# 基于弹幕评论的大数据分析平台软件设计方案

开拓者开发团队

2018年6月18日

**弹幕评论的实时爬取及数据持久化**

**斗鱼弹幕服务器简介**

**斗鱼弹幕服务器**

斗鱼弹幕服务器每天接受来自全球各地若干亿次获取弹幕请求，峰值约50Gbps的数据传输量，为广大热爱视频弹幕直播的观众提供稳定高质量的服务。（引用）且对第三方接入友好，保证了此论文中弹幕获取的可行性，合法性。此次弹幕的爬取基于“斗鱼弹幕服务器第三方接入通讯协议”，利用Java多线程编程来实时爬取海量斗鱼弹幕数据。

**弹幕获取的可行性**

此次弹幕的爬取基于“斗鱼弹幕服务器第三方接入通讯协议”，利用Java多线程编程来实时爬取海量斗鱼弹幕数据。

**斗鱼弹幕评论的爬取**

**获得TCP连接**

由于斗鱼弹幕服务器通讯协议是一种基于TCP服务的应用层协议，故获得弹幕的第一步便是获得TCP连接。核心代码如下：

//获取弹幕服务器访问hostName=“openbarrage.douyutv.com”

String host = InetAddress.getByName(hostName).getHostAddress();

//建立socke连接

sock = new Socket(host, port);

//设置socket输入及输出

bos = new BufferedOutputStream(sock.getOutputStream());

bis= new BufferedInputStream(sock.getInputStream());

**保持心跳保证TCP的长连接服务**

斗鱼第弹幕第三方接入协议规定，获取弹幕信息的客户端必须要与服务器建立TCP连接，并发送登录请求验证相关信息。通过验证登录的客户端才能从后台获取到弹幕信息。而为了保持TCP的长连接服务，客户端需每隔45s发送一条心跳消息以保持长连接。核心代码如下：

//获取与弹幕服务器保持心跳的请求数据包

byte[] keepAliveRequest = DyMessage.getKeepAliveData((int)(System.currentTimeMillis() / 1000));

try{

//向弹幕服务器发送心跳请求数据包

bos.write(keepAliveRequest, 0, keepAliveRequest.length);

bos.flush();

logger.debug("Send keep alive request successfully!");

} catch(Exception e){

e.printStackTrace();

logger.error("Send keep alive request failed!");

}

**弹幕消息格式的解析**

1. **消息格式如下，例如rid为房间id，txt为弹幕内容**

type@=chatmsg/rid@=58839/gid@=-9999/ct@=8/uid@=123456/nn@

=test/txt@=666/cid@=1111/ic@=icon/sahf@=0/level@=1/nl@=0/nc

@=0/cmt@=0/gt@=0/col@=0/rg@=0/pg@=0/dlv@=0/dc@=0/bdlv@

=0/gatin@=0/gatout@=0/chtin@=0/chtout@=0/repin@=0/repout@=

0/bnn@=test/bl@=0/brid@=58839/hc@=0/ol@=0/rev@=0/hl@=0/ifs

@=0/p2p@=0/el@=eid@AA=1@ASetp@AA=1@ASsc@AA=1@AS/

1. **格式解析核心代码**

public void getServerMsg(){

//初始化获取弹幕服务器返回信息包大小

byte[] recvByte = new byte[MAX\_BUFFER\_LENGTH];

//定义服务器返回信息的字符串

String dataStr;

try {

//读取服务器返回信息，并获取返回信息的整体字节长度

int recvLen = bis.read(recvByte, 0, recvByte.length);

//根据实际获取的字节数初始化返回信息内容长度

byte[] realBuf = new byte[recvLen];

//按照实际获取的字节长度读取返回信息

System.arraycopy(recvByte, 0, realBuf, 0, recvLen);

//根据TCP协议获取返回信息中的字符串信息

dataStr = new String(realBuf, 12, realBuf.length - 12);

//循环处理socekt黏包情况

while(dataStr.lastIndexOf("type@=") > 5){

//对黏包中最后一个数据包进行解析

MsgView msgView = new MsgView(StringUtils.substring(dataStr, dataStr.lastIndexOf("type@=")));

//分析该包的数据类型，以及根据需要进行业务操作

parseServerMsg(msgView.getMessageList());

//处理黏包中的剩余部分

dataStr = StringUtils.substring(dataStr, 0, dataStr.lastIndexOf("type@=") - 12);

}

//对单一数据包进行解析

MsgView msgView = new MsgView(StringUtils.substring(dataStr, dataStr.lastIndexOf("type@=")));

} catch (Exception e) {

e.printStackTrace();

