Apache Flink实战开发

Objective(本课目标)

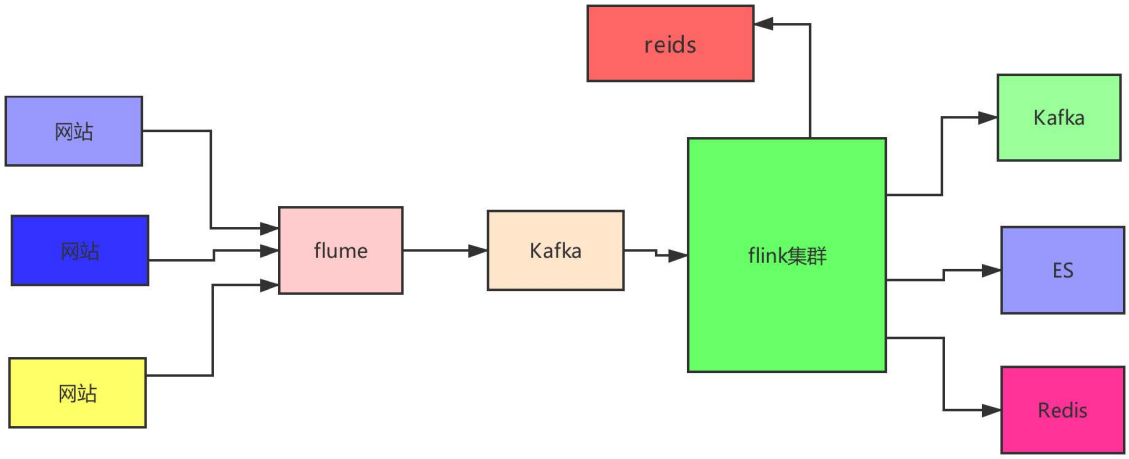
- [x] 掌握实时ETL开发
- [x] 掌握实时报表开发
- [x] 基于yarn集群运行Flink程序

1. 实时ETL

1.1 需求背景

- 产生的日志数据是嵌套json格式，需要拆分
- 针对日志数据中的国家字段进行大区转换
- 把数据回写到Kafka

1.2 项目架构



1.3 方案设计

日志格式：

```
{"dt":"2021-06-03 10:51:52","countryCode":"IN","data":
[{"type":"s4","score":0.1,"level":"A"},{"type":"s1","score":0.2,"level":"B"}]}
{"dt":"2021-06-03 10:51:54","countryCode":"IN","data":
[{"type":"s1","score":0.5,"level":"C"},{"type":"s4","score":0.1,"level":"C"}]}
{"dt":"2021-06-03 10:51:56","countryCode":"HK","data":
[{"type":"s3","score":0.3,"level":"A"},{"type":"s4","score":0.3,"level":"A"}]}
{"dt":"2021-06-03 10:51:58","countryCode":"TW","data":
[{"type":"s1","score":0.8,"level":"B"},{"type":"s2","score":0.1,"level":"D"}]}
{"dt":"2021-06-03 10:52:00","countryCode":"SA","data":
[{"type":"s1","score":0.8,"level":"C"},{"type":"s5","score":0.1,"level":"D"}]}
```

