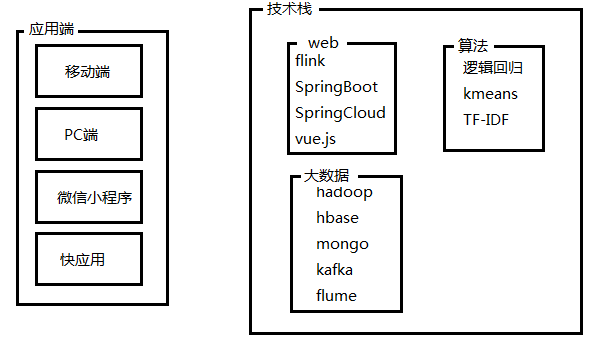
# 基于Flink流处理的动态实时亿级全端用户画像系统

## 项目概述

### 1.1整体概述

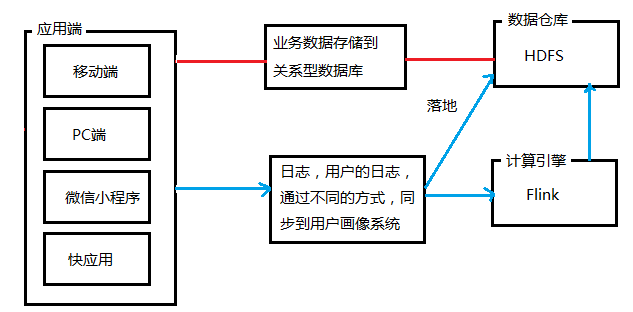


### 1.2项目价值

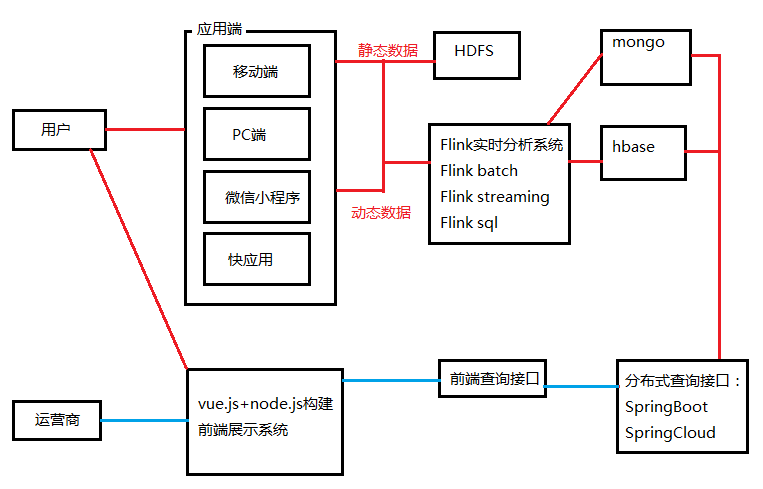
这套系统针对亿级用户的用户画像，帮助运营商很好的了解用户，根据这套系统针对性的营销，包括广告、增加用户的粘度、增加系统的趣味性等等，这套系统的商业价值非常高。根据企业的实际应用，进行少量的二次开发，便可以在线上应用。

