运营商网络流量流向异常账号监测系统详细设计说明书

目录

[1引言](#_Toc6549_WPSOffice_Level1) [4](#_Toc6549_WPSOffice_Level1)

[1.1编写目的](#_Toc29022_WPSOffice_Level1) [4](#_Toc29022_WPSOffice_Level1)

[1.2背景](#_Toc11904_WPSOffice_Level1) [4](#_Toc11904_WPSOffice_Level1)

[1.3术语定义及说明](#_Toc3461_WPSOffice_Level1) [4](#_Toc3461_WPSOffice_Level1)

[2需求概述](#_Toc20514_WPSOffice_Level1) [5](#_Toc20514_WPSOffice_Level1)

[2.1任务和目标](#_Toc16503_WPSOffice_Level1) [5](#_Toc16503_WPSOffice_Level1)

[2.2集群设计](#_Toc357_WPSOffice_Level1) [5](#_Toc357_WPSOffice_Level1)

[2.3数据来源](#_Toc6534_WPSOffice_Level1) [5](#_Toc6534_WPSOffice_Level1)

[1.3.1数据描述](#_Toc29022_WPSOffice_Level2) [5](#_Toc29022_WPSOffice_Level2)

[1.3.2样本数据描述](#_Toc11904_WPSOffice_Level2) [5](#_Toc11904_WPSOffice_Level2)

[2.4检测方法](#_Toc20051_WPSOffice_Level1) [6](#_Toc20051_WPSOffice_Level1)

[3设计概述](#_Toc20110_WPSOffice_Level1) [6](#_Toc20110_WPSOffice_Level1)

[3.1数据存储模块详细设计](#_Toc28523_WPSOffice_Level1) [6](#_Toc28523_WPSOffice_Level1)

[3.2数据处理模块详细设计](#_Toc28632_WPSOffice_Level1) [7](#_Toc28632_WPSOffice_Level1)

[3.2.1数据预处理](#_Toc3461_WPSOffice_Level2) [7](#_Toc3461_WPSOffice_Level2)

[3.2.2数据分析](#_Toc20514_WPSOffice_Level2) [8](#_Toc20514_WPSOffice_Level2)

[3.3数据展示模块详细设计](#_Toc11890_WPSOffice_Level1) [9](#_Toc11890_WPSOffice_Level1)

[3.3.1数据可视化](#_Toc16503_WPSOffice_Level2) [9](#_Toc16503_WPSOffice_Level2)

[3.3.2界面设计](#_Toc357_WPSOffice_Level2) [10](#_Toc357_WPSOffice_Level2)

[4数据库设计](#_Toc14389_WPSOffice_Level1) [11](#_Toc14389_WPSOffice_Level1)

[4.1表设计](#_Toc31400_WPSOffice_Level1) [11](#_Toc31400_WPSOffice_Level1)

[4.2查5分钟内的异常数据](#_Toc17100_WPSOffice_Level1) [11](#_Toc17100_WPSOffice_Level1)

[5. 流程图](#_Toc6534_WPSOffice_Level2) [12](#_Toc6534_WPSOffice_Level2)

[5.1 Topology图](#_Toc29022_WPSOffice_Level3) [12](#_Toc29022_WPSOffice_Level3)

[5.2持续读hdfs流程图](#_Toc11904_WPSOffice_Level3) [13](#_Toc11904_WPSOffice_Level3)

[6类及方法的定义](#_Toc28636_WPSOffice_Level1) [14](#_Toc28636_WPSOffice_Level1)

[6.1 flume](#_Toc31849_WPSOffice_Level1) [14](#_Toc31849_WPSOffice_Level1)

[6.1.1 Product实体类](#_Toc20051_WPSOffice_Level2) [14](#_Toc20051_WPSOffice_Level2)

[private String time; //数据统计时间](#_Toc3461_WPSOffice_Level3) [14](#_Toc3461_WPSOffice_Level3)

[private String userAccount; //宽带账户](#_Toc20514_WPSOffice_Level3) [14](#_Toc20514_WPSOffice_Level3)

[private String userIP; //用户以太网IP](#_Toc16503_WPSOffice_Level3) [15](#_Toc16503_WPSOffice_Level3)

[private String qqid; //QQ号](#_Toc357_WPSOffice_Level3) [15](#_Toc357_WPSOffice_Level3)

[private String natIP; //内网ip](#_Toc6534_WPSOffice_Level3) [15](#_Toc6534_WPSOffice_Level3)

[private String cookieValue; //cookie值](#_Toc20051_WPSOffice_Level3) [15](#_Toc20051_WPSOffice_Level3)

[private String devName; //设备名称](#_Toc20110_WPSOffice_Level3) [15](#_Toc20110_WPSOffice_Level3)

[private String osName; //操作系统名称](#_Toc28523_WPSOffice_Level3) [15](#_Toc28523_WPSOffice_Level3)

[6.2.2 ProductData类](#_Toc20110_WPSOffice_Level2) [15](#_Toc20110_WPSOffice_Level2)

[e.printStackTrace();](#_Toc28523_WPSOffice_Level2) [15](#_Toc28523_WPSOffice_Level2)

[6.2 storm](#_Toc28407_WPSOffice_Level1) [15](#_Toc28407_WPSOffice_Level1)

[6.2.1 ProductDataSpout](#_Toc28632_WPSOffice_Level2) [15](#_Toc28632_WPSOffice_Level2)

[6.2.2 SplitBolt](#_Toc11890_WPSOffice_Level2) [16](#_Toc11890_WPSOffice_Level2)

[6.2.3 NatIPCountBolt](#_Toc14389_WPSOffice_Level2) [16](#_Toc14389_WPSOffice_Level2)

[6.2.4 QqidCountBolt](#_Toc31400_WPSOffice_Level2) [16](#_Toc31400_WPSOffice_Level2)

