

基于 ThinkPHP 和 Redis 的高并发微信公众号开发的研究与实现

李建华 夏 汛 罗明全

(泸州职业技术学院信息工程系 四川 泸州 646600)

摘 要 随着“互联网+”的快速发展,微信公众号已经逐渐融入大家的生活。微信公众号基础平台仅提供一些较为常用的服务,而对于一些特殊的需求,则需要使用微信平台封装的接口进行二次开发,扩展公众号功能,定制个性化服务。针对这种情况,以某公司价值文化宣传为例,考虑到用户需求特殊和开发周期等要求,设计并实现基于 ThinkPHP 和 Redis 快速搭建高并发微信公众号二次开发平台。实验结果表明,使用 ThinkPHP 框架可以减少开发难度,使用 Redis 缓存技术可以提高系统并发度。其开发技术具有通用性,能够在高并发的微信公众号二次开发方面有所借鉴。

关键词 ThinkPHP Redis 公众号二次开发 快速建站 高并发

中图分类号 TP311 **文献标识码** A **DOI**:10.3969/j.issn.1000-386x.2019.02.019

RESEARCH AND IMPLEMENTATION OF HIGH CONCURRENT WECHAT PUBLIC ACCOUNT DEVELOPMENT BASED ON THINKPHP AND REDIS

Li Jianhua Xia Xun Luo Mingquan

(Department of Information and Engineering, Luzhou Vocational and Technical College, Luzhou 646600, Sichuan, China)

Abstract With the rapid development of the Internet plus, the WeChat public account has gradually integrated into everyone's life. The WeChat public account basic platform only provides some common services. For some special needs, it needs to use the interface encapsulated by the WeChat platform for secondary development, expand the public number function, and customize personalized services. In view of this, we took a company's value culture propaganda as an example. Considering the special requirements of users and the development cycle, we designed and implemented a secondary development platform of high-concurrency WeChat public account based on ThinkPHP and Redis. The experimental results show that using the ThinkPHP framework can reduce the difficulty in development, and the Redis cache technology can improve the system concurrency. The development technology is universal and can be used for reference in the secondary development of the high-concurrency WeChat public account.

Keywords ThinkPHP Redis Secondary development of WeChat public account Rapid establishment High concurrence

0 引 言

随着信息技术的发展,互联网在给我们提供丰富信息的同时也给我们生活带来了巨大改变,信息时代下人们已经离不开互联网。与此同时,类似于 QQ、微信、微博等社交网络应用充斥着整个互联网,其中以微信应用最为典型。微信公众号是微信公众平台的简称,利用微信公众号可以进行自媒体活动,用户可在微

信平台上和目标受众用户用语言、文字、视频等多种方式进行交流,形成一种微营销方式,使得微信公众号在消息互动的同时也在传递着商业价值。

据《2017 年微信经济数据报告》可知,截至 2017 年底微信公众号已超过 1 000 万个^[1],微信公众号已经成为了公司对外宣传的标配。公司通过微信公众号向受众用户及时推送消息和业务,极大地拉近了企业和用户之间的距离;用户通过关注企业微信公众号,可以充分了解企业业务动向和产品文化。

微信公众号基础平台已经为用户提供了一些基本的功能,如文章推送、粉丝管理等。但是对于部分特殊功能,则需要开发者自行扩展。微信公众平台已将自带的资源服务封装成接口,这就使得二次开发成为可能。同时随着粉丝数量越来越庞大,一些类似抽奖、抢红包等高并发功能的二次开发成为必要。