### 1.3项目架构

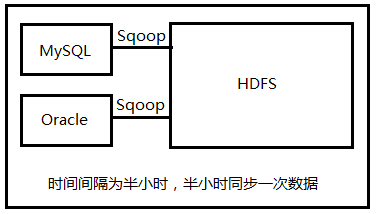
1. 数据来源与落地



（2）处理流程



（3）数据同步



（4）静态信息和动态信息说明

**静态信息数据来源**

用户填写的个人资料，或者由此通过一定的算法，计算出来的数据

如果有不确定的，可以建立模型来判断，比如用户的性别注册没有填写，可以建立模型，根据用户的行为来判断用户性别是什么，或者它的概率

**动态信息数据来源**

用户行为产生的数据：注册、游览、点击、购买、签收、评价、收藏等等。

用户比较重要的行为数据：游览商品，收藏商品、加入购物车、关注商品

根据这些行为特性可以计算出：用户注册时间、首单时间、潮妈族、纠结商品、最大消费、订单数量、退货数量等等。

## 第2章 系统设计

### 2.1用户/用户详情补充表结构定义

用户表：用户ID、用户名、密码、性别、手机号、邮箱、年龄、注册时间、收货地址、终端类型

|  |
| --- |
| CREATE TABLE IF NOT EXISTS `userinfo` (  `userid` int(20) NOT NULL COMMENT '用户ID',  `username` varchar(50) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '用户名称',  `password` varchar(50) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '用户密码',  `sex` int(1) NULL DEFAULT NULL COMMENT '性别',  `telphone` varchar(50) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '电话',  `email` varchar(50) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '邮箱',  `age` int(20) NULL DEFAULT NULL COMMENT '年龄',  `registerTime` timestamp(0) NULL DEFAULT NULL COMMENT '注册时间',  `usertype` int(1) NULL DEFAULT NULL COMMENT '终端类型，0：PC，1：移动端，2小程序',  PRIMARY KEY (`userid`) USING BTREE  ) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8\_general\_ci ROW\_FORMAT = Dynamic; |

用户详情补充表：学历、收入、职业、婚姻、是否有小孩、是否有车有房、使用手机品牌、用户id

|  |
| --- |
| CREATE TABLE IF NOT EXISTS `userdetail` (  `userdetailid` int(20) NOT NULL COMMENT '主键',  `userid` int(20) NULL DEFAULT NULL COMMENT '用户ID',  `edu` int(1) NULL DEFAULT NULL COMMENT '学历',  `profession` varchar(20) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '职业',  `marriage` int(20) NULL DEFAULT NULL COMMENT '婚姻状况，1：未婚，2：已婚，3：离异，4：未知',  `haschild` int(1) NULL DEFAULT NULL COMMENT '1：没有孩子，2：有孩子，3：未知',  `hascar` int(1) NULL DEFAULT NULL COMMENT '1：有车，2：无车，3：未知',  `hashourse` int(1) NULL DEFAULT NULL COMMENT '1：有房，2：无房，3：未知',  `telphonebrand` varchar(50) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL COMMENT '手机品牌',  PRIMARY KEY (`userdetailid`) USING BTREE  ) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8\_general\_ci ROW\_FORMAT = Dynamic; |

### 2.2商品信息/类型表结构定义

用户订单表：订单id 商品id，商品类别id，订单时间、支付时间、支付方式、使用优惠劵金额、退款金额

商品表

商品类别表

|  |
| --- |
| CREATE TABLE IF NOT EXISTS `productinfo` (  `id` int(20) NOT NULL,  `producttypeid` int(20) NULL DEFAULT NULL,  `productname` varchar(50) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL,  `productdescription` varchar(1500) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL,  `price` int(20) NULL DEFAULT NULL,  `num` int(20) NULL DEFAULT NULL,  `createtime` timestamp(0) NULL DEFAULT NULL,  `updatetime` timestamp(0) NULL DEFAULT NULL,  `matchartid` int(20) NULL DEFAULT NULL,  `producturl` varchar(20) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL,  PRIMARY KEY (`id`) USING BTREE  ) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8\_general\_ci ROW\_FORMAT = Dynamic; |
| CREATE TABLE IF NOT EXISTS `producttype` (  `id` int(20) NOT NULL AUTO\_INCREMENT,  `producttypename` varchar(50) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL,  `producttypedescription` varchar(200) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL,  `producttypeleave` int(2) NULL DEFAULT NULL,  PRIMARY KEY (`id`) USING BTREE  ) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8\_general\_ci ROW\_FORMAT = Dynamic; |