[6.2.5 CookieDevOsCountBolt](#_Toc17100_WPSOffice_Level2) [16](#_Toc17100_WPSOffice_Level2)

[6.2.6 ComboneBolt](#_Toc28636_WPSOffice_Level2) [16](#_Toc28636_WPSOffice_Level2)

[6.2.7 SaveBolt](#_Toc31849_WPSOffice_Level2) [16](#_Toc31849_WPSOffice_Level2)

[6.3 util工具包](#_Toc5213_WPSOffice_Level1) [16](#_Toc5213_WPSOffice_Level1)

[6.3.1 DBUtils](#_Toc28407_WPSOffice_Level2) [16](#_Toc28407_WPSOffice_Level2)

[c3p0数据库连接池类](#_Toc28632_WPSOffice_Level3) [16](#_Toc28632_WPSOffice_Level3)

[6.3.2 UUIDUtils](#_Toc5213_WPSOffice_Level2) [16](#_Toc5213_WPSOffice_Level2)

[6.4 web](#_Toc7150_WPSOffice_Level1) [16](#_Toc7150_WPSOffice_Level1)

# 1引言

本详细设计说明书是针对运营商网络流量流向异常账号监测系统而编写，目的是对该项目进行详细设计，在概要设计的基础上进一步明确系统结构，详细的介绍各个系统模块，为后面的实现和测试做准备。本详细设计说明书的预期读者为本项目的小组成员以及对该系统感兴趣，在以后想对系统进行扩展和维护的人员。

## 1.1编写目的

为开发及维护人员提供编码的依据。

## 1.2背景

随着时代的发展，目前有很多不法分子根据运营商系统的漏洞，通过多账号共享的问题，给运营商利益带来了巨大的挑战。我们就是运用大数据技术，从海量的数据中挖掘出异常的网络流量流向异常账号，并对其限制，从而缓解对运营商利润的冲击。

## 1.3术语定义及说明

|  |  |
| --- | --- |
| 术语 | 解释 |
| Hadoop | 分布式系统的基础架构 |
| Flume | 一个高可用的，高可靠的，分布式的海量日志采集、聚合和传输的系统 |
| Kafka | 一个分布式的，高吞吐，可扩展的消息发布-订阅系统，用于缓存数据 |
| Storm | 一个开源的，分布式，高容错的实时计算系统 |
| 数据可视化 | 用Servlet+Jsp+Ajax+Echarts+MySQL技术，把统计出的数据展示出来 |

# 2需求概述

## 2.1任务和目标

每5分钟检测一下运营商宽带的账号信息，将一天内每5分钟的异常账号绘制成图表。

## 2.2集群设计

硬件：16台服务器，2台主节点，10台从节点，4台作业提交节点。

软件：HDFS、Storm、Flume、Kafka、Zookeeper、MySQL。

服务器的配置：Intel(R)Xeon(R) CPU E5645 @2.40GHz

内存：100G

硬盘：2T

每天处理的数据量：500G

## 2.3数据来源

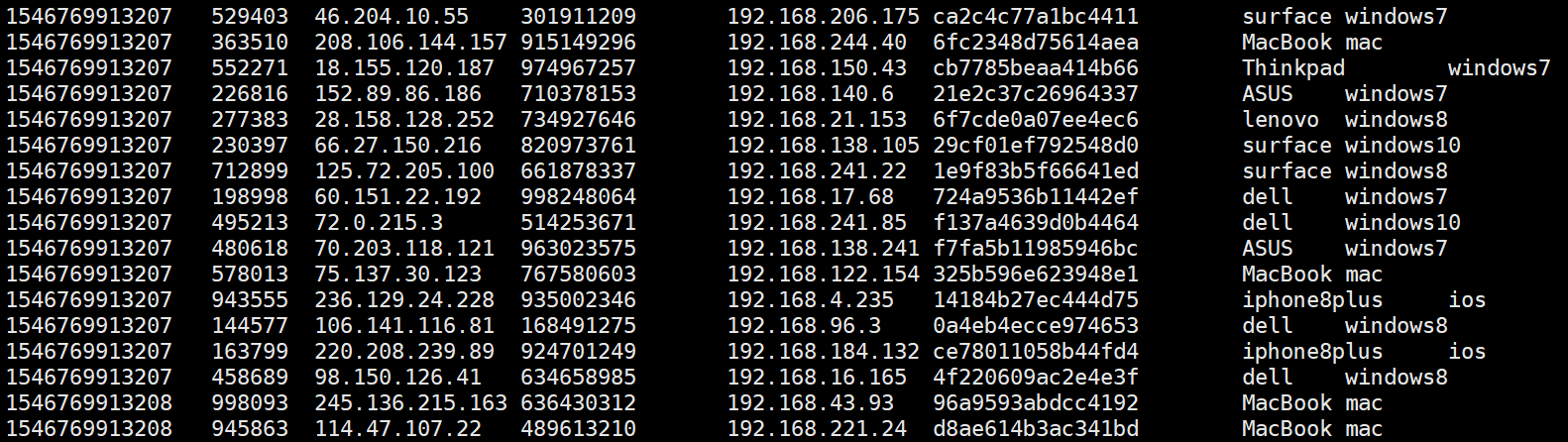
本数据来源于运营商宽带账号数据。

### 1.3.1数据描述

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 描述 |
| 1 | time | String | 数据统计时间 |
| 2 | userAccount | String | 宽带账户 |
| 3 | userIP | String | 用户以太网IP |
| 4 | qqid | String | QQ号 |
| 5 | natIP | String | 内网IP |
| 6 | cookieValue | String | Cookie值 |
| 7 | devName | String | 设备名称 |
| 8 | osName | String | 操作系统名称 |