本文以某企业为例,为了有效地对外宣传其企业文化,扩大市场知名度,需要每位员工发现身边的新鲜事,并在信息平台上发布文章,经过领导审核之后,通过微信公众号推送出去。其他员工对该文章进行转发,本公司员工以外的粉丝阅读/转发之后,该员工获得相应的积分(本公司员工之间相互阅读/转发不积分)。同时系统需要具有统计功能,管理员能够按照时间段、部门、转发文章数量进行统计,以作为季度考核的指标。同时该公司员工规模庞大(几千人),可能会出现同一时刻多人阅读/转发同一篇文章,并发度要求高。通过分析,发现微信公众平台自带的功能并不能满足用户需求,需要对微信公众平台进行扩展。本文针对用户需求,采用开源框架 ThinkPHP 和 Redis 缓存技术,快速搭建一套高并发的微信公众号二次开发平台。测试结果表明,采用 ThinkPHP 框架能够减少开发难度和缩减开发周期,采用 Redis 缓存技术能够实现 1 000 人左右的并发数,基本上能满足用户需求。本文介绍的开发技术具有通用性,在快速搭建高并发的微信公众号二次开发方面有一定的借鉴意义,能够节约开发时间和成本。

1 关键技术介绍

1.1 ThinkPHP 框架介绍

ThinkPHP 是一款基于 Apache2 开源协议的 Web 应用程序框架,自从 2006 年诞生以来,就受到了开发者的极大关注^[2]。其作为轻量级的 PHP 框架,目前发展迅速,已经是三大主流框架之一(Laravel、Yii 和 ThinkPHP)。ThinkPHP 采用 MVC (Model-View-Controller) 开发模式思想,将传统的混杂开发模式转变成界面显示,逻辑控制和数据处理三层分离,每层专注于自己的开发,并形成模块化的程序块,代码重用度较高。各层之间耦合度较低,仅仅通过简单的接口进行数据流通。MVC 也是现在较为主流的开发模式,它使得 Web 开发更简单快捷,能够实现一处开发,多处使用的效果,在各个开发语言中都能见到其身影,比较适合大中型项目应用的开发。

1.2 Redis 缓存技术介绍

Redis 缓存也是缓存的一种形式,它是一个开源的基于内存存储的数据库,可支持如链表、集合等多种数据类型^[3]。可以使用简单的 Redis 命令快速处理大量的数据,实现对数据高并发读写。传统的 Web 数据库数据一般存储在硬盘中,当同时有大量用户操作数据库时,数据库的连接池会承受较大的压力,并且读写硬盘需要较大的 IO 开销,带来较大的访问延迟,用户的体验感会大打折扣。Redis 是一个内存数据库,数据存储在内存,所以相比传统的数据库技术,内存数据库在读写速度方面更有优势。根据官网测试数据显示:在 Linux2.6 下,50 个并发进程执行 100 000 次请求,其读取速度在十万次/秒左右,写入速度在八万转/秒左右。由此可见,Redis 读写效率较高。

2 系统设计与实现

2.1 系统总体设计

考虑到建站的快速性和访问的并发性,本文采用 ThinkPHP 和 Redis 技术相结合,实现一个高并发的微信公众号二次开发框架,供读者参考。本系统根据用户角色划分不同权限控制,主要分为前台模块和后台两个模块。后端模块主要采用开源框架 Hui-admin^[4]实现对用户、文章、组织架构等相关数据管理,所有的操作都在 PC 端完成。同时实现对关系数据库 MySQL 和内存数据 Redis 的管理。前端主要实现文章管理和用户个人信息管理,通过与企业服务号对接,实现微信公众平台和第三方服务器之间数据流通。前端和后端通过 Redis 实现快速交互,整个架构如图 1 所示。