### 2.3订单信息表

|  |
| --- |
| CREATE TABLE IF NOT EXISTS `orderinfo` (  `id` int(20) NOT NULL AUTO\_INCREMENT,  `productid` int(20) NULL DEFAULT NULL,  `producttypeid` int(20) NULL DEFAULT NULL,  `createtime` timestamp(0) NULL DEFAULT NULL,  `amount` double(20, 2) NULL DEFAULT NULL,  `paytype` int(2) NULL DEFAULT NULL,  `paytime` timestamp(0) NULL DEFAULT NULL,  `paystatus` int(2) NULL DEFAULT NULL,  `couponamount` double(20, 2) NULL DEFAULT NULL,  `totalamount` double(20, 2) NULL DEFAULT NULL,  `refundamount` double(20, 2) NULL DEFAULT NULL,  `num` int(20) NULL DEFAULT NULL,  `userid` int(20) NULL DEFAULT NULL,  PRIMARY KEY (`id`) USING BTREE  ) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8\_general\_ci ROW\_FORMAT = Dynamic; |

### 2.4败家指数

败家指数 = 支付金额平均值\*0.3、最大支付金额\*0.3、下单频率\*0.4

### 2.5用户日志行为结构

浏览商品行为：商品id 商品类别id 浏览时间、停留时间、用户id 终端类别,用户ip

收藏商品行为：商品id 商品类别id 操作时间、操作类型（收藏，取消）、用户id、终端类别、用户ip

购物车行为：商品id 商品类别id 、操作时间、操作类型（加入，删除）、用户id、终端类别、用户ip

关注商品:商品id 商品类别id 操作时间、操作类型（关注，取消）、用户id、终端类别、用户ip

## 第3章 项目搭建

### 3.1HDFS环境搭建

详细过程，参见

<https://github.com/byf312358196/it.doc/tree/master/11.%E5%A4%A7%E6%95%B0%E6%8D%AE>

《大数据入门.docx》--3.6HDFS环境搭建

[root@master /soft/hadoop]#vi /etc/profile

|  |
| --- |
| export JAVA\_HOME=/soft/jdk  export PATH=$PATH:$JAVA\_HOME/bin  export HADOOP\_HOME=/soft/hadoop  export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin |

[root@master /soft/hadoop]#vi etc/hadoop/core-site.xml

|  |
| --- |
| <configuration>  <property>  <name>fs.default.name</name>  <value>hdfs://master:9000</value>  </property>  <property>  <name>hadoop.tmp.dir</name>  <value>/data/hadoop/tmp</value>  </property>  </configuration> |

[root@master /soft/hadoop]#vi etc/hadoop/hdfs-site.xml

|  |
| --- |
| <configuration>  <property>  <name>dfs.replication</name>  <value>1</value>  </property>  <property>  <name>dfs.permissions</name>  <value>false</value>  </property>  </configuration> |

[root@master /soft/hadoop]#vi etc/hadoop/mapred-site.xml

|  |
| --- |
| <configuration>  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>  <property>  <name>mapreduce.jobhistory.address</name>  <value>master:10020</value>  </property>  </configuration> |

[root@master /soft/hadoop]#vi etc/hadoop/yarn-site.xml

|  |
| --- |
| <configuration>  <!-- Site specific YARN configuration properties -->  <property>  <name>yarn.resourcemanager.hostname</name>  <value>master</value>  </property>  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property>  <property>  <name>mapreduce.job.ubertask.enable</name>  <value>true</value>  </property>  </configuration> |

[root@master /soft/hadoop]#vi etc/hadoop/slaves

|  |
| --- |
| master |

启动HDFS

|  |
| --- |
| [root@master /soft/hadoop/etc/hadoop]#jps  3343 Jps  [root@master /soft/hadoop/etc/hadoop]#start-dfs.sh  Starting namenodes on [master]  master: starting namenode, logging to /soft/hadoop-2.7.3/logs/hadoop-root-namenode-master.out  master: starting datanode, logging to /soft/hadoop-2.7.3/logs/hadoop-root-datanode-master.out  Starting secondary namenodes [0.0.0.0]  0.0.0.0: starting secondarynamenode, logging to /soft/hadoop-2.7.3/logs/hadoop-root-secondarynamenode-master.out  [root@master /soft/hadoop/etc/hadoop]#jps  3458 NameNode  3735 SecondaryNameNode  3576 DataNode  3848 Jps  [root@master /soft/hadoop/etc/hadoop]#start-yarn.sh  starting yarn daemons  starting resourcemanager, logging to /soft/hadoop-2.7.3/logs/yarn-root-resourcemanager-master.out  master: starting nodemanager, logging to /soft/hadoop-2.7.3/logs/yarn-root-nodemanager-master.out  [root@master /soft/hadoop/etc/hadoop]#jps  3458 NameNode  4291 Jps  3735 SecondaryNameNode  3576 DataNode  3996 NodeManager  3901 ResourceManager |