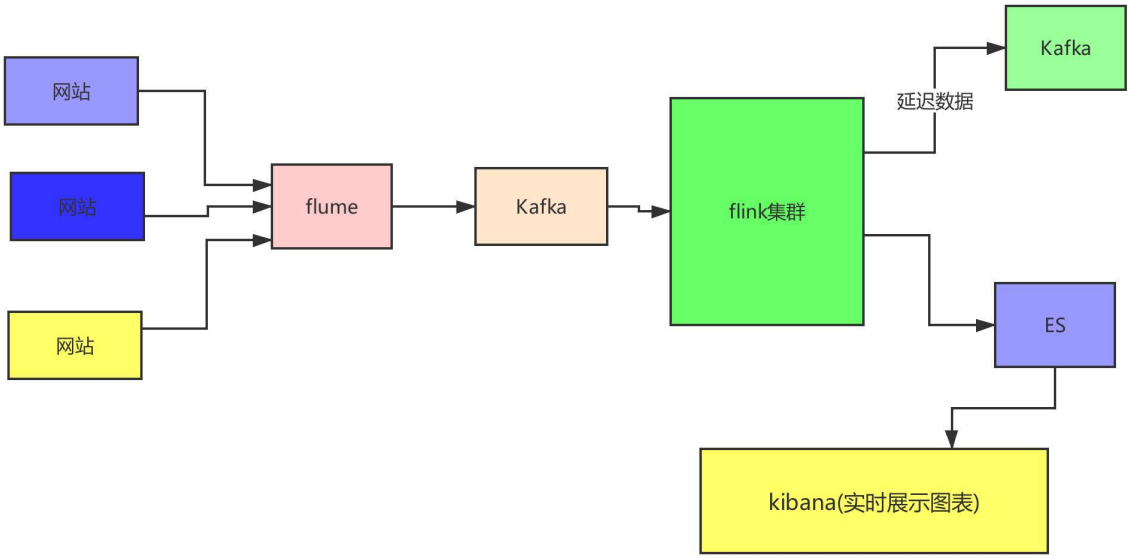
```
{"dt":"2021-06-03 10:52:02","countryCode":"KW","data":
[{"type":"s1","score":0.5,"level":"C"},{"type":"s2","score":0.5,"level":"A+"}]}
{"dt":"2021-06-03 10:52:04","countryCode":"US","data":
[{"type":"s1","score":0.5,"level":"C"},{"type":"s5","score":0.8,"level":"D"}]}
{"dt":"2021-06-03 10:52:06","countryCode":"PK","data":
[{"type":"s4","score":0.5,"level":"A"},{"type":"s1","score":0.3,"level":"A+"}]}

清洗格式为:
"dt":"2021-06-20 19:15:21","cc":"TW","type":"m1","score":0.5,"level":"C"
"dt":"2021-06-20 19:15:21","cc":"TW","type":"m2","score":0.3,"level":"B"

reids码表格式：码表里面的数据是动态的。
source: 自定义source
hset areas AREA_US US
hset areas AREA_CT TW,HK
hset areas AREA_AR PK,KW,SA, XX
hset areas AREA_IN IN

操作:
HKEYS areas
HGETALL areas
hget areas AREA_US

代码实现
ETL
```

2. 实时报表

2.1 需求背景

- 主要针对直播/短视频平台审核指标的统计
  - 统计不同大区每1 min内过审(上架)的视频数据量
  - 统计【不同大区】【不同类型】的每1 min内过审(上架)的视频数据量
    - watermark保证数据的有序和延迟多久有效
    - 迟到的数据写到kafka做数据补全

2.2 项目架构



2.3 方案设计

日志格式：

```
{"dt":"2021-06-03
11:32:05","type":"child_unshelf","username":"shenhe5","area":"AREA_IN"}
{"dt":"2021-06-03
11:32:06","type":"child_unshelf","username":"shenhe1","area":"AREA_IN"}
{"dt":"2021-06-03
11:32:06","type":"child_unshelf","username":"shenhe1","area":"AREA_CT"}
{"dt":"2021-06-03
11:32:07","type":"unshelf","username":"shenhe3","area":"AREA_AR"}
{"dt":"2021-06-03
11:32:07","type":"chlid_shelf","username":"shenhe4","area":"AREA_IN"}
{"dt":"2021-06-03
11:32:08","type":"child_unshelf","username":"shenhe5","area":"AREA_IN"}
{"dt":"2021-06-03
11:32:08","type":"chlid_shelf","username":"shenhe4","area":"AREA_AR"}
{"dt":"2021-06-03
11:32:09","type":"child_unshelf","username":"shenhe5","area":"AREA_US"}
{"dt":"2021-06-03 11:32:09","type":"black","username":"shenhe1","area":"AREA_ID"}
{"dt":"2021-06-03 11:32:10","type":"shelf","username":"shenhe5","area":"AREA_ID"}
{"dt":"2021-06-03
11:32:10","type":"chlid_shelf","username":"shenhe2","area":"AREA_CT"}
{"dt":"2021-06-03
11:32:11","type":"chlid_shelf","username":"shenhe4","area":"AREA_ID"}
{"dt":"2021-06-03 11:32:11","type":"black","username":"shenhe4","area":"AREA_IN"}
{"dt":"2021-06-03 11:32:12","type":"shelf","username":"shenhe1","area":"AREA_AR"}
{"dt":"2021-06-03
11:32:12","type":"unshelf","username":"shenhe1","area":"AREA_CT"}
{"dt":"2021-06-03
11:32:13","type":"child_unshelf","username":"shenhe2","area":"AREA_AR"}
{"dt":"2021-06-03 11:32:13","type":"black","username":"shenhe4","area":"AREA_CT"}
{"dt":"2021-06-03 11:32:14","type":"black","username":"shenhe3","area":"AREA_IN"}
{"dt":"2021-06-03
11:32:14","type":"child_unshelf","username":"shenhe4","area":"AREA_IN"}
```

