

文件共享管理系统的设计与实现

摘要：本文件共享管理系统解决了用户在搜索文件不需要下载文件到本地硬盘后才能查看文件的详细内容的弊端；解决用户在搜索关键字不明确条件下无法搜索到自己需要的文件弊端；解决了系统用户并发量增加后服务器宕机而无法提供服务的弊端；解决服务器面对大量业务处理数据读取速度过慢的弊端；解决用户登录密码在 http 协议下被轻易窃取的弊端。

本文件共享管理系统，通过构建服务器集群和分布式数据库集群来满足互联网用户的并发访问，提高用户请求的响应时间；系统依赖于缓存服务器提高对数据的访问速度；通过对用户上传文件的处理使得文件可以被用户在线浏览；保证用户文件的安全性采用单独的服务器进行存储；通过使用全文检索框架对用户输入关键字智能分析。

本系统采用恰当的技术来解决上述问题从而为用户提供良好的操作体验。

关键字：全文检索；安全；在线预览；文件存储

The Design and Implementation of File Sharing Management System

Abstract: This file sharing management system to solve the user in the search for files do not need to download the file to the local hard disk to check the file details of defects;To solve the user in the search keywords don't conditions to search the drawbacks of the file to your questions;Solved the system user concurrency value increased after the server goes down to the disadvantages of service;Solve the server in the face of the disadvantages of a large number of business processing data read speed too slow;To solve the user login password under the HTTP protocol is easy to steal.

This file sharing management system, this paper builds and distributed database cluster server cluster to meet concurrent access to Internet users, to improve the response time of user request;System is dependent on the cache server to increase the speed of access to data;Based on the user to make the file upload files can be read by users online;To ensure the safety of the user files using a separate server to store;Through the use of full text retrieval framework for user input keyword intelligence analysis.

This system USES the appropriate technology to solve the above problems so as to provide good experience for the user.

Key words: Full Text Retrieval;Safety;.Online Preview;File Storage

目 录

1 绪论	1
1.1 项目背景	1
1.2 开发目的与意义	1
2 系统开发工具与技术	2
2.1 MySQL 数据库	2
2.2 Memcached 技术	2
2.3 Lucene 技术	2
2.4 Nginx 技术	2
2.5 Servlet 技术	3
2.6 JSP 技术	3
2.7 Struts 技术	3
2.8 Hibernate 技术	3
2.9 Spring 技术	4
3 系统规划与系统分析	1
3.1 系统的总体结构	1
3.2 数据库需求分析	1
4 系统设计与实现	4
4.1 系统用户功能设计	4
4.2 数据库表存储设计	4
4.3 系统的架构设计	8
4.4 系统 web 界面设计	9
5 系统测试	16
5.1 性能测试	16
5.2 性能测试结果	17
5.3 功能测试	17
5.4 功能测试结果	17
6 软件开发中主要解决的问题	20
6.1 帐号登录的安全性	21
6.2 搜索的精准与快速	21
6.3 文件的处理	22
6.4 敏感词过滤	22
6.5 网站高并发	22
6.6 服务器集群下状态共享	23
6.7 高并发下磁盘读写速度	23
6.8 分布式数据库事务一致性	23
6.9 集群下 quartz 协调处理	24
7 结论	24
参考文献	25
致 谢	错误！未定义书签。

1 绪论

1.1 项目背景

随着计算机技术的飞速发展，计算机应用的普及，计算机软件系统对社会，经济带来越来越重要的影响，极大程度上提高了人们的工作效率。尤其近些年来，随着互联网+的提出，我国政府将互联网作为当前信息化发展的核心特征并与工业，商业，金融等服务业全名融合。不仅提高了各行各业的服务质量和工作效率，而且降低了各行各业的不必要的成本开销，这对促进整个社会的良性发展有着推动作用。

马云说过计算机所处的时代正由 IT 转向 DT 时代，现在的软件系统已经不仅仅是为局部的几十个人使用了，而是面对整个互联网用户提供服务，传统的架构模式已经无法满足时代的需要，如何设计一个良好的软件架构面向整个互联网用户提供服务，这就是本系统的研发所在。

1.2 开发目的与意义

一个系统要想获得用户的青睐和高比例的留存率，能否为用户提供满意的服务是一个很重要的因素，本系统做到想用户所想，在页面上减少用户对本系统的学习成本，高质量的响应用户的搜索内容，使得用户能够搜索到符合自己满意的文件减少用户不必要的下载和流量消耗。互联网系统必须保证用户信息的安全性，如果不能给用户安全感，这个系统就是一个失败的项目，系统的开发目的就是有人愿意使用。传统系统我们经常会遇到当有很多人和自己同时登陆系统操作时，自己的操作结果往往要等待几分钟才有响应，这样的系统在设计上就已经是失败的，一个好的系统必须可以在任何情况下都可以在用户可以等待的范围内及时响应。

本系统的开发就是对为了响应互联网+的号召，对传统系统的一次改变；保证用户的信息的安全性，防止用户的个人信息在网络传输中被人截取利用；保证用户的响应都可以在合理的时间内进行响应；保证用户能够在下载文件前就充分知道文件内容是自己所需要的等。

2 系统开发工具与技术

本文件共享管理系统的开发工具使用 Eclipse，数据库采用的是 MySQL，服务器采用 Tomcat6.0，Nginx。在开发中采用 Java 语言进行开发，项目整体使用 Struts2，hibernate，spring 三大框架作为开发的基本环境，使用 Lucene 全文检索框架进行文件的搜索，MyCat 中间件处理分布式数据库和分布式事务问题等问题，OpenOffice 技术对 office 文件转换为 swf 文件时数据内容的提取，页面采用 JSP+HTML+CSS+DIV 等技术，Ajax 进行请求的异步发送，Java Mail 技术对用户账号进行验证，Memcached 技术实现内存级缓存，同时在项目开发中使用 SVN 进行项目的版本控制。下面经对这些平台和技术进行简单的介绍。

2.1 MySQL 数据库

MySQL 是一个采用了关系模型来组织数据的关系型数据库，而且它的源代码是开放的，谁都可以在一定的条件下无偿使用，而且可以对应用程序进行改造。它使用普及率广泛；对数据的处理十分迅速；支持在多种 OS 中运行；支持多种开放语言。

2.2 Memcached 技术

Memcached 是一个性能很高的内存对象缓存系统，它通过在内存里维护一个统一的巨大的 hash 表，来存储各种格式的数据，包括图像、音频以及数据库检索的结果等。简单的说 Memcached 就是将数据调用到内存中存储起来，然后需要对应数据时从内存中读取而不是从硬盘中读取，从而大大提高读取速度。Memcached 以守护程序的方式可以运行在一个或多个服务器上，随时接收客户端的连接和操作。

2.3 Lucene 技术

Lucene 是一个开放源代码的全文检索引擎框架，它提供了完整的查询引擎和索引引擎，部分文本分析引擎。全文检索是计算机程序通过扫描文章中的每一个词，对每一个词建立一个索引，指明该词在文章中出现的次数和位置。当用户查询时根据建立的索引查找，类似于通过字典的检索字表查字的过程。

2.4 Nginx 技术

Nginx 是一个高性能的 HTTP 和反向代理的服务器，反向代理方式是指以代理服务器来接受 internet 上的连接请求，然后代理服务器将请求转发给内部网络

上的服务器，并将从服务器上得到的结果返回给 internet 上请求连接的客户端，此时代理服务器对外部网络的客户端而言像一个服务器。

2.5 Servlet 技术

servlet 是一个用 Java 编写的服务器端程序。servlet 运行在 Java 服务器上。Java servlet 可以动态的扩展服务器的能力，采用请求-响应模式提供 web 服务。

当 web 服务器接收到一个 http 请求时，他会先判断请求内容,如果请求的内容是静态网页数据,web 服务器会自行处理并返回响应信息;如果请求的内容涉及动态数据,web 服务器会将请求转给 servlet 容器, servlet 容器就会找到对应的处理该请求的 servlet 实例,最后 servlet 实例将结果送回 web 服务器,再由 web 服务器传回客户端。

2.6 JSP 技术

JSP 全称是 Java Server Pages, 它是一种用于开发动态 web 资源的技术。

JSP 技术的特点在于, 开发者写 jsp 就像在写 html, 但它相比 html 只能为用户提供静态数据, Jsp 技术允许在页面中嵌套 java 代码, 为用户提供动态数据。

不管是在开发中使用 JSP 或是 Servlet, 它们都可以用于开发动态 web 资源。但由于这 2 门技术各自的特点, 在长期的软件开发中, 工程师逐渐把 servlet 作为 web 应用中的控制器组件来使用, 而把 JSP 技术作为数据显示模板来使用。