关闭防火墙

|  |
| --- |
| [root@master /soft/hadoop/etc/hadoop]#systemctl stop firewalld.service  [root@master /soft/hadoop/etc/hadoop]#systemctl status firewalld.service  ● firewalld.service - firewalld - dynamic firewall daemon  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)  Active: inactive (dead)  Docs: man:firewalld(1) |

### 3.2HBase环境搭建

安装ZooKeeper

|  |
| --- |
| [root@master /soft]#tar zxvf zookeeper-3.4.8.tar.gz  [root@master /soft]#ln -s zookeeper-3.4.8 zookeeper  [root@master /soft/zookeeper/conf]#cp zoo\_sample.cfg zoo.cfg  [root@master /soft/zookeeper/conf]#vi zoo.cfg  ...  dataDir=/data/zookeeper/tmp  ... |

安装HBase

|  |
| --- |
| [root@master /soft]#tar zxvf hbase-1.2.6-bin.tar.gz  [root@master /soft]#ln -s hbase-1.2.6 hbase  export ZK\_HOME=/soft/zookeeper  export PATH=$PATH:$ZK\_HOME/bin  export HBASE\_HOME=/soft/hbase  export PATH=$PATH:$HBASE\_HOME/bin |

启动

|  |
| --- |
| [root@master /soft/hbase/conf]#zkServer.sh start  ZooKeeper JMX enabled by default  Using config: /soft/zookeeper/bin/../conf/zoo.cfg  Starting zookeeper ... STARTED  [root@master /soft/hbase/conf]#zkServer.sh status  ZooKeeper JMX enabled by default  Using config: /soft/zookeeper/bin/../conf/zoo.cfg  Mode: standalone  [root@master /soft/hbase/conf]#start-hbase.sh  master: starting zookeeper, logging to /soft/hbase/bin/../logs/hbase-root-zookeeper-master.out  starting master, logging to /soft/hbase/logs/hbase-root-master-master.out  starting regionserver, logging to /soft/hbase/logs/hbase-root-1-regionserver-master.out  [root@master /soft/hbase/conf]#jps  3458 NameNode  5332 Jps  3735 SecondaryNameNode  5111 HMaster  5207 HRegionServer  3576 DataNode  3996 NodeManager  3901 ResourceManager  4798 QuorumPeerMain |

### 3.3MongoDB环境搭建

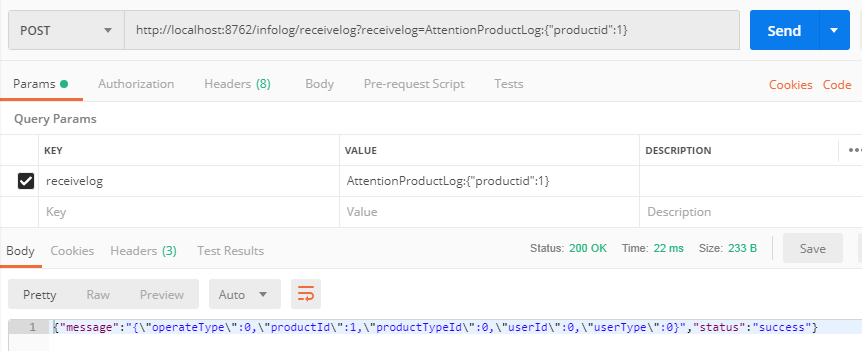
|  |
| --- |
| [root@master /soft]#tar zxf mongodb-linux-x86\_64-rhel70-3.4.21.tgz  [root@master /soft]#ln -s mongodb-linux-x86\_64-rhel70-3.4.21 mongodb  [root@master /soft]#cd mongodb  [root@master /soft/mongodb]#mkdir data  [root@master /soft/mongodb]#vi /etc/profile  [root@master /soft/mongodb]#source /etc/profile  [root@master /soft/mongodb]#mongod --dbpath=data  [root@master /root]#mongo  > show dbs;  admin 0.000GB  local 0.000GB |