}

}

**数据持久化**

**弹幕存储难点**

由于爬取弹幕评论时，某个时间段内存在弹幕数量的峰值，且爬取过程数据量大，同步进行等特征。实时海量数据的插入必定会造成数据库系统的I/O操作过于频繁，导致客户端服务器性能下降，Mysql服务奔溃并且丢失大量弹幕数据。故弹幕存储的过程不是简单的插入操作，是一个系统化的工程。

**解决方案**

结合爬取的弹幕评论数据量大，同步性等特征。本系统采用Spring data jpa作为弹幕评论存储的持久层框架，采用java消息队列作为弹幕数据的消息中间件结合Spring data jpa中的批量插入方法实现海量数据的异步，安全的存储。

1. 弹幕存储异步的实现

消息队列

1. 减少频繁的I/O操作

合理的批量插入

**可视化管理平台的搭建及基本功能的实现**

**技术框架的使用**

**Maven**

Maven是一个项目管理和综合工具。[Maven](http://www.yiibai.com/maven" \t "_blank)提供了开发人员构建一个完整的生命周期框架。开发团队可以自动完成项目的基础工具建设，Maven使用标准的目录结构和默认构建生命周期。

Maven是基于项目对象模型（pom），可以通过一小段描述信息管理项目的构建，报告和文档的软件项目管理工具。java可以说是Java开发中必须可少的开源工具。

**Spring MVC**

Spring MVC 是一个模型 - 视图 - 控制器（MVC）的Web框架建立在中央前端控制器servlet（DispatcherServlet），它负责发送每个请求到合适的处理程序，使用视图来最终返回响应结果的概念。

Spring 框架提供了构建 Web 应用程序的全功能 MVC 模块。Spring MVC 分离了控制器、模型对象、过滤器以及处理程序对象的角色，这种分离让它们更容易进行定制。是现阶段最流行的JavaWeb开发框架。

**Mybatis-Plus**

MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis 可以使用简单的 XML 或注解来配置和映射原生信息，将接口和 Java 的 POJOs(Plain Old Java Objects,普通的 Java对象)映射成数据库中的记录。

本文中使用的Mybatis-Plus(简称MP)是一个 Mybatis 的增强工具,在 Mybatis 的基础上只做增强不做改变,是目前非常流行的持久层框架。

**H-ui**

H-ui前端框架是在bootstrap的思想基础上基于 HTML、CSS、JAVASCRIPT开发的轻量级web前端框架,开源免费,简单灵活,兼容性好,满足大多数中国网站。故本次框架搭建选择H-ui作为可视化平台的前端框架。

**Echarts**

ECharts，一个使用 JavaScript 实现的开源可视化库，可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器（IE8/9/10/11，Chrome，Firefox，Safari等），底层依赖轻量级的矢量图形库 [ZRender](https://github.com/ecomfe/zrender" \t "_blank)，提供直观，交互丰富，可高度个性化定制的数据可视化图表。

由于本系统是基于弹幕的大数据分析分析可视化平台，需要丰富的可视化图表类型从不同维度展示大数据分析结果，故在平台的搭建过程中自然少不了Echarts这样优秀的开源可视化函数库。

**平台的搭建步骤**

本平台首先通过项目管理工具Maven创建并管理项目中需要的第三方Jar包，通过集成Spring MVC框架来分离控制器、模型对象、过滤器，并作为系统处理web请求的基础，采用Mybatis-Plus实现数据持久层的设计，并通过配置文件与Spring框架完美结合，基本完成底层框架的搭建。最后引入H-ui框架与echarts完成前台设计。

**基于角色访问控制技术的权限系统的设计**

**常见系统权限设计的介绍**

权限管理系统中，目前一般有三种常用的访问控制方法。自主型访问控制方法，常用于大多数信息；系统强制型访问控制方法，主要用于多层次安全级别的军事应用；基于角色的访问控制方法，是目前公认的解决大型企业统一资源访问控制的有效方法。

**基于角色访问控制技术的权限系统设计思路**

基于对本系统的综合考量与实际场景的需求分析，认为设计功能级权限管理更适用于本系统。由此采用基于角色访问控制技术来实现权限管理。也就是通过区分平台用户的角色使得不同的角色拥有不同的功能模块来实现权限管理。

权限的验证逻辑在用户登录平台时，系统通过查看当前登录用户的角色类型，从而确定该角色包含那些功能的权限。通过将这些功能加载到系统的功能列表，从而实现本系统的权限设计。

