标签部署文档

深海大数据平台

标签画像系统

<部署手册>

版本	V1.0.2
组织	京东科技
发布日期	2021-09-29

目录

1.　法律声明

2.　产品简介

2.1. 产品概述

标签系统可以为业务人员提供一站式的数据分析、用户画像分析以及可视化门户等功能需求。通过标签系统的建设，可以统一全行的管理应用分析体系以及管理分析指标体系，全面提升各业务条线的管理和经营决策能力。整合不同业务条线的数据资源，满足全行数据资源调度需求，建立统一的数据服务平台，提升营管部的数据服务管理能力。

2.2. 产品部署架构

3.　部署环境需求

3.1. 硬件资源需求

机器类型	**配置**	**台数**
虚机	4C*8G*500G硬盘	3

3.2. 部署组件依赖需求

模块	**依赖**	**部署方式**
产品依赖	门户	提前准备
	作业管理	提前准备
环境中间件	JDK	手工部署
	Mysql	天玑
	Hadoop	天玑
	Hbase	天玑
	zookeeper	天玑
	clickHouse	天玑
	spark	天玑
	hive	天玑
	minio	手工部署
	nginx	提前准备
	redis	天玑

4.　　部署前准备

4.1. 部署包信息

序号	应用包	说明	版本
1	avatar-pass.zip	商业化-标签画像系统-统一画像Web后台	v2
2	avatar-data-task.zip	任务控制位图计算任务	v2
3	tag-calculate.zip	商业化-标签画像系统-统一画像计算服务	v2
4	tag-service-bitmap.zip	商业化-标签画像系统-群体下载服务	v2
5	tag-interface-value.zip	商业化-标签画像系统-标签取值接口服务	v2
6	tag-calculate-hit.zip	标签画像群体命中存储服务商业化版本	v2
7	tag-interface-hit.zip	群体命中接口服务商业化版本	v2
8	avatar-task-scheduler.zip	数据推送服务-数据任务调度	v2
9	avatar-offline-main-1.0.0.jar	商业化-标签画像系统-JDT用户画像离线任务	v2
10	avatar-data-crowder-1.0.0.jar	数据推送服务-库表建群与回流5K	v2
11	avatar-offline-sinker-1.0.0.jar	数据推送服务-数据任务调度	v2
12	RoaringBitmap-0.6.66.jar guava-20.0.jar	依赖包	

4.2. 安装用户说明

用户	是否需要sudo权限	备注
admin	是	

4.3. 部署路径说明

安装机器需要提前创建目录

组件	路径		
全部	部署路径	/export/App/{模块名}	
	日志路径	/export/App /{模块名}/logs	

5.　　安装部署步骤操作指南

5.1. 初始化

5.1.1.　　jdk安装

(注:jdk版本是基于linux的64位的1.8.0_60，不能低于1.8.0_60版本，部署TAG的所有服务器都需要安装jdk)

1）检测环境变量

```
$ java -version
```

2）若没有安装,上传jdk-8u261-linux-x64.tar.gz到/export/servers目录下，执行 tar -zxvf jdk-8u261-linux-x64.tar.gz解压

3）添加以下环境变量（jdk安装路径作为依据）：

```
export JAVA_HOME=/export/servers/jdk1.8.0_261

export JAVA_BIN=/export/servers/jdk1.8.0_261/bin

export PATH=$JAVA_HOME/bin:$PATH
```

export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar

export JAVA_HOME JAVA_BIN PATH CLASSPATH JAVA_OPTS

4）重新刷新

执行source /etc/profile，使环境变量生效

5）检测是否安装成功

$ java －version

出现java版本号即为安装成功

5.1.2.　　Kerberos客户端配置

1. kerberos客户端安装

yum install -y krb5-libs krb5-workstation

2. 下载krb5.conf放置在本机目录: /etc/krb5.conf

下载keytab(如: dwetl.keytab)文件到本机目录/etc/下

3. 同步keytab文件

查看/export/kerberos/52/xxx 是否有keytab文件

5.1.3.　　数据初始化

使用navicat等工具远程连接给定数据库

```
mysql －h IP －u user －p

  • create database if not exists avatar;
  • create database if not exists bdp_portal_asm_sge;
  • source /home/supdev/avatar.sql
  • source /home/supdev/ bdp_portal_asm_sge.sql
```

5.2. 后端服务部署

部署顺序依次为：

统一画像Web后台(avatar-pass)

任务控制位图计算任务(avatar-data-task)

画像计算服务(tag-calculate)

群体下载服务(tag-service-bitmap)

标签取值接口服务(tag-interface-value)

画像群体命中存储服务(tag-calculate-hit)

群体命中接口服务(tag-interface-hit)

数据推送服务-数据任务调度(avatar-task-scheduler)

5.2.1.　　统一画像Web后台

上传应用包

上传avatar-pass.zip压缩包到/export/App目录下(若目录不存在, 创建目录mkdir －p /export/App)

解压安装包unzip avatar-pass.zip,

更改配置文件

vim avatar-pass.jar 找到配置文件application.properties查看对应的激活文件 spring.profiles.active=open

退出编辑器，找到对应的application-open.properties 修改配置项

修改app配置文件

```
#
cd /export/App/avatar-pass
#jar
vim avatar-pass.jar
#
/application-open.properties
#  esc :wq!

#
sh /export/App/avatar-pass/bin/stop.sh
sh /export/App/avatar-pass/bin/start.sh
```