2.7 Struts 技术

struts2 是一个遵循 MVC 结构的框架, 采用了更加松耦合的设计, 使得系统的 action 不与 servlet API 耦合, 使系统可以从 B/S 结构转换为 C/S 结构; action 不需要与 WebWork 耦合, 代码的重用率更高; 对 JSP,Velocity,FreeMarker 等表现层技术提供支持; 使用拦截器作为增强处理, 以用户的业务逻辑控制为目标, 创建一个控制器代理。

2.8 Hibernate 技术

hibernate 是轻量级 javaEE 应用的持久层解决方案。Hibernate 对 JDBC 访问数据库的代码做了封装, 负责 java 对象的持久化, 大大简化了数据访问层繁琐的重复性代码; Hibernate 是一个基于 jdbc 的主流持久化框架, 是一个优秀的 orm 实现, 建立面向对象的域模型和关系数据模型之间的映射, 连接 java 应用和数据库的中间件, 它很大程度的简化了 dao 层编码工作, 封装对数据库的访问细节,

使业务逻辑层更专注于实现业务逻辑；Hibernate 的性能非常好，因为它是一个轻量级框架。映射的灵活性很出色。它支持很多关系型数据库，跨平台性很强，从一对一到多对多的各种复杂关系；hibernate 的一级缓存，二级缓存，查询缓存，使访问数据的效率提高很大。

2.9 Spring 技术

Spring 是根据大量开发中出现的通用步骤对这些通用步骤进行抽取的框架，因此它完成了，他留给开发者的仅仅是与特定应用相关的部分，开发者不需要再编写系统开发中需要解决的通用问题，从而大大提高开发者的开发效率。

Spring 为企业应用的开发提供了一个轻量级的解决方案。该解决方案包括：基于依赖注入的核心机制，基于 AOP 的声明式事务管理，与多种持久层技术的整合，以及优秀的 web MVC 框架等。Spring 致力于 Java EE 应用各层的解决方案，而不仅仅是专注于某一层的方案。

Spring 具有低侵入式的设计，代码的污染极低；独立于各种应用服务器，基于 Spring 框架编写的应用，可以真正实现一处编写，到处运行的承诺；Spring 的 DI 容器降低了业务对象替换的复杂性，提高了组件之间的解耦；Spring 的 AOP 支持允许将一些通用任务如安全，事务，日志等进行集中式处理，从而提供了更好的复用；Spring 的 ORM 和 DAO 提供了与第三方持久层框架的良好整合，并简化了底层的数据库访问。

3 系统规划与系统分析

3.1 系统的总体结构

根据文件共享管理系统的设计需求，确定本系统平台的整体运作模式要求用户通过 web 端进入文件共享系统的首页系统搜索自己需要的文件，用户根据系统呈现的数据选择满足自己需要的，进而点击该文件在线预览以验证该文件是否符合自己的需要，从而再进行文件的下载。最后用户也可以上传自己的文件资源供其他用户检索下载，从而达到资源共享的目的。

按照每个用户可以操作的功能来理清系统整体功能，系统功能模块图如图 3.1 所示的。

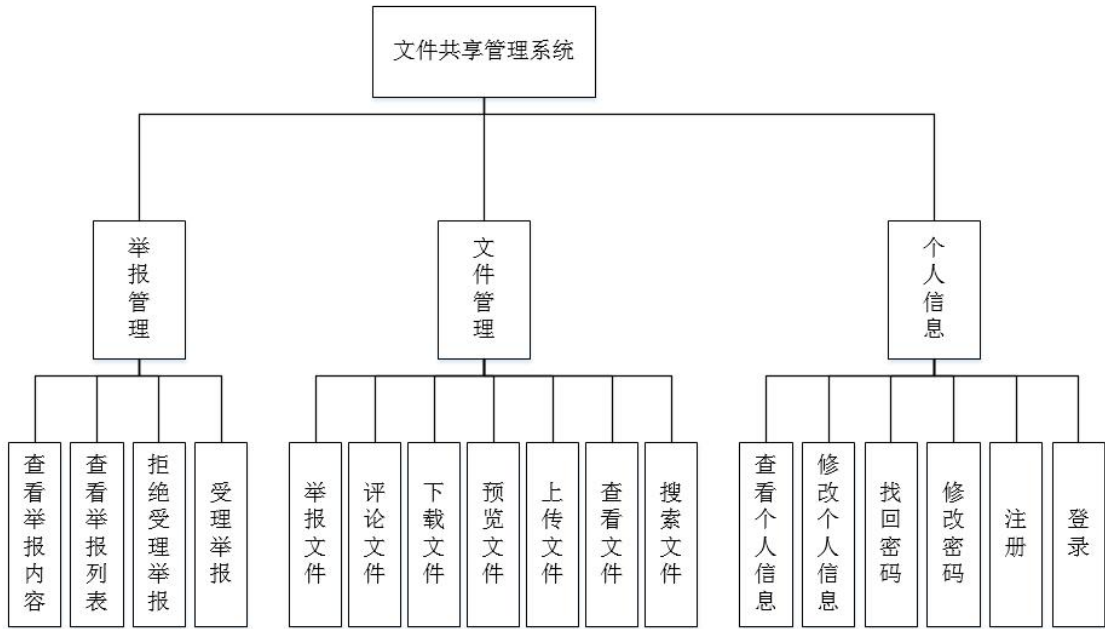


图 3.1 系统功能图

3.2 数据库需求分析

开发应用系统创建一个合格的数据库是一个复杂而有重要的任务，数据库表设计的好坏直接影响到系统进入开发阶段的开发效率。所以我们设计一个完整的数据库应用环境时这要求我们更加广泛关注问题本身。同时针对数据设计本身，我们要避免两个方面的缺陷，即冗余和完整性，一个好的设计方案不可能或者很少有重复的信息，当然，也不可能使单位的某个方面难以建模。作为数据库设计的概念阶段，充分考虑系统数据需求，保证设计的数据完整刻画用户对系统的数据的需求，还需要以系统的方式定义数据库用户的数据需求，并定义满足这些需

求的数据库结构。

本系统的概念设计如下：

- 实体：共有 8 个实体，分别是用户、管理员、文件、文件地址、评论、举报、用户文件状态和注册。
- 联系：各个实体通过主外键属性关联。
- 属性：各个实体属性如表 3.1 所述。

表 3.1 文件管理系统实体属性表

实体名称	实体属性
管理员	adminId, 账号, 密码, 昵称
用户	userId, 账号, 密码, 头像, 电子邮箱, 生日, 是否冻结, 恶意行为次数, 昵称
文件	fileId, 文件标题, 文件介绍, 下载次数, 好评次数, 差评次数, 上传时间, 文件总大小, 文件是否被封, 上传用户
文件地址	fileResourceId, 文件名称, 文件链接地址, 文件预览地址, 文件大小, 上传文件
评论	commentId, 评论内容, 评论等级, 评论时间, 评论人, 评论文件
举报	reportId, 举报用户, 被举报文件, 举报内容
用户文件状态	usestateId, 用户, 下载文件, 是否评论, 是否举报
注册	usereamilId, 注册账号, 注册密码, 邮箱地址, 创建时间

