# 论坛附件系统设计方案

概述

本文档描述了论坛中用来处理用户上传的图片，文本以及其他数据的设计方案。

1. 文档历史

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 作者 | 更新的概要 | 注意 |
| 2014-09-09 | Kid | 无 | 初稿 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目的

用户开发和使用本系统的用户进行沟通，确定产品的形态和功能，描述产品的设计思想。

适用对象

开发和使用本系统的相关人员。

备注

目录

[论坛附件系统设计方案 1](#_Toc404073313)

[1附件上传系统 3](#_Toc404073314)

[总流程图 3](#_Toc404073315)

[附件请求系统 7](#_Toc404073316)

[总体架构： 7](#_Toc404073317)

[附件统计系统 9](#_Toc404073318)

[总体架构 9](#_Toc404073319)

[安装使用文档 11](#_Toc404073320)

[参考文档 16](#_Toc404073321)

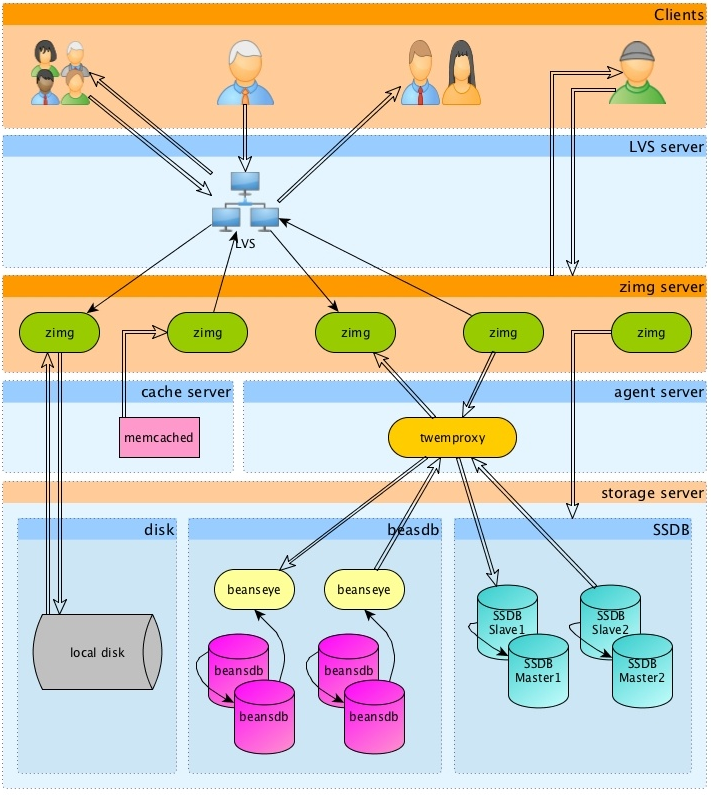
# 1附件上传系统

## 总流程图

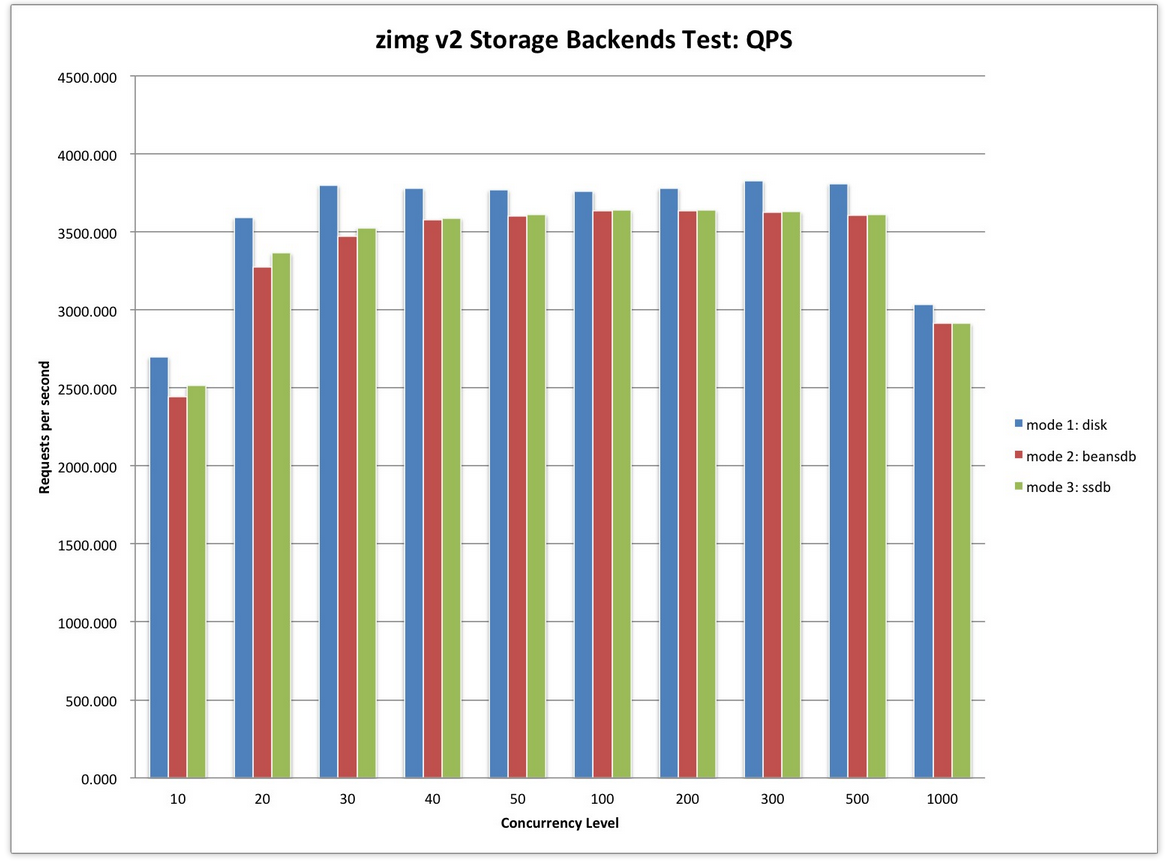


附件上传系统基于开源的ZIMGv3.1图片系统进行二次开发而来，ZIMG图片系统具有高并发，灵活，轻量的特点，可以满足目前的业务要求，同时可以通过源码快速的进行二次开发。

ZIMG的体系架构：



ZIMG的性能测试数据：



ZIMG已经实现了图片的全部处理逻辑，包括图片的上传，压缩，裁剪和存储，以及图片的访问等功能。

详细请参考文档： http://zimg.buaa.us/