application-open.properties

```
#datasource-config
spring.datasource.driver-Class-Name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://10.222.10.111:80/avatar?
characterEncoding=utf8&useSSL=true&allowMultiQueries=true
spring.datasource.username=root
spring.datasource.password=0gvzJr66iNs5


#kafka
#bdp.mqTypekafkajmq
bdp.mqType=kafka
bdp-rpc-use=dubbo
bdp.kafka.bootstrap-servers=component.app.kafka1.cluster.nosecured:9092,component.app.kafka2.cluster.nosecured:
9092,component.app.kafka3.cluster.nosecured:9092
spring.kafka.listener.type=batch
spring.kafka.consumer.max-poll-records=10000



#
#external-plat-config
##5K Metadata
5k.metadata.app=avatar
5k.metadata.authorization=Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJhdWQiOiJnYV9hdmF0YXIiLCJpc3MiOiJqZGZtZ3QuY29tIiwibmFtZSI6ImdhX2F2YXRhciIsImV4cCI6NDc2Mzc2MTI0MywiaWF0IjoxNjA4
MDg3NjQzfQ.L49i8UFEVsjaJHtEL2XaQJQGzCr3_M3Q6Y5Y9cZTBC8
5k.metadata.url=http://10.222.26.159/api/v1/metastore-5k/
##
#external-plat-config
#
bdp.routerurl.gaia= http://internal.bdp-gaia-api-gateway/common-authority/
bdp.routerurl.asm= http://linking-asm-core.101bank.sh/restful
#hadoop
bdp.hadoop.cluster.id=1

##uas,
sso.login.url = http://192.168.0.40:8080/sso/login
sso.exclude.path = /static,/operation/crowd/isHit,/operation/tag/mappingTagValue,/operation/crowd/download
sso.cookie.name = jdfcloud
sso.app.key = test3
sso.app.token = 347c6161e79f4b6a8873202dd5fe7e8f
sso.app.url = http://192.168.0.40:8080/
```

```
login.sso.invalid = true

##uas,
rh.login.address=http://test-ocean-ssa.jdfcloud.com:8080/sso/login
rh.sso.exclude.path=
rh.cookieName=jdfcloud
rh.ssoAppKey=test3
rh.ssoAppToken=347c6161e79f4b6a8873202dd5fe7e8f
rh.ssoAppUrl=http://test-ocean-ssa.jdfcloud.com:8080/
#validate=true
rh.login.validate=true


##asm
app.permissions.appCode=avatar
app.permissions.safeKey=4faae9abc4e44e8397f422b3e4a2f8f3


#uas
bdp.uas.appurl= http://internal.uas-web
bdp.uas.asmurl= http://linking-asm-core.101bank.sh
bdp.uas.cookiecipherkey=CW9bG0LwftWBr9W5v3gWug==
bdp.uas.cookiedomain=101bank.sh
bdp.uas.cookiemaxage=1728000000
bdp.uas.homeurl=http://linking.101bank.sh/board
#uas bdp.uas.invalid = false
bdp.uas.invalid=false
bdp.uas.loginsystem=ocean
bdp.uas.salt=89d33aa477b446c0a27cdd41ab1a18f3
bdp.uas.uasurl=http://uas-web.ql
bdp.uas-asm.exeludepath=/doc.html, /static,/rest,/assets,/data/sync,/com.jd.api,/sheduler,/job,/restful,/uas
/account/login,/ava/userGrantTag/getFlowDetails,/ava/token/getTokenGrantFlow,/ava/token/getSysGrantFlowDetails,
/ava/userGrantRole/getSysGrantFlowDetails,/ava/mock/jobMqMock


##
operateFlowApi.alias=test-operateFlowApi:0.0.1
flow.systemId=2020020

flow.encryption=cKdgHo4WDaQ=

##
hr.app.code=001
hr.safety.key=abc123
hr.jsf.alias=ZYX_HR_USER_SERVICE_JSF


#biz-config
##OSSminoossTpye=minoJSS
tag.selected.max.count=20
tag.enum.template.download.url=http://test.storage.jd.com/avatar-common/tag-enum-template/%E6%9E%9A%E4%B8%BE%E5%
80%BC%E6%A8%A1%E6%9D%BF20210121.xlsx?
Expires=3758684532&AccessKey=rqXviWEZl7RdTt4t&Signature=ESsWSZaXJMaqnzbdT6WkHU47Cv4%3D
##jss OSSminoossTpye=minoJSS
jss.bucket.upload=group-file
jss.path.upload=upload
jss.bucket.export=ava-export
jss.path.export.excel=excel
local.path.upload=/export/upload
local.path.export=/export/data/excel/
##r2m
r2m.queue.sourceToTarget=clickhouse-source-2-target
calculate.tag.bitmap.batch=5
scheduled.delay.initial.crowd.bitmap=5000
scheduled.delay.fixed.crowd.bitmap=10000
scheduled.delay.initial.push.wide=5000
scheduled.delay.fixed.push.wide=10000
scheduled.delay.initial.tag.offline=5000
scheduled.delay.fixed.tag.offline=10000
scheduled.delay.initial.tag.online=5000
scheduled.delay.fixed.tag.online=10000
```

```
#spsring
spring.task.scheduling.pool.size=10
spring.task.scheduling.thread-name-prefix=ava-scheduled-task

#uwc
uwc.url=http://uwc-server.jdd-dev.local:8021
uwc.app.code=avatar-paas
uwc.safety.key=91949acd75594fb499c94ed2abd15b5e

##
tag.manage.ip=10.222.30.164
tag.manage.dbname=tag_manage
tag.manage.username=root
tag.manage.password=123456
tag.manage.port=8083
jdd.tag.manage.ip=10.222.30.164
jdd.tag.manage.dbname=jdd_tag_manage
jdd.tag.manage.username=root
jdd.tag.manage.password=123456
jdd.tag.manage.port=8083
thor.appcode=avatar_pass
thor.secretkey=c821a89948228a0af4a9499aaaad8812
thor.alias=test

#dubbo
tag.value.jsf.alias=avatar-test
jsf.alias.tag=tag
jsf.alias.avatar=avatar


#MQ
gravity.job.status=ocean_gravity_job_status
jdd.oa.integration=jddOAIntegration_dev
hit.jsf.alias=avatar-hit
#
dd.noticeId=~test3
dd.aspId=110.200.0000002.102
dd.secret=2bab3bc067c56c3480e2dc0be86aa86d
dd.message.alias=line
dd.message.token=timline.jd.com

#dubbojsf dubbo
bdp.rpc.use=dubbo
bdp.dubbo.registryUrl=component.app.zk01.nonsecured:2181,component.app.zk02.nonsecured:2181,component.app.zk03.nonsecured:2181
dubbo.alias=dubbo.test
dubbo.check=false




#redis r2m.jd.jr.provider-type jedis,r2m
r2m.jd.jr.provider-type=jedis
r2m.jd.jr.app-name=
r2m.jd.jr.password=JDDocean2020
r2m.jd.jr.nodes[0].host=component.storage.redis01.cluster.auth
r2m.jd.jr.nodes[0].port=7001
r2m.jd.jr.nodes[1].host=component.storage.redis02.cluster.auth
r2m.jd.jr.nodes[1].port=7001
r2m.jd.jr.nodes[2].host=component.storage.redis03.cluster.auth
r2m.jd.jr.nodes[2].port=7001
r2m.jd.jr.nodes[3].host=component.storage.redis04.cluster.auth
r2m.jd.jr.nodes[3].port=7002
r2m.jd.jr.nodes[4].host=component.storage.redis05.cluster.auth
r2m.jd.jr.nodes[4].port=7002
r2m.jd.jr.nodes[5].host=component.storage.redis06.cluster.auth
r2m.jd.jr.nodes[5].port=7002

#minio oss.type MINIOJSS
oss.type=MINIO
```

```
minio.endpoint=http://internal.minio-server
minio.accesskey=minioadmin
minio.secretkey=minioadmin
minio.region=cn-north-1
minio.bucketName=tag

minio.local.path=/export/data/avatar-paas
```