通过对文件管理系统的数据库概念描述，根据其中的实体，联系和属性，构建文件管理系统数据实体联系图，如图 3.2 所示。

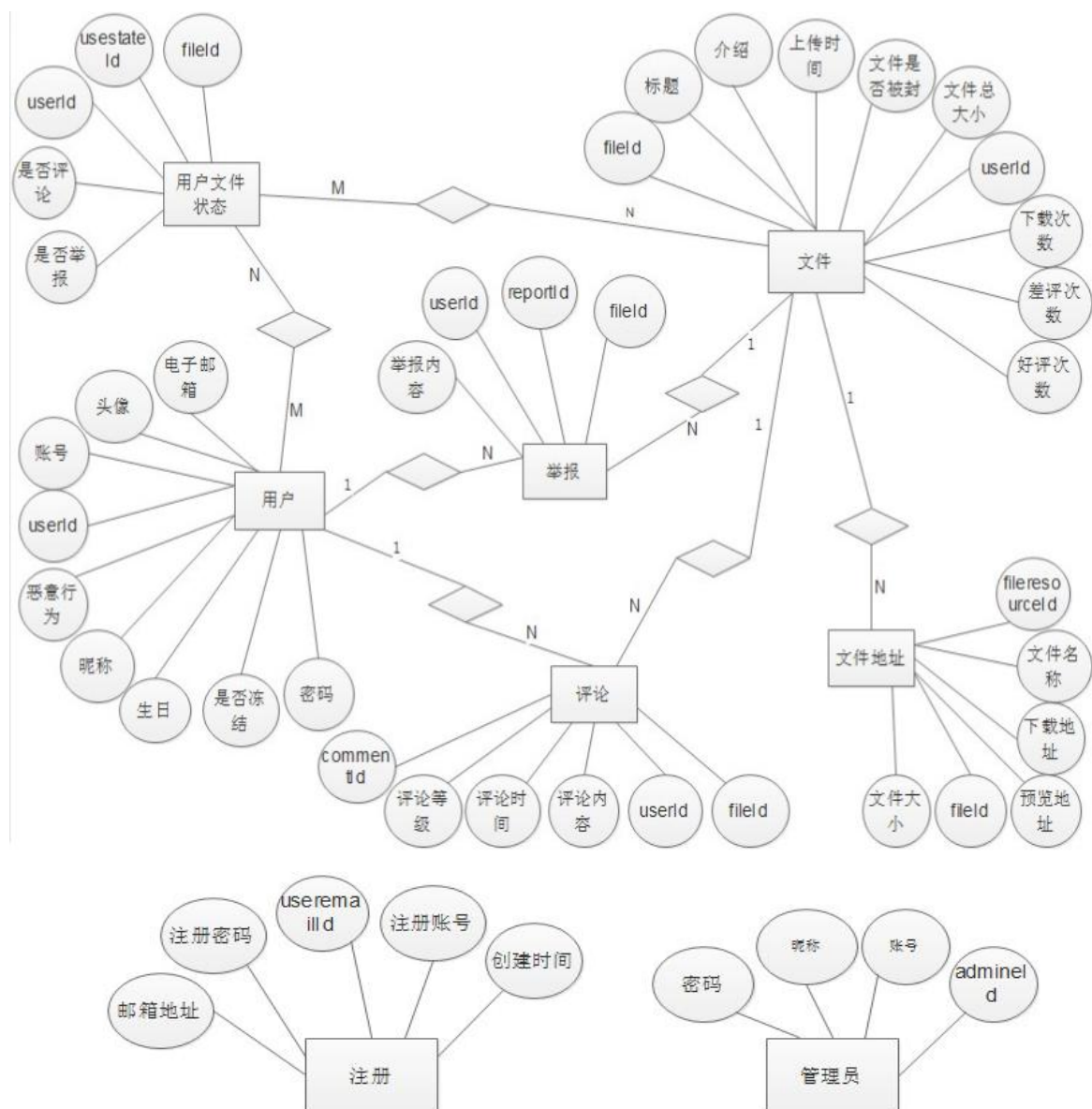


图 3.2 数据库实体 ER 图

4 系统设计与实现

4.1 系统用户功能设计

系统用户分为三种角色：游客、用户、管理员。这三个角色的功能如下所述：

(1)游客

搜索资料：游客可以在搜索框输入要搜索文件的关键字进行搜索；

查看文件信息：游客可以点击搜索结果列表的文件查看该文件的详细信息以及其他用户对该文件的评论；

预览文件：游客可以对查看的文件进行在线预览；

(2)用户

搜索资料：用户可以在搜索框输入要搜索文件的关键字进行搜索；

查看文件信息：用户可以点击搜索结果列表的文件查看该文件的详细信息以及其他用户对该文件的评论；

下载文件：用户可以下载该文件；

评论文件：用户可以评论下载过的文件，但只能评论一次；

举报文件：用户可以举报下载过的文件，但只能举报一次；

上传文件：用户可以上传 office 文件供其他用户下载；

用户注册：用户通过邮箱验证的方式进行注册；

用户登录：用户登录成功的同时系统会向用户手机发送短信告知；

查看个人信息：用户可以查看个人信息；

修改个人信息：用户可以修改个人信息；

找回密码：用户通过注册的邮箱信息重设密码；

(3)管理员

查看投诉列表：管理员查看用户投诉列表；

处理投诉：对用户的投诉信息进行查看并处理；

用户登录：用户登录成功的同时系统会向用户手机发送短信告知；

查看个人信息：用户可以查看个人信息；

4.2 数据库表存储设计

在第三节介绍了数据库表的概念设计之后，接下来讲解每个表的实际业务意义。各信息实体具体描述如下：

管理员表(见表 4.1 所示): 用于存储管理员的账户信息。管理员表有 4 个字段, 分别是 adminId, 账号, 密码, 昵称, 其中 adminId 为数据库自增字段, 也是该表主键。

表 4.1 管理员表 ADMIN

字段名称	字段类型	字段长度	是否为主键	是否为空	字段说明
ADMINID	varchar	10	是	否	主键 id
ACCOUNT	varchar	10	否	否	账号
PASSWORD	varchar	10	否	否	密码
NAME	varchar	10	否	否	昵称

用户表(见表 4.2 所示): 用于存储用户的账户信息。用户表有 9 个字段, 分别是 userId, 账号, 密码, 昵称, 头像, 生日, 电子邮箱, 账号是否冻结(用户是否可以登录进系统), 恶意行为次数(记录用户违规操作, 上传不健康文件), 其中 userId 为数据库自增字段, 也是该表主键。

表 4.2 用户表 USER

字段名称	字段类型	字段长度	是否为主键	是否为空	字段说明
USERID	varchar	10	是	否	主键 id
ACCOUNT	varchar	10	否	否	账号
PASSWORD	varchar	10	否	否	密码
IMAGE	varchar	10	否	否	头像
EMAIL	varchar	10	否	否	电子邮箱
BIRTHDAY	datetime	10	否	否	生日
STATE	int	10	否	否	账号是否被冻结
BADBEHAVIOR	int	10	否	否	恶意行为次数
NAME	varchar	10	否	否	昵称

文件表(见表 4.3 所示): 用于存储用户上传的文件信息。文件表有 10 个字段, 分别是 fileId, 文件标题, 文件介绍, 下载次数, 好评次数, 差评次数, 上传时间, 文件总大小, 文件是否被封(文件是否可以被所有用户检索到), 上传用户, 其中 fileId 为数据库自增字段, 也是该表主键。

表 4.3 文件表 FILE

字段名称	字段类型	字段长度	是否为主键	是否为空	字段说明
FILEID	varchar	10	是	否	主键 id
TITLE	varchar	10	否	否	文件标题
INTRODUCE	varchar	10	否	否	文件介绍
DOWNCOUNT	bigint	10	否	否	下载次数
GOODCOMM ENT	bigint	10	否	否	好评次数
BADCOMME NT	bigint	10	否	否	差评次数
DATE	datetime	10	否	否	上传时间
TOTALSIZE	varchar	10	否	否	文件总大小
ISUSE	bit	10	否	否	是否可见
USERID	varchar	10	否	否	上传者 id

文件地址表(见表 4.4 所示): 用于存储文件信息的实际存储地址。文件地址表有 6 个字段, 分别是 fileresourceId, 文件名称, 文件链接地址, 文件预览地址, 文件大小, 文件(这个文件地址属于哪一个文件的), 其中 fileresourceId 为数据库自增字段, 也是该表主键。

表 4.4 文件地址表 FILERESOURCE

字段名称	字段类型	字段长度	是否为主键	是否为空	字段说明
FILERESOU RCEID	varchar	10	是	否	主键 id
NAME	varchar	10	否	否	名称
FILESRC	varchar	10	否	否	链接地址
SWFSRC	varchar	10	否	否	预览地址
SIZE	varchar	10	否	否	文件大小
FILEID	varchar	10	否	否	文件 id

评论表(见表 4.5 所示): 用于用户对文件的评论信息。评论表有 6 个字段, 分别是 commentId, 评论内容, 评论等级(该文件时好或差), 评论时间, 评论用户, 评论文件, 其中 commentId 为数据库自增字段, 也是该表主键。

表 4.5 文件评论表 COMMENT

字段名称	字段类型	字段长度	是否为主键	是否为空	字段说明
COMMENTID	varchar	10	是	否	主键 id
TEXT	varchar	10	否	否	评论内容
RATE	int	10	否	否	评论等级
DATE	datetime	10	否	否	评论时间
USER	varchar	10	否	否	评论用户 id
FILEID	varchar	10	否	否	被评论文件 id

举报表(见表 4.6 所示): 用于用户对文件的举报信息。举报表有 4 个字段, 分别是 reportId, 举报用户, 被举报文件, 举报内容, 其中 reportId 为数据库自增字段, 也是该表主键。

表 4.6 文件举报表 REPORT

字段名称	字段类型	字段长度	是否为主键	是否为空	字段说明
REPORTID	varchar	10	是	否	主键 id
USERID	varchar	10	否	否	举报用户 id
USERFILEID	varchar	10	否	否	被举报文件 id
REASON	varchar	10	否	否	举报内容

用户文件状态表(见表 4.7 所示): 用于记录用户是否下载过对应文件且是否对文件进行过评论或举报。保证只有下载过对应文件的用户可以举报和评论且只能举报或评论一次。用户文件状态表有 5 个字段, 分别是 usestateId, 用户, 文件, 是否评论, 是否举报, 其中 usestateId 为数据库自增字段, 也是该表主键。

表 4.7 文件文件状态表 USESTATE

字段名称	字段类型	字段长度	是否为主键	是否为空	字段说明
USESTATEID	varchar	10	是	否	主键 id
USERID	varchar	10	否	否	用户 id
USERFILEID	varchar	10	否	否	文件 id
STATE	int	10	否	否	是否评论
REPORT	int	10	否	否	是否举报