现在需要完善的是需要扩展上传接口，支持文本和一些其他文件，比如bt种子文件的上传等等

上传接口：

定义了用户上传数据的格式和协议

协议：http协议

格式：POST方式或者raw方式

目前ZIMG将上传日志数据存储在nosql中，不利于统计，所以需要在MYSQL中在存储一份上传的日志，用来做数据分析适用，需要在处理完成图片之后入库，可以在zimg中扩展，也可以通过zimg调用另外一个程序的接口来处理。

DB入库扩展

返回接口：

目前ZIMG的返回接口是html格式的，不利于用户上传之后对结果的分析，需要改成返回JSON格式的数据。

格式大概如下：

{

‘code’:0,1…n,

‘msg’:’ok|error’,

‘url’:’url’ //MD5 值

}

参考文档：

V3.1文档：<http://zimg.buaa.us/documents/#design-and-architecture>

V3.1说明文档：<http://zimg.buaa.us/documents/design_of_zimg_new_generation/#top>

V2存储介绍：<http://zimg.buaa.us/documents/Distributed_Image_Storage_Server_zimg/>

V1设计文档：http://zimg.buaa.us/documents/Architecture\_Design\_of\_Image\_Server/#top

# 附件请求系统

## 总体架构：



附件请求系统可以通过zimg来扩展，也可以通过搭建nginx来处理

接收到用户的请求之后，在缓存中查询文件的实际路径，如果不成功，查询DB ，获取到文件的信息之后，根据文件的信息进行处理。如果是图片，检查图片是否存在，如果不存在，则用源图进行裁剪或者压缩生成新的图片尺寸并保存，将文件存入到文件系统中，并返回结果，同时异步调用统计接口进行统计操作。