启动服务

sh bin/start.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ start.sh）

停止服务

sh bin/stop.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ stop.sh）

5.2.2. 任务控制位图计算任务

上传应用包

1. 上传avatar-data-task.zip压缩包到/export/App目录下
2. 解压安装包unzip avatar-data-task.zip

更改配置文件

vim ldata-task-web.jar 找找到对应的application-xxx.properties 修改配置项

```
#
db:
  url: jdbc:mysql://10.222.10.111:80/avatar?useUnicode=true&characterEncoding=utf-
8&autoreconnect=true&serverTimezone=UTC
  username: root
  password: 0gvzJr66iNs5

#r2mjedis provider-type:jedisr2m
r2m:
  jd:
    jr:
      provider-type: jedis
      app-name: avatar-data-task-web
      password: JDDocean2020
      nodes:
        - {host: component.storage.redis01.cluster.auth,port: 7001}
        - {host: component.storage.redis02.cluster.auth,port: 7001}
        - {host: component.storage.redis03.cluster.auth,port: 7001}
        - {host: component.storage.redis04.cluster.auth,port: 7002}
        - {host: component.storage.redis05.cluster.auth,port: 7002}
        - {host: component.storage.redis06.cluster.auth,port: 7002}


#rpc dubbojsf
bdp:
  rpc:
    use: dubbo
  dubbo:
    registryUrl: component.app.zk01.nonsecured:2181,component.app.zk02.nonsecured:2181,component.app.zk03.
nonsecured:2181

jsf:
  alias: pre
  avatar:
    alias: avatar-pre

#
server:
  port: 15883


logging:
  level:
    root: info
```

sh bin/start.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ start.sh）**启动服务**

停止服务

sh bin/stop.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ stop.sh）

5.2.3. 统一画像计算服务

上传应用包

1. 上传tag-calculate.zip压缩包到/export/App
2. 解压安装包unzip tag-calculate.zip

更改配置文件

vim tag-calculate.jar 找到配置文件application.properties查看对应的激活文件 spring.profiles.active=open

退出编辑器，找到对应的application-open.properties 修改 mysql 、redis、kafka,dubbo、minio、clickhouse 、hbase 配置项

