

清 华 大 学

# 综 合 论 文 训 练

题目：基于垂直云计算技术的  
教育网设计与实现

系 别：信息学院自动化系

专 业：自动化专业

姓 名：王 旭

指导教师：曹军威 研究员

2012 年 6 月 7 日



## 关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内 容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签 名：\_\_\_\_\_ 导师签名：\_\_\_\_\_ 日 期：\_\_\_\_\_



## 中文摘要

垂直云计算是使云计算技术适应于某些特殊应用而进行的优化与组合。依照垂直云计算的架构，云计算供应商可以针对行业特点提供独特的功能，更好地服务于用户需求。现在针对医疗保健的垂直医疗云平台是已建立的最完善垂直云计算应用。

本文将重点关注教育行业，以设计和实现服务于教育行业的大型网络资源共享平台为目标，利用云计算、Web Service 网络服务等多项技术，通过实现云平台、网络服务器、客户端三方的交互，对基于垂直云平台的下一代教育行业的网络应用进行了创新性探索。主要完成了以下工作：

1. 在实验室环境下搭建了三层网络结构，分别是云计算平台、网络服务器和客户端。针对教育行业特有的需求和应用提供服务，基本可以满足于中小学教育的整个过程；

2. 应用 Web Service 技术搭建了面向服务的架构，完成了 13 个独立的网络功能模块。每一个功能模块都是基于标准因特网协议的，且能够适应不同的开发平台和编程语言；

3. 在云平台建立了分布式文件存储系统，可以满足大量数据存储的需求；同时使用虚拟化技术，通过虚拟机的调度实现计算资源更加高效的利用。

**关键词：**垂直云计算；云计算；教育云平台；面向服务架构



## ABSTRACT

Vertical cloud, is the optimization of cloud computing for a particular specific use application. The cloud provider can offer specialized functions and options that best meet industry-use and specification. Today, the vertical health-care is one of the most well-established vertical clouds.

In this paper, we will focus on the education, and propose the design and implementation of resource sharing platform using the technologies of vertical cloud computing and Web Service. The main contribution of our work is the design of the next generation of education using the vertical cloud computing.

1. Our system is built in laboratory environment and consists of three layers, namely cloud computing platform, web servers and clients. It can provide services meeting the unique needs in primary and secondary education.
2. We use Web Services to establish a service-oriented architecture, making 13 functional building-blocks accessible over standard Internet protocols independent of platforms and programming languages.
3. We built a distributed data storage system in cloud platform. We also use virtual machines to make the tasks scheduling more efficient.

**Key words:** Vertical cloud computing; Cloud computing; Cloud computing for education; Service-oriented architecture





# 目 录

<b>第 1 章 引言</b> .....	<b>1</b>
1.1 项目背景及可行性分析 .....	1
1.1.1 垂直云计算技术.....	1
1.1.2 SOA 架构 .....	2
1.2 国内外相关研究成果 .....	4
1.2.1 Google 协作平台 .....	4
1.2.2 医疗卫生垂直云平台 .....	5
1.3 论文结构 .....	6
<b>第 2 章 需求分析及系统设计</b> .....	<b>7</b>
2.1 需求分析 .....	7
2.1.1 教育行业特点分析.....	7
2.1.2 系统设计中面临的困难.....	8
2.1.3 解决措施及关键技术.....	8
2.2 垂直云平台设计架构 .....	9
2.3 系统功能预期.....	10
<b>第 3 章 网络服务器的设计与实现</b> .....	<b>11</b>
3.1 开发环境 .....	11
3.1.1 硬件环境.....	11
3.1.2 软件环境 .....	11
3.2 数据库的设计与维护 .....	11
3.2.1 用户权限信息数据库.....	11
3.2.2 文件管理信息数据库 .....	13
3.3 网络服务器功能的实现 .....	15
3.3.1 网络服务功能的组织架构 .....	15
3.3.2 网络服务的创建.....	16
3.4 Web 服务的发布和功能测试 .....	22
3.5 客户端功能实现与测试 .....	24

<b>第 4 章 云平台的功能实现</b> .....	<b>27</b>
4.1 分布式文件存储系统.....	27
4.1.1 使用 Apache 搭建网络服务器 .....	27
4.1.2 分布式文件系统的组织结构 .....	28
4.2 虚拟机资源调度.....	30
4.2.1 在虚拟机上部署网络服务器 .....	30
4.2.2 资源调度基础功能的实现 .....	32
<b>第 5 章 总结与展望</b> .....	<b>34</b>
5.1 本文工作总结.....	34
5.2 未来工作展望.....	34
<b>插图索引</b> .....	<b>37</b>
<b>表格索引</b> .....	<b>39</b>
<b>参考文献</b> .....	<b>41</b>
<b>致 谢</b> .....	<b>43</b>
<b>声 明</b> .....	<b>45</b>
<b>附录 A 书面翻译</b> .....	<b>47</b>

# 第1章 引言

## 1.1 项目背景及可行性分析

该课题以设计和实现服务于教育行业的大型网络资源共享平台为目标，利用云计算、web service 网络服务等多项技术，通过实现云平台、网络服务器、客户端三方的交互，对基于垂直云平台的下一代教育行业的变革进行了创新性探索。

该课题主要依托于以下几方面的背景：

### 1.1.1 垂直云计算技术

随着信息技术的飞速发展，计算机和互联网的信息存储、处理能力受到了越来越大的挑战；与此同时，互联网上还存在着很多闲置的计算资源未被利用。所以，如果用户能够通过互联网使用闲置共享的计算资源，以并行计算和分布式处理技术为核心进行任务调度，就可以大大减轻用户对于硬件资源的依赖。

基于这种发展现状，云计算技术应运而生，并且已经迅速成为高效的计算资源调度方法[1]。云计算以并行计算、分布式存储和虚拟化技术为技术核心，将网络资源整合成一个庞大的信息处理和数据存储“资源池”[2]。通过对于资源和服务的集中调度和管理，云计算平台能够按需地为用户提供服务，包括共享的硬件、软件、数据信息等等。基于云计算模式，服务供应商能够将计算能力作为资源按需地提供给所有用户[3]。随着云计算技术的日趋完善，亚马逊、google 等大型企业将云计算的应用再次推上了新的高度。越来越多的行业和公司也在应用云计算技术为用户提供更好地服务。

随着云计算的飞速发展和普及，很多云计算供应商根据一些行业的具体特点和需求搭建云平台，这就发展成为“垂直云计算”框架。垂直云计算是针对于某些特定应用的、基于云计算技术的优化调度[4]。例如教育、政府部门、医疗卫生等公共服务行业，均有自己独特的需求和特点。通过在云平台的功能实现中加入更多行业需求特点，可以实现更加高效和智能的资源调度。相信随着网络科技的进一步发展，会有越来越多的垂直云计算平台服务于大家的生活。

基于以上背景，本课题希望将云计算应用到教育领域，是因为垂直云计算有以下三个特点[5]，保证了其可行性：

- 个性化：云计算能够按用户所需提供服务，而且可以使资源利用情况做

到动态收发。由此垂直云计算平台可以为教育行业中不同角色、不同特点的用户提供个性化服务，可适应教育行业特点。

- 低成本：学校不需要购置昂贵的大型服务器，也可以通过云平台共享的计算资源提供服务。这就保证了云平台良好的应用市场和前景。
- 高性能：通过充分利用网络共享的闲置计算资源，可以为各个学校提供更加高效的计算服务，能够满足教育行业用户多、需求量大的特点。

### 1.1.2 SOA 架构

随着网络的普及和 web2.0 时代的到来，网络用户已不再仅仅满足于传统的信息获取、信息发送等单一功能，而希望网络服务更加复杂化、个性化和智能化。例如希望网络服务能够根据用户的喜好而量身定制，网络服务的结构也能学习和适应用户的个性化需求等等。

在这样的需求背景下，再加上云计算的蓬勃发展，一方面我们需要对现有的丰富的资源和服务进行有效地组合，以实现我们需求的新功能；另一方面由于开发环境、操作系统等差异，使得资源和服务的整合变得十分困难。应对这种情况，Web Service 的出现很好地解决了这一问题。它通过统一的架构和标准，为跨平台的操作提供了可能性，极大地增强了网络服务的交互操作性[6]。

Web Service 是最终面向用户的服务整合手段。它统一了服务组成部分之间的接口，实现了服务元件之间的松耦合，从而将服务功能分模块进行封装。经典的 SOA 架构包括三个部分[7]，分别是服务注册中心、服务提供者和服务请求者；同时在功能实现中遵循三套协议，分别为：WSDL、UDDI 和 SOAP。其典型的架构如图 1.1 所示。

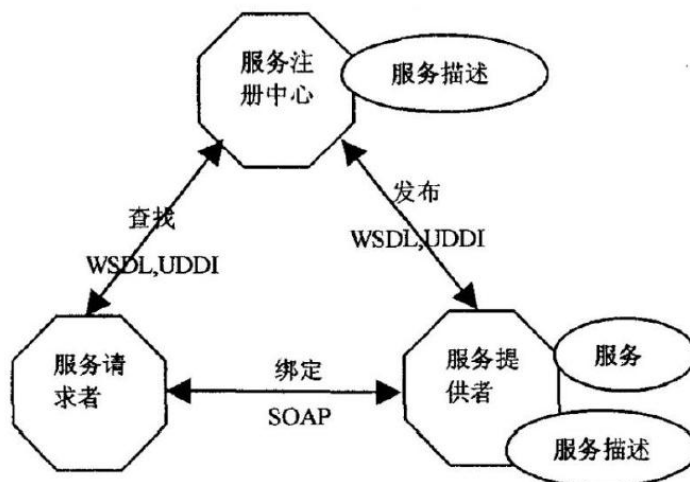


图 1.1 Web Service 的经典架构

如图 1.1 所示，Web Service 的服务体系分为三部分，分别是通信协议、服务发现和服务描述，分别对应以下三个核心协议：

- SOAP (Simple Object Access Protocol, 简单对象访问协议) 是在现有网络传输协议之上、基于 XML 的协议。该协议定义了调用 Web Service 服务的语法规则，可摆脱平台和语言对信息传输的限制。发送方可以将自己的任何信息包装在一个 XML 文档中。只要该文档的语言符合 SOAP 规则，接收方就可以通过解读该 XML 文档得到发送方传递的消息。
- UDDI (Universal Description, Discovery and Integration, 统一描述、发现与集成) 为用户提供了统一的、规范的方法来查找符合要求的服务供应商，也为服务供应商对自己提供的服务进行描述提供了统一规范。服务供应商可以将符合该协议的服务描述发布到注册中心供用户查询。而注册中心中包含了该网络服务的三种信息，分别为白页 (White Pages) 黄页 (Yellow Pages) 和绿页 (Green Pages)，以此对外提供查询服务。
- WSDL (Web Service Description Language, 网络服务描述语言) 同样基于 XML 语言。该描述向用户说明了如何与特定的 Web Service 进行沟通、需要怎样的信息与细节。通过解析 Web Service 提供的 WSDL，用户可以了解使用什么通信协议、如何实现服务及网络链接地址等等信息。可以说 WSDL 就是用户使用 Web Service 的说明书。其典型实现如图 1.2 所示。