调用统计接口为url方式

需要做异常处理，当ssdb或者memcache异常不能访问的时候，需要提供默认的图片，而不是404错误。

# 附件统计系统

## 总体架构



统计接口通过调用数据分析程序完成数据分析的过程，并将数据分析的结果记录到流水表中，分析的数据包括来源的url，文件的md5sum值，提取时间，所在版块等等

统计进程通过定时执行程序来进行统计计算，并将结果存入到统计表中。

统计的维度包括时间，当日，当周，当月等等

点击量

板块

……

统计查询接口：用户提交查询条件，调用查询程序返回统计结果展示在web

# 安装使用文档

step1：安装基础依赖文件

sudo yum install openssl-devel cmake libevent-devel libjpeg-devel giflib-devel libpng-devel libwebp-devel ImageMagick-devel libmemcached-devel

step2：安装openssl

wget http://www.openssl.org/source/openssl-1.0.1i.tar.gz

tar zxvf openssl-1.0.1i.tar.gz

cd openssl-1.0.1i

./config shared --prefix**=**/usr/local --openssldir**=**/usr/ssl

make **&&** make install

step3：安装cmake

wget http://www.cmake.org/files/v3.0/cmake-3.0.1.tar.gz

tar xzvf cmake-3.0.1.tar.gz

cd cmake-3.0.1

./bootstrap --prefix**=**/usr/local

make **&&** make install

step4：安装libevent

wget http://cloud.github.com/downloads/libevent/libevent/libevent-2.0.21-stable.tar.gz

tar zxvf libevent-2.0.17-stable.tar.gz

cd libevent-2.0.17-stable

./configure --prefix**=**/usr/local

make **&&** make install

step5：安装libjpeg-turbo

wget https://downloads.sourceforge.net/project/libjpeg-turbo/1.3.1/libjpeg-turbo-1.3.1.tar.gz

tar zxvf libjpeg-turbo-1.3.1.tar.gz

cd libjpeg-turbo-1.3.1

./configure --prefix**=**/usr/local --with-jpeg8

make **&&** make install

step6：安装webp

wget http://downloads.webmproject.org/releases/webp/libwebp-0.4.1.tar.gz

tar zxvf libwebp-0.4.1.tar.gz

cd libwebp-0.4.1

./configure

make

sudo make install

step7：安装imagemagick

wget http://www.imagemagick.org/download/ImageMagick-6.8.9-7.tar.gz

tar xzvf ImageMagick-6.8.9-7.tar.gz

cd ImageMagick-6.8.9-7

./configure --prefix**=**/usr/local

make **&&** make install

step8：安装libmemcached

wget https://launchpad.net/libmemcached/1.0/1.0.18/+download/libmemcached-1.0.18.tar.gz

tar zxvf libmemcached-1.0.18.tar.gz

cd libmemcached-1.0.18

./configure -prefix**=**/usr/local

make **&&**　make install

step9：安装memcached

wget <http://www.memcached.org/files/memcached-1.4.21.tar.gz>

tar memcached-1.4.21.tar.gz

cd memcached-1.4.21

./configure --prefix=/usr/local/memcached

make && make test

sudo make install

step10：安装zimg3

tar zxvf zimg.s8bin.tar.gz

cd zimg.s8bin/bin/

vim conf/zimg.lua

修改下面的参数：

mysql\_ip = '10.146.180.13'

--mysql端口

mysql\_port = 3306

--mysql用户名

mysql\_user = 'testuser'

--mysql密码

mysql\_pwd = 'root1234'

--mysql数据库名称

mysql\_db = 'test'

修改红色的部分为对应的服务器信息，同时在对应的DB服务的库下面执行下面的SQL语句：

CREATE TABLE `fileinfo` (

`id` int(11) unsigned NOT NULL auto\_increment,

`src` int(11) NOT NULL,

`md5sum` char(32) NOT NULL,

`filetype` tinyint(1) NOT NULL,

`uid` int(11) unsigned NOT NULL,

`tid` int(11) unsigned NOT NULL,

`pid` int(11) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

step11：新建对应的目录

midir –p /data/www1/img

step11:新建memcache运行的用户

useradd memcache

step12：启动memcache

/usr/local/memcached/bin/memcached –umemcache

step13：启动zimg3

cd zimg.s8bin/bin/

./zimg3 conf/zimg.lua &

# 参考文档

http://www.php-fig.org/