### 3.4Kafka环境搭建

|  |
| --- |
| [root@master /soft]#tar zxf kafka\_2.11-2.3.0.tgz  [root@master /soft]#ln -s kafka\_2.11-2.3.0 kafka  [root@master /soft]#cd kafka  [root@master /soft/kafka]#cd config/  [root@master /soft/kafka/config]#vi server.properties  ...  Log.dir=/soft/kafka/kafka-logs  ...  [root@master /soft/kafka/config]#vi /etc/profile  export KAFKA\_HOME=/soft/kafka  export PATH=$PATH:$KAFKA\_HOME/bin  [root@master /soft/kafka/config]#kafka-server-start.sh server.properties  [root@master /soft/kafka]#kafka-topics.sh --create --topic attentionProductLog --zookeeper 127.0.0.1:2181 --replication-factor 1 --partitions 1  Created topic attentionProductLog.  [root@master /soft/kafka]#kafka-topics.sh --create --topic buyCartProductLog --zookeeper 127.0.0.1:2181 --replication-factor 1 --partitions 1  Created topic buyCartProductLog.  [root@master /soft/kafka]#kafka-topics.sh --create --topic collectProductLog --zookeeper 127.0.0.1:2181 --replication-factor 1 --partitions 1  Created topic collectProductLog.  [root@master /soft/kafka]#kafka-topics.sh --create --topic scanProductLog --zookeeper 127.0.0.1:2181 --replication-factor 1 --partitions 1  Created topic scanProductLog. |
| [root@master /root]#kafka-console-producer.sh --broker-list localhost:9092 --topic attentionProductLog  hello  >>hello  >hi  [root@master /soft/kafka]#kafka-console-consumer.sh --bootstrap-server 192.168.0.108:9092 --topic attentionProductLog  hello  hello  hi  {"operateType":0,"productId":1,"productTypeId":0,"userId":0,"userType":0} |

[http://localhost:8762/infolog/receivelog?receivelog=AttentionProductLog:{"productid":1}](http://localhost:8762/infolog/receivelog?receivelog=AttentionProductLog:{\"productid\":1})

{"message":"{\"operateType\":0,\"productId\":1,\"productTypeId\":0,\"userId\":0,\"userType\":0}","status":"success"}



## 详细设计

### 4.1代标签代码编写

年代：40年代 50年代 60年代 70年代 80年代 90年代 00后 10后

统计每个年代群里的数量，做到近实时统计，每半小时会进行一次任务统计

flink结合hbase保存年代标签代码编写

### 4.2年代群体数量统计代码

flink结合mongo保存年代群体数量

### 4.3手机运营商标签代码编写

### 4.4邮件运营商标签代码编写

### 4.5还原真实消费信息表结构定义

### 4.6败家指数计算规则定义

败家指数 = 支付金额平均值\*0.3、最大支付金额\*0.3、下单频率\*0.4

败家指数之最终得分计算代码编写

败家指数之最终得分计算代码编写

### 4.7用户行为日志结构讲解以及实体定义

### 4.8基于springboot+springcloud之2.0版本构建实时数据收集服务之注册中心代码编写

### 4.9基于springboot+springcloud之2.0版本构建实时数据收集服务之服务搭建代码编写

### 4.10基于springboot+springcloud之2.0版本构建实时数据收集服务代码编写

### 4.11实时收集服务整合kafka代码编写

### 4.12实时品牌偏好设计以及代码编写实现实时更新用户品牌偏好

### 4.13实时终端偏好代码编写