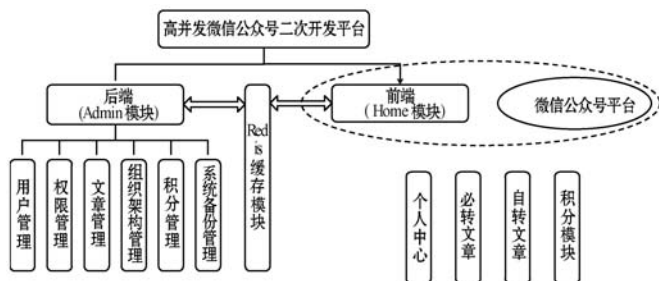


图 1 系统架构图

2.2 系统实现

2.2.1 微信公众号与第三方系统对接

在微信公众号二次开发过程中,微信官方平台服务器其实就是一个消息转发器,所有的用户请求和第三方服务器的响应都经过微信服务器进行转发^[5],数据流如图 2 所示。

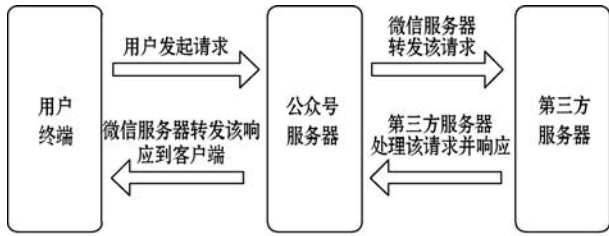


图2 微信公众号二次开发数据流

- (1) 用户在微信终端向微信公众号发送一条消息(发起请求)。
- (2) 微信公众号服务器接收到该消息之后,将此消息转发给第三方服务器(转发请求)。
- (3) 第三方服务器解析公众号服务器发来的请求,并将响应数据打包成消息,返回给公众号服务器(响应请求)。
- (4) 用户收到响应数据之后,按照既定格式展现到页面上(转发响应)。

首先在通过认证的微信公众号(服务号)页面添加菜单项,然后将该菜单和本平台入口文件(URL)进行绑定。在公共配置文件/Thinkphp/Application/Common/Config下的 config.php 文件中配置微信公众号相关参数。其中 WX_APPID 是微信公众号中第三方用户唯一凭证,相当于第三方系统接入的微信平台的登录用户名。

```
<? php
return array(
//mysql 数据库定义
'DB_TYPE' => 'mysql',
'DB_HOST' => 'localhost',
'DB_NAME' => 'db_xxxx',
'DB_USER' => 'root',
'DB_PWD' => '123456',
'DB_PORT' => 3306, //端口号一般 3306
'DB_PREFIX' => 'tb_', //表前缀
...
'WX_APPID' => 'xxxxxx', //微信公众号 ID
'WX_SECRET' => 'xxxxxx', //应用密钥
'WEB_URL' => 'http://xxx.com', //第三方系统
);
? >
```

2.2.2 获取微信用户相关信息

微信公众号平台使用 OAuth 2.0 开放协议进行身份认证,该协议允许第三方应用获取该用户在某一网站上存储的私密资源,无需使用用户名和密码进行身份验证,微信公众号使用 OAuth 2.0 进行验证一般需要三步(类似于三次握手协议):

第一步:用户经过第三方服务器向微信公众号服务器发送一个访问授权请求(该请求附带 WX_APPID、重定向地址、响应消息类型等参数),该请求到达微信服务器之后,微信服务器解析该请求,将当前的 URL 重定向到第三方服务器,并向第三方服务器返回该用户的 CODE 值。

第二步:第三方服务器收到该 CODE 值之后,将 CODE、WX_SECRET 和 WX_APPID 等作为参数,再次向微信公众号服务器发起请求,请求返回 access token 参数。

第三步:第三方服务器收到该 access token 参数之后,将 access token 和 WX_APPID 等作为参数,最后一次向微信公众号平台发起请求,请求该用户在微信平台上存储的私密信息(如微信号、头像、性别等)。整个过程如图3所示。

