=/usr/local/hive
```

将$HIVE_HOME/lib和$HADOOP_HOME/share/hadoop/mapreduce中的jar包复制到$SQOOP_HOME/lib下

```
cp -r $HIVE_HOME/lib/*.jar $SQOOP_HOME/lib
cp -r $HADOOP_HOME/share/hadoop/mapreduce/*.jar $SQOOP_HOME/lib
```

验证安装成功

```
cd SQOOP_HOME/bin
sqoop version
```

8 数据导入本地mysql

本地新建hive数据库，并新建buy_data表

```
create database hive;
use hive;
CREATE TABLE `buy_data` (
  `id` bigint DEFAULT NULL,
  `user_id` int DEFAULT NULL,
  `item_id` int DEFAULT NULL,
  `category_id` int DEFAULT NULL,
  `type` varchar(32) DEFAULT NULL,
  `timestamp` int DEFAULT NULL
);
```

sqoop导入

```
bin/sqoop export --connect jdbc:mysql://localhost:3306/hive --table buy_data --username root --
password password --export-dir hdfs://172.29.4.17:9000/tmp/buy_gen/part-m-00000
```

如报错java.lang.ClassNotFoundException，大概率需要复制/tmp/sqoop-mjh/complie/(乱码)/下的buy_data.jar，该文件为编译中生成

再次运行sqoop导入，即可在mysql中查看导入的buy_data表内容

B.流数据技术文档

Kafka相关代码，将consumer接受到的数据转储到mongodb中

```
from kafka import KafkaConsumer
import pymongo

consumer = KafkaConsumer(
    'foobar',
    bootstrap_servers='172.29.4.17:9092',
    security_protocol='SASL_PLAINTEXT',
    sasl_mechanism='PLAIN',
    sasl_plain_username='student',
    sasl_plain_password='nju2021',
)

# 多个 consumer 可以重复消费相同的日志，每个 consumer 只会消费到它启动后产生的日志，不会拉到之前的余量
dataClient = pymongo.MongoClient(host="localhost:27017", username="root", password="hyzyj2007")
db = dataClient['dataIntegration']
collection = db['robots']

for msg in consumer:
    line = msg.value.decode("utf-8")
    collection.insert_one({"value": line})
```