### 1.3.2样本数据描述

该系统所需要的数据样本如下：

图3-1 数据样本

## 2.4检测方法

5分钟内，同一个账号（UserAccount）下，只要满足下面的任意一个条件，表示出现共享账户问题：

（1）natIP去重求和数 > 5

（2）qqid去重求和数 > 20

（3）cookieValue + devName + osName去重求和数 > 5

# 3设计概述

## 3.1数据存储模块详细设计

该模块的详细设计，考虑两种情况；

其一：通过Flume从运营商业务系统中采集数据存储到HDFS，通过Storm实时计算处理框架计算出5分钟内的异常账号信息，将其存储到MySQL中。

其二：通过Flume从运营商业务系统中采集数据，只不过不是存储到HDFS，而是通过Flume整合Kafka，Kafka整合Storm,为了效率将其存储到Kafka消息缓存队列中，直接通过Storm从Kafka缓存中拿取数据，提高效率，省去了写HDFS带来的性能上的开销，最后同样是通过Storm计算出5分钟内的异常账号信息，将其存储到MySQL数据库中。

下面对该系统中数据流向进行介绍。

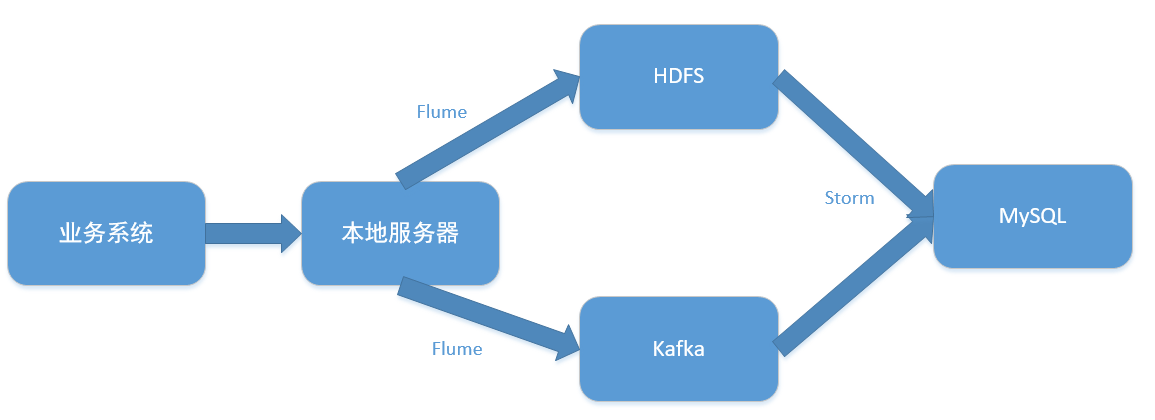


图3-2 数据流向图

紧接着，对Flume做一个具体的介绍。日志收集系统Flume 的核心就是把模拟的数据收集在一起，然后发送到目的地。Flume为了能保证传输成功，在把传送的数据传到到目的地之前，先把数据缓存起来，等到数据真正到达到目的地之后，立刻就会删除缓存的数据。Event是Flume 中数据的一个基本单位，如果说它是一个文本文件，那么通常情况下就是一行记录，就是事务中的一个基本单位。一个Event 事件就是一个数据流的最小单元，从外部的数据源而来，去向外部的目的地。

在这个日志收集系统Flume 中，它运行的核心是一个 Agent。这个系统对数据进行收集，其中包含有三个主要的核心组件，它们分别是 Source、Channel、Sink。通过这些组件，一个Event事件从Flume中的Source流向Channel，最后到Sink。

Source 组件可以接收从外部发送而来的数据。不同的Source组件之间，是可以接受不相同的数据格式。如果源数据是在目录池里，那么可以监控指定文件夹中的新文件的各种变化。如果发现目录中有新的文件产生，那么就会马上读取其它的内容。

Channel 是一个存储地也相当于缓冲池，用来接收Source的输出，一直到Sink消费掉缓冲池Channel中的数据。Channel当中的数据一直等到进入到下一个Channel缓冲池中或者进入到终端才会被删除。如果Sink组件写入失败之后，还可以自动重启，并且还不会造成数据的丢失，所以很可靠。

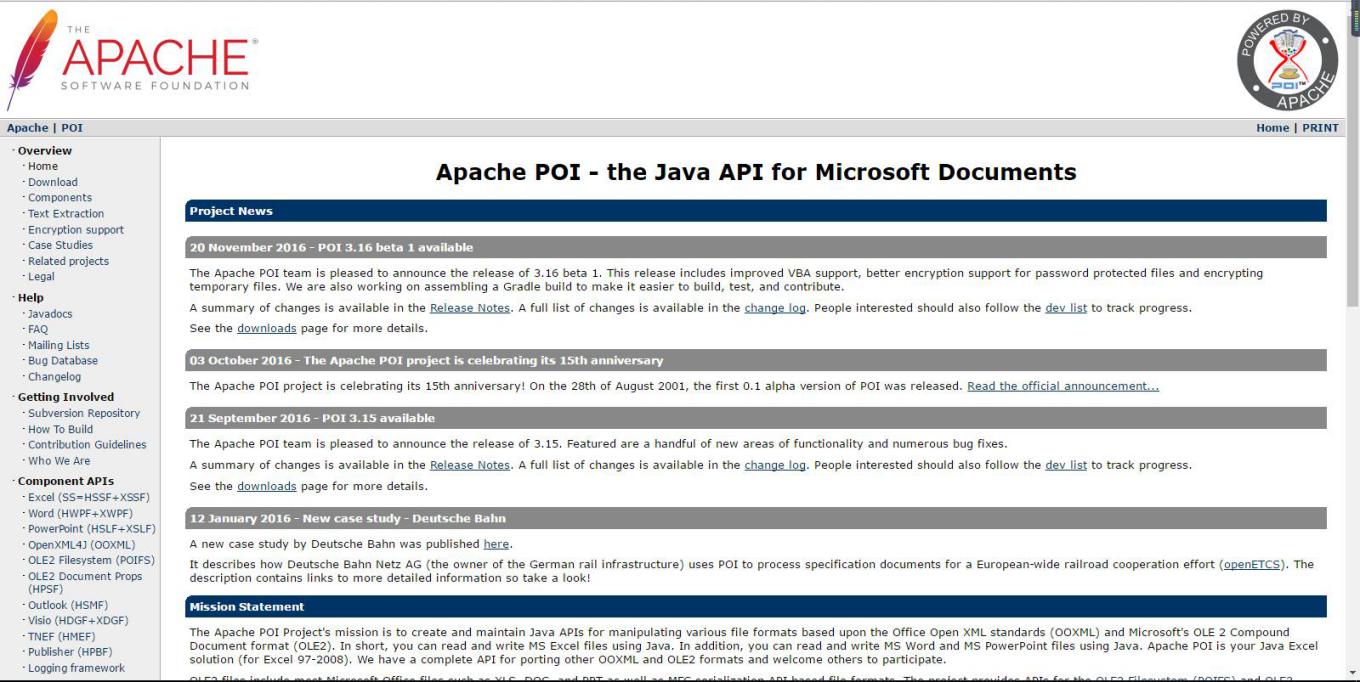
组件Sink会消费缓冲池Channel中的数据，接着发送给外部的源或者是其他Source。例如该系统最后把数据发送给Kafka。