注册表(见表 4.8 所示): 用于保存用户注册但是还未生效的账号信息, 当注册表的数据超过 24 小时将会删除, 保证账号的可用性。注册表有 5 个字段, 分别是 usereamid, 注册账号, 注册密码, 邮箱地址, 创建时间, 其中 usereamid 为数据库自增字段, 也是该表主键。

表 4.8 注册表 REGIST

字段名称	字段类型	字段长度	是否为主键	是否为空	字段说明
USEREAMILID	varchar	10	是	否	主键 id
ACCOUNT	varchar	10	否	否	注册账号
PASSWORD	varchar	10	否	否	注册密码
EMAIL	varchar	10	否	否	邮箱地址
DATE	datetime	10	否	否	创建时间

4.3 系统的架构设计

本系统的后台集群架构的设计如图 4.1 所示。

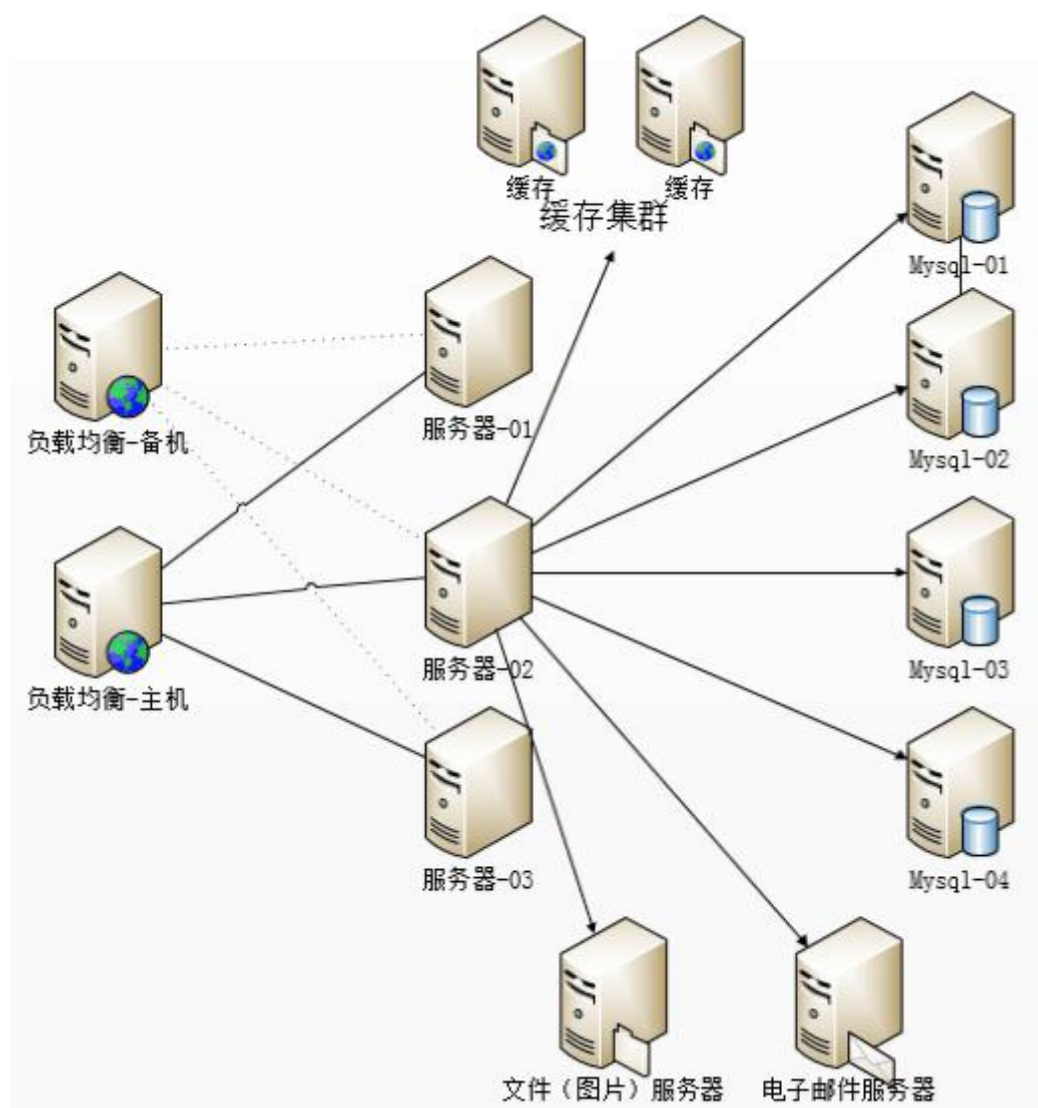


图 4.1 系统架构模块图

用户请求发送到负载均衡主机，主机把请求分发到后台的应用服务器进行处理，服务器对文件的下载和上传操作通过文件服务器进行存储。

当用户搜索文件时，服务器向 lucene 服务器发出请求搜索数据，而不是搜索数据库。

当用户对数据字典的数据访问通过缓存服务器获取，不再通过数据库获取。

4.4 系统 web 界面设计

系统首页：所有用户包括登录用户和未登录用户都可以访问系统首页，系统首页以百度搜索为参考，用户可以在首页输入框输入自己要检索的文件信息。系统会根据用户输入的关键字来检索系统中已经上传的文件的文件标题和文件简介是否有符合用户搜索的，并将满足的记录以列表的形式显示给检索的用户，并对关键字进行高亮显示，排序方式会根据文件被搜索的权重进行相关对排序。原型图如图 4.2，4.3 所示。



图 4.2 系统首页



图 4.3 系统搜索列表首页

文件简介页面：任何人都可以点击搜索返回文件列表的某一条数据进入该文件的信息页面，可以查看文件大小，文件好评和差评数量，文件简介，文件名称，文件作者，用户对文件评论，文件上传时间，文件的下载路径，用户可以对文件进行举报和评论。如果是未登录的用户点击举报和评论，下载文件按钮。系统会自动跳转到系统登录页面，要求用户只能登录后才可以举报，下载和评论文件。如图 4.4 所示。

走向架构师之路

下 载 豆: 1
资料大小: 0.670

下载 举报 评价



intsmaze

2016-05-22

已有 1 人下载, 其中 1 人发表了评价, 好评率 100. %
力荐: 1 差评: 0

简介: 讲解如何搭建服务器集群已经反向代理服务器的配置, 同时讲解了任务调度框架在集群下的问题。

本资料共包含以下附件:
5-通向架构师的道路之tomcat集群.docx 预览
1-通向架构师的道路之Apache整合Tomcat.docx 预览

Down友的评价

下载记录 (897人)

全部评价



刘洋

力推

2016-05-22 18:24:12 说:

讲的很好, 我的问题立马就解决了, 晚上不用加班了。

赞 (0) 回复

请点选一个评级:

☐ 力荐 资料与描述一致, 且很有价值。

☐ 差评 资料与描述完全不一致, 并说明理由

请说明一个理由:

提 交

图 4.4 文件简介页面

文件预览页面: 所有用户都可以在文件信息页面点击预览按钮来线预览文件的内容, 不需要下载。如图 4.5 所示。

10

请点选一个评级: ☒ 力荐 资料与描述一致, 且很有价值。
☐ 差评 资料与描述完全不一致, 并说明理由

请说明一个理由:

阿瑟发是打发斯蒂芬

提交

图 4.7 文件评论页面

文件上传页面: 已经登录的用户可以进行文件上传操作, 否则点击跳转到用户登录页面。可以上传多个文件, 但是文件类型必须是 office 类型, 否则提示上传失败, 同时必须注明上传文件的标题和简介。如果用户输入的标题和简介中含有敏感词汇会被和谐。文件上传成功后, 会跳转到文件信息页面。如图 4.8 所示。

返回首页 | 刘洋 我的文件 退出

上传资料

请勿上传任何含有病毒、木马、政治敏感、色情以及侵犯他人版权的资料!

文件: D:\ITinformation\走向架构师之路\1-通向架构师的道路之 浏览...

文件: D:\ITinformation\走向架构师之路\5-通向架构师的道路之 浏览...