<pre> POST /testservice SOAPAction: "http://www.soap.com/test" Content-Type: text/xml; charset="utf-8" Content-Length: nnnn  &lt;SOAP:Envelope xmlns:SOAP=   "http://schemas.xmlsoap.org/soap/envelope/"&gt;   &lt;SOAP:Header&gt;     &lt;!-- header --&gt;   &lt;/SOAP:Header&gt;   &lt;SOAP:Body&gt;     &lt;!-- body --&gt;   &lt;/SOAP:Body&gt; &lt;/SOAP:Envelope&gt; </pre>	<pre> &lt;message name="GetInstruInfoInput"&gt;   &lt;part name="InstruName" type="xsd:string"/&gt;   &lt;part name="InstruId" type="xsd:int"/&gt; &lt;/message&gt;  &lt;message name="GetInstruInfoOutput"&gt;   &lt;part name="InstruInfo" type="fixsd:InstruInfoType"/&gt; &lt;/message&gt;  &lt;message name="LoginInput"&gt;   &lt;part name="body" element="userxsd:Username"/&gt; &lt;/message&gt;  &lt;portType name="InstruServicePortType"&gt;   &lt;operation name="GetInstruInfo"&gt;     &lt;input message="tns:GetInstruInfoInput"/&gt;     &lt;output message="tns:GetInstruInfoOutput"/&gt;   &lt;/operation&gt;   &lt;operation name="Login"&gt;     &lt;input message="tns:LoginInput"/&gt;   &lt;/operation&gt; &lt;/portType&gt; </pre>
--	---

图 1.2 SOAP 典型的消息结构及 WSDL 对接口的描述

依托于 SOA 架构，该教育云平台可以摆脱操作系统、开发平台的限制，有着更加良好的扩展性和适应性，可以实现更好的资源共享和服务相互调用，为云平台实现广泛的应用和推广提供了可能性。

## 1.2 国内外相关研究成果

随着云计算的日趋普及，国内外很多企业也纷纷把目光投向了相关领域的研究工作。对于教育、医疗、卫生等公共服务行业，Google、微软等大型公司也正在充分发挥云计算的优势来提供更好的服务。在此主要列举一些在教育和医疗方面的垂直云计算的应用。

### 1.2.1 Google 协作平台

Google 协作平台是 Google 公司推出的一个基于云计算技术的在线应用程序 [9]，可以方便地将各种信息（视频、照片、文本等）收集在统一的地方，快速便捷地创建团队网站，并可在页面中嵌入各种类型的内容。基于 Google 协作平台开发的教育交流平台，可以说是云计算在教育领域发展的先行者。



图 1.3 Google 协作平台示例站点

由于 Google 协作平台更多地关注于团队网站的开发，所以通过使用 Google 协作平台开发的教育辅助网站带有很强的团队交流建设的特点。有利于增强老师、

同学和家长之间的交流，也有利于各年级、各班之间的信息共享和互动。由于是基于云计算技术开发的平台，所以它可以为同学们提供个性化学习的空间，有利于培养同学们的创新精神和自主学习的能力。

然而，由于该平台并非专门为教育行业而开发，所以难以为教育行业提供全程的、有针对性的服务。

### 1.2.2 医疗卫生垂直云平台

另一个相关工作就是应用于医疗卫生行业的垂直云平台。随着网络技术与应用的普及，电子医疗保健成为了病人接受治疗的过程中十分重要的组成部分。应用于医疗行业的垂直云平台更是为医院和患者提供了更加便捷和高质量的服务。

微软公司开发了一个名为 **Health Vault** 的网络平台[10]，存储和维护用户的个人健康及身体状况的相关信息。该平台完全应对于医疗卫生行业的特点，能够与医院网站配合，收集、存储和共享用户的个人健康信息，同时保证私人信息的安全，实现了功能更加强大的“电子病历”的功能。

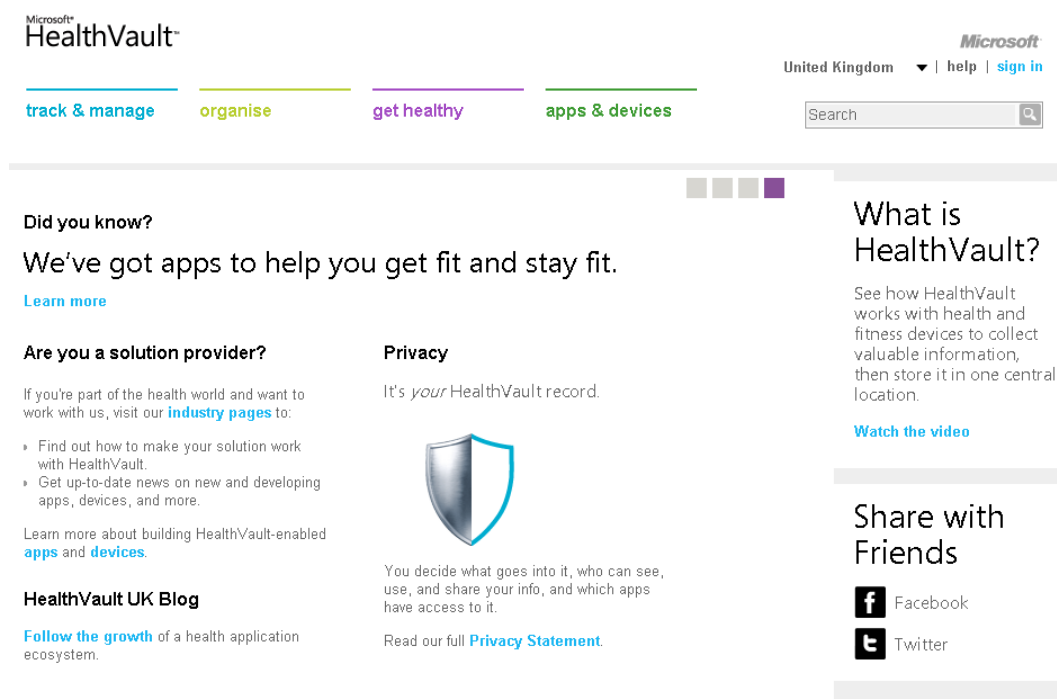


图 1.4 微软公司开发的 Health Vault 医疗垂直云平台

另外，Google 公司也在此之前提供了个人健康信息集中管理和信息共享的网络服务，名叫 Google Health[11]。该平台为用户提供服务，可以管理自己的病历，

并可自愿将自己的个人身体情况资料包括有关家族病史、过敏史和患病历史的信息交付给谷歌保存，以便本人或医院通过新服务随时取回。

然而，由于平台的影响力不足、使用人数过少，该平台在 2012 年 1 月 1 日由谷歌宣布停止运营。

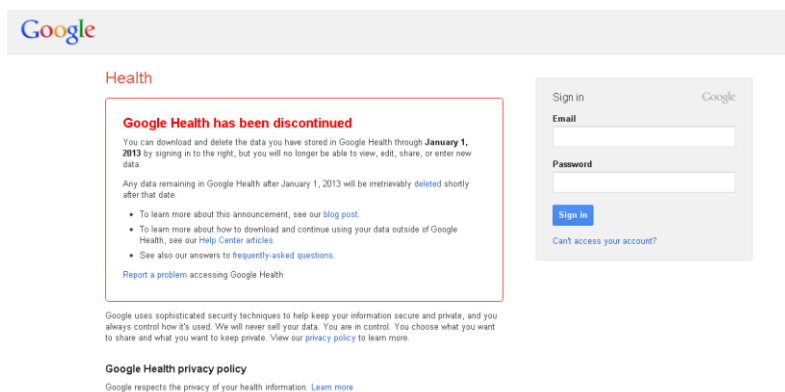


图 1.5 Google Health 平台已停止运营

### 1.3 论文结构

本文通过对垂直云计算、面向服务架构的讨论，在实验室的环境下建立应用于教育行业的教育垂直云平台。利用云计算、Web Service 网络服务等多项技术，通过实现云平台、网络服务器、客户端三方的交互，对基于垂直云平台的下一代教育行业的变革进行了创新性探索。

第一章介绍了该课题的两方面技术背景，分别是云计算技术和 Web Service 技术，并进行了一定的可行性分析。此外还列举了几个国内外相关研究工作，并分析了各自的特点。

第二章详细分析了教育行业所特有的几个应用背景和需求，并应对每种需求提出相应的解决方法。最后设计出整体的教育行业垂直云计算组成架构。

第三章详细阐述了网络服务器的设计与实现过程，包括开发环境、组织架构、数据库维护和网络服务的创建。之后对开发实现的网络服务器功能进行了测试。在最后介绍了服务器端的功能实现，并展示了服务器程序的测试结果。

第四章分别就云平台中分布式文件管理系统和虚拟机调度两个功能模块进行了介绍，从功能的实现过程、最终结果的测试等几个方面进行了详细的阐述。

第五章对该课题的工作进行了总结，并对未来工作的发展方向进行了展望。



## 第2章 需求分析及系统设计

### 2.1 需求分析

该课题希望设计一个面向中小学教育行业的垂直云计算平台。所以在设计之初需充分考虑教育行业不同于其他行业的特点和需求，并进行有针对性地完善和解决。最终才能够设计出中小学教育网垂直云计算平台的架构。