## 3.2数据处理模块详细设计

### 3.2.1数据预处理

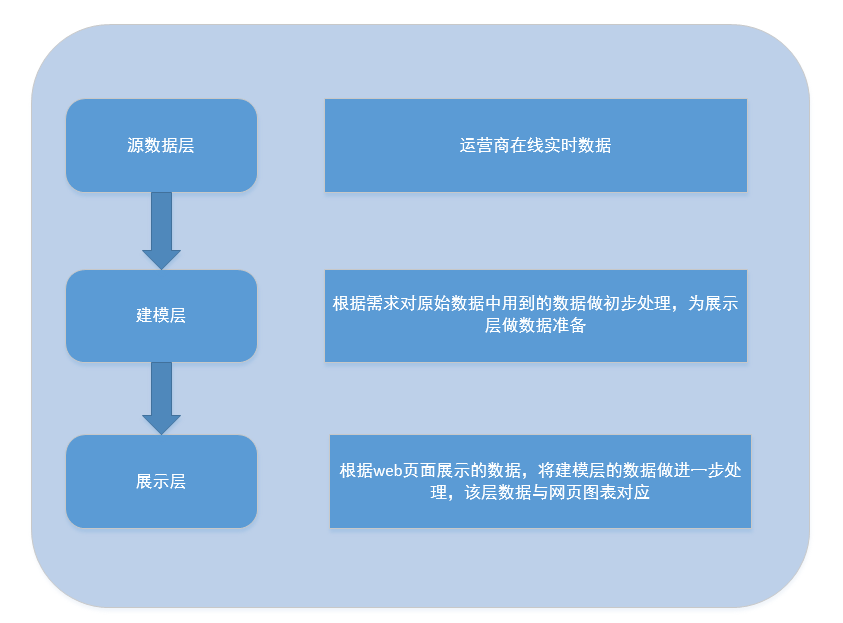
业务系统导出数据为xlsx、xls、csv等Excel格式的数据，不能直接用于数据分析。因此该项目使用Apache POI来解析这类格式的数据，将其转换为纯文本数据。

Apache POI是创建和维护操作各种符合Office Open XML（OOXML）标准和微软的OLE2复合文档格式（OLE2）的Java API。用它可以使用Java读取和创建,修改MS Excel文件.而且,还可以使用Java读取和创建MS Word和MSPowerPoint文件。Apache POI提供Java操作Excel解决方案（适用于Excel97-2008）。



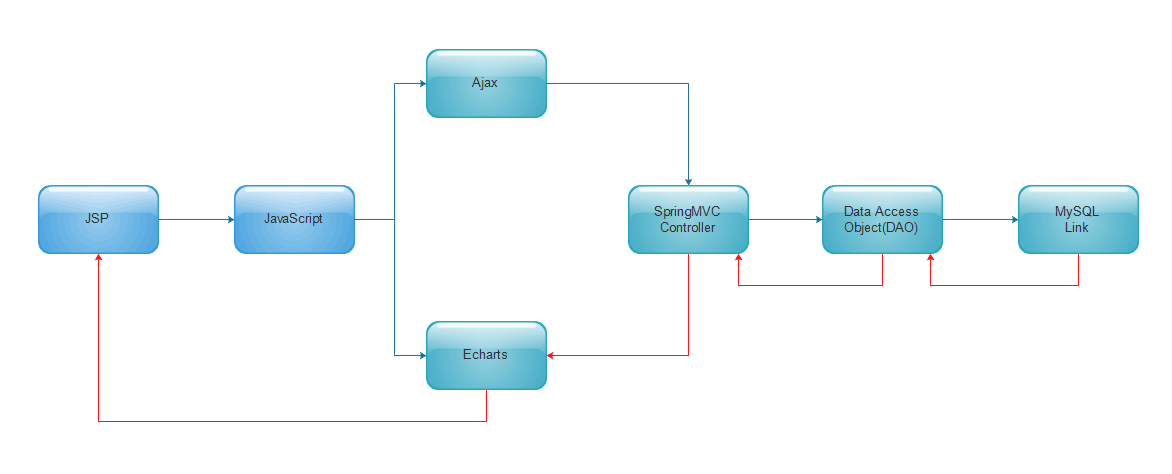
### 3.2.2数据分析

数据处理分为三个阶段：源数据层、建模层、展示层。详细如下图：



## 3.3数据展示模块详细设计

### 3.3.1数据可视化

在JSP中内嵌JavaScript，然后在JavaScript中写入Ajax，Ajax中的URL指定到SpringMVC的Controller中的XX.do，XX.do执行数据的获取-DAO(Data Access Object)，DAO通过MySQL的连接从MySQL中获取到数据。若以上步骤成功，Ajax则会获得一个JSON数据并传输到Echarts上。最终Echarts在JSP上显示。