修改app配置文件

```
#
cd /export/App/tag-calculate
#jar
vim tag-calculate.jar
#
/application-open.properties
#  esc :wq!

#
sh /export/App/tag-calculate/bin/stop.sh
sh /export/App/tag-calculate/bin/start.sh
```

```
#log-config
logging.level.root=info
logging.file=tag-calculate
logging.path=./logs

#datasource-config
spring.datasource.driver-Class-Name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://10.222.10.111:80/avatar?
characterEncoding=utf8&useSSL=true&allowMultiQueries=true
spring.datasource.username=root
spring.datasource.password=0gvzJr66iNs5


#r2m-config
r2m.zkConnection=component.bdp.zk01.secured:2181,component.bdp.zk02.secured:2181,component.bdp.zk03.secured:2181
r2m.appName.avatar=avatar-paas
r2m.password.avatar=B0l8kVs94WlEoOW9

r2m.tag-cache.address=10.222.55.19:8175
r2m.tag-cache.appName=Tag-realtime
r2m.tag-cache.token=acf7b3a5db
r2m.tag-cache.password=UnLSNaw8IZR7dyvUlZmgtPbjKBAzmA5uJCyxt5fBWsk=

r2m.tag-cache2.address=10.222.55.19:8175
r2m.tag-cache2.appName=Tag-realtime
r2m.tag-cache2.token=acf7b3a5db
r2m.tag-cache2.password=UnLSNaw8IZR7dyvUlZmgtPbjKBAzmA5uJCyxt5fBWsk=

#jmq-config kafka
jmq.user=avatar
jmq.password=8AA7CCCB
jmq.sendTimeout=5000
jmq.address=jmq-testcluster.jd.local:50088
jmq.epoll=false
jmq.app=avatar

#kafka
mqType=kafka
bdp-rpc-use=dubbo
bdp.kafka.bootstrap-servers=component.app.kafka1.cluster.nosecured:9092,component.app.kafka2.cluster.nosecured:
9092,component.app.kafka3.cluster.nosecured:9092
bdp.fmq-particle.status-topic=ocean_gravity_job_status
bdp.jsf.alias.avatarPass=avatarPass




#jss-config  minio
jss.accessKey=rqXviWEZl7RdTt4t
jss.secretKey=q1w28PTpEGhgc01i86MCTJn41TdVgZj6IbLFMzMk
jss.hostName=test.storage.jd.com
jss.connectionTimeout=50000

#clickhouse-config  clickhous
clickhouse.local.ip=component.bdp.ck01:8123,component.bdp.ck02:8124;component.bdp.ck03:8123,component.bdp.ck04:
8124;component.bdp.ck05:8123,component.bdp.ck06:8124
clickhouse.user=default
clickhouse.passwd=0gvzJr66iNs5
clickhouse.socket.timeout=6000
clickhouse.session.timeout=6000
clickhouse.connect.timeout=3000

#hbase-config hbase
hbase.datasource.url=yz-222-97-17.h.chinabank.com.cn,yz-222-97-2.h.chinabank.com.cn,yz-222-97-24.h.chinabank.
com.cn
hbase.rpc.timeout=1000
hbase.client.operation.timeout=1000
hbase.client.scanner.timeout.period=1000



##jmq
jmq.crowd.topic=crowd_generate_status
##jss
jss.enterprise.bucket=bitmap-rule
jss.enterprise.bucket.path=business-tag
jss.fileBucket.bucket=crowd-file
jss.fileBucketPath.bucket.path=business-tag
jss.tag.user.bucket=bitmap
jss.tag.user.path=tag
jss.crowd.user.bucket=bitmap
jss.crowd.user.path=crowd
jss.avatar.bucket=group-file
jss.avatar.path=download
file.local.path=./tmp/

##R2M
r2m.queue.sourceToTarget=clickhouse-source-2-target
##R2M
proxy.busi.relTable=business_tag.dmxupf_biz_sf_user_lable_all_user_offset_data_s_d_v1
proxy.busi.relCalCol=busi_offset
proxy.busi.xn.relMatchCol=xn_id
proxy.busi.rl.relMatchCol=rl_id
proxy.busi.relGroupByCol=id_type
proxy.busi.relTableDtColumn=dt

spring.task.scheduling.pool.size=3
spring.task.scheduling.thread-name-prefix=ava-scheduled-task

jsf.alias.avatar=avatar-dev
jsf.alias.tag=tag-dev

file.path.temp=D:/temp/csv/

#dubbo
bdp.rpc.use=dubbo
bdp.dubbo.registryUrl=component.app.zk01.nonsecured:2181,component.app.zk02.nonsecured:2181,component.app.zk03.
nonsecured:2181
dubbo.alias=dubbo.test

#redis
r2m.jd.jr.provider-type=jedis
r2m.jd.jr.app-name=
r2m.jd.jr.password=JDDocean2020
r2m.jd.jr.nodes[0].host=component.storage.redis01.cluster.auth
r2m.jd.jr.nodes[0].port=7001
r2m.jd.jr.nodes[1].host=component.storage.redis02.cluster.auth
r2m.jd.jr.nodes[1].port=7001
r2m.jd.jr.nodes[2].host=component.storage.redis03.cluster.auth
r2m.jd.jr.nodes[2].port=7001
r2m.jd.jr.nodes[3].host=component.storage.redis04.cluster.auth
```

```
r2m.jd.jr.nodes[3].port=7002
r2m.jd.jr.nodes[4].host=component.storage.redis05.cluster.auth
r2m.jd.jr.nodes[4].port=7002
r2m.jd.jr.nodes[5].host=component.storage.redis06.cluster.auth
r2m.jd.jr.nodes[5].port=7002

#minl
oss.type=MINIO
minio.endpoint=http://10.222.11.50:9000
#minio.accesskey=CzMinIO
#minio.secretkey=CzzzAdmin0219
minio.accesskey=minioadmin
minio.secretkey=minioadmin
minio.region=cn-north-1
minio.bucketName=bucketName

minio.local.path=/export/data/tag-calculate
```

sh bin/start.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ start.sh）**启动服务**

停止服务

sh bin/stop.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ stop.sh）

5.2.4. 群体下载服务

上传应用包

1. 上传tag-service-bitmap.zip压缩包到/export/App
2. 解压安装包unzip tag-service-bitmap.zip

更改配置文件

vim tag-service-bitmap.zip 找到配置文件application.properties查看对应的激活文件 spring.profiles.active=open

退出编辑器，找到对应的application-open.properties 修改配置项

更改配置文件命令

```
#
cd /export/App/tag-service-bitmap-war
#jar
vim tag-service-bitmap-war.jar
#
/application-open.properties
#  esc :wq!

#
sh /export/App/tag-service-bitmap-war/bin/stop.sh
sh /export/App/tag-service-bitmap-war/bin/start.sh
```

需要修改的配置参数

```
#kafka
myType=kafka
bdp.kafka.bootstrap-servers=component.app.kafka1.cluster.nosecured:9092,component.app.kafka2.cluster.nosecured:
9092,component.app.kafka3.cluster.nosecured:9092
#bdp.dubbo.registryUrl=component.app.zk01.nonsecured:2181,component.app.zk02.nonsecured:2181,component.app.zk03.
nonsecured:2181
bdp.fmq-particle.status-topic=ocean_gravity_job_status

#dubbo
bdp.rpc.use=dubbo
bdp.dubbo.registryUrl=zookeeper://component.app.zk01.nonsecured:2181
dubbo.alias=dubbo.test
dubbo.check=false

#redis
r2m.jd.jr.provider-type=jedis
r2m.jd.jr.app-name=
r2m.jd.jr.password=JDDocean2020
r2m.jd.jr.nodes[0].host=component.storage.redis01.cluster.auth
r2m.jd.jr.nodes[0].port=7001
r2m.jd.jr.nodes[1].host=component.storage.redis02.cluster.auth
r2m.jd.jr.nodes[1].port=7001
r2m.jd.jr.nodes[2].host=component.storage.redis03.cluster.auth
r2m.jd.jr.nodes[2].port=7001
r2m.jd.jr.nodes[3].host=component.storage.redis04.cluster.auth
r2m.jd.jr.nodes[3].port=7002
r2m.jd.jr.nodes[4].host=component.storage.redis05.cluster.auth
r2m.jd.jr.nodes[4].port=7002
r2m.jd.jr.nodes[5].host=component.storage.redis06.cluster.auth
r2m.jd.jr.nodes[5].port=7002

#minlo
oss.type=MINIO
minio.endpoint=http://10.222.11.50:9000
#minio.accesskey=CzMinIO
#minio.secretkey=CzzzAdmin0219
minio.accesskey=minioadmin
minio.secretkey=minioadmin
minio.region=cn-north-1
minio.bucketName=bucketName

minio.local.path=/export/data/service-bitmap
```

sh bin/start.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ start.sh）启动服务

停止服务

sh bin/stop.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ stop.sh）

5.2.5. 标签取值接口服务

上传应用包

1. 上传 tag-interface-value.zip 压缩包到/export/App
2. 解压安装包unzip ag-interface-value.zip