#### 2.1.1 教育行业特点分析

经过归纳总结，现在的中小学教育有以下几个方面的特点，我们需要在系统设计中加以充分考虑[12]：

1) 用户数量巨大。随着九年义务教育的普及，中小学在校生人数在近些年呈现出大幅度上涨的趋势。根据人口普查结果显示，2011年中小学在校生约2.08亿人。如果再加上教师与学校管理人员，这个数目将会更加庞大。这种特点一方面说明教育行业市场前景广阔，另外也对我们的系统性能提出了更加严峻的挑战。

2) 资源重复利用。全国中小学现在普遍使用纸质教材，每年同年级的学生教材差别不大。然而教材的使用效率很低，多数教育资源只能使用一次，这就造成了大量的资源浪费现象。

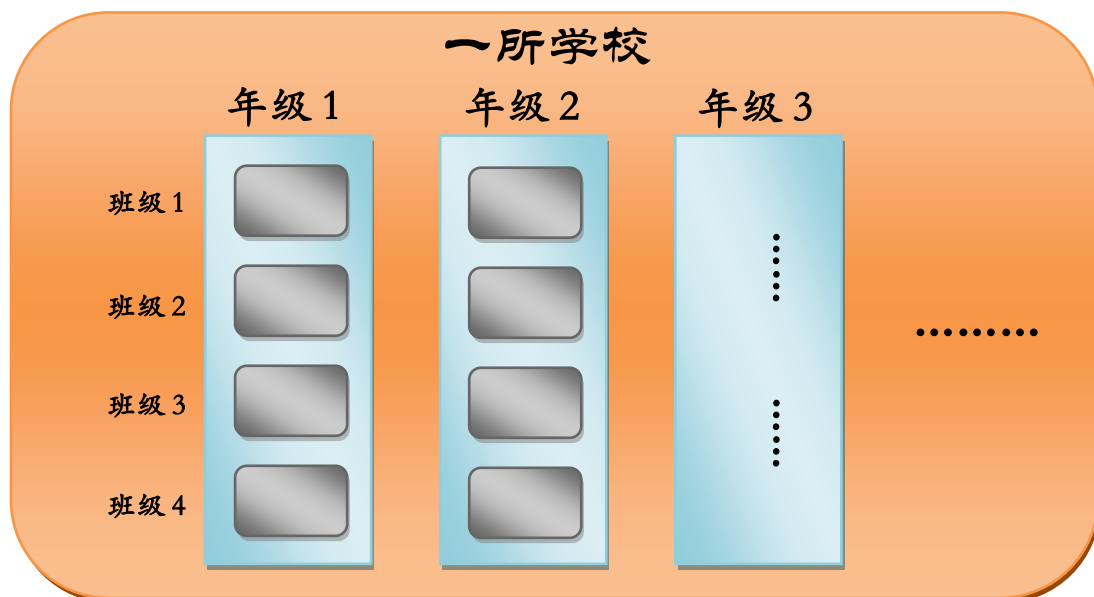


图 2.1 典型的学校管理体系模型

3) 人员角色等级分明。教育行业的一大特点就是层级管理十分鲜明。每个学校分多个年级、每个年级分多个班级、每个班级中老师带领多个同学完成教学过程，整个权限管理体系呈树状结构。一个典型的学校管理模型如图 2.1 所示。

4) 文件需要频繁交互。教育过程中一个很重要的环节就是老师布置的作业与考试，使同学更好地认识自己的学习水平、老师更好地了解同学的发展缺陷。所以这就使中小学教育行业比其他行业需要更加频繁的文件交互。

5) 需要个性化发展。中小学教育不能针对每个同学都千篇一律，需要有同学们自由个性化发展的空间。在教育的过程中需要照顾到每个同学不同的特点和需求，提供有针对性的服务。

### 2.1.2 系统设计中面临的困难

基于以上中小学教育行业所具有的五個方面的特点，我们可以预期在系统设计过程中需要解决以下几个方面的问题：

1) 大量用户的分时段访问。由于中小学教育市场潜力巨大，所以系统需要能够应对大量用户访问的压力，需要系统具有较高的性能；与此同时由于教育行业的特点，用户的访问可能是比较集中的、分时段的，在很多时间系统资源可能处于闲置状态。所以大量用户分时段访问的问题是我们首先要解决的。

2) 大量的数据冗余。由于教育行业中资源可广泛的重复利用，所以使用计算机网络建立的应用程序可以很好地解决资源浪费的问题。但与此同时也很容易使得网络存储的数据发生大量冗余，造成网络存储资源的浪费。如何更好地进行系统设计、将冗余程度减到最低是我们应该考虑的问题。

3) 严密的权限管理。教育系统对权限管理的要求十分严格，不同等级的用户需要有不同等级的权限，整个权限管理系统呈树状结构。维护好严密的权限管理系统，将是维持良好的教学过程的前提条件。

4) 个性化的私人信息存储。教学过程中为了使学生有个性化发展的舞台，需要为每个同学开辟个性化的私人信息存储的空间。同学们可以将自己的笔记、个人文件存储其中。为实现此功能，应用系统需要能够存储海量的数据。

### 2.1.3 解决措施及关键技术

应对于教育行业的几个突出特点，针对以上几个方面需要解决的问题，我们在设计该垂直云计算平台时将采用几下几个方面的核心技术：

1) 应用云计算虚拟化技术，进行计算资源灵活调度。将网络服务器搭建在

虚拟机上，通过对虚拟机的操作达到能够应对大量用户访问的压力、同时资源尽可能减少浪费的目的。

2) 应用 Web Service 技术，实现网络服务器功能模块之间的松耦合，使系统具有良好的可扩展性，并可使学校间跨平台地进行功能互相调用。相同数据不必每个学校都存放一份，解决数据冗余严重的问题。

3) 应用分布式数据存储技术，可以为每个用户开辟个人文件存储空间，解决大量数据存储的问题。

4) 学校网络服务器需要维护和管理用户权限数据库。将用户个人信息和文件信息存储在数据库当中，保证严明的权限管理系统和文件管理系统。

## 2.2 垂直云平台设计架构

基于以上对于教育行业特点及需求的分析，以及需要解决的几项技术问题，设计基于垂直云平台技术的教育网络架构设计如图 2.2 所示：

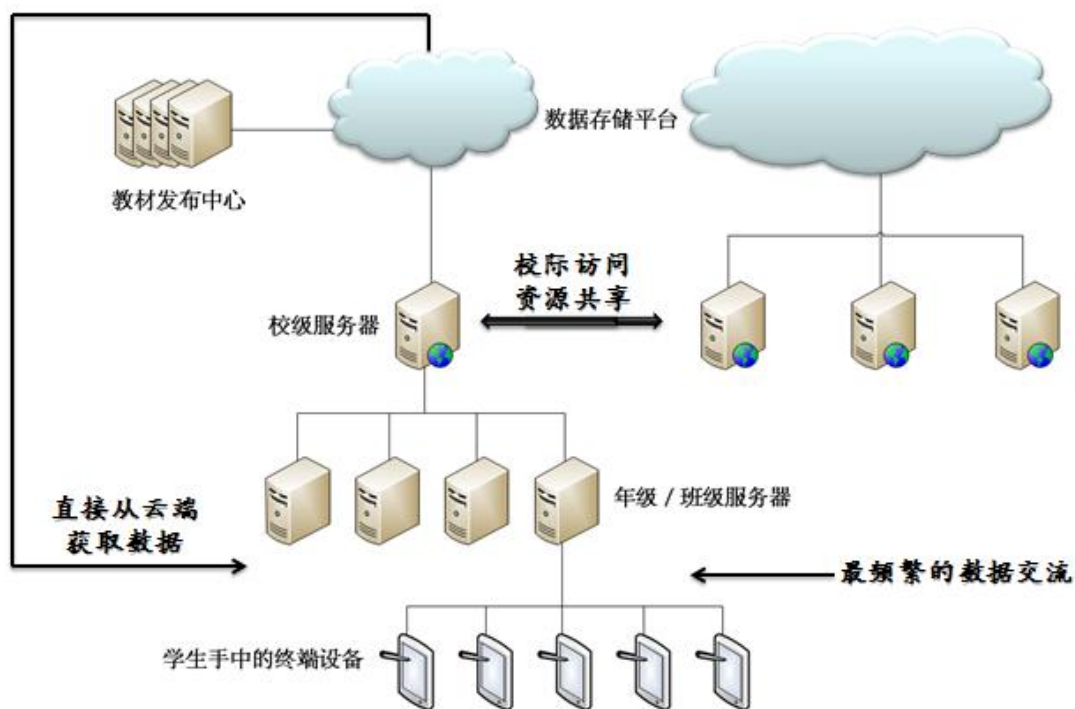


图 2.2 教育网垂直云计算平台架构

该系统架构可以分为三个层次，每层的功能和实现如下：

- 云平台：这是该架构的最顶端的一层。该层需要建立数据存储系统，实现数据的分布式存储和计算资源的优化调度。如 2.2 图中所示，每个学校可以搭建自己的云平台，亦可以多个临近的学校共享同一个文件存储的云平台，尽量减少数据冗余的情况。
- 网络服务器：每个学校需要搭建自己的网络服务器。该层需要建立权限管理系统，维护好学校所有用户的权限管理，可以使用户依照自己的权限等级访问云平台中存储的不同数据。此外网络服务器应用 Web Service 技术，实现服务模块的松耦合，具有良好的扩展性；能够实现学校之间的校际访问和资源共享，摆脱系统环境和开发平台的制约，进一步遏制数据冗余和服务冗余的情况。
- 客户端：随着智能手机和掌上电脑的发展和普及，我们希望未来的教育能够摆脱书本的束缚，学生的全部教育过程都在智能终端上进行。例如，客户端可以开发在 Android 操作系统上，使学生们可以使用智能终端下载教材、读书做题、提交作业等等。在本课题实验室开发环境中，我们在 Android 系统上开发了测试用客户端程序。

