一、搭建kafka环境

1 下载kafka\_2.11-0.10.2.0

kylin支持的kafka版本为0.10及以上，同时发现s12181上的kafak用的是2.11的scala. (集群上默认的kafka版本低于0.10，kafka-client的jar包版本也不满足kylin的要求，需要逐个节点进行替换升级)，配置好的kylin及kafka均在/usr/mqb/目录下(20170215-apache-kylin-1.6.0-cdh5.7-bin, kafka\_2.11-0.10.2.0), kylin-2.0技术预览版本的spark引擎目前不支持kafka 流式构建。

2 修改kafka配置文件

zookeeper集群共有3个节点(s12181, s12191, s12192), 进入kafka\_2.11-0.10.2.0\config目录修改对应文件。

2.1修改对应节点的server.properties

broker.id=81

listeners=PLAINTEXT://s12181:9092

zookeeper.connect= s12181:2181,s12191:2181,s12192:2181

delete.topic.enable=true

log.dirs= /usr/mqb/ kafka\_2.11-0.10.2.0/ kafka-logs

2.2 修改producer.properties

bootstrap.servers=s12181:9092, s12191:9092, s12192:9092

compression.type=snappy

2.3 修改consumer.properties

zookeeper.connect= s12181:2181,s12191:2181,s12192:2181

3 启动kafka并测试

在每个节点上启动kafka

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

./kafka-topics.sh --zookeeper s12181:2181 --list

./kafka-topics.sh --zookeeper s12181:2181 --create --replication-factor 2 --partitions 3 --topic test

./kafka-topics.sh --zookeeper s12181:2181 --list

test-relicated-topic

./kafka-topics.sh --zookeeper s12181:2181 --describe --topic test

Topic:test PartitionCount:3 ReplicationFactor:2 Configs:

Topic: test Partition: 0 Leader: 91 Replicas: 91,92 Isr: 91,92

Topic: test Partition: 1 Leader: 92 Replicas: 92,81 Isr: 92,81

Topic: test Partition: 2 Leader: 81 Replicas: 81,91 Isr: 81,91

./kafka-console-producer.sh --broker-list s12181:9092, s12191:9092, s12192:9092 --topic test

This is a message

This is another message

./kafka-console-consumer.sh --bootstrap-server s12181:9092 --topic test --from-beginning

This is a message

This is another message

./kafka-topics.sh --zookeeper s12192:2181 --delete --topic test

Topic test-relicated-topic is marked for deletion.

Note: This will have no impact if delete.topic.enable is not set to true.

二、kylin stream cube 构建

1 修改kylin配置文件, 重启kylin

分别在setenv.sh和kylin.sh中添加如下内容（设置kafka环境变量），之后重启各个kylin节点

export KAFKA\_HOME=/usr/mqb/kafka\_2.11-0.10.2.0

2 创建一个kafka样例topic, 并生成样例数据

./kafka-topics.sh --create --zookeeper s12181:2181 --replication-factor 2 --partitions 3 --topic kylindemo

# KafkaSampleProducer 约每秒产生100条数据

./kylin.sh org.apache.kylin.source.kafka.util.KafkaSampleProducer --topic kylindemo --broker s12181:9092

./kafka-console-consumer.sh --bootstrap-server s12181:9092 --topic kylindemo --from-beginning

3 构建cube

3.1 点击 “Model” -> “Data Source”， 然后点击“Add Streaming Table”图标

3.2 粘贴JSON message，例如：

{"amount":80.46377518390474,"category":"ELECTRONIC","order\_time":1487835028996,"device":"Other","qty":0,"user":{"id":"260063ad-a5be-4c36-9e20-5d5a8be97b68","age":21,"gender":"Male"},"currency":"USD","country":"JAPAN"}

3.3 输入STREAMING\_SALES\_TABLE作为表明，点击next

3.4 配置Kafka集群信息

Topic: Kylindemo

ID Host Port

81 s12181 9092

91 s12191 9092

92 s12192 9092

3.5 定义数据模型(SSModel)

维度：

["CATEGORY","ORDER\_TIME","DEVICE","USER\_AGE","USER\_GENDER","CURRENCY","COUNTRY","YEAR\_START","QUARTER\_START","MONTH\_START","WEEK\_START","DAY\_START","HOUR\_START","MINUTE\_START"]

度量：

1 AMOUNT

2 QTY

Partition Date Column

DEFAULT.STREAMING\_SALES\_TABLE.MINUTE\_START

3.6 定义CUBE模型(SSCube)

don’t use “order\_time” as dimension as that is pretty fine-grained; suggest to use “mintue\_start”, “hour\_start” or other, depends on how you will inspect the data.