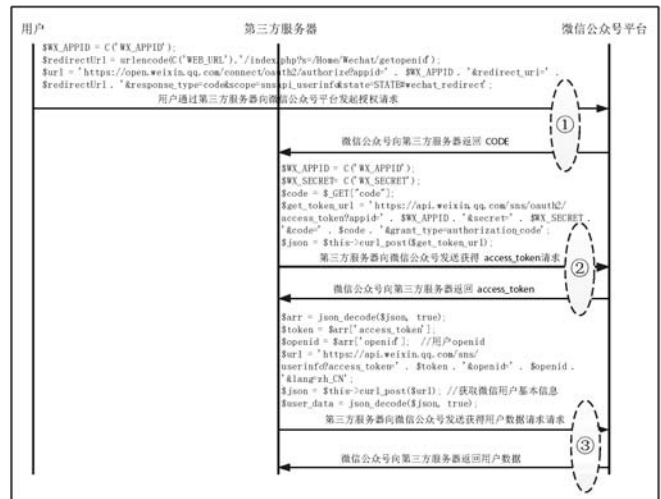


图3 OAuth 2.0 验证流程图

在 OAuth 2.0 验证过程中,使用到了 curl_post() 方法。该方法是一个访问 HTTP 协议接口,用来取得某 URL 对应的页面内容。核心代码实现如下:

```
$ ch = curl_init(); //初始化 CURL
curl_setopt( $ ch, CURLOPT_SSL_VERIFYPEER, false); //屏蔽检测
curl_setopt( $ ch, CURLOPT_URL, $ url);
curl_setopt( $ ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt( $ ch, CURLOPT_CONNECTTIMEOUT, 120); //设置请求超时时间
file_contents = curl_exec( $ ch); //获取 URL 内容
curl_close( $ ch); //关闭连接
```

file_contents 变量中存储的就是 ch 变量(URL) 对应页面内容打包的 JSON 数据,再经过 json_encode 方法将 JSON 数据转化,即可得到用户信息数组。

第三方服务器经过授权之后,能够得到用户的 OpenID 和相关的信息,将 OpenID 值写进 Redis 缓存数

据和 MySQL 数据库的字段,表明该用户已经和第三方系统进行了绑定,可以直接和第三方服务器进行数据交换。

2.2.3 Redis 安装与配置

Windows 环境下,安装 Redis 后,需要手动添加 php 的 redis 拓展。首先到 Redis 官网下载 php_igbinary.dll 和 php_redis.dll 两个库文件,然后在 php.ini 文件中新增 extension = php_igbinary.dll 和 extension = php_redis.dll 两个扩展,实现 PHP 解析器对 Redis 的支持。

Redis 一般是通过命令来操作数据,为了简化缓存读写操作,ThinkPHP 把所有的缓存机制封装成了一个 S 方法,该方法使用简单,跟 Session 使用类似。通过封装成方法,用户不必关注数据操作的具体细节,而重在注重业务处理逻辑,大大提高开发效率。安装好 Redis 之后,在/Application/Common/Common/function.php 中配置 Redis 数据库,如下所示:

```
function REDISCONFIG() {
    //数据缓存配置
    REDIS = S( array( 'type' => 'redis', 'host' => '127.0.0.1', 'port' => '6379', 'prefix' => 'lz_', ));
    return $ REDIS;
}
```

2.2.4 文章转发操作

本系统需要记录用户文章转发之后的阅读量,该阅读量需要记录在首次转发者的记录中。所以需要在文章内容页面记录转发数据(文章 ID 以及转发者的 OpenID),这里使用微信 JS-SDK 提供的接口。