## 2.3 系统功能预期

基于以上的系统框架和各层的功能配合，我们希望在实验室环境下实现该系统的以下几个部分功能，服务于中小学教学的全部过程：

- 用户权限和信息的管理：用户可以使用自己的用户名密码登陆并进行权限验证、完善用户个人信息。管理员用户可以有更高权限的操作；
- 学生查询和下载教材：学生可以查询适合自己本学期学习的教材，并从云平台数据存储系统中下载；
- 老师布置作业和共享文件：老师可以向自己教授班级的同学布置作业，或共享一些辅助教学的参考文件；
- 学生上交作业：学生在终端上可下载对应老师上传的作业，完成后可上传到相应位置，老师之后可以进行审阅批改；
- 私人文件上传下载：老师和学生都可以申请私人空间，用于存储私人文件，如课堂笔记等等。私人空间只有自己有权查看和管理。

## 第3章 网络服务器的设计与实现

### 3.1 开发环境

实际的应用中，云平台、网络服务器、客户端应该分布在很多不同的机器上。在实验室的环境下，处于调试方便的考虑，选择在一台电脑上进行开发与调试。

#### 3.1.1 硬件环境

CUP: Intel(R) Core(TM) i3 CPU 2.27GHz  
内存: 2.00GB

#### 3.1.2 软件环境

操作系统: Windows XP Professional 32 bit  
浏览器: MyEclipse Web Browser  
IDE: MyEclipse 8.6  
JKD: jdk 1.6.0  
数据库: MySQL 5.0  
Web Service: Tomcat 5.5

### 3.2 数据库的设计与维护

由于学校的网络服务器需要根据用户的不同角色和权限提供不同的服务，所以需要在服务器后台维护用户权限的数据库；另外由于教育网络需要大量的文件交互，包括私人文件上传与下载、老师布置作业、同学下载教材等等，所以网络服务器后台也需要维护文件信息的列表。

在该课题中使用的是 MySQL 5.0 数据库，主要建立了以下两类数据库：

#### 3.2.1 用户权限信息数据库

首先需要维护用户的权限和个人身份信息。根据系统需求，可以设定用户有三种身份，分别为学生、教师和管理员。通过这些实体的特点可以设计出表的结构。其逻辑关系如图 3.1 所示：

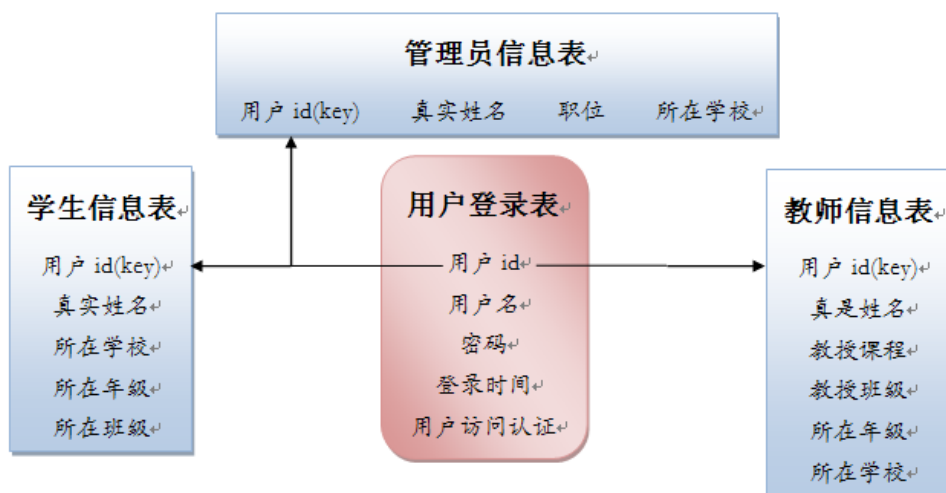


图 3.1 用户权限管理数据库逻辑结构

其中用户登录表记录的是用户登录所需的信息，用户 id 为主键。一旦用户登陆成功，则将产生一个随机字符串作为用户访问认证，在返回给用户的同时修改表格中“用户访问认证”对应的值。登陆过的用户只要在之后的请求中提交用户访问认证即可验证身份（类似 cookies 的设置），用户退出时表中对应值清空。表 3.1 为 MySQL 中对用户登录表的描述，其它表格相类似，不再一一列举。

表 3.1 表 user\_account 的结构

字段名	数据类型	是否为空	是否主键	默认值	描述
id	int(11)	NO	PRI	NULL	用户 ID 号
username	varchar(20)	NO		NULL	登陆用户名
password	varchar(20)	NO		NULL	登录密码
time	datetime	YES		0	登录时间
access	varchar(40)	YES		NULL	用户身份识别码

其它三个表单，“学生信息表”、“教师信息表”和“管理员信息表”分别存储着每个用户的个人信息资料，以备之后搜索文件时匹配用户之间的逻辑关系。用户的个人信息需要在自己账号登录后进行完善。

### 3.2.2 文件管理信息数据库

在维护用户权限的同时，也需要在网络服务器层维护文件信息，包括文件的名称、作者、可见权限、下载上传地址等等。建立统一的文件索引体系，有利于实现数据的分布式存储，可将文件放在许多个不同的文件服务器上。根据教育过程中文件管理的特点，建立以下四个表格管理不同的文件信息：

1) 教材信息表 (textbook\_info 表)：该表格保存所有的教材信息，包括教材的名称、作者、出版社、下载地址，以及教材的适用年级和课程等信息。该表格所有人均可见，同学们可以依此在学期初查找适合本学期的教材并下载。

表 3.2 表 textbook\_info 的结构

字段名	数据类型	是否为空	是否主键	默认值	描述
number	int(11)	NO	PRI	NULL	教材编号
course	varchar(20)	NO		NULL	教材所对应的课程
grade	varchar(20)	NO		NULL	教材所对应的年级
title	varchar(40)	NO		NULL	教材书名
author	varchar(40)	NO		NULL	教材作者
press	varchar(40)	NO		NULL	出版社
link	varchar(100)	YES		NULL	下载地址

2) 共享文件信息表 (shared\_files 表)：该表格保存老师布置的作业及上传的共享资料等文件信息，作为共享文件的查找索引。由于权限设定，只有老师才能够上传共享文件，同学们只能看到和自己相关的文件信息，并可根据中的下载地址下载文件。

表 3.3 表 shared\_files 的结构

字段名	数据类型	是否为空	是否主键	默认值	描述
code	int(11)	NO	PRI	NULL	共享文件自动编号
id	Int(11)	NO		NULL	上传文件老师的 ID
grade	varchar(20)	NO		NULL	文件对应的年级
class	varchar(20)	NO		NULL	文件对应的班级
title	varchar(40)	NO		NULL	文件名

times	Int(11)	NO	0	文件下载次数
link	varchar(100)	YES	NULL	文件下载地址

3) 学生上交作业表 (homework\_files 表): 该表格存放学生上交的作业文件信息。表格内容包括上交同学的 id 号、对应课程、文件名、老师批阅情况和下载链接等等。老师可以查找所教授班级对应课程的所有上交作业, 同学们仅可见自己上交的作业列表。

表 3.4 表 homework\_files 的结构

字段名	数据类型	是否为空	是否主键	默认值	描述
code	int(11)	NO	PRI	NULL	作业文件自动编号
course	varchar(20)	NO		NULL	作业对应的课程
id	int(11)	NO		NULL	上传同学的 ID 号
title	varchar(40)	NO		NULL	作业的文件名
time	datetime	YES		NULL	作业上传时间
score	int(11)	YES		0	作业得分
link	varchar(100)	YES		NULL	作业下载地址

4) 私人文件信息表 (private\_files 表): 该表格存放所有私人空间中的文件信息列表, 以用户的 id 号为主键。包括文件的名称、上传时间、下载链接等信息。每个用户仅能浏览自己所用 id 号上传的文件列表。

表 3.5 表 private\_files 的结构

字段名	数据类型	是否为空	是否主键	默认值	描述
id	int(11)	NO		NULL	文件上传人的 ID 号
title	varchar(40)	NO		NULL	文件名称
time	datetime	YES		NULL	文件上传时间
link	varchar(100)	NO		NULL	文件下载地址



### 3.3 网络服务器功能的实现

在数据库创建完毕后，便可以进行网络服务功能的具体编写与实现。由于采用 Web Service 技术，所以各个服务功能模块之间相互独立，最终由多个功能模块的先后调用实现复杂的服务功能。

#### 3.3.1 网络服务功能的组织架构

在本课题实验室环境中，一共完成了 13 个独立功能模块的编写，其功能基本覆盖了中小学教育的整个过程。具体来说，13 个功能模块的逻辑关系和组织架构如图 3.2 所示。