更改配置文件

application-open.ymlhbase-v3 dubbokafkaredis

```
#application-open.ymlhbase-v3 dubbokafkaredis
vim  tag-interface-value.jar
/application-open.yml
```

修改相应配置

applicaiton-open.yml

```
#hbase-v3
hbase-v3:
  uri: component.bdp.zk01.secured:2181,component.bdp.zk02.secured:2181,component.bdp.zk03.secured:2181

bdp:
  rpc:
    use: dubbo
          #dubbo
        dubbo:
    registryUrl: component.app.zk01.nonsecured:2181
  #kafka
  fmq-particle:
    status-topic: ocean_gravity_job_status
  kafka:
    bootstrap-servers: component.app.kafka1.cluster.nosecured:9092,component.app.kafka2.cluster.nosecured:9092,
component.app.kafka3.cluster.nosecured:9092
    registurl: component.bdp.zk01.secured:2181
# kafka jmq
mqType: kafka

#redis
r2m:
  jd:
    jr:
      provider-type: jedis
      app-name: bdp-ric
      password: JDDocean2020
      nodes:
        - {host: component.storage.redis01.cluster.auth,port: 7001}
        - {host: component.storage.redis02.cluster.auth,port: 7001}
        - {host: component.storage.redis03.cluster.auth,port: 7001}
        - {host: component.storage.redis04.cluster.auth,port: 7002}
        - {host: component.storage.redis05.cluster.auth,port: 7002}
        - {host: component.storage.redis06.cluster.auth,port: 7002}
```

:wq! 保存退出

```
:wq!
```

application.ymldubbo

```
/applicaiton.yml
```

```
############### dubbo
dubbo:
  alias:
  application:
    name: bdp-tag-app-download-service
  protocol:
    name: dubbo
    port: 38000
  registry:
    protocol: zookeeper
    address: ${bdp.dubbo.registryUrl}
  check: false
```

启动服务

sh bin/start.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ start.sh）

停止服务

sh bin/stop.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ stop.sh）

5.2.6. 群体命中存储服务

上传应用包

1. 上传tag-calculate-hit.zip压缩包到/export/App
2. 解压安装包unzip tag-calculate-hit.zip

更改配置文件

vim tag-calculate-hit.zip 找到配置文件application.properties查看对应的激活文件 spring.profiles.active=open

退出编辑器，找到对应的application-open.properties 修改配置项

更改配置文件命令

```
#
cd /export/App/tag-calculate-hit
#jar
vim tag-calculate-hit.jar
#
/application-open.properties
#  esc :wq!

#
sh /export/App/tag-calculate-hit/bin/stop.sh
sh /export/App/tag-calculate-hit/bin/start.sh
```

需要修改的配置参数

```
#bitmap
bitmap.local.path=G:\\export\\bitmap


#dubbo
bdp.rpc.use=dubbo
bdp.dubbo.registryUrl=component.app.zk01.nonsecured:2181,component.app.zk02.nonsecured:2181,component.app.zk03.
nonsecured:2181
dubbo.alias=dubbo.test

#redis
r2m.jd.jr.provider-type=jedis
r2m.jd.jr.app-name=
r2m.jd.jr.password=JDDocean2020
r2m.jd.jr.nodes[0].host=component.storage.redis01.cluster.auth
r2m.jd.jr.nodes[0].port=7001
r2m.jd.jr.nodes[1].host=component.storage.redis02.cluster.auth
r2m.jd.jr.nodes[1].port=7001
r2m.jd.jr.nodes[2].host=component.storage.redis03.cluster.auth
r2m.jd.jr.nodes[2].port=7001
r2m.jd.jr.nodes[3].host=component.storage.redis04.cluster.auth
r2m.jd.jr.nodes[3].port=7002
r2m.jd.jr.nodes[4].host=component.storage.redis05.cluster.auth
r2m.jd.jr.nodes[4].port=7002
r2m.jd.jr.nodes[5].host=component.storage.redis06.cluster.auth
r2m.jd.jr.nodes[5].port=7002

#minlo
oss.type=MINIO
minio.endpoint=http://internal.minio-server
#minio.accesskey=CzMinIO
#minio.secretkey=CzzzAdmin0219
minio.accesskey=minioadmin
minio.secretkey=minioadmin
minio.region=
minio.bucketName=bucketName

minio.local.path=/export/data/calculate-hit
```

注意：涉及minio的项目需要创建临时文件夹，命令如下：

创建临时文件夹

```
mkdir -p /export/data/avatar-paas
mkdir -p /export/data/service-bitmap
mkdir -p /export/data/tag-calculate
mkdir -p /export/data/calculate-hit
mkdir -p /export/data/data-crowder
mkdir -p /export/data/offine-sinker
```

sh bin/start.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ start.sh）启动服务

停止服务

sh bin/stop.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ stop.sh）

5.2.7. 群体命中接口服务

上传应用包

1. 上传tag-interface-hit.zip压缩包到/export/App
2. 解压安装包unzip tag-interface-hit.zip

更改配置文件

配置文件修改

```
#
cd /export/App/ tag-interface-hit
#jar
vim  tag-interface-hit.jar
#
/application-open.properties
#  esc :wq!

#
sh /export/App/tag-calculate/bin/stop.sh
sh /export/App/tag-calculate/bin/start.sh
```