资料标题: 走向架构师之路

资料简介: 讲解如何搭建服务器集群已经反向代理服务器的配置, 同时讲解了任务调度框架在集群下的问题。

提交

图 4.8 文件上传页面

登录页面: 验证用户的信息是否正确, 正确就进入系统首页, 否则返回该页, 提示登录错误信息。这里验证用户信息包括账号和密码是否正确以及该用户的状态是否是有效, 如果无效就无法登录。只有登录的用户才可以下载, 举报和评论下载的文件以及上传文件, 当未登录的用户点击上传文, 下载, 评论和举报按钮, 系统会跳转到登录页面。如图 4.9 所示。



图 4.9 系统登录页面

注册页面：用户输入用户名和邮箱，系统通过 ajax 异步查询数据库，如果用户名和邮箱已经存在了，则在页面提示已经被使用，否则提示可以使用（如图 4.10 所示）。注册信息通过后，点击立即注册，就向注册邮箱发送激活信息（如图 4.11 所示），这时候就会跳转到邮箱页面。点击进入邮箱（如图 4.12 所示），点击激活链接就成功激活了（如图 4.13 所示）。

邮箱注册

已有账号? [去登录](#)

用户名

电子邮箱

密码

确认密码

立即注册

已有账号? [去登录](#)

图 4.10 注册页面

我们已向发送账号激活邮件，请登录邮箱点击激活账号。

进入邮箱查看

没有收到重置密码邮件？你可以：

- 1.到邮箱中的垃圾邮件、广告邮件目录中找找
- 2.再次尝试重设密码
- 3.与我们的客服联系，电话：400-885-9898

图 4.11 邮箱激活页面



图 4.12 邮箱页面

进入首页

没有收到重置密码邮件？你可以：

- 1.到邮箱中的垃圾邮件、广告邮件目录中找找
- 2.再次尝试重设密码
- 3.与我们的客服联系，电话：400-885-9898

图 4.13 注册成功页面

密码找回：如果用户忘记密码，可以通过注册时绑定的邮箱进行密码的重设。

如图 4.14 所示。操作流程同 4.10 用户注册。

密码找回

电子邮箱

密 码

确认密码

立即找回

图 4.14 密码找回页面

个人信息页面：登录用户可以查看并修改个人信息，用户只能修改自己的昵称，头像，生日。同时用户可以看见自己的违规次数。如果昵称含有敏感词将会被系统和谐掉以*替换。如果上传的头像不是图片类型则提示上传失败。如图 4.15 所示。

个人信息

账号:

intsmaze

电子邮箱:

597676318@qq.com

昵称

刘洋

头像



浏览...

未选择文件。

生日:

2015-11-25

恶意行为:

0

完成修改

修改密码

图 4.15 个人信息页面

管理员登录页面：系统有一个单独的页面面向管理员，管理员可以登录系统。如图 4.16 所示。

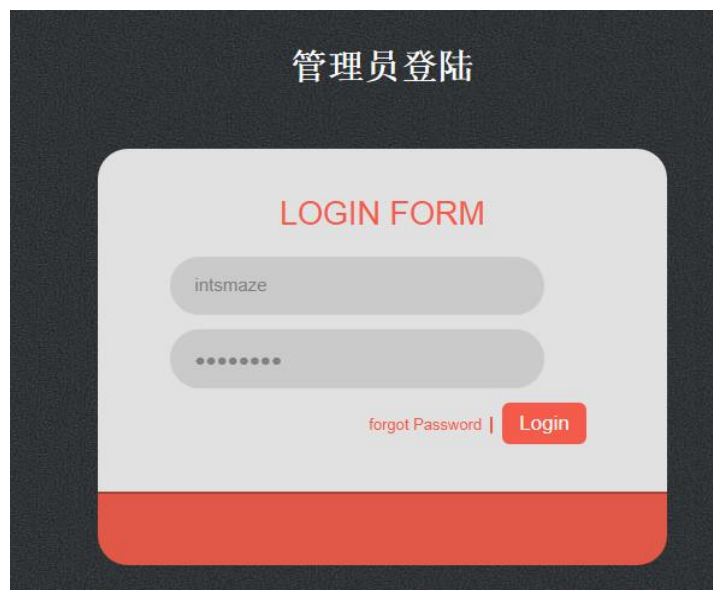


图 4.16 管理员页面

投诉处理页面：管理员登录系统后，将会看到未处理的用户举报列表，管理员选择查看被投诉的文件，看文件是否符合投诉条件，符合则点击受理，该文件的所有举报都被处理，该文件无法被用户查询到了，同时记录上传该文件者的不良行为，当不良行为到达一定次数后，就会被封号。点击拒绝，该文件的所有举报都被处理了。如图 4.17 所示。

 投诉受理管理

投诉原因	被投诉文件名	投诉人名称	投诉时间	查看文件信息	操作
文件包含不良内容	架构之美	user	时间	查看	受理 拒绝
上传的文件没有内容	架构之美	user	时间	查看	受理 拒绝

总共 2 条记录，当前第 1 页，共 1 页，找到相关结果约2个

图 4.17 管理员受理页面

5 系统测试

5.1 性能测试

网站性能测试的主要衡量指标包括网站的请求响应时间，网站的并发数，网

站的吞吐量等。

1，响应时间：系统执行一个完成操作整体消耗的时间，这个时间包括从客户端发出请求开始到服务器上的系统收到然后把响应数据发出去所需要的时间。

2，并发数：系统能够同时处理来自客户端的请求数目，这个指标反映系统的负载特性。

3，吞吐量：指系统在单位时间内处理的请求数量，这个指标可以反应出系统的整体处理能力。

5.2 性能测试结果

使用如上的测试方法本系统的性能测试结果如表 5.1 所示。

表 5.1 性能测试结果报告

并发数	响应时间（ms）	TPS	错误率（%）	Load	内存（GB）	备注
10	500	20	0	5	8	性能测试
20	800	20	0	5	8	性能测试
30	1000	30	0	10	10	性能测试
40	1200	45	20	30	16	性能测试
60	2000	30	40	50	16	性能测试
80	超时	0	100	不详	不详	性能测试

5.3 功能测试

在测试中，根据系统的需求分析说明书以及系统的详细设计说明书，理解不同模块的输入输出条件和逻辑结构，准备每个功能的测试用例，来验证每个功能模块的执行结果是否正确。

5.4 功能测试结果

由于系统的模块可限制多，这里主要对部分关键功能进行测试用例的描述，具体测试结果如下：

对登录过程的测试用例表如表 5.2 所示。

表 5.2 登陆测试用例表

用例说明	输入	预期结果	测试结果
用户登录	正确的账号和密码	跳转系统首页	符合预期
用户登录	错误的账号或密码	停留在登录页	符合预期
用户登录	冻结的账号和密码	停留在登录页	符合预期

对文件下载过程的测试用例表如表 5.3 所示。

表 5.3 下载文件测试用例表

用例说明	输入	预期结果	测试结果
用户未下载该文件对文件评论	提交评论	提示用户没有下载该文件无法对该文件进行评论	符合预期
未登录用户评论文件	提交评论	跳转到系统登录页面	符合预期
用户下载该文件对文件评论	提交评论	提交评论成功，文件评论列表显示用户的评论信息	符合预期
用户评论过该文件再次评论	提交评论	提示用户评论过该文件无法再次评论	符合预期

对文件举报过程的测试用例表如表 5.4 所示。

表 5.4 举报文件测试用例表

用例说明	输入	预期结果	测试结果
用户未下载该文件 举报该文件	提交举报	提示用户没有下载该文件 无法对该文件进行举报	符合预期
未登录用户举报该文	提交举报	跳转到系统登录页面	符合预期
用户下载该文件举报该文	提交举报	提交举报成功，管理员受理列表显示该用户的举报内容	符合预期
用户举报过该文件 举报该文	提交举报	提示用户举报过该文件无法再次举报	符合预期

对密码找回过程的测试用例表如表 5.5 所示。

表 5.5 密码找回测试用例表

用例说明	输入	预期结果	测试结果
密码找回	输入错误的电子邮箱地址	停留在原界面，提示用户邮箱地址有误	符合预期
密码找回	输入的新密码为空	停留在原界面，提示用户修改的密码格式不正确	符合预期
密码找回	输入的新密码两次不相同	停留在原界面，提示用户两次密码不相同	符合预期
密码找回	输入新密码两次都相同	跳转到激活页面，点击按钮进入邮箱激活	符合预期

对文件上传过程的测试用例表如表 5.6 所示。

表 5.6 文件上传测试用例表

用例说明	输入	预期结果	测试结果
登录登录用	点击上传文	跳转到系统登录页	符合预期

户上传文件	件按钮	面	
登录用户上传文件	点击上传文件按钮	跳转到文件上传页面	符合预期
登录用户上传文件	输入上传文件格式非 office 格式	停留在上传文件页面提示用户上传文件的格式不符合要求	符合预期
登录用户上传文件	输入上传文件的大小超过 100M	停留在上传文件页面提示用户上传文件的大小超过要求	符合预期
登录用户上传文件	输入的文件格式为 office，文件大小小于 100M，文件标题和简介为空	停留在上传文件页面提示用户上传文件的标题和简介的长度不符合	符合预期
登录用户上传文件	输入的文件格式为 office，文件大小小于 100M，文件标题长度小于 10，简介长度小于 50。	上传文件成功，跳转到该文件详情页面	符合预期