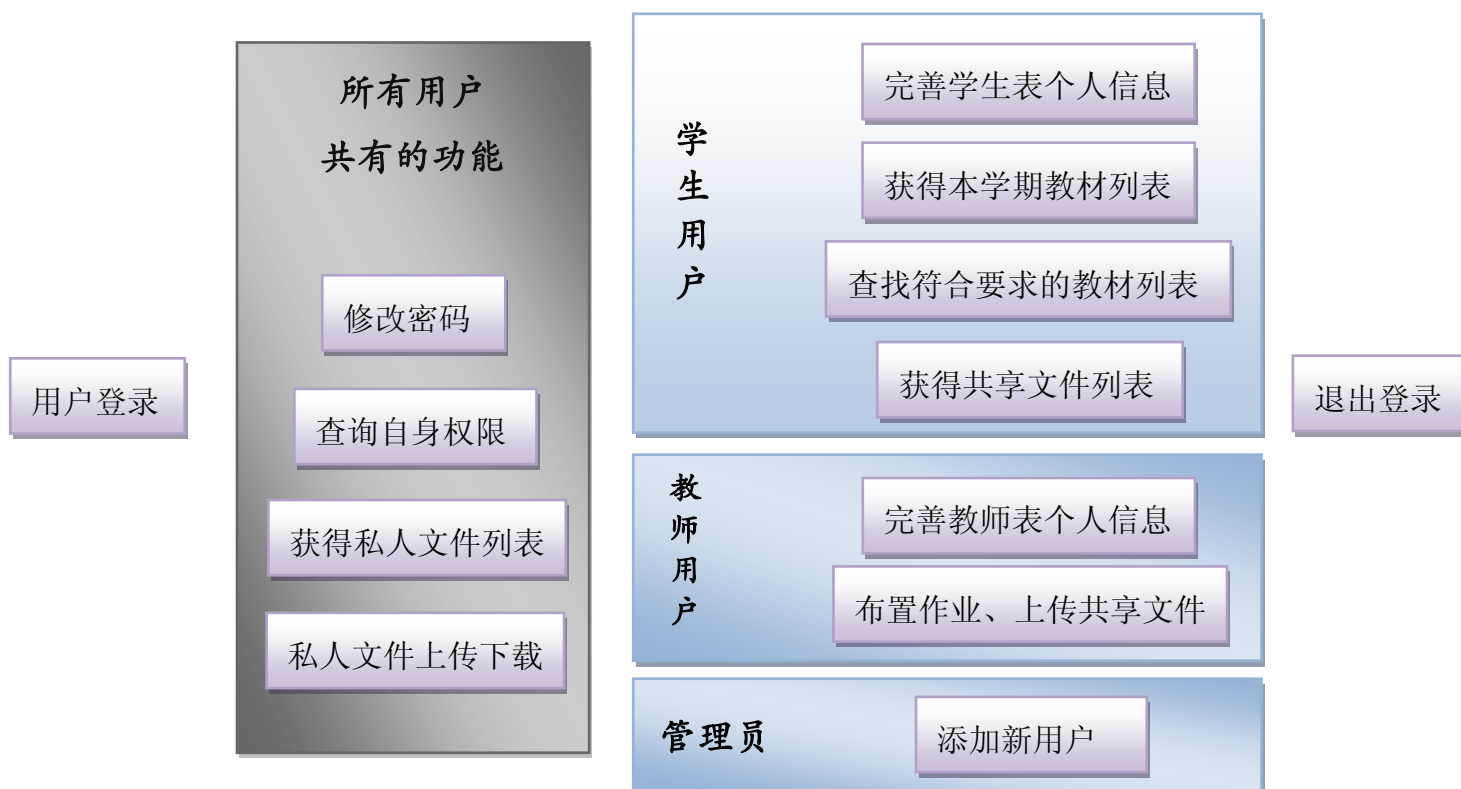


图 3.2 网络服务各功能模块的组织结构

如图中所示，在 13 个功能中，修改密码、查询权限、管理私人文件等四个功能是所有用户共有的。此外学生用户可以完善个人信息、查找并下载适用的教材、获取所学课程的共享文件列表，教师用户可以完善个人信息、上传作业或共享文件、批改作业等，管理员可以添加新的账户。

这 13 个功能模块通过与后台数据库的交互，能够很好地完成教学过程中最典型的一些环节。以学生用户的操作流程为例，该系统可实现的功能为：

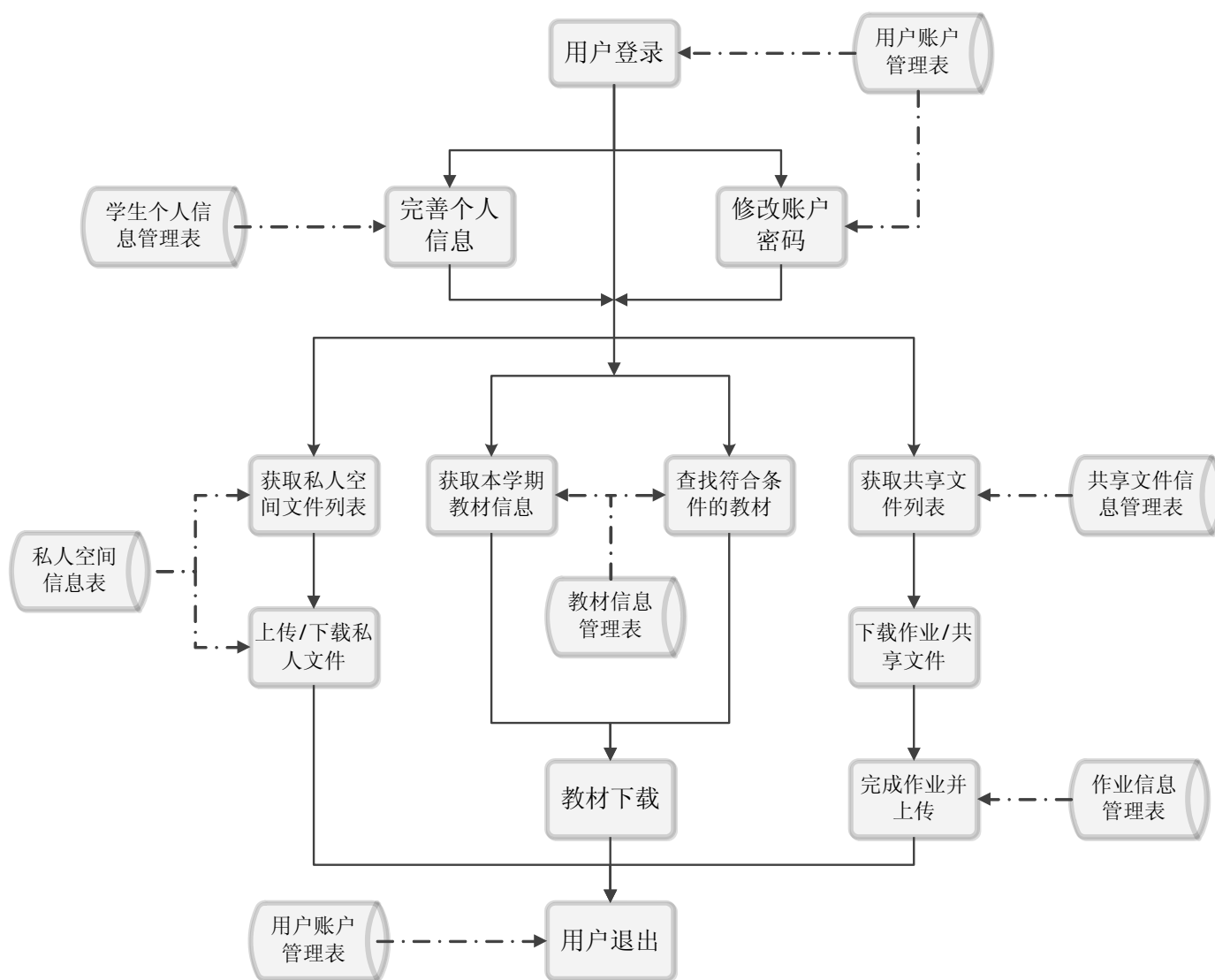


图 3.3 学生用户可实现的功能组合

接下来将详细介绍一下每个功能模块的具体实现。

### 3.3.2 网络服务的创建

本课题中的网络服务基于 JAVA 语言编写创建，基于 JAX-WS 架构实现。JAX-WS 规范是一组基于 XML 语言的 Web Service 的 JAVA API，可将远程调用转换为基于 XML 的协议，并自动将对于 API 的调用转换成对应的 SOAP 消息。

Web Service 中实现了 13 个功能模块，可以分为用户账号与个人信息管理、文件存储管理两个部分。第一部分的功能如表 3.6 所示。

表 3.6 用户账号和个人信息管理功能模块

对外接口函数	实现功能	用户上传信息	对数据库的操作
Login	用户登录	用户名、密码	账号管理表
Quit	用户登出	身份识别码	账号管理表
ChangeMyPassowrd	修改登录密码	身份识别码、新密码	账号管理表
AddNewMember	添加新用户	身份识别码、新用户的账号信息	管理员信息表
GetMyPermission	查询用户权限	身份识别码	账号管理表
FillStudentInfo	填写学生个人信息	身份识别码、个人信息各项内容	学生信息表
FillTeacherInfo	填写教师个人信息	身份识别码、个人信息各项内容	教师信息表

该部分各模块的具体实现如下：

**Login:** 用户登录。用户提交用户名密码，核对账号管理表正确后生成身份识别字符串返回用户，并记录登录时间和身份识别字符串。

```
public String userlogin (String input_name, String input_password)
```

- 若用户名不存在，返回“用户名不存在！”；
- 若密码不存在，返回“密码输入有误！”；
- 若用户名密码在数据库中存在，返回身份认证字符串并修改数据库中数据；
- 若数据库链接有误，返回“数据库连接有误！”

**ChangeMyPassword:** 用户修改密码。用户提交身份识别字符串和新密码，确认无误后修改账号管理表。

```
public String password (String input_access, String input_password1,
String input_password2)
```

- 若权限认证有误，返回“用户登录有异常！”；
- 若两次密码输入不同，返回“两次密码输入不同！”；
- 若有权限且两次密码相同，返回“密码修改成功”并修改数据库中数据；
- 若用户输入格式有误，返回“输入格式有误！”

**AddNewMember:** 管理员添加新账户。管理员提交身份识别字符串以及要添加的用户各项信息。核对管理员信息表确认管理员权限等级后在对应表中加入用户信息。

```
public String addnewmember (String input_access, String input_id, String input_username, String input_password)
```

- 若权限认证有误，返回“用户登录有异常！”；
- 若用户没有管理员权限，返回“您的当前用户没有权限！”；
- 若输入的 id 已经存在，返回“该用户 id 已存在！”；
- 若输入的用户名已经存在，返回“该用户名已存在”；
- 若输入符合要求且有权限，返回“用户添加成功”并在数据库中添加数据；
- 若用户输入不符合格式，返回“输入格式有误！”

**Quit:** 用户退出登录。用户提交身份识别字符串，核对后将账号管理表中的登录时间和身份识别置为 NULL。

```
public String userquit (String input_access)
```

- 若权限认证有误，返回“用户登录有异常！”；
- 若用户已经登录，返回“用户已成功退出！”并清空数据库中的权限认证；
- 若数据库连接有误，返回“数据库连接有误！”

**GetMyPermission:** 用户查询自己的权限。用户提交身份识别字符串，核对后将用户权限等级返回，分别为 teacher、student 和 admin 三种类型的字符串。

```
public String userpermission (String input_access)
```

- 若权限认证有误，返回“用户登录有异常！”；
- 若用户已经登录且 id 小于等于 8 位，则返回“admin”；
- 若用户已经登录且 id 为 9 位，则返回“teacher”；
- 若用户已经登录且 id 大于等于 10 位，则返回“student”；
- 若数据库连接有误，返回“数据库连接有误！”