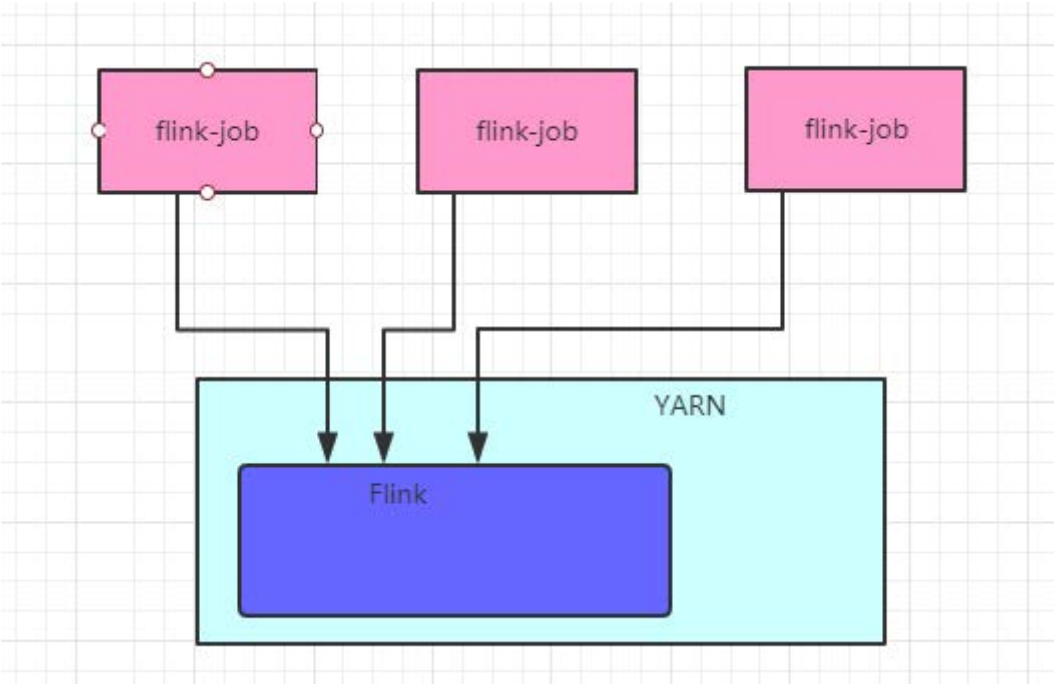
```
{"dt":"2021-06-03 11:32:15","type":"black","username":"shenhe4","area":"AREA_US"}
{"dt":"2021-06-03
11:32:15","type":"child_unshelf","username":"shenhe3","area":"AREA_CT"}
{"dt":"2021-06-03 11:32:16","type":"black","username":"shenhe4","area":"AREA_AR"}
{"dt":"2021-06-03 11:32:16","type":"shelf","username":"shenhe3","area":"AREA_ID"}
{"dt":"2021-06-03 11:32:17","type":"shelf","username":"shenhe3","area":"AREA_ID"}
{"dt":"2021-06-03
11:32:17","type":"unshelf","username":"shenhe2","area":"AREA_IN"}
{"dt":"2021-06-03
11:32:18","type":"unshelf","username":"shenhe1","area":"AREA_IN"}
{"dt":"2021-06-03
11:32:18","type":"child_unshelf","username":"shenhe3","area":"AREA_AR"}
{"dt":"2021-06-03
11:32:19","type":"unshelf","username":"shenhe3","area":"AREA_IN"}
{"dt":"2021-06-03
11:32:19","type":"unshelf","username":"shenhe5","area":"AREA_IN"}
{"dt":"2021-06-03 11:32:20","type":"shelf","username":"shenhe4","area":"AREA_IN"}
```