6 软件开发中主要解决的问题

本系统是一个面向互联网用户的项目，本系统不仅要保证用户的账号安全不被盗用，还有保证本系统中内容的健康性，不散播低俗淫秽内容和用户体验的友

好和及时性。另外本系统是一个分布式系统，相比以往的传统系统有很多挑战需要解决。

6.1 帐号登录的安全性

因为系统的网络连接采用基于 socket 的 http 协议，所以用户的输入的账号密码在网络传输过程中存在被人劫持而暴露用户密码的风险。采用传统方式 web 页面对前端用户输入的明文密码进行加密，然后在网络中传输加密后的密码，服务器接收到加密码后与数据库中的加密的密码进行比较来验证用户是否可以登录，但是这种方式下盗号者完全不需要知道用户的明文密码是多少，只需要劫持到网络传输中的加密后密码，以后直接发加密码即可登录系统。

因为以上原因，我们想到的是，用户在登录页面，服务器根据用户存在与服务器的 session 生成一个随机码保存在服务器中，同时把这个值传给登录页面，登录页面获取这个值后拼接到用户的密码明文后面进行加密后通过网络传输到服务器，服务器接受到加密后密码后，通过服务器存储的随机值与数据库的原始密码也拼接起来加密后进行比较来验证用户登录，这样就可以保证用户每一次登录时通过网络发送的加密码不一样，这样就算被劫持，也无法被人使用。

6.2 搜索的精准与快速

如何人性化为用户提供精准服务，采用全文检索——垂直化搜索引擎主要针对系统内部的自有数据的检索，通过对系统内部数据建立索引提供给用户进行检索。

它既能满足用户对全文检索，模糊匹配的需求，解决数据库 like 查询效率低下的问题，又能够解决分布式环境下，由于采用分库分表或使用 NoSQL 数据库，导致无法进行多表关联或者进行复杂查询的问题。

lucene 将文档中的词作为关键字，建立词与文档的映射关系，通过对倒排索引的检索，可以根据词快速获取包含这个词的文档列表。

能对句子或段落进行切割，从中取出包含固定语义的词。

当输入一个关键字进行搜索时，可能会命中许多文档，搜索引擎给用户的价值就是快速地找到需要的文档，根据排序算法，将相关度更大的内容排在前面，命中多次的文档比命中一次的文档有更高的相关性。

6.3 文件的处理

传统系统用户只能下载文件后才能查看文件内容，并不能让用户在文件下载之前就能查看文件的内容，经常导致用户下载文件后发现文件不是自己想要的，不仅影响用户的体验，同时也给网站带来流量开销和连接的压力。为了实现文件预览，我们需要将文件转换为 swf 文件，实现 flash 播放达到预览，在这里面我们不仅要考虑不同版本 office 文件处理问题，还要考虑如果文件里面有图片如何准确提取出来把数据正确的存储到 swf 文件中。在这我们通过 java 代码使用 OpenOffice 服务把文件转换为 swf 文件，使用 FlexPaper, swfTools 在线预览，从而达到用户不需下载文件就能看到文件内容。

6.4 敏感词过滤

由于本系统收录的敏感词库中包含上万个敏感词，系统要对用户输入的文件信息进行处理，检查用户输入的信息是否含有敏感词，但是这里带来一个效率问题，采用传统思路，系统会对用户输入的一个文本数据比对上万次，来验证文本数据是否包含敏感词库的词。显然这是十分耗时的。NFA 是非确定性的状态机，DFA 是确定性的状态机。确定性和非确定性的最大区别就是：从一个状态读入一个字符，确定性的状态机得到一个状态，而非确定性的状态机得到一个状态的集合。如果我们把 NFA 的起始状态 S 看成一个集合 $\{S\}$ 的话，对于一个状态集合 S' ，给定一个输入，就可以用 NFA 计算出对应的状态集合 T' 。因此我们在构造 DFA 的时候，只需要把起始状态对应到 S' ，并且找到所有可能在 NFA 同时出现的状态集合，把这些集合都转换成 DFA 的一个状态，那么任务就完成了。因为 NFA 的状态是有限的，所以 NFA 所有状态的集合的幂集的元素个数也是有限的，因此使用这个方法构造 DFA 是完全可能的。

6.5 网站高并发

一台 tomcat 服务器最大能支持的稳定并发数在 100—120 之间，因为本系统是面向互联网用户的，所以在同一时刻本系统面对的并发数远远大于 120，当实际并发数大于 tomcat 服务器的支持数量后，会导致服务器因为压力过大而无法及时处理其他请求，甚至导致服务器宕机。考虑到高并发访问的情况，本系统使用 Nginx 服务器的负载均衡技术构建一个由多台服务器组成的服务器集群，将来自客户端的并发访问请求分发到多台服务器上处理，避免单一服务器出现负载压

力过大的情况。

为了保证当任意一台或多台服务器宕机，Nginx 服务器将请求提交给集群中其他任意一台可用服务器能够正确处理，本系统需要设置每一台服务器不保存请求的状态，这样所有的服务器完全对等，服务器就可以成功处理其他服务器之前处理的请求了。

6.6 服务器集群下状态共享

集群中所有的应用服务器都是无状态的，但是在业务上系统总是有状态的，因为系统需要记录用户当前的登录状态来确定用户是否可以执行接下来的操作。web 应用中将这些多次请求使用的上下文对象称作会话 (session)，单机情况下，session 可由部署在服务器上的 web 容器管理。在使用负载均衡的集群环境中，请求由负载均衡服务器分发到集群上任意一台应用服务器上，如何保证任意一台应用服务器对每次请求依然能够获得正确的 session 是一个挑战。

通过将一部分数据存储存储在 cookie 中，来解决分布式环境下 session 操作同步的问题。但是这样做的弊端有很多，比如 cookie 的安全性，cookie 存储数据的大小的限制。

本系统的解决方案是将 session 统一存储在缓存集群 memcache 上，这样保证较高的读，写性能，再从安全性上看，利用缓存的失效机制，达到控制 session 有效期的的作业。

6.7 高并发下磁盘读写速度

系统在面对高并发的情况下，大量的读，写请求涌向数据库，磁盘的处理速度与内存显然不在一个量级，为了提高系统对数据的读取速度，通常将常用的数据放在内存中，这样就避免了应用程序读取硬盘而增加时间开销。

memcache 是一个高性能的内存对象缓存系统，为了提高在内存中查找数据的速度，memcache 在内存中维护一张巨大的 HashTable，使得对数据查询的时间复杂度降低到 $O(1)$ 。内存的空间总是有限的，当内存没有更多的空间来存储新数据时，memcache 会使用 LRU 算法，将最近不常访问的数据淘汰掉，以腾出空间来存放新的数据。

6.8 分布式数据库事务一致性

软件系统对单个数据库内部的多个 DML 操作都会组成一个局部事务，因为

这些操作不会跨越多个事务性资源，软件系统可以直接使用底层数据库的事务支持；但是当软件系统的功能操作涉及对多个数据库的修改时，就面临多个数据库事务的一致性问题，这些数据库的操作要么全部成功要么全部不成功，这就是分布式事务，分布式事务处理的对象是全局事务。

使用 JTA 编程就可以用一种与事务管理器无关的方式来开始，提交或回滚事务，Java EE 应用服务器通过 java 事务服务来实现事务管理器。JTA 事务由 java EE 事务管理器负责控制，它可以保证多个数据库更新的一致性，通过 JTA 即可实现全局事务控制。

6.9 集群下 quartz 协调处理

在集群环境下，大家会碰到一直困扰的问题，即多个应用程序下如何用 quartz 协调处理自动化工作。比如有 A，B，C 三台服务器同时作为集群服务器对外统一提供服务；A，B，C 三台机器上各有一个 Quartz，他们会按照即定的时间频率自动执行各自的任務。这样的架构其实有点像多线程，那多线程里就会存在“资源竞争”的问题，即可能产生脏读，脏写，由于三台应用服务器里都有 Quartz，因此会存在重复处理任务的现象。quartz 的任务实例化如数据库，基于数据库引擎及 High-Available 的策略（集群的一种策略）自动协调每个节点的 quartz，当任一节点的 quartz 非正常关闭或出错时，另几个节点的 quartz 会自动启动；这样每台作为集群点的应用程序上都可以布署 quartz；无需开发人员更改原已经实现的 quartz，使用 Spring 类反射的机制对原有程序作切面重构。

7 结论

去年 10 月我拿到论文题目，第一反应就是做一个类似百度文库，面向所有互联网用户的系统，而不是一个内部系统。确定了目标之后，在后面的两个月我

遇到了很多挑战，同时也开阔了我的视野。

我意识到仅仅对登录的密码进行加密并不能提高密码的安全性，我也意识到当从操作一个数据库到操作多个数据库的挑战，也意识到如何提高系统的并发能力，在系统集群的环境下会遇到哪些问题。

在这个过程中我参考了很多架构设计的书，吸取了很多有丰富经验的架构师的架构心得。

最后我明白了我们在开发一个系统时不要一开始就企图去设计一个大型的网站。大型网站不是设计出来的，而是逐步根据不同的情况下发展演化出来的。每一种架构都没有好坏之分，只存在是否在当前系统面临的压力下是最好的一种方式。就像阿里巴巴去 IOE，并不是 IOE 不适合阿里巴巴的系统，而是阿里巴巴的系统在发展到一定程度后，IOE 在很多业务情况下已经无法满足阿里巴巴系统的需要了。

我们以后在开发软件系统时不要为了技术而技术，不能因为某个技术很火就盲目的在系统中采用该技术，我们要记住网站技术是为了业务而存在的。

参考文献

- [1]王志刚,江友华.MySQL 高效编程.北京:人民邮电出版社,2011.01:9-10.
- [2]苗泽.nginx 高性能 web 服务器详解[M].北京:电子工业出版社,2013.10.
- [3]陈康贤.大型分布式网站架构与设计实践[M].北京:电子工业出版社,2014.09.