application-open.properties

```
# jsf
jsf.threads=2
jsf.queues=10
jsf.epoll=false
jsf.timeout=200
jsf.threadPool=fixed
# cluster
shard.total=1
#calculate
shard.size=16
#calculate
shard.jsf.alias=avatar-c
shard.jsf.timeout=200
shard.jsf.retries=3
shard.jsf.serialization=protobuf
shard.jsf.loadBalance=leastactive
# R2M
r2m.zkConnection=component.bdp.zk01.secured:2181
#offsetpin
r2m.appName.offset2pin=TagBM_C2_1
r2m.password.offset2pin=f4ls71E1bRr1Ttm6
r2m.token.offset2pin=affa31fca6
#pinoffset
r2m.appName.pin2Offset=TagBM_C2_2
r2m.password.pin2Offset=GhFIRboWtmVqG7Pr
r2m.token.pin2Offset=ae71869fcc
#offsetID
r2m.appName.offset2business=offset2business
r2m.password.offset2business=l0zcz6etXKVibnAd
r2m.token.offset2business=accbbd8373
#IDoffset
r2m.appName.business2offset=business2offset
r2m.password.business2offset=c8bkhWdgnl15yFtH
r2m.token.business2offset=ac2a4d70f5
#
r2m.appName.avatar=avatar-paas
r2m.password.avatar=B0l8kVs94WlEoOW9
r2m.token.avatar=a3de95919a
#log config
logging.path=./logs
logging.level.root=debug
logging.queue.size=10
#caffeine config,expire:S
cache.caffeine.max=100
cache.caffeine.expire=60
sys.subject.id=1
```

```
#
monitor.log.scheduled=0 */1 * * * ?
monitor.log.key=tag-interface-hit-log-flag
#
down.ids.num.min=5
down.ids.num.max=5000

bdp.rpc.use=dubbo
bdp.dubbo.registryUrl=zookeeper://component.app.zk01.nonsecured:2181
dubbo.alias=dubbo.test

#redis
r2m.jd.jr.provider-type=jedis
r2m.jd.jr.app-name=requirepass
r2m.jd.jr.password=JDDocean2020
r2m.jd.jr.nodes[0].host=10.222.109.186
r2m.jd.jr.nodes[0].port=7001
r2m.jd.jr.nodes[1].host=10.222.109.180
r2m.jd.jr.nodes[1].port=7001
r2m.jd.jr.nodes[2].host=10.222.109.213
r2m.jd.jr.nodes[2].port=7001
r2m.jd.jr.nodes[3].host=10.222.109.186
r2m.jd.jr.nodes[3].port=7002
r2m.jd.jr.nodes[4].host=10.222.109.180
r2m.jd.jr.nodes[4].port=7002
r2m.jd.jr.nodes[5].host=10.222.109.213
r2m.jd.jr.nodes[5].port=7002
```

启动服务

sh bin/start.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ start.sh）

停止服务

sh bin/stop.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ stop.sh）

5.2.8. 数据推送服务-数据任务调度

上传应用包 avatar-task-scheduler.zip

1. 上传avatar-task-scheduler..zip压缩包到/export/App
2. 解压安装包unzip avatar-task-scheduler.zip

更改配置文件

```
#jar application.yml ,redishdfs  :wq!  ESC :qjar
vim task-scheduler-web.jar
/application.yml
```

```
#jar application-open.yml ,kafkamysqluwc :wq!  ESC :qjar
vim task-scheduler-web.jar
/application-open.yml
```

application.yml

```yaml
spring:
  profiles:
    active: open
  mvc:
    static-path-pattern: /**
  main:
    allow-bean-definition-overriding: true
  # kafka start ---------------------------
  kafka:
    bootstrap-servers: ${bdp.kafka.bootstrap-servers}
    consumer:
      auto-commit-interval: 100
      auto-offset-reset: earliest
      enable-auto-commit: true
      group-id: bdp-integration
      key-deserializer: org.apache.kafka.common.serialization.StringDeserializer
      topics: ${bdp.fmq-particle.status-topic}
      value-deserializer: org.apache.kafka.common.serialization.StringDeserializer
  # kafka end ----------------------------
# kafka jmq
mqType: ${mqTypeSwitch}


r2m:
#  address: 10.222.55.19:8175
  appName:
  token: a5301175f2
  password: JDDocean2020
  providerType: jedis
  nodes:
    - { host: component.storage.redis01.cluster.auth,port: 7001 }
    - { host: component.storage.redis02.cluster.auth,port: 7001 }
    - { host: component.storage.redis03.cluster.auth,port: 7001 }
    - { host: component.storage.redis01.cluster.auth,port: 7002 }
    - { host: component.storage.redis02.cluster.auth,port: 7002 }
    - { host: component.storage.redis03.cluster.auth,port: 7002 }

task:
  time:
    hbase:
      start: 0
      end: 24
  alarm:
    mails: liangfawen@jd.com;fujialong3@jd.com;tianye51@jd.com;biyulong@jd.com;yuanzengbao@jd.com;zhufeng21@jd.com;quyibo1@jd.com;guocongyuan@jd.com
    erps: liangfawen;biyulong;fujialong3;tianye52;yuanzengbao;zhufeng21;quyibo1;guocongyuan;
hdfs:
  jarPath: hdfs://component.bdp.nn01:8020/ns1/user/yhhx/job/jars
  #dbName: dmx_dev  #
```

启动服务-标签取值接口服务

sh bin/start.sh （首次启动需格式化shell文件 sed -i ’s/\r//’ start.sh）

配置文件说明

5.2.9 JDT用户画像离线任务

上传应用包

上传avatar-offline-main-1.0.0.jar到/export/App

更改配置文件

vim avatar-offline-main-1.0.0.jar 找到配置文件application.conf application.yml

```
#
cd /export/App
#jar
vim avatar-offline-main-1.0.0.jar
#
/application.conf
#  esc :wq!
```

application.conf文件内容

```
ClickHouse {
  url = "component.bdp.ck01,component.bdp.ck02,component.bdp.ck03,component.bdp.ck04,component.bdp.ck05,
component.bdp.ck06"
  insertUrl = "component.bdp.ck01,component.bdp.ck02,component.bdp.ck03,component.bdp.ck04,component.bdp.ck05,
component.bdp.ck06"
  port = 8123
  insertPort = 8123
  user = "default"
  password = "V1sBUm+u"
  cluster = "oceantestself"
  storagePolicy = "default"
  ReplicatedEngine {
    zkPrefix = "/clickhouse/tables/{shard}"
    replicaName = "{replica}"
  }
  remainDt = 2
}
```