3. Flink on Yarn模式安装

- 首先安装好Hadoop（yarn）
- 上传一个flink的包（什么参数都不要配置）

3.1 方式一

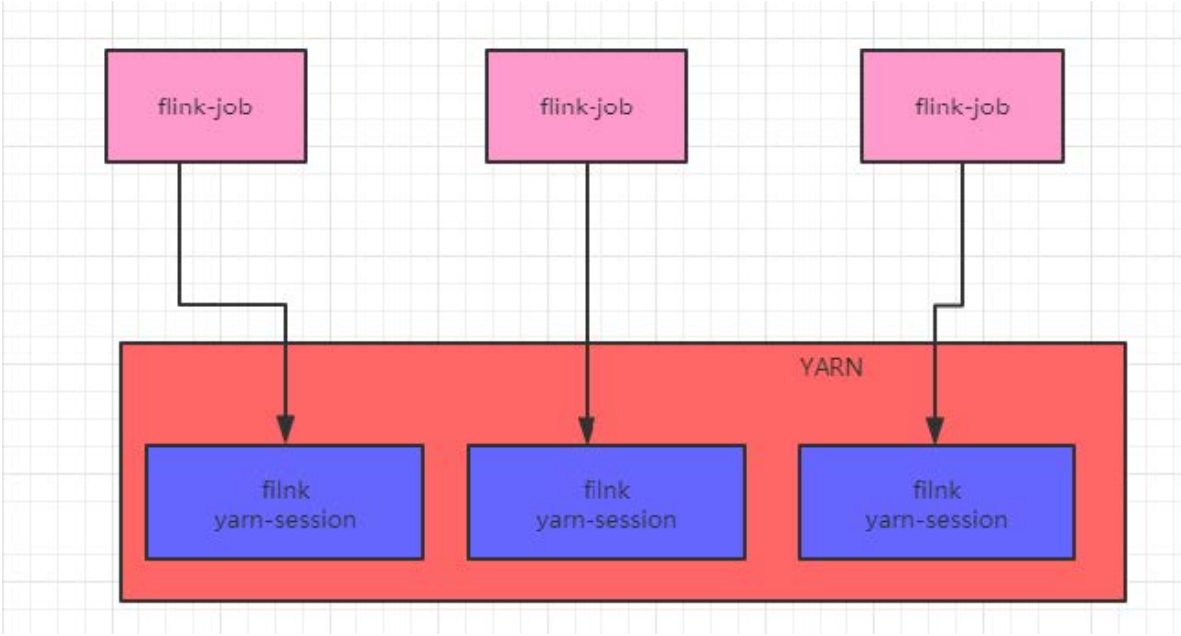
- 在YARN里面启动一个flink集群，然后我们再往flink集群提交任务，除非把Flink集群停了，不然资源不会释放
- 占用资源优先给Flink用