- [4]罗刚.解密搜索引擎技术实战:lucene&java 精华版[M].第二版.北京:电子工业出版社,2014.01.
- [5]李智慧.大型网站技术架构:核心原理与案例分析[M].北京:电子工业出版社,2013.09.
- [6]刘琼. Peer-to-Peer 文件共享系统的测量研究[D].北京:中国科学院软件研究所,2011 年.
- [7]倪高鹏. 基于 Memcached 的缓存系统设计与实现[D]. 大连理工大学:丁锋,2012 年.
- [8]王利萍. 基于 Nginx 服务器集群负载均衡技术的研究与改进[D]. 山东大学:袁东风, 2015 年.
- [9]李永春;丁华福. Lucene 中文分析器在书目搜索应用中的比较研究[J]. 业务研究, 2014 年, 4 期:23-25.
- [10]刘志鹏;卫晨;丁华福. 基于 Quartz 与 Spring 的动态任务调度系统的设计与实现[J]. 计算机光盘软件与应用,2014 年,13 期:44-45.
- [11]李汝光;徐骏;丁华福. 基于 JTA 的跨数据库分布式事务的实现[N]. 常州工学院学报, 2012 年, 4 期.
- [12]Charles A.Bell,Mats Kindahl,Lars Thalmann .MySQL High Availability[M].北京:东南大学出版社, 2015.02.
- [13]Yassine Rhazali, Youssef Hadi,Abdelaziz Mouloudi. Model Transformation with ATL into MDA from CIM to PIM Structured through MVC[J].Procedia Computer Science, 2016, Vol.83.
- [14]Chunyan GUO, Konglai ZHU,Zongguo MA.The Design of B2B E-commerce System Based on MVC Model and J2EE[J].Management Science and Engineering, 2010, Vol.4 (4), pp.113-119
- [15]Andy Zaidman, Nick Matthijssen, Margaret-Anne Storey, Arie Deursen .Understanding Ajax applications by connecting client and server-side executiontraces[J].Empirical Software Engineering,2013, Vol.18 (2), pp.181-218.

附：开题报告正文

文件共享管理系统的设计与实现

1 文件共享管理系统研究背景

在日常生活中，通常会遇到很多问题需要获得相应的知识进行解决。比如想

要写一个劳动合同，但是不知道具体的格式，从而导致写的合同不规范，影响工作效率。因此，我设计了能够满足这种需求的系统——文件共享管理系统。

通过文件共享管理系统，广大用户可以上传自己的资料供其他需要的用户进行搜索并下载作为参考模板。这样可以极大的提高用户在日常生活中解决问题所耗费的精力，提高用户效率并为用户提供专业的知识科普。

2 本课题的开发思想与解决方案

文件共享系统不仅要能够存储用户文件，提供用户搜索。在面对互联网环境下我们要考虑的有如下几点：

如何人性化为用户提供精准服务，采用全文检索，进行相关度排序，把相关度高的搜索结果排在上面，对存储文本数据和用户搜索关键词进行智能分词提高搜索命中率。

如何存储海量的数据单独的文件服务器存储海量数据，缓解应用服务器的存储压力。

如何屏蔽不健康信息，使用 DFA 算法，根据系统收录的敏感词词库中智能屏蔽用户上传数据中含有的敏感词。

如何避免必要的流量开销，减少用户不必要的下载次数。使用 FlexPaper，swfTools，OpenOffice 实现 office 文件的在线预览，从而达到用户不需下载文件就能看到文件内容。

如何提高系统的并发量而不宕机，一个应用服务器支持的稳定的并发连接是有限的，当客户端的请求并发量超过了一个应用服务器提供的稳定连接后，该系统通过架构 nginx 服务器做反向代理来搭建应用服务器集群，这样系统支持的稳定连接就随着应用服务器的增加线性上升。

如何避免不必要的数据（硬盘）查询次数，用户对系统的操作就是对数据库数据的获取操作，传统单机系统就是通过把一些常用，少变得数据放到缓存（内存）中，比起直接再从数据库（硬盘）中获取数据，以指数级提高了效率。但是在本应用服务器集群中，采用传统方案会导致各自服务器中缓存的数据不一致，导致从缓存获取数据的命中率降低。这时候我们使用 redis 技术，搭建 redis 集群，把应用服务器集群要存储到缓存的数据存储到 redis 服务器中，以后获取缓存数据从 redis 服务器中获取，这样就保证了缓存数据的一致。

如何在集群服务下 session 共享，使得用户在第一次连接 A 服务器后，第二次连接 B 服务器后，服务器还能够记住用户。这里同样要把 session 存储到 redis 服务器中。

如何提高数据库的执行效率，系统的业务不同，不同的表的查询次数压力不同，为了提高数据库的执行效率，高效响应各种查询语句，根据业务压力，把不同的表划分到不同的数据库，从而分拆数据库的压力。