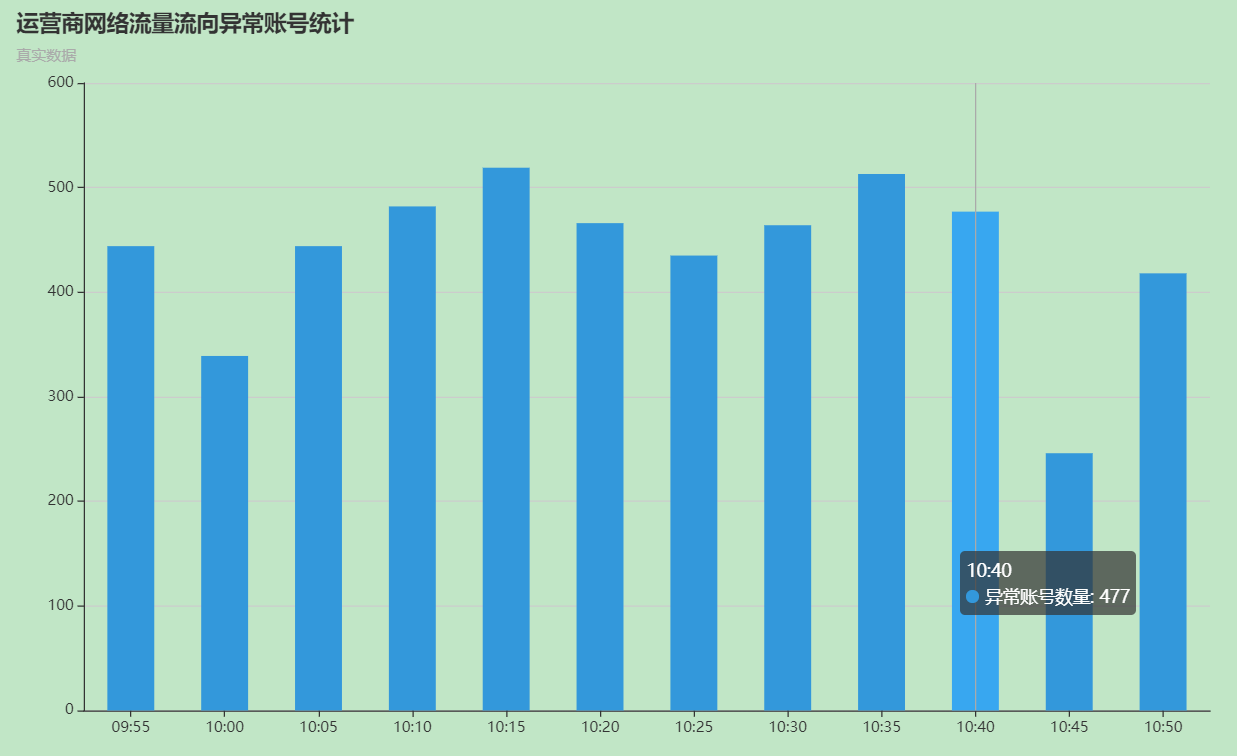
JavaScript:一种[脚本语言](http://baike.baidu.com/view/76320.htm" \t "http://baike.baidu.com/_blank)，是一种动态型、弱型、基于原型的语言，内置支持类型。它的[解释器](http://baike.baidu.com/view/592974.htm" \t "http://baike.baidu.com/_blank)被叫做JavaScript引擎，是[浏览器](http://baike.baidu.com/view/7718.htm" \t "http://baike.baidu.com/_blank)的一部分，非常广泛用于[客户端](http://baike.baidu.com/view/930.htm" \t "http://baike.baidu.com/_blank)的脚本语言，用来给[HTML](http://baike.baidu.com/view/692.htm" \t "http://baike.baidu.com/_blank)网页增加动态功能，增强页面的友好度。js语言和他语言一样，有它自己的基本数据类型，表达式和算术运算符及程序的基本程序框架。js提供了四种基本的数据类型和两种特殊数据类型用来处理数据和文字。变量提供存放信息的地方，表达式则可以完成较复杂的信息处理

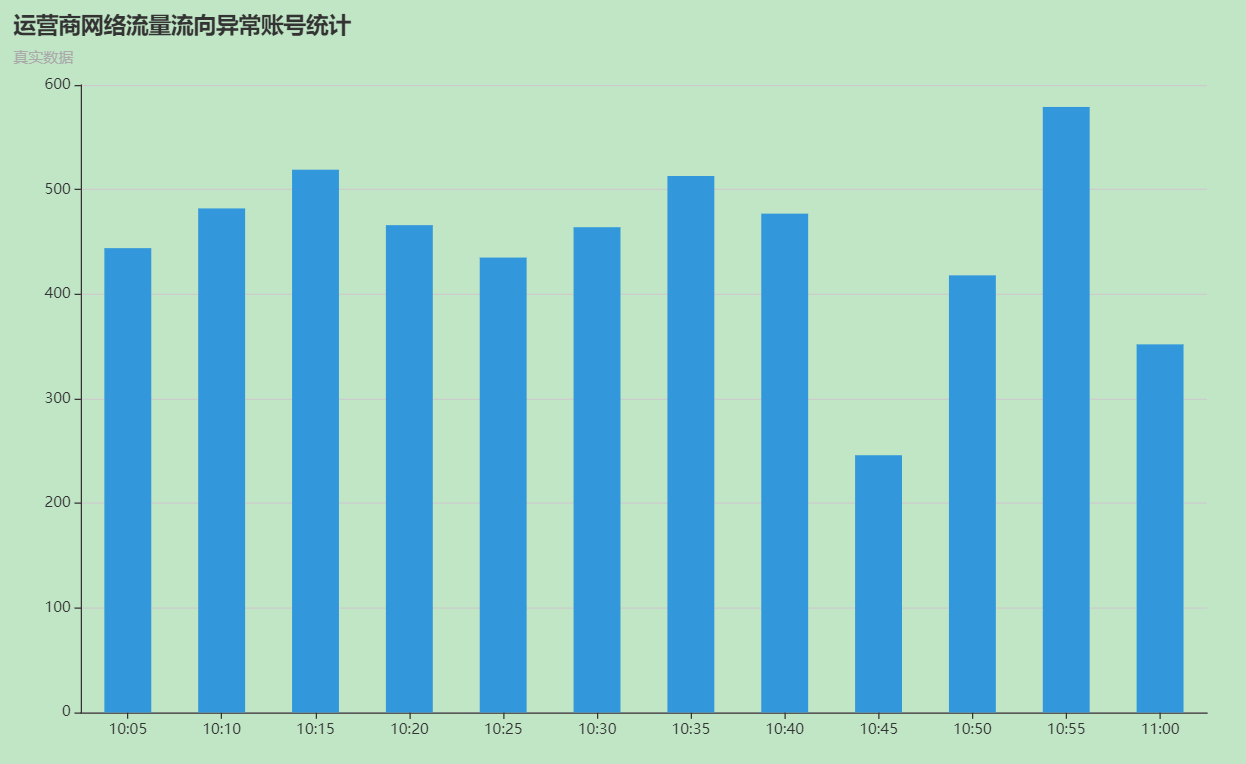
Echarts: 一个用javascript写的图表库，能非常流畅的运行在 PC 和移动设备上，兼容性非常好，底层依赖轻量级的 Canvas 类库 ZRender，提供直观，生动，交互性高，可高度个性化定制的数据可视化图表。