本文中的基于角色访问控制技术的权限系统主要基于四张数据表实现。在系统开发初期首先定义系统权限表（菜单表），即将系统存在的所有功能菜单的路由存储在菜单表中。其次通过大致划分系统的功能模块将不同的菜单授权到不同的角色中，并将授权信息存储在角色权限表中。最后在添加用户时，根据需求选择该用户的角色。添加完成后，不同的用户即拥有对于系统不同的权限。

**表的设计**

1. 系统角色表（sys\_role）结构

`id` int(11) NOT NULL AUTO\_INCREMENT COMMENT '主键',

`rolecode` varchar(4) NOT NULL COMMENT '角色编码',

`rolename` varchar(255) NOT NULL COMMENT '角色名称',

`roletype` char(1) DEFAULT 'a' COMMENT '角色类型',

`sequence` int(11) DEFAULT '0' COMMENT '显示顺序',

`status` char(1) DEFAULT 'a' COMMENT '是否启用',

`remark` varchar(255) DEFAULT NULL COMMENT '备注',

`uid` int(11) NOT NULL COMMENT '操作员id',

`uptime` datetime DEFAULT NULL COMMENT '操作时间',

1. 系统用户表（sys\_user）结构

`id` int(11) NOT NULL AUTO\_INCREMENT COMMENT '主键',

`tel` varchar(50) NOT NULL COMMENT '手机号码',

`pwd` varchar(200) NOT NULL COMMENT '密码',

`rolecode` varchar(4) NOT NULL COMMENT '角色编码',

`name` varchar(50) NOT NULL COMMENT '真实姓名',

`qq` varchar(20) DEFAULT NULL COMMENT 'QQ号码',

`uid` int(11) NOT NULL COMMENT '操作员id',

`uptime` datetime DEFAULT NULL COMMENT '操作时间',

1. 系统权限表（sys\_privilege）

`id` int(11) NOT NULL AUTO\_INCREMENT COMMENT '主键',

`privilegecode` varchar(4) NOT NULL COMMENT '权限编码',

`privilegename` varchar(255) NOT NULL COMMENT '权限名称',

`parentcode` varchar(4) NOT NULL COMMENT '父编码',

`url` varchar(255) DEFAULT NULL COMMENT '地址',

`iconfont` varchar(30) DEFAULT NULL,

`isshow` char(1) DEFAULT 'a' COMMENT '是否显示',

`sequence` int(11) DEFAULT '0' COMMENT '显示顺序',

`remark` varchar(255) DEFAULT NULL COMMENT '备注',

`uid` int(11) NOT NULL COMMENT '操作员id',

`uptime` datetime DEFAULT NULL COMMENT '操作时间',

1. 角色权限表（sys\_role\_privilege）

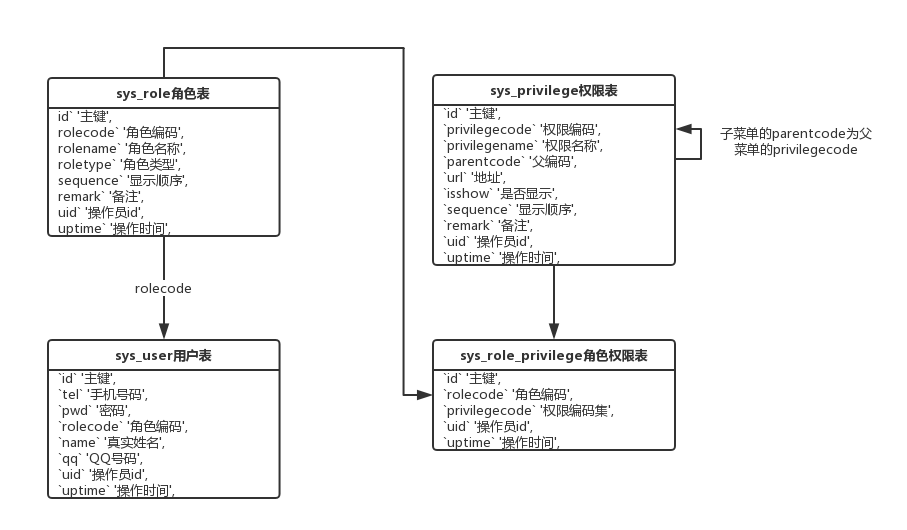
`id` int(11) NOT NULL AUTO\_INCREMENT COMMENT '主键',

`rolecode` varchar(4) NOT NULL COMMENT '角色编码',

`privilegecode` varchar(4) NOT NULL COMMENT '权限编码集',

`uid` int(11) NOT NULL COMMENT '操作员id',

`uptime` time DEFAULT NULL COMMENT '操作时间',



**授权逻辑流程**