如何保证保证一个业务逻辑处理中对每个独立数据库的操作能够同时回滚或执行成功保持一致性。该系统采用 JTA 分布式事务规范，可以保证数据库的跨源操作的一致性。

如何保证管理员能够准时执行某些功能保证系统的可控性，该系统在某些业务功能上使用任务调度框架定时进行处理。

如何保证用户账号安全性，该系统通过使用 javaMail 技术进行用户密码修改和找回密码的验证，通过第三方短信验证接口在该账号登陆系统时向用户手机发送短信告知。

3 系统功能模块

本系统的功能模块如图 1 所示。

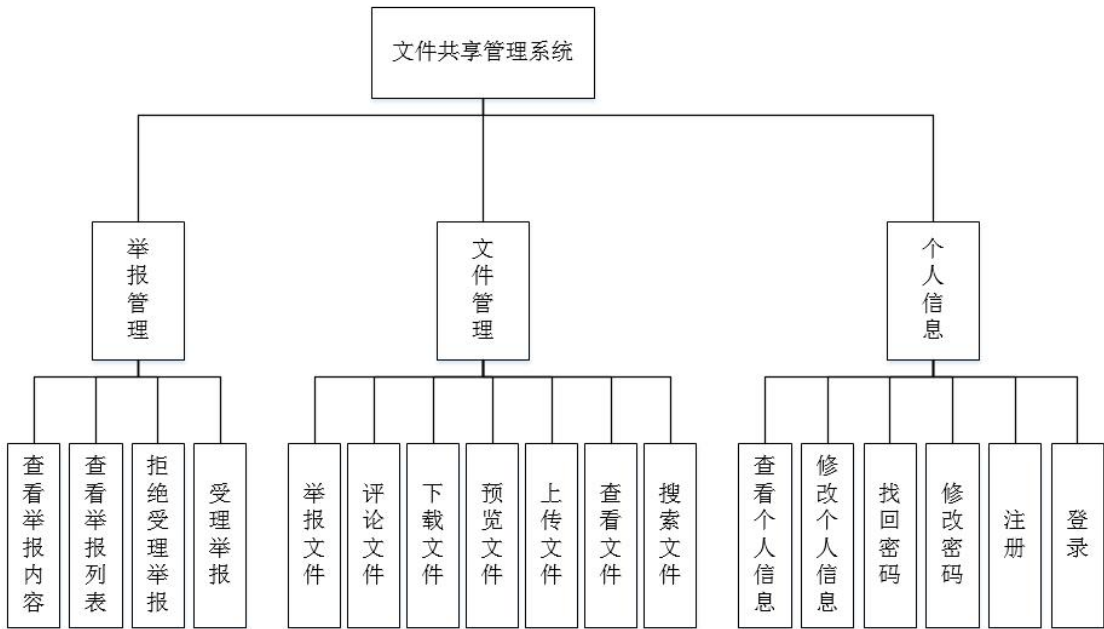


图1 系统功能模块图

系统用户分为三种：游客、用户、管理员。这三个模块的功能如下所述：

(1)游客

搜索资料：用户可以在搜索框输入要搜索文件的关键字进行搜索；

查看文件信息：用户可以点击搜索结果列表的文件查看该文件的详细信息以及其他用户对该文件的评论；

(2)用户

搜索资料：用户可以在搜索框输入要搜索文件的关键字进行搜索；

查看文件信息：用户可以点击搜索结果列表的文件查看该文件的详细信息以及其他用户对该文件的评论；

下载文件：用户可以下载该文件；

评论文件：用户可以评论下载过的文件，但只能评论一次；

举报文件：用户可以举报下载过的文件，但只能举报一次；

上传文件：用户可以上传 office 文件供其他用户下载；

用户注册：用户通过邮箱验证的方式进行注册；

用户登录：用户登录成功的同时系统会向用户手机发送短信告知；

查看个人信息：用户可以查看个人信息；

修改个人信息：用户可以修改个人信息；

找回密码：用户通过注册的邮箱信息重设密码；

(3)管理员

查看投诉列表：管理员查看用户投诉列表；

处理投诉：对用户的投诉信息进行查看并处理；

用户登录：用户登录成功的同时系统会向用户手机发送短信告知；

查看个人信息：用户可以查看个人信息；

3 主要技术简介：

3.1 MySQL 数据库

MySQL 是一个采用了关系模型来组织数据的关系型数据库，而且它的源代码是开放的，谁都可以在一定的条件下无偿使用，而且可以对应用程序进行改造。它使用普及率广泛；对数据的处理十分迅速；支持在多种 OS 中运行；支持多种开放语言。

3.2 Memcached 技术

Memcached 是一个性能很高的内存对象缓存系统，它通过在内存里维护一个统一的巨大的 hash 表，来存储各种格式的数据，包括图像、音频以及数据库检索的结果等。简单的说 Memcached 就是将数据调用到内存中存储起来，然后需要对应数据时从内存中读取而不是从硬盘中读取，从而大大提高读取速度。Memcached 以守护程序的方式可以运行在一个或多个服务器上，随时接收客户端的连接和操作。

3.3 Lucene 技术

Lucene 是一个开放源代码的全文检索引擎框架，它提供了完整的查询引擎和索引引擎，部分文本分析引擎。全文检索是计算机程序通过扫描文章中的每一个词，对每一个词建立一个索引，指明该词在文章中出现的次数和位置。当用户查询时根据建立的索引查找，类似于通过字典的检索字表查字的过程。

3.4 Nginx 技术

Nginx 是一个高性能的 HTTP 和反向代理的服务器，反向代理方式是指以代理服务器来接受 internet 上的连接请求，然后代理服务器将请求转发给内部网络上的服务器，并将从服务器上得到的结果返回给 internet 上请求连接的客户端，此时代理服务器对外部网络的客户端而言像一个服务器。

3.5 Servlet 技术

servlet 是一个用 Java 编写的服务器端程序。servlet 运行在 Java 服务器上。Java servlet 可以动态的扩展服务器的能力，采用请求-响应模式提供 web 服务。

当 web 服务器接收到一个 http 请求时，他会先判断请求内容,如果请求的内容是静态网页数据，web 服务器会自行处理并返回响应信息;如果请求的内容涉及动态数据，web 服务器会将请求转给 servlet 容器，servlet 容器就会找到对应的处理该请求的 servlet 实例，最后 servlet 实例将结果送回 web 服务器，再由 web 服务器传回客户端。

3.6 JSP 技术

JSP 全称是 Java Server Pages，它是一种用于开发动态 web 资源的技术。

JSP 技术的特点在于，开发者写 jsp 就像在写 html，但它相比 html 只能为用户提供静态数据，Jsp 技术允许在页面中嵌套 java 代码，为用户提供动态数据。

不管是在开发中使用 JSP 或是 Servlet，它们都可以用于开发动态 web 资源。但由于这 2 门技术各自的特点，在长期的软件开发中，工程师逐渐把 servlet 作为 web 应用中的控制器组件来使用，而把 JSP 技术作为数据显示模板来使用。

3.7 Struts 技术

struts2 是一个遵循 MVC 结构的框架，采用了更加松耦合的设计，使得系统的 action 不与 servlet API 耦合，使系统可以从 B/S 结构转换为 C/S 结构；action 不需要与 WebWork 耦合，代码的重用率更高；对 JSP，Velocity，FreeMarker 等表现层技术提供支持；使用拦截器作为增强处理，以用户的业务逻辑控制为目标，创建一个控制器代理。

3.8 Hibernate 技术

hibernate 是轻量级 javaEE 应用的持久层解决方案。Hibernate 对 JDBC 访问数据库的代码做了封装，负责 java 对象的持久化，大大简化了数据访问层繁琐的重复性代码；Hibernate 是一个基于 jdbc 的主流持久化框架，是一个优秀的 orm 实现，建立面向对象的域模型和关系数据模型之间的映射，连接 java 应用和数据库的中间件，它很大程度的简化了 dao 层编码工作，封装对数据库的访问细节，使业务逻辑层更专注于实现业务逻辑；Hibernate 的性能非常好，因为它是一个轻量级框架。映射的灵活性很出色。它支持很多关系型数据库，跨平台性很强，从一对一到多对多的各种复杂关系；hibernate 的一级缓存，二级缓存，查询缓存，使访问数据的效率提高很大。

3.9 Spring 技术

Spring 是根据大量开发中出现的通用步骤对这些通用步骤进行抽取的框架，因此它完成了，他留给开发者的仅仅是与特定应用相关的部分，开发者不需要再编写系统开发中需要解决的通用问题，从而大大提高开发者的开发效率。

Spring 为企业应用的开发提供了一个轻量级的解决方案。该解决方案包括：基于依赖注入的核心机制，基于 AOP 的声明式事务管理，与多种持久层技术的整合，以及优秀的 web MVC 框架等。Spring 致力于 Java EE 应用各层的解决方案，而不仅仅是专注于某一层的方案。

Spring 具有低侵入式的设计，代码的污染极低；独立于各种应用服务器，基于 Spring 框架编写的应用，可以真正实现一处编写，到处运行的承诺；Spring

的 DI 容器降低了业务对象替换的复杂性,提高了组件之间的解耦;Spring 的 AOP 支持允许将一些通用任务如安全,事务,日志等进行集中式处理,从而提供了更好的复用;Spring 的 ORM 和 DAO 提供了与第三方持久层框架的良好整合,并简化了底层的数据库访问。

4 进度安排

第 12--16 周 查阅资料写开题报告,收集有关文献

第 17 周 开题报告答辩

第 2—9 周 根据模型框架进行编码设计与实现

第 10—13 周 完成毕业论文的初稿

第 14 周 撰写毕业论文及准备毕业答辩

5 参考文献

- [1]王志刚,江友华.MySQL 高效编程.北京:人民邮电出版社,2011.01:9-10.
- [2]苗泽.nginx 高性能 web 服务器详解[M].北京:电子工业出版社,2013.10.
- [3]陈康贤.大型分布式网站架构与设计实践[M].北京:电子工业出版社,2014.09.
- [4]罗刚.解密搜索引擎技术实战:lucene&java 精华版[M].第二版.北京:电子工业出版社,2014.01.
- [5]李智慧.大型网站技术架构:核心原理与案例分析[M].北京:电子工业出版社,2013.09.
- [6]刘琼. Peer-to-Peer 文件共享系统的测量研究[D].北京:中国科学院软件研究所,2011 年.
- [7]倪高鹏. 基于 Memcached 的缓存系统设计与实现[D]. 大连理工大学:丁锋,2012 年.
- [8]王利萍. 基于 Nginx 服务器集群负载均衡技术的研究与改进[D]. 山东大学:袁东风, 2015 年.
- [9]李永春;丁华福. Lucene 中文分析器在书目搜索应用中的比较研究[J]. 业务研究, 2014 年, 4 期:23-25.
- [10]刘志鹏;卫晨;丁华福. 基于 Quartz 与 Spring 的动态任务调度系统的设计与实现[J]. 计算机光盘软件与应用,2014 年,13 期:44-45.
- [11]李汝光;徐骏;丁华福. 基于 JTA 的跨数据库分布式事务的实现[N]. 常州工学院学报, 2012 年;4 期.
- [12]Charles A.Bell,Mats Kindahl,Lars Thalmann .MySQL High Availability[M].北京:东南大学出版社, 2015.02.

- [13]Yassine Rhazali, Youssef Hadi,Abdelaziz Mouloudi. Model Transformation with ATL into MDA from CIM to PIM Structured through MVC[J].Procedia Computer Science, 2016, Vol.83.
- [14]Chunyan GUO, Konglai ZHU,Zongguo MA.The Design of B2B E-commerce System Based on MVC Model and J2EE[J].Management Science and Engineering, 2010, Vol.4 (4), pp.113-119
- [15]Andy Zaidman, Nick Matthijssen, Margaret-Anne Storey, Arie Deursen .Understanding Ajax applications by connecting client and server-side executiontraces[J].Empirical Software Engineering,2013, Vol.18 (2), pp.181-218.