维度：

ID Name Table Name Type

1 CATEGORY DEFAULT.STREAMING\_SALES\_TABLE normal Column CATEGORY

2 DEVICE DEFAULT.STREAMING\_SALES\_TABLE normal Column DEVICE

3 USER\_AGE DEFAULT.STREAMING\_SALES\_TABLE normal Column USER\_AGE

4 USER\_GENDER DEFAULT.STREAMING\_SALES\_TABLE normal Column USER\_GENDER

5 CURRENCY DEFAULT.STREAMING\_SALES\_TABLE normal Column CURRENCY

6 COUNTRY DEFAULT.STREAMING\_SALES\_TABLE normal Column COUNTRY

7 DAY\_START DEFAULT.STREAMING\_SALES\_TABLE normal Column DAY\_START

8 HOUR\_START DEFAULT.STREAMING\_SALES\_TABLE normal Column HOUR\_START

9 MINUTE\_START DEFAULT.STREAMING\_SALES\_TABLE normal Column MINUTE\_START

度量：

Name Expression Parameters Return Type

SUM\_AMOUNT SUM Value:AMOUNT, Type:column decimal(19,4)

SUM\_QTY SUM Value:QTY, Type:column bigint

Auto Merge Thresholds

0.5 (Hours)

4 (Hours)

1 (Days)

7 (Days)

28 (Days)

Hierarchy Dimensions

["DAY\_START","HOUR\_START","MINUTE\_START"]

3.7 构建CUBE

1) 通过WEB页面点击build

2) 使用restful API (7071端口为nigix负载均衡器的端口，kylin的默认端口为7070)

curl -X PUT --user ADMIN:KYLIN -H "Content-Type: application/json;charset=utf-8" -d '{ "sourceOffsetStart": 0, "sourceOffsetEnd": 9223372036854775807, "buildType": "BUILD"}' <http://s12083:7071/kylin/api/cubes/SSCube/build2>

第一次构建经过4.82 mins cube 构建成功，其中包含3611条数据。SQL select count(\*) from STREAMING\_SALES\_TABLE 首次运行时间 3.42s , 再次运行时间 0.29s 。

二次构建：(运行过程中最多使用10240M , 9vcore。目前yarn共管理24g, 32vcore)

经过5.13 mins cube 构建成功，其中包含44584条数据。SQL select count(\*) from STREAMING\_SALES\_TABLE 首次运行时间 0.35s , 再次运行时间 0.29s 。

三次构建：（5分钟左右以后， 让kafka累计一定数据）

经过5.10 mins cube 构建成功，其中包含86482条数据。SQL select count(\*) from STREAMING\_SALES\_TABLE 首次运行时间 0.32s , 再次运行时间 0.35s 。

四次构建：(10分钟后, 停止kafka生产者)

经过5.12 mins cube 构建成功，其中包含113310条数据。SQL select count(\*) from STREAMING\_SALES\_TABLE 首次运行时间 0.41 , 再次运行时间 0.16s 。

第五次构建：（kafka生产者停止后，即在第4次构建后没有再次启动生产者）

提示没有数据，此时kylin的cube包含的数据量为247987，信息如下：{"url":"http://kylin\_lb/kylin/api/cubes/SSCube/build2","exception":"No new message comes, startOffset = endOffset:247987"}

第六次构建：（kafka 11:31 开始运行， 13:25停止）

经过6.98 mins cube 构建成功，其中包含639384条数据，此时cube共包含887371条数据。SQL select count(\*) from STREAMING\_SALES\_TABLE 首次运行时间 0.34 , 再次运行时间 0.14s 。