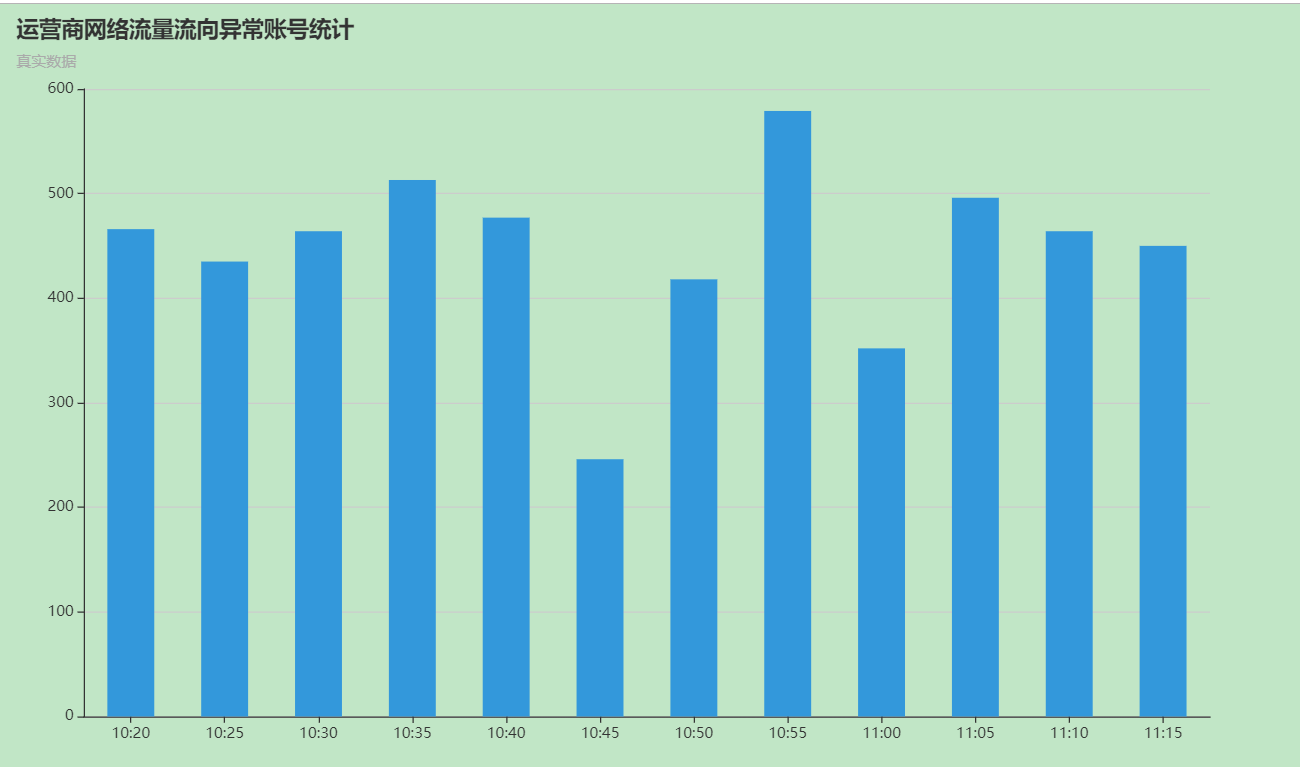
mvc框架: 英文名字是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，用一种业务逻辑、数据、界面 显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

Ajax：即“Asynchronous Javascript And XML”（异步JavaScript和XML），是指一种创建交互式网页应用的网页开发技术。AJAX是一种用于创建快速动态网页的技术。通过在后台与服务器进行少量数据交换，AJAX可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

### 3.3.2界面设计







# 4数据库设计

## 4.1表设计

create table if not exists abnormalData(

id int primary key auto\_increment,

time varchar(50),

userAccount varchar(25),

userIP varchar(15),

qqid varchar(15),

natIP varchar(15),

cookieValue varchar(64),

devName varchar(50),

osName varchar(50)