**FillStudentInfo:** 完善学生表的个人信息。用户提交身份识别字符串，及学生的姓名、年级、班级和学校，经过权限认证后填入学生信息表。

```
public String fillstudentinfo (String input_access, String input_name,
String input_grade, String input_class, String input_school)
```

- 若权限认证有误，返回“用户登录有异常！”；
- 若用户已经登录且 id 非学生用户，则返回“您不是学生用户！”；
- 若用户是学生用户但个人信息已经存在，则返回“您的信息已经存在！”；
- 若用户是学生用户且个人信息未填写，则返回“您的个人信息填写完成！”并在数据库中添加数据；
- 若填写数据格式有误，返回“输入格式有误！”

**FillTeacherInfo:** 完善教师表的个人信息。用户提交身份识别字符串，及教师的姓名、课程、年级、学校、班级（1）、班级（2）和班级（3），经过权限认证后填入教师信息表。

```
public String fillteacherinfo (String input_access, String input_name,
String input_course, String input_grade, String input_school, String
input_class1, String input_class2, String input_class3)
```

- 若权限认证有误，返回“用户登录有异常！”；
- 若用户已经登录且 id 非教师用户，则返回“您不是教师用户！”；
- 若用户是教师用户但个人信息已经存在，则返回“您的信息已经存在！”；
- 若用户是教师用户且个人信息未填写，则返回“您的个人信息填写完成！”并在数据库中添加数据；
- 若填写数据格式有误，返回“输入格式有误！”

文件存储管理部分共有 6 个功能模块，其功能如表 3.7 所示：

表 3.7 文件存储管理部分功能模块

对外接口函数	实现功能	用户上传信息	对数据库的操作
GetMyFiles	获得本学期教材列表	身份识别码	教材信息表
SearchFiles	查找符合条件的教材	身份识别码，教材编号	教材信息表
GetMyPrivateFiles	私人文件列表	身份识别码	私人文件信息表
GetSharedFiles	共享文件列表	身份识别码	共享文件信息表

UpLoadSharedFiles	上传共享文件	身份识别码	共享文件信息表
UpLoadPrivate	上传私人文件	身份识别码	私人文件信息表

该部分各模块的具体实现如下：

**GetMyFiles:** 获得本学期教材列表。用户提交身份识别字符串，通过查询账号管理表和学生信息表获得年级，从教材信息表中获取符合该年级的所有教材信息并返回。

```
public String[][] mytextbook (String input_access)
```

- 若权限认证有误，返回“用户登录有异常！”；
- 若用户不是学生账号，返回“该用户不是学生用户！”；
- 若用户个人信息表格尚未填写，返回“该用户信息未完善！”；
- 若符合要求则按用户年级查找到教材，返回字符串数组存放教材的信息；
- 若填写数据格式有误，返回“输入格式有误！”

**SearchFiles:** 用户查找符合要求的教材。用户提交身份识别字符串以及要搜索教材的编号，从教材信息表中获取记录后返回用户。

```
public String[][] searchfiles (String input_access, int input_number)
```

- 若权限认证有误，返回“用户登录有异常！”；
- 若用户已登录，返回字符串数组存放教材信息查询结果；
- 若填写数据格式有误，返回“输入格式有误！”

**GetMyPrivateFiles:** 获得私人上传文件列表。用户提交身份识别字符串，通过搜索私人空间表来获得用户私人空间文件，依次返回文件所有信息，包括文件的下载地址。

```
public String[][] getprivate (String input_access)
```

- 若权限认证有误，返回“用户登录有异常！”；
- 若用户已登录，返回用户个人空间中文件的文件名和下载链接；
- 若填写数据格式有误，返回“输入格式有误！”

**GetSharedFiles:** 获得共享文件列表。用户提交身份识别字符串，通过搜索共享文件表获得满足条件的文件，依次返回文件的所有信息包括下载地址。

```
public String[][] mytextbook (String input_access)
```

- 若权限认证有误，返回“用户登录有异常！”；
- 若用户不是学生账号，返回“该用户不是学生用户！”；
- 若用户个人信息表格尚未填写，返回“该用户信息未完善！”；
- 若符合要求则按用户学校、年级、班级查找到共享文件，返回字符串数组存放文件的信息、课程、上传老师姓名和下载地址；
- 若填写数据格式有误，返回“输入格式有误！”

**UploadSharedFiles:** 老师上传共享文件。用户提交身份识别字符串，通过搜索账号管理表和教师信息表获得老师信息，在共享文件表中添加相应记录并返回上传文件的地址。

```
public String uploadsharedfiles (String input_access, String  
input_class, String input_title)
```

- 若权限认证有误，返回“用户登录有异常！”；
- 若用户不是教师，返回“您不是教师用户！”；
- 若用户信息没有填写，返回“该用户信息未完善！”；
- 若用户不是该班级的教师，返回“您没有该班级的操作权限！”；
- 若用户上传的文件名已存在，返回“该文件名已存在！”
- 若符合要求则按用户学校、年级、班级确定文件地址，返回文件上传地址；
- 若填写数据格式有误，返回“输入格式有误！”

**UploadPrivateFiles:** 用户上传私人文件。用户提交身份识别字符串，通过搜索账号管理表得到用户 ID，在私人空间表中添加相应的字段并返回上传到的地址。

```
public String uploadsharedfiles (String input_access, String  
input_title)
```

- 若权限认证有误，返回“用户登录有异常！”；
- 若用户上传的文件名已存在，返回“该文件名已存在！”

- 若符合要求且用户已登录，返回文件上传地址；
- 若填写数据格式有误或数据库连接有误，返回“数据库连接有误！”

### 3.4 Web 服务的发布和功能测试

在编写好各模块的功能之后，可以利用 MyEclipse 中的 JAX-WS 架构，将对 JAVA 中 API 的调用转化为对应的 SOAP 消息。每个 Web Service 后台程序都会根据框架自动生成 Delegate.java 的类函数，作为外部调用的接口。最终程序编写完成后需要配置好 tomcat 5.5 的环境，在 MyEclipse 开发环境中选择 Run -> Run As -> MyEclipse Server Application，在运行之后该网络服务平台即发布成功。

服务发布成功后通过访问 <http://localhost:8080/education/loginPort> 即可得到如图 3.4 所示的网页。

#### Web Services

Endpoint	Information
Service Name: (http://User_Login/) loginService Port Name: (http://User_Login/) loginPort	Address: <a href="http://localhost:8080/education/loginPort">http://localhost:8080/education/loginPort</a> WSDL: <a href="http://localhost:8080/education/loginPort?wsdl">http://localhost:8080/education/loginPort?wsdl</a> Implementation class: User_Login.loginDelegate
Service Name: (http://Change_Password/) changepasswordService Port Name: (http://Change_Password/) changepasswordPort	Address: <a href="http://localhost:8080/education/changepasswordPort">http://localhost:8080/education/changepasswordPort</a> WSDL: <a href="http://localhost:8080/education/changepasswordPort?wsdl">http://localhost:8080/education/changepasswordPort?wsdl</a> Implementation class: Change_Password.changepasswordDelegate
Service Name: (http://Add_Member/) addmemberService Port Name: (http://Add_Member/) addmemberPort	Address: <a href="http://localhost:8080/education/addmemberPort">http://localhost:8080/education/addmemberPort</a> WSDL: <a href="http://localhost:8080/education/addmemberPort?wsdl">http://localhost:8080/education/addmemberPort?wsdl</a> Implementation class: Add_Member.addmemberDelegate
Service Name: (http://User_Quit/) quitService Port Name: (http://User_Quit/) quitPort	Address: <a href="http://localhost:8080/education/quitPort">http://localhost:8080/education/quitPort</a> WSDL: <a href="http://localhost:8080/education/quitPort?wsdl">http://localhost:8080/education/quitPort?wsdl</a> Implementation class: User_Quit.quitDelegate
Service Name: (http://Get_Permission/) permissionService Port Name: (http://Get_Permission/) permissionPort	Address: <a href="http://localhost:8080/education/permissionPort">http://localhost:8080/education/permissionPort</a> WSDL: <a href="http://localhost:8080/education/permissionPort?wsdl">http://localhost:8080/education/permissionPort?wsdl</a> Implementation class: Get_Permission.permissionDelegate
Service Name: (http://Student_Info/) studentinfoService Port Name: (http://Student_Info/) studentinfoPort	Address: <a href="http://localhost:8080/education/studentinfoPort">http://localhost:8080/education/studentinfoPort</a> WSDL: <a href="http://localhost:8080/education/studentinfoPort?wsdl">http://localhost:8080/education/studentinfoPort?wsdl</a> Implementation class: Student_Info.studentinfoDelegate
Service Name: (http://Teacher_Info/) teacherinfoService	Address: <a href="http://localhost:8080/education/teacherinfoPort">http://localhost:8080/education/teacherinfoPort</a> WSDL: <a href="http://localhost:8080/education/teacherinfoPort?wsdl">http://localhost:8080/education/teacherinfoPort?wsdl</a>

图 3.4 Web Service 发布成功后访问所得网页

可以看到在网页中包含了 13 个服务功能模块，以及每个模块的服务名称、端口名称、访问地址、接口地址和 WSDL 访问链接等信息。我们打开用户登录功能对应的 WSDL 链接，可以看到发布的 WSDL 文件如图 3.5 所示，该 WSDL 文件的发布地址为 <http://localhost:8080/education/loginPort?wsdl>。



```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-UI
<!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-UI
- <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xml
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:xsd="http
targetNamespace="http://User_Login/">
+ <types>
+ <message name="userlogin">
+ <message name="userloginResponse">
+ <portType name="loginDelegate">
+ <binding name="loginPortBinding" type="tns:loginDelegate">
+ <service name="loginService">
</definitions>

```

图 3.5 用户登录功能的 WSDL 文档

可以看到每个功能模块都已经成功发布了对应的 WSDL 文档。文档是基于 XML (extensible markup language, 可扩展标记语言) 描述的, 具有良好的可读性, 可以摆脱平台和编程语言的束缚。可以看到在 WSDL 文档中定义了该 Web 服务的地址、名称、参数格式、返回内容格式等等接口信息, 以使用户在得到该 WSDL 文件后可以通过解读其中含义而使用此项 Web 服务。