3.2 方式二

- 每次提交一个任务就在yarn上面启动一个flink小集群（推荐使用）

- 任务运行完了资源就自动释放



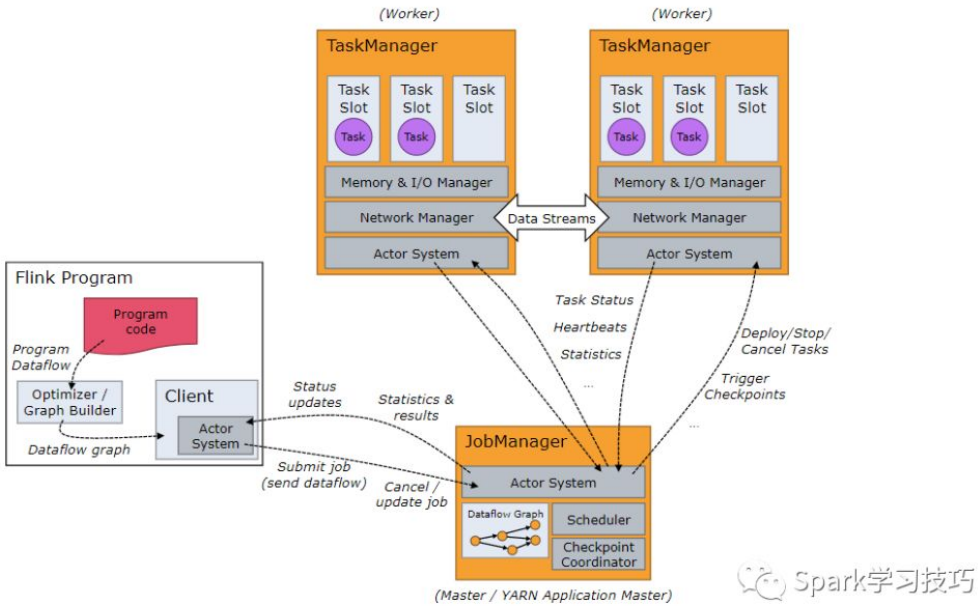
3.3 不同模式的任务提交

```
第一种方式实现:
启动一个一直运行的flink集群
/bin/yarn-session.sh -n 2 -jm 1024 -tm 1024 [-d]

-n: 启动几个taskManager
-jm: jobManager内存
-tm: taskManager内存

执行任务
flink run WordCount.jar -input hdfs://hadoop4:8020/word -output
hdfs://hadoop4:8020/result

把任务附着到一个已存在的flink yarn session
yarn-session.sh -id application_153382341238_7712

停止任务 【web界面或者命令行执行cancel命令】



第二种方式实现:
flink run -m yarn-cluster

启动集群，执行任务
提交程序:
flink run -m yarn-cluster -yqu default -ynm bw-ats1 -yn 2 -ys 2 -yjm 1024 -ytm
1024 -c com.bw.flink.wordCount WordCount.jar

-yqu: 指定队列
-ynm: 指定任务名称
-yn: 指定TaskManager的数量
```

```
-ys: 每个taskManager里面有几个slot
-yjm: jobmanager的内存
-ytm: taskmanager的内存
-c:指定运行的class

注意-1: client端必须要设置YARN_CONF_DIR或者HADOOP_CONF_DIR或者HADOOP_HOME环境变量，通过这个环境变量来读取YARN和HDFS的配置信息，否则启动会失败
注意-2: 如果想要flink在hadoop中运行，需要添加flink-slidae-hadoop2-uber-1.10.2.jar


help信息
yarn-session.sh

用法:
必选
    -n,--container <arg> 分配多少个yarn容器 (=taskmanager的数量)可选
    -D <arg> 动态属性
    -d,--detached 独立运行
    -jm,--jobManagerMemory <arg> JobManager的内存 [in MB]
    -nm,--name 在YARN上为一个自定义的应用设置一个名字
    -q,--query 显示yarn中可用的资源 (内存, cpu核数)
    -qu,--queue <arg> 指定YARN队列.
    -s,--slots <arg> 每个TaskManager使用的slots数量
    -tm,--taskManagerMemory <arg> 每个TaskManager的内存 [in MB]
    -z,--zookeeperNamespace <arg> 针对HA模式在zookeeper上创建NameSpace
    -id,--applicationId <yarnAppId> YARN集群上的任务id，附着到一个后台运行的 yarn session中
```

3.4 Flink on YARN集群部署

- Flink on yarn运行原理



- 其实Flink on YARN部署很简单，就是只要部署好hadoop集群即可，我们只需要部署一个Flink客户端，然后从flink客户端提交Flink任务即可。

- 1.上传客户端的jar包，配置信息等到HDFS
- 2.到resourcesManager进行注册和申请container
- 3.启动AppMaster，也就是启动JobManager
- 4.向ResourceManger申请运行任务的资源
- 5.开始运行TaksManager