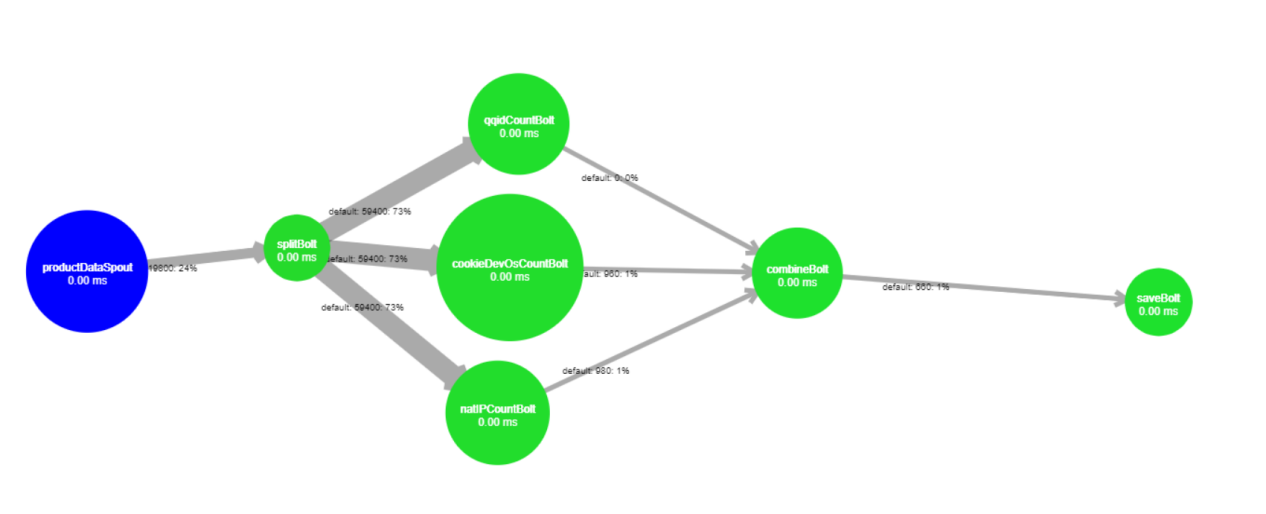
);