在所有功能模块的 WSDL 文件已经成功发布之后, 网络服务器部分的开发工作就已经告一段落。我们可以如图 3.6 所示从 Web Service Explorer 中以 WSDL 通信的方式进行检测。

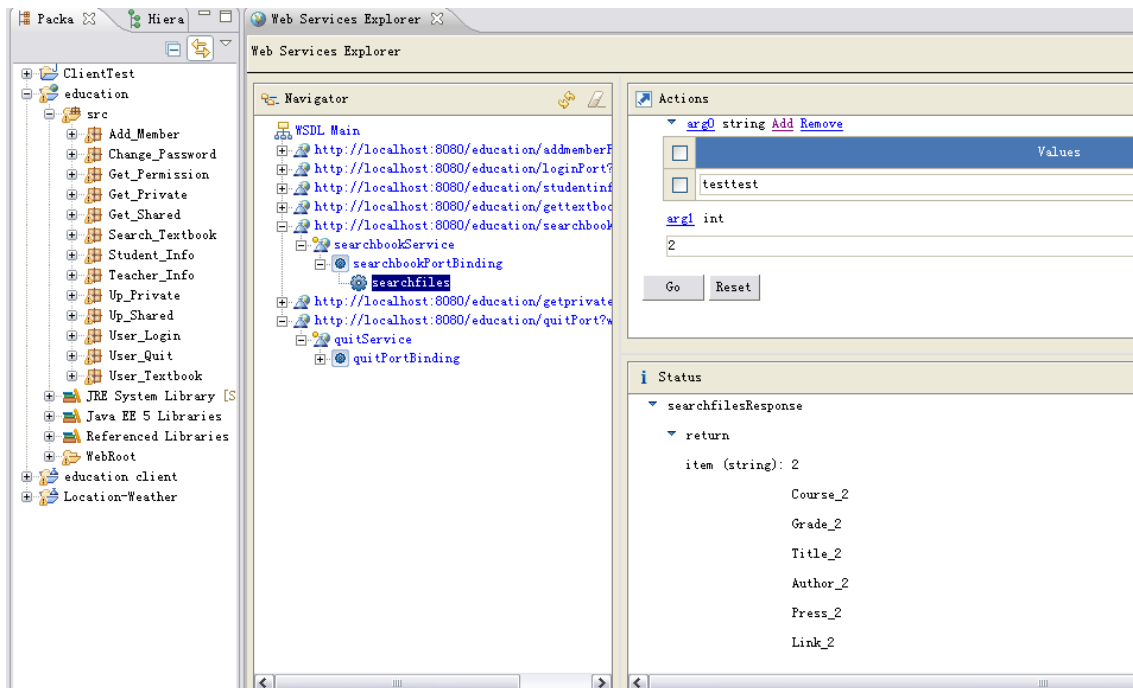


图 3.6 利用 Web Service Explorer 测试结果

该浏览器视图的左侧是后台运行的程序包，以及已经连接上的 WSDL 节点。通过进行添加用户、用户登录、用户信息完善等一系列步骤，之后进行如图所示的教材搜索工作。用户上传自己的身份识别码，以及要查找的教材编号，返回值如图右下方所示，为查找到的教材的课程、年级、书名、作者等信息。

### 3.5 客户端功能实现与测试

网络平台建立之后，下一步需要开发相应的 web service client 程序，以实现客户端与网络服务器之间的交互。在本课题中，我们在基于 Android 系统的 ipad 上开发了与网络服务相对应的 JAVA 客户端，并完成了简单的图形界面。学生们可以在课堂上利用安装了客户端的智能终端设备进行教学环节，摆脱传统纸质教材的束缚。下面对客户端的功能进行简单的介绍和测试。

客户端程序在 ipad 上运行之后，可自动连接到配置好的网络服务器端。首先是需要输入用户名和密码进行身份验证，其界面如图 3.7 所示。



图 3.7 客户端程序用户登陆界面

用户需要输入正确的用户名和密码进行登录。用户名和密码将发送到网络服务器端与数据库中的数据进行对比验证，如果完全正确则用户登陆成功，登录后的主界面如图 3.8 所示。

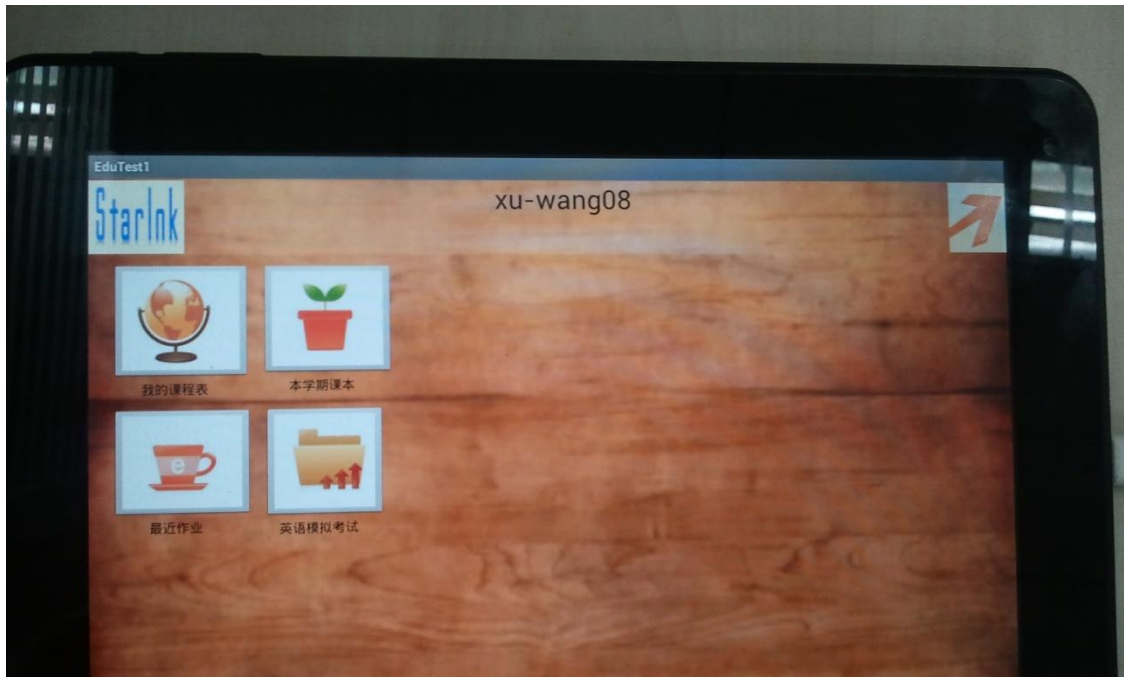


图 3.8 用户登录后客户端主界面

现阶段客户端已经完成了四个功能的调试，分别是查找本学期教材、查看老师布置的作业、上传私人文件以及完善个人信息。其中查找本学期的教材演示效果如图 3.9 所示。

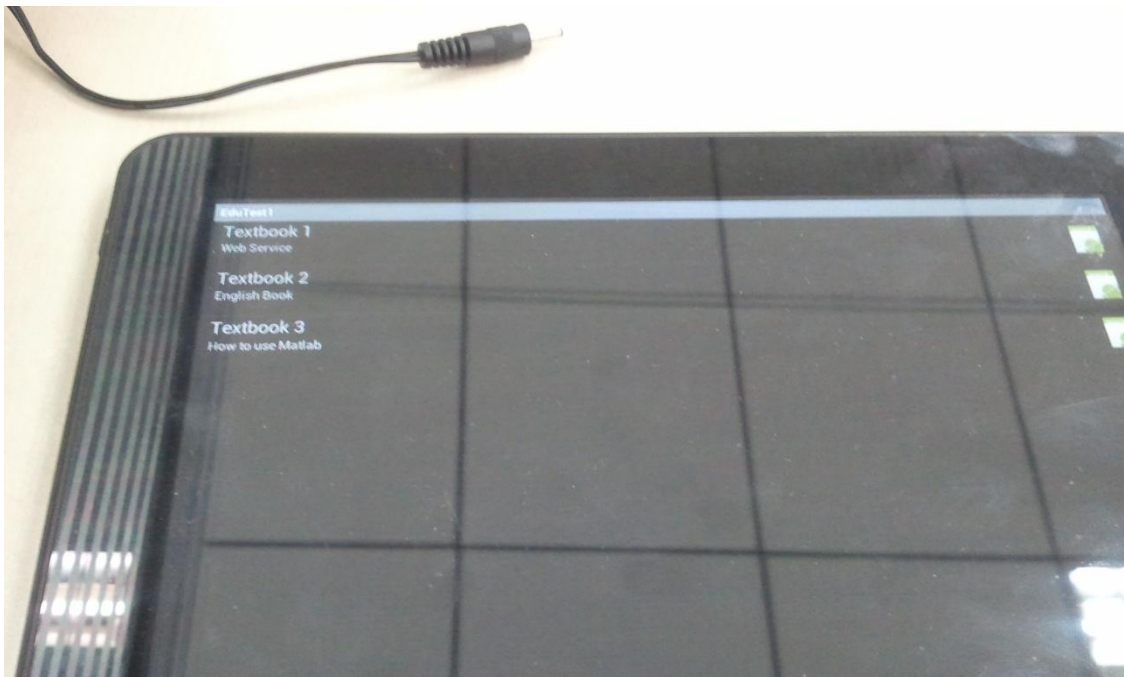


图 3.9 教材查找服务显示结果

学生请求查询本学期教材的服务时，会由客户端程序将自己的身份验证信息发送到服务器端。服务器在数据库中查找学生用户的个人信息，确定学生的在学年级，之后再从统计教材信息的表格中找出所有该年级同学使用的教材，将列表中教材的信息以二维字符串数组的形式返回到客户端程序上，之后由客户端程序显示在界面上。同时网络服务器将每本教材的下载链接也会发送到客户端，同学可根据需要进行教材下载。