vim avatar-offline-main-1.0.0.jar 找到配置文件config.properties

config.properties文件内容

```
#hbase
hbase.zookeeper.quorum=component.app.zk01.nonsecured
hbase.tag.pin.tablename=smart_tag:jdd_ava_pin_tag
hbase.tag.pin.family=t

#redis
r2m.jd.jr.provider-type=jedis
r2m.jd.jr.app-name=
r2m.jd.jr.password=JDDocean2020
r2m.jd.jr.nodes[0].host=component.storage.redis01.cluster.auth
r2m.jd.jr.nodes[0].port=7001
r2m.jd.jr.nodes[1].host=component.storage.redis02.cluster.auth
r2m.jd.jr.nodes[1].port=7001
r2m.jd.jr.nodes[2].host=component.storage.redis03.cluster.auth
r2m.jd.jr.nodes[2].port=7001
r2m.jd.jr.nodes[3].host=component.storage.redis04.cluster.auth
r2m.jd.jr.nodes[3].port=7002
r2m.jd.jr.nodes[4].host=component.storage.redis05.cluster.auth
r2m.jd.jr.nodes[4].port=7002
r2m.jd.jr.nodes[5].host=component.storage.redis06.cluster.auth
r2m.jd.jr.nodes[5].port=7002

r2m.address=10.222.55.19:8175
r2m.paas.app=jbirt_dataservice_cache
r2m.paas.token=a6dde094d2
r2m.paas.password=YZuUcmLIdNnH9sPQHSnVBs2YjvcwtleQ0qBtKTiqR/8=

r2m.pinoffset.app=jbirt_dataservice_cache
r2m.pinoffset.token=a6dde094d2
r2m.pinoffset.password=YZuUcmLIdNnH9sPQHSnVBs2YjvcwtleQ0qBtKTiqR/8=


r2m.offset2business.app=offset2business
r2m.offset2business.token=accbbd8373
r2m.offset2business.password=DayL7UBxa4fihe8VffIaxTF1s6s2uHSyXZmhUxsyFIs=


r2m.business2offset.app=business2offset
r2m.business2offset.token=ac2a4d70f5
r2m.business2offset.password=FoEZ/JvOPi3gJgO0CZ2f8+dcTaEdevBEY3tEkpGnsRo=

sink.address.batchSize=10000
sink.address.multiple=1
```

部署文件

1、 上传到hdfs的 hdfs://ns1/user/yhhx/job/jars

修改app配置文件

```
#
cd /export/App/


#hdfs
hadoop fs ls /ns1/user/yhhx/job/jars/
#hdfs
hadoop fs -mkdir -p /ns1/user/yhhx/job/jars/
#jar
hadoop fs -rm -r /ns1/user/yhhx/job/jars/avatar-offline-main-1.0.0.jar
#jar
hadoop fs -put ./avatar-offline-main-1.0.0.jar  /ns1/user/yhhx/job/jars/
#
hadoop fs -chmod 777 /ns1/user/yhhx/job/jars/avatar-offline-main-1.0.0.jar
```

5.2.10 数据推送服务-库表建群与回流5K

上传应用包

上传avatar-offline-sinker-1.0.0.jar到/export/App

更改配置文件

vim avatar-offline-sinker-1.0.0.jar 找到配置文件application.conf

application.yml

```
#
cd /export/App
#jar
vim avatar-offline-sinker-1.0.0.jar
#
/application.conf
#  esc :wq!
```

application.conf文件内容

```
Hdfs {
  jarPath = "hdfs://ns1/user/yhhx/job/jars"
  dbName = "dmx_dev"
}
R2m {
  url = "component.storage.redis01.cluster.auth:7001,component.storage.redis02.cluster.auth:7001,component.
storage.redis03.cluster.auth:7001,component.storage.redis04.cluster.auth:7002,component.storage.redis05.cluster.
auth:7002,component.storage.redis06.cluster.auth:7002"
  expireTime = 86400
  providerType = "redis" //
  AvatarPaas {
    app = "avatar-paas"
    token = "masterauth"
    password = "JDDocean2020"
  }
}
Vop {
  vopZk = ""
  vopApp = ""
  vopToken = ""
  vopCluster = ""
  vopNameSpace = ""
  vopGroup = ""
  vopUser = ""
  vopImage5k = ""
  vopImageM6 = ""
}
Jss {
  host = "http://10.222.11.50:9000"
  access = "minioadmin"
  secret = "minioadmin"

  region="cn-north-1"
  ossType="MINIO"
  bucketName="buchetName"
}

ClickHouse {
  //  9000 13000
  url = "component.bdp.ck01,component.bdp.ck02,component.bdp.ck03"
  insertUrl = "component.bdp.ck01`component.bdp.ck02,component.bdp.ck02`component.bdp.ck03,component.bdp.
ck03`component.bdp.ck01"
  port = 8123
  insertPort = 8123
  user = "default"
  password = "0gvzJr66iNs5"
  cluster = "oceantestself"
  storagePolicy = "default"
  ReplicatedEngine {
    zkPrefix = "/clickhouse/tables/{shard}"
    replicaName = "{replica}"
  }
  tblLife = 3
}

HBase {
  quorum = "component.bdp.zk01.secured,component.bdp.zk02.secured,component.bdp.zk03.secured"
  zkPort = 2181
  znode = "/hbase"
  namenode = "hdfs://component.bdp.nn01:50070"
  hfilePath = "/user/yhhx/"
  hfilePrefix = "ava_hfile"
  hfileNum = 200
  numMappers = 200
  brandWidth = 40
}
```

部署文件

上传到hdfs的 hdfs://ns1/user/yhhx/job/jars

修改app配置文件

```
#
cd /export/App/
#hdfs
hadoop fs -mkdir -p /ns1/user/yhhx/job/jars/
#jar
hadoop fs -rm -r /ns1/user/yhhx/job/jars/avatar-offline-sinker-1.0.0.jar
#jar
hadoop fs -put ./avatar-offline-sinker-1.0.0.jar  /ns1/user/yhhx/job/jars/
#
hadoop fs -chmod 777 /ns1/user/yhhx/job/jars/avatar-offline-sinker-1.0.0.jar
```

