zookeeper-3.4.10:

設置文件conf/zoo.cfg dataDir=/hadoop/tmp/zookeeper

//zookeeper 集群：

在datadir下添加myid 檔，寫入zookeeper集群的主機編號，只需單個序號數字

zoo.cfg添加

server.0=127.0.0.1:8080:7770

server.1=127.0.0.1:8081:7771

server.2=127.0.0.1:8082:7772

啓動zookeeper命令:zkServer.sh start

監控zookeeper命令:zkCli.sh

kafka-1.1.0:

設置文件server.properties log.dirs=c:/tmp/kafka-logs

//kafka集群：

設置broker.id,在zkcli.sh 中使用 ls /brokers/ids 查看集群的brokers數量，實現一個product 多個consumer 同時接收

zookeeper.connect=192.168.159.132:2181

listeners=PLAINTEXT://192.168.159.132:9092

啓動kafka命令:

zkServer.sh start

kafka-server-start.sh /hadoop/kafka/config/server.properties

pyspark-2.3.1:

set spark\_home in pycharm direct to pyspark

spark集成kafka:添加spark-streaming-kafka-0-8-assembly\_2.10-2.2.2.jar到spark/jars，jars同scala版本和pyspark 版本

broker 使用主機名稱稱表示或直接只用IP ，根據網路情況選擇

spark-2.3.1:

設置文件spark-defaults.conf.template改名爲spark-defaults.conf

spark-submit --master local F:\laikaiwaa\project\web\hadoop\code\_spark\sparkstreming\streaming.py

hadoop-2.7.6:

Windows : hadoop-common-2.6-src winutil.ext hadoop.dll

主機名稱寫入salve文件:main suba

設置文件hadoop/etc/hadoop/core-site.xml:

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>/hadoop/tmp</value>

</property>

設置文件hadoop/etc/hadoop/hdfs-site.xml:

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>singlehost:9001</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:///hadoop/tmp/dfs/name</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:///hadoop/tmp/dfs/data</value>

</property>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

<name>dfs.webhdfs.enabled</name>

<value>true</value>

</property>

設置文件hadoop/etc/hadoop/hadoop-env.sh

export JAVA\_HOME=/jdk8

安裝ssh服務：sudo apt-get install openssh-server

ssh狀態 sudo ps -e |grep ssh

啟動服務: sudo service ssh start

sudo /etc/init.d/ssh restart

設置ssh：ssh-keygen -t rsa -P ""

将服务端的公钥拷贝到客户端

cat /root/.ssh/id\_rsa.pub >> /root/.ssh/authorized\_keys

設置免密碼登錄:

ssh-copy-id -i .ssh/id\_rsa.pub root@172.17.0.2

hadoop初始化:

hadoop namenode -format

啓動Hadoop:start-all.sh

停止Hadoop:stop-all.sh

Hadoop 安全模式（被spark 读取）：

hadoop dfsadmin -safemode leave

hadoop 写入权限:

设置变量HADOOP\_USER\_NAME为有写入权限的用户名

hive-1.2.2:

設置文件hive-env.sh:

export HIVE\_CONF\_DIR=/hadoop/hive/conf

設置文件hive-site.xml(hive-default.xml.template):

<property>

<name>hive.metastore.warehouse.dir</name>

<value>/hadoop/tmp/hive/warehouse</value>

<property>

<name>hive.exec.scratchdir</name>

<value>/hadoop/tmp/hive</value>

</property>

<property>

<name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:mysql://localhost:3306/hive</value>

</property>

<property>

<name>javax.jdo.option.ConnectionDriverName</name>

<value>com.mysql.jdbc.Driver</value>

</property>

<property>

<name>javax.jdo.option.ConnectionUserName</name>

<value>hive</value>

</property>

<property>

<name>javax.jdo.option.ConnectionPassword</name>

<value>hivepassword</value>

</property>

mysql

CREATE USER ‘hive’@‘localhost’ IDENTIFED BY ‘hivepassword’；

grant all privileges on \*.\* to 'hive'@'localhost' IDENTIFED BY 'hivepassword';

flush privileges;

schematool -dbType mysql -initSchema

復制 mysql-connector-java-5.1.46-bin.jar 到hive\lib,spark\jars

查詢語句對保留字用反引號 ``

hbase-2.0.1:

hbase-site.xml:

<property>

<name>hbase.rootdir</name>

<value>hdfs://master:9000/opt/hbase/hbase\_db</value>

</property>

<property>

<name>hbase.cluster.distributed</name>

<value>true</value>

</property>

<property>

<name>hbase.zookeeper.quorum</name> <!-- list of zookooper -->

<value>master,slave1,slave2</value>

</property>

<property>

<name>hbase.zookeeper.property.dataDir</name>

<value>/opt/hbase/zookeeper</value>

</property>

<property>

<name>zookeeper.znode.parent</name>

<value>/hbase/hbasenode</value>

</property>

设置conf中hbase-env.sh的 export JAVA\_HOME=/hadoop/jdk8

啓動hbase:

start-hbase.sh

hbase shell

hadoop namenode -format

hadoop fs -chmod -R 777 /tmp

http://localhost:50070