为了能准确统计文章转发数量,转发时需要使用微信 JS-SDK 获取 signature 签名。需要注意的是,这里要求提前得到 jsapi_ticket,它是公众号调用微信 JS 接口时使用的一种网络凭证^[6]。正常情况下,jsapi_ticket 的有效期为 7 200 秒,通过 access_token 来获取。

```
if(S('ticketStr')){//如果 ticketStr 存在
    $ ticketStr = S('ticketStr');
} else {
    $ accessToken = $ this -> curl_post("https://api.weixin.qq.com/cgi-bin/token? grant_type = client_credential&appid = ". C('WX_APPID')." &secret = ". C('WX_SECRET'));
    $ accessTokenArr = json_decode( accessToken );
    $ accessTokenStr = $ accessTokenArr -> access_token;
    $ ticketJson = $ this -> curl_post("https://api.weixin.qq.com/cgi-bin/ticket/getticket? access_token = ". $ accessTokenStr. "&type = jsapi");
    $ ticketArr = json_decode( $ ticketJson );
    $ ticketStr = $ ticketArr -> ticket;
```

```
S('ticketStr', $ ticketStr, 7100);
}
$ str = "jsapi_ticket = ". $ ticketStr. "&noncestr = XXXXXXXX
XXXXXXXXX&timestamp = 1421142450&url = http://xxxx. xxxx.
com". $ _SERVER["REQUEST_URI"];
$ signature = sha1( $ str );
$ this -> assign( 'signature', $ signature );
```

接下来,在文章详情页面使用 signature 签名,进行文章转发操作步骤。

(1) 在文章内容页面引入 jweixin - 1. 2. 0. js 文件。

(2) 所有需要使用 JS-SDK 的页面必须先进行配置,配置文件如下所示:

```
wx.config( {
    debug: true,
    appId: '{ :C('WX_APPID') }',
    timestamp: 1421142450,
    nonceStr: 'XXXXXXXXXXXXXXXXXX', // 必填,生成签名的随机串
    signature: '{ $ signature }', // 必填,微信加密签名
    jsApiList: [
        'onMenuShareTimeline',
        'onMenuShareAppMessage',
        'onMenuShareQQ',
    ]
});
```

(3) 分享接口实现:在这里我们使用三种分享方式:“分享到朋友圈”、“分享给朋友”和“分享到 QQ”,这三种分享方式在实现上异曲同工,其中“分享到朋友圈”接口实现逻辑如表下所示。分享之后第三方服务器需要拿到被分享文章的 id 和被分享人的 OpenID。

```
wx.onMenuShareTimeline( {
    title: '{ $ info. title }',
    link: "",
    imgUrl: '{ :C('WEB_URL') } /Public/{ $ info. cover }',
    success: function ( ) {
        var $ arid = $. trim( $ '#ar_id'). val();
        var $ weid = $. trim( $ '#we_id'). val();
        $. post( '{ :U('Article/forwarded') }',
            { arId: $ arid, weId: $ weid }, function() {} );
    },
    cancel: function ( ) {
    }
});
```

2.2.5 Redis 数据流的控制

(1) Redis 数据流 首先在 PC 后台,用户发布文章,管理员进行审核,审核通过之后将文章 ID 和文章

内容分别写进 articleID 和 articleContent_ar_id 表中。articleContent_ar_id 表在高并发环境下用户方便快速读取文章内容并呈现, articleID 表方便在文章修改时判断文章是否存在。

在微信前端,首先判断用户是否已绑定自己的微信号,如果没有绑定,则通过注册方式绑定。在注册的同时将自己的工号和 OpenID 写进 num-openid 表和 staffArray 表,主要用来判断访问者是否是本公司员工。如果已经绑定,则阅读文章(本公司员工互相阅读文章,文章阅读数不变)并将文章转发到微信好友、微信朋友圈或者 QQ 中。如果自己的好友(非本公司员工)阅读文章之后,则该员工的文章阅读数增加 1,同时将访问者的 OpenID 写进文章 id 为 ar_id 的缓存表中,防止重复阅读增加阅读数。文章的访问数和自己的阅读数是数据统计的依据,同时也是评优评奖的重要指标,整个流程如图 4 所示。