5.2.11 数据推送服务-离线推送CK和HBASE

上传应用包

1、 上传avatar-data-crowder-1.0.0.jar到/export/App

application.yml

```
#
cd /export/App
#jar
vim avatar-data-crowder-1.0.0.jar
#
/application.conf
#  esc :wq!
```

更改配置文件

application.conf文件内容

```
Hdfs {
  path = "hdfs://ns1/user/yhhx/job/jars"
}
R2m {
  url = "component.storage.redis01.cluster.auth:7001,component.storage.redis02.cluster.auth:7001,component.storage.redis03.cluster.auth:7001,component.storage.redis01.cluster.auth:7002,component.storage.redis02.cluster.auth:7002,component.storage.redis03.cluster.auth:7002"
  expireTime = 86400
  providerType = "redis"  // ccc ,redis ,zk
  AvatarPaas {
    app = "avatar-paas"
    token = "requirepass"
    password = "JDDocean2020"
  }
}
Vop {
  vopZk = ""
  vopToken = ""
  vopCluster = ""
  vopApp = ""
  vopNameSpace = ""
  vopGroup = ""
  vopUser = ""
  vopImage5k = ""
}
Jss {
  host = "http://10.222.11.50:9000"
  access = "minioadmin"
  secret = "minioadmin"

  region="cn-north-1"
  ossType="MINIO"
  bucketName="buchetName"
}
ClickHouse {
  //  9000 13000
  url = "component.bdp.ck01,component.bdp.ck02,component.bdp.ck03"
  insertUrl = "component.bdp.ck01`component.bdp.ck02,component.bdp.ck02`component.bdp.ck03,component.bdp.ck03`component.bdp.ck01"
  port = 8123
  insertPort = 8123
  user = "default"
  password = "0gvzJr66iNs5"
  cluster = "clickhouse-server"
  storagePolicy = "default"
  ReplicatedEngine {
    zkPrefix = "/clickhouse/tables/{shard}"
    replicaName = "{replica}"
  }
  remainDt = 2
}
```

部署文件

上传到hdfs的 hdfs://ns1/user/yhhx/job/jars

修改app配置文件

```
#
cd /export/App/
#hdfs
# hadoop fs -mkdir -p /ns1/user/yhhx/job/jars/
#jar
hadoop fs -rm -r /ns1/user/yhhx/job/jars/avatar-data-crowder-1.0.0.jar
#jar
hadoop fs -put ./avatar-data-crowder-1.0.0.jar  /ns1/user/yhhx/job/jars/
#
hadoop fs -chmod 777 /ns1/user/yhhx/job/jars/avatar-data-crowder-1.0.0.jar
```

部署依赖包

RoaringBitmap-0.6.66.jar

guava-20.0.jar

修改app配置文件

```
#
cd /export/App/
#hdfs
# hadoop fs -mkdir -p /ns1/user/yhhx/job/jars/
#jar
hadoop fs -rm -r /ns1/user/yhhx/job/jars/RoaringBitmap-0.6.66.jar
#jar
hadoop fs -put ./RoaringBitmap-0.6.66.jar  /ns1/user/yhhx/job/jars/
#
hadoop fs -chmod 777 /ns1/user/yhhx/job/jars/RoaringBitmap-0.6.66.jar
```

修改app配置文件

```
#
cd /export/App/
#hdfs
# hadoop fs -mkdir -p /ns1/user/yhhx/job/jars/
#jar
hadoop fs -rm -r /ns1/user/yhhx/job/jars/guava-20.0.jar
#jar
hadoop fs -put ./guava-20.0.jar  /ns1/user/yhhx/job/jars/
#
hadoop fs -chmod 777 /ns1/user/yhhx/job/jars/guava-20.0.jar
```

5.2.12 系统初始化

```
curl -H "Content-Type: application/json"  \
  -H 'Cookie: qid_uid=0a49ba41-84d9-4dd1-b837-e03bdba07f62; qid_fs=1631203650577; ocean_ticket=Y
/413VjHTN9ue9zBTokW6NBd2Px39+X0XPrZfIA6/M5eOHnygTdEtQ==; qid_ad=linking-tag.101bank.sh%7C-%7Cnjd%7C-%7C0;
qid_ls=1632365160832; qid_ts=1632367068188; qid_sq=7; qid_sid=0a49ba41-84d9-4dd1-b837-e03bdba07f62-7' \
-X POST "http://localhost:19086/init/id/initSubject"

curl -H "Content-Type: application/json"  \
  -H 'Cookie: qid_uid=0a49ba41-84d9-4dd1-b837-e03bdba07f62; qid_fs=1631203650577; ocean_ticket=Y
/413VjHTN9ue9zBTokW6NBd2Px39+X0XPrZfIA6/M5eOHnygTdEtQ==; qid_ad=linking-tag.101bank.sh%7C-%7Cnjd%7C-%7C0;
qid_ls=1632365160832; qid_ts=1632367068188; qid_sq=7; qid_sid=0a49ba41-84d9-4dd1-b837-e03bdba07f62-7' \
-X POST "http://localhost:19086/init/id/initIDMapping"
```

5.3 前端部署

5.3.1 安装node和pm2

部署包
node-v12.16.1.tar.gz
pm2.tar.gz

部署路径
/export/servers 上传后解压tar包

设置path
/etc/profile文件 追加 /export/servers/node-v12.16.1/bin 和 /export/servers/pm2/bin
例如:
export PATH=$JAVA_HOME/bin:$PATH:/export/servers/node-v12.16.1/bin:/export/servers/pm2/bin

5.3.2 项目部署

初次部署:
1.把data-mass-tag-v2.zip 上传到 /export/servers/node_apps 并解压
2.进入data-mass-tag-v2项目，进入bin目录，sh start.sh 启动项目

二次部署:
1.进入data-mass-tag-v2项目，替换dist目录