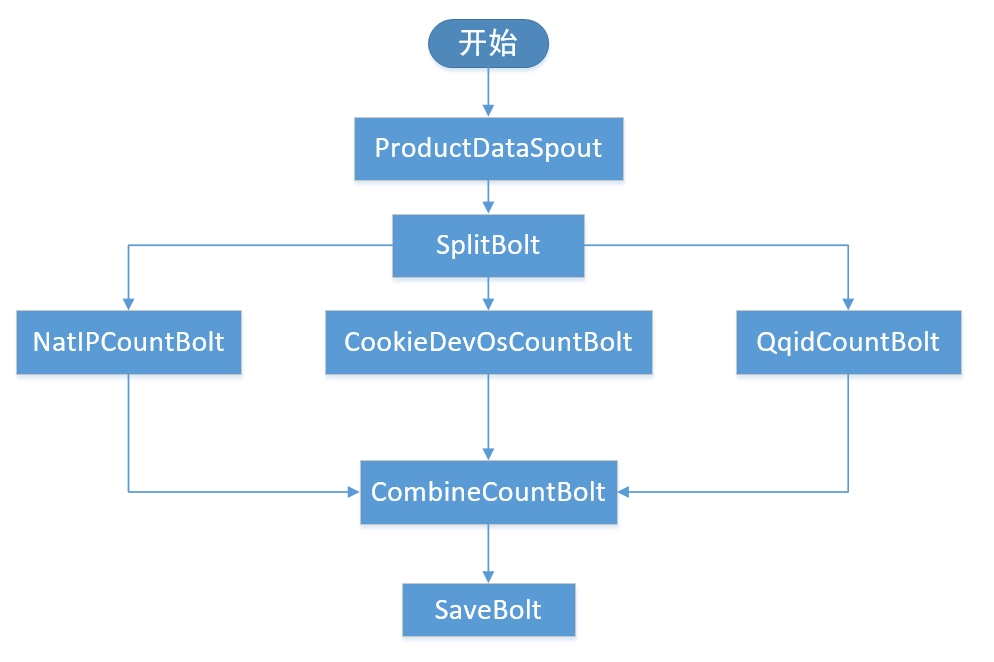
## 4.2查5分钟内的异常数据

每隔五分钟查一次数据库表的总量，减去上次查到的，就是异常的数量。

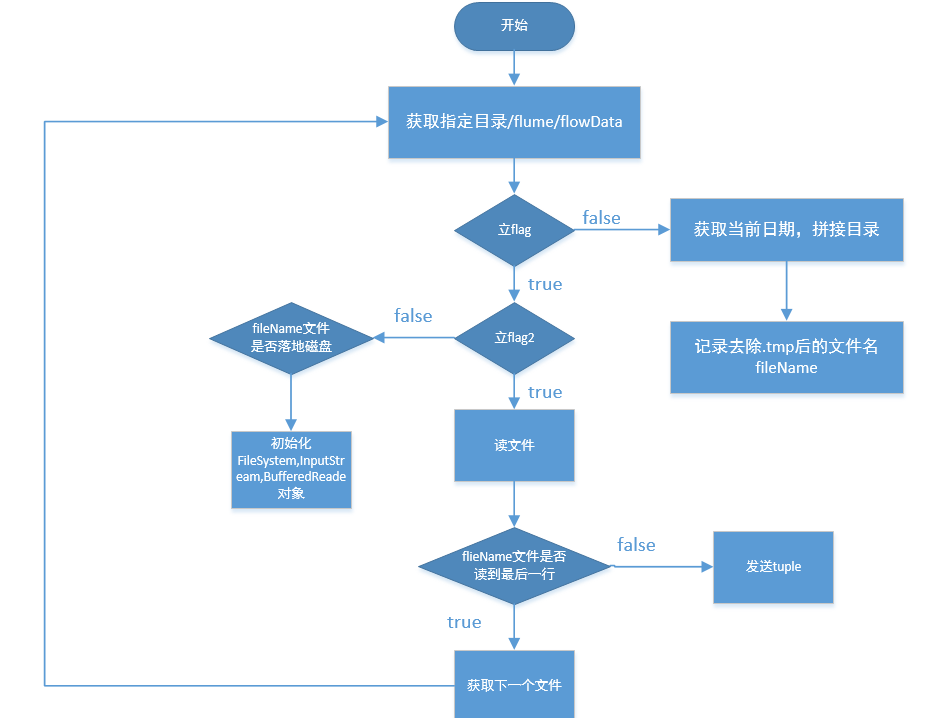
# 流程图

## Topology图

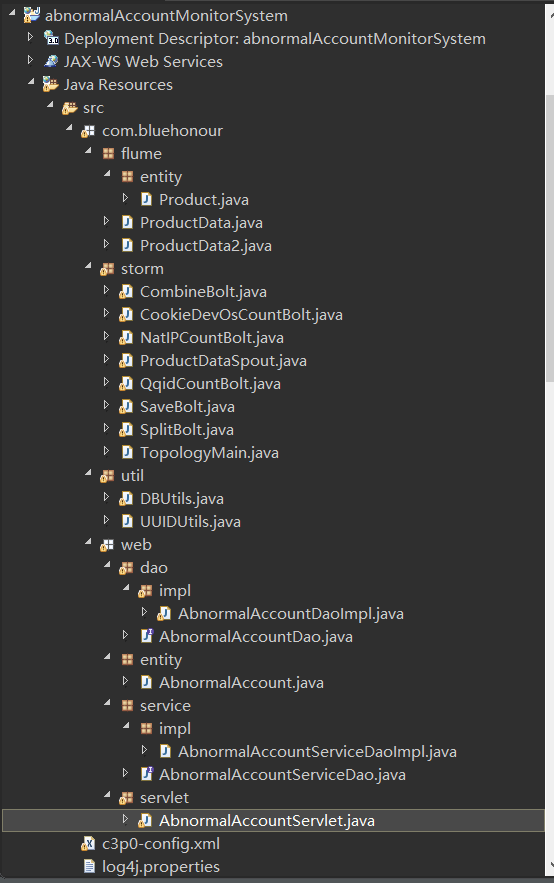




## 5.2持续读hdfs流程图



# **6类及方法的定义**



## 6.1 flume

### 6.1.1 Product实体类

**private** **String** time; //数据统计时间

**private** **String** userAccount; //宽带账户

**private** **String** userIP; //用户以太网IP

**private** **String** qqid; //QQ号

**private** **String** natIP; //内网ip

**private** **String** cookieValue; //cookie值

**private** **String** devName; //设备名称

**private** **String** osName; //操作系统名称

### 6.2.2 ProductData类

### 目的：生产数据

public List<Product> createData(int num)：一次性成产多少条数据

public void tcpSendSocket(String address, int port) throws IOException：通过tcp的方式，指定ip地址和端口，不断生成数据，flume采集

内部封装了实现多线程的一个内部类，通过new 对象的方式向不同的flume采集节点发送数据。

class TcpSendSocket2 implements Runnable {

private ProductData data = new ProductData();

String address;

int port;

public TcpSendSocket2(String address, int port){

this.address = address;

this.port = port;

}

@Override

public void run() {

while(true){

try{

data.tcpSendSocket(address,port);

Thread.sleep(1000);

} catch (Exception e){

e.printStackTrace();

}

}

}

}

## 6.2 storm

### 6.2.1 ProductDataSpout

（1）open()：初始化一个到达的task任务

（2） nextTuple()：实时读取从hdfs上生成的文件，并发送tuple到下一级SplitBolt

（3）declareOutputFields：声明当前Spoutd的tuple的发送流

（4） public void listDir()：获取当前时间目录下的.tmp后缀的文件，记录该没有.tmp的文件

### 6.2.2 SplitBolt

根据”\t”对tuple进行切分，分流发送给以下三个CountBolt

### 6.2.3 NatIPCountBolt

### 6.2.4 QqidCountBolt

### 6.2.5 CookieDevOsCountBolt

这干个Bolt内部都维护了一个map集合

public static Map<String, Set<String>> map = new HashMap<>();

key:账号，value:5分钟内登录该共享账号的条件类型

当set集合中的个数满足一定条件，说明该账号是异常的，就将其发送到下一级ComboneBolt

### 6.2.6 ComboneBolt

以上三个CountBolt类可能因为条件不一样而出现账号相同的情况，该类目的是为了去除重复的账号信息，发送该账号到下一级SaveBolt类。

### 6.2.7 SaveBolt

该类是为了将异常的账号持久化到MySQL数据库中，为之后storm处理实时数据做准备。

## 6.3 util工具包

### 6.3.1 DBUtils

c3p0数据库连接池类

public static Connection getConn()：获取c3p0连接池对象

### 6.3.2 UUIDUtils

生成UUID的工具类

public static String generateUUID(int num)：生成指定长度的UUID

## 6.4 web

### AbnormalAccountServlet

内部维护了一个全局的List<AbnormalAccount>集合，用于做缓存用

1. protected void service(HttpServletRequest request, HttpServletResponse response)：使用发射机制内部决定调用哪一个方法
2. public void queryAbnormalAccount(HttpServletRequest request, HttpServletResponse response)：查询异常账号的数量，向客户端返回一个json格式的数组

内部维护了一局部的List<abnormalAccount>集合，存储最多12条异常账户对象，含时间和对应的异常账户的数量。

通过ajax每5分钟异步请求该servlet类，每请求一次查询一次数据库，获取数据库表中的总数据量，减去上一次查询的总量，就得到了异常账户的数量，时间是ajax每请求一次获取一次本地时间，存入全局的List集合，局部的List集合每请求一次就从全局的List集合缓存中拿取数据，通过一个ObjectMapper类，将集合转成衣蛾json数组，发出去。