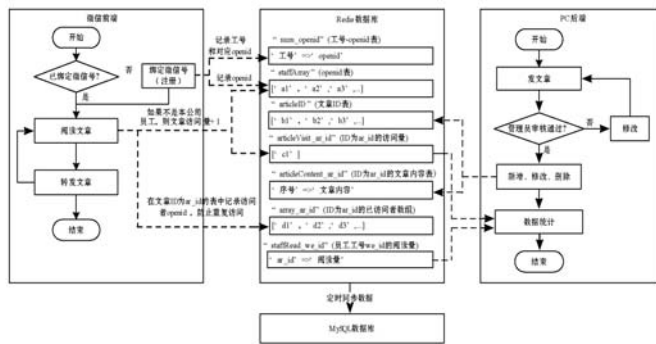


图4 Redis数据流图

(2) 数据备份 Redis 是缓存数据库,具有一定的生命期,为了保证数据安全,本项目采用定时备份的方式将 Redis 数据库的关键数据及时备份到 MySQL 数据库中。

3 测试

软件测试是软件开发过程中不可缺少的环节,通过测试,目的是为了发现系统与用户需求不符或矛盾的地方,进而进行更改和完善^[7]。功能测试和性能测试是一般软件项目需要关注的两个测试方面。功能测试是软件测试中最基本的测试,主要验证系统是否符合需求说明。性能测试主要测试系统在某些极端条件下是否满足需求,其中压力测试是一种典型的性能测试。在 Web 系统中,压力测试主要是在系统同时接收大量用户和请求时,检测 Web 系统的响应。

本系统测试环境:ubuntu1204 虚拟机,200 GB 硬盘,8 GB 内存,2 Gbit/s 带宽。Apache2.2, PHP5.3.10, MySQL5.5。

3.1 功能测试

按照用户的需求文档,设计了 30 个测试用例,主要对文章审核流程和文章转发统计这两个业务逻辑复杂点进行测试。测试结果良好,测试结果和预期效果一致,测试通过率 100%。

3.2 性能测试

对于 Web 服务器来说,用户请求响应时间是一个重要指标,它是评价系统性能的关键参数。本系统的性能关键点(并发点)主要是文章阅读和转发操作,使用 Apache 自带的 ab 并发测试命令对该并发点进行测试,测试请求数为 10 000,结果发现在并发数 1 000 左右的时候,系统平均响应时间在 16.7 ms 左右。也就是说 1 000 人同时阅读文章或者转发文章的时候,平均等待时间在 16.7 ms 左右。目前该公司人数在 4 000 左右,该并发数和响应时间基本上能够满足性能要求。

4 结语

随着“互联网+”的发展,微信公众号已经成了一种大众化的沟通工具。为了满足用户的特殊需求,微信公众号二次开发成为必要。同时随着粉丝数量越来越庞大,一些类似抽奖、抢红包等高并发的二次开发也迫在眉睫。本文以某企业为例,采用开源框架 ThinkPHP 框架和 Redis 缓存技术,搭建一套基于微信公众号的高并发二次开发通用平台,该开发技术具有通用性,在快速开发高并发的微信公众号方面有一定的借鉴意义,能够节约时间和经济成本。

参考文献

- [1] 2017 年微信数据报告[EB/OL]. <https://wenku.baidu.com/view/a8122d5077c66137ee06eff9aef8941ea66e4b11.html>.
- [2] 王俊芳,李隐峰,王池. 基于 MVC 模式的 ThinkPHP 框架研究[J]. 电子科技,2014,27(4):151-153,158.
- [3] 郎泓钰,任永功. 基于 Redis 内存数据库的快速查找算法[J]. 计算机应用与软件,2016,33(5):40-43,52.
- [4] H-ui. admin V3.1 后台模版[EB/OL]. <http://www.h-ui.net/H-ui.admin.shtml>.
- [5] 谢远超. 微信公众号信息服务平台的设计与实现[D]. 广州:中山大学,2014.
- [6] 微信公众平台技术文档[EB/OL]. https://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1445241432.
- [7] 王葵蓁. 软件测试理论初步框架[J]. 计算机科学,2014,41(3):12-16,35.