除此之外，客户端程序还利用手写板的功能开发了课堂笔记的应用，同学们可以在听课或自习的过程中对课本进行手写批注，之后可以将做过笔记的教材保存在个人私有空间上。

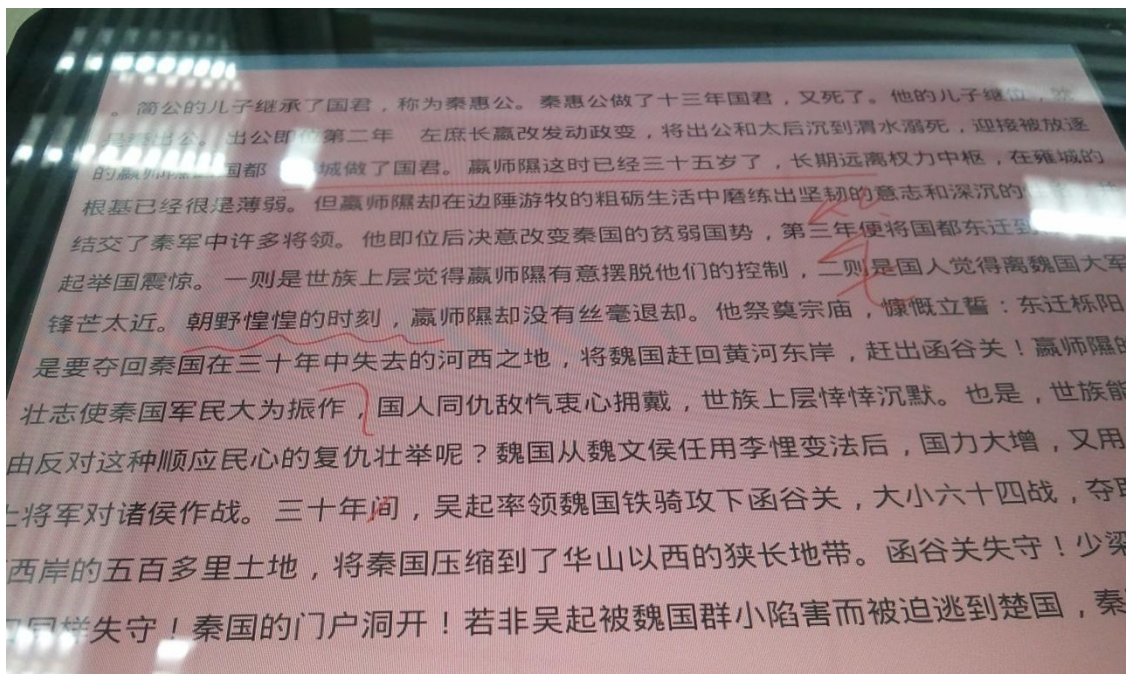


图 3.10 对教材批注功能的演示

## 第4章 云平台的功能实现

如第 2 章需求分析与系统架构设计中所述，该课题的垂直云平台搭建可分为三层：云平台、网络服务器和客户端。第 3 章主要讲述了网络服务器的架构与具体实现，本章将具体阐述云平台的主要功能和具体实现方式。云平台主要有两个功能，分别为分布式文件存储系统和虚拟机计算资源的调度。

### 4.1 分布式文件存储系统

如前文所述，教育网络中需要存储大量的数据，所以传统的物理存储资源直接连接在本地节点上的设计方法是无法适应的。我们需要通过计算机网络与数据存储节点相连，在网络服务器的数据库部分维护统一的文件索引列表（即第 3 章中提到的文件信息管理数据库），以实现文件的分布式存储。

#### 4.1.1 使用 Apache 搭建网络服务器

首先我们需要构建单独的数据存储节点。在本课题中我们使用 Apache http Server 搭建节点网络服务器。Apache 是世界排名第一的网络服务软件，可以运行在几乎所有计算机平台之上。该软件有着结构简单、速度快、性能稳定的优点[14]，成为我们开发网络服务器的第一选择。

Apache 安装成功之后，需要对网络服务器的特性进行相应的配置，以适应我们的需要。由于它是采用纯文本文件配置的方式，所以修改配置文件后再重新启动即可完成配置工作。

Apache 软件有以下几个配置文件，目录通常存放在“\Apache\conf”中。

- **Httpd.conf**: 最主要的配置文件。对服务器的特性有着很大的影响，往往对此文件有较多修改；
- **Access.conf**: 上一文件中 AccessConfig 指令的缺省文件，主要控制用户权限，如对文件的读写限制等；
- **Srm.conf**: 主要描述了服务器向用户浏览器提供的文件类型指令。是 Httpd.conf 中 ResourceConfig 缺省的文件。
- **Mime.type**: 记载网络服务器可以识别出的所有 MIME 格式。

在本项目所需的配置中，我们主要会改变 Httpd.conf 文件中的相关配置。以

记事本的方式打开该配置文件，进行如下几个方面的配置：

- 配置指令 **KeepAlive on**：使搭建的网络服务器支持断点续传的功能；
- 配置指令 **MaxKeepaliveRequest 200**：表示该服务器可支持最大的用户连接数为 200；
- 配置指令 **MaxRequestPerChild 0**：该配置修改的是同一时间子进程的数量，设置为零可使网络服务器更加稳定；
- 配置指令 **Prot 80**：设置外部访问的端口号，默认为 80，可在输入时缺省端口号；
- 配置指令 **ias/private/ “C:/education/private”**：改变文件存储路径，表示 `http://localhost/private` 访问的实际地址是 `C:\education\private` 而不是 `C:\Apache\htdocs\private`。

完成相关配置后，运行 Apache 服务器，可以看到如图 4.1 所示的文件服务器已经搭建完毕，文件根目录下有三个文件夹，分别存放学生上交的作业、老师共享的文件和每个人的私人空间。之后只要定位到目标文件明确的路径即可实现文件的上传和下载功能。

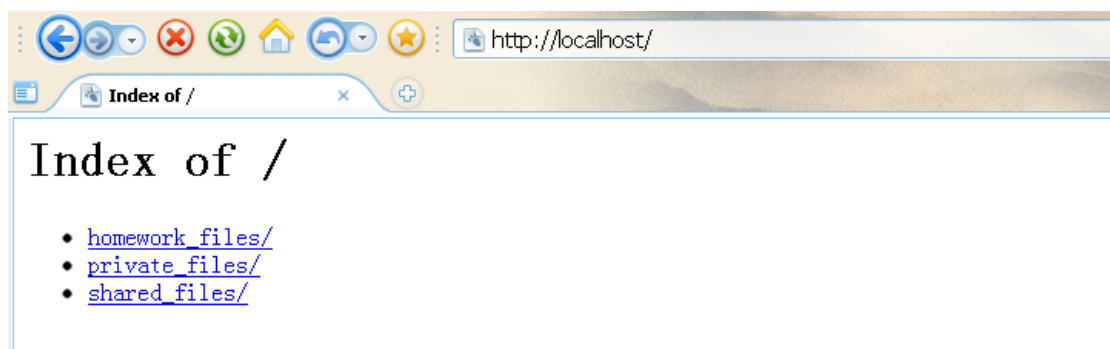


图 4.1 网络服务器搭建成功后可访问的文件目录

#### 4.1.2 分布式文件系统的组织结构

在完成了每个数据存储节点的服务器搭建后，系统中可以通过计算机网络实现数据的分布式存储和文件的统一检索管理。具体来说，该项目的文件存储系统组织结构如图所示。

如图 4.2 所示，网络中的数据存储在许多个物理位置不同的数据存储节点上。用户在进行文件管理时，先向网络服务器发送服务请求，通过调用 Web Service 服务模块与文件存储管理数据库进行沟通。通过数据库中表格的索引向用户返回文件存储的具体位置，之后用户将遵循相应目录下载或上传目标文件。

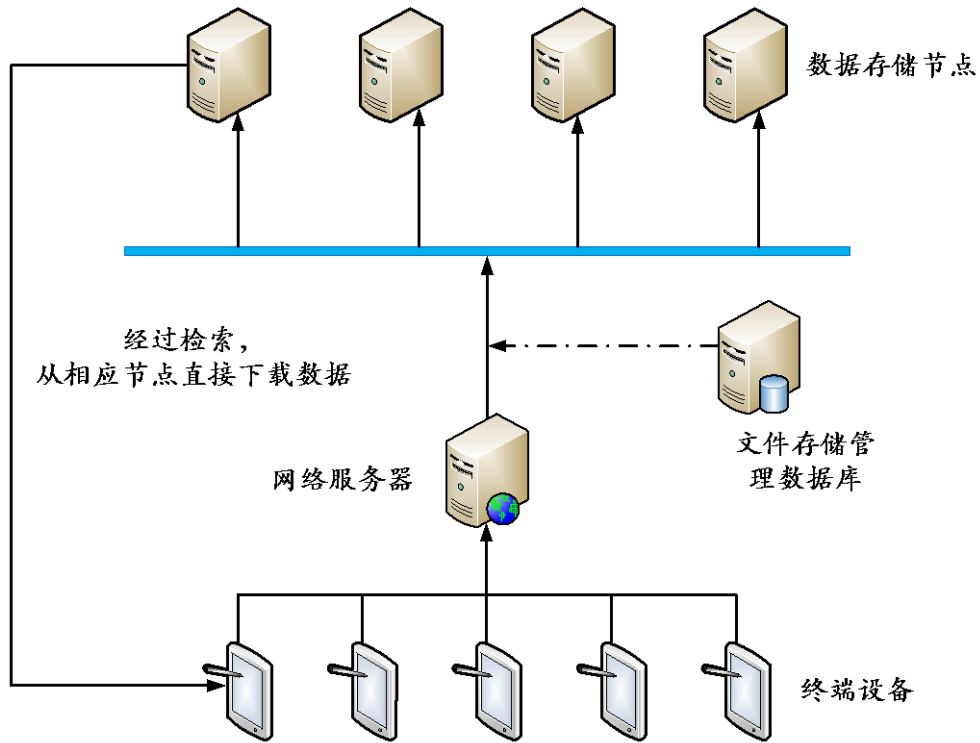


图 4.2 分布式文件系统组织架构

例如在数据库中“教材信息表”中，有一个字段是“下载链接”，作为每个文件存储位置的索引。当学生要下载本学期教材时，客户端将通过调用 GetMyFiles 服务模块，得到所有符合要求的教材列表的信息，其中包括“下载链接”。之后用户可根据自己的需要进行下载。其演示结果如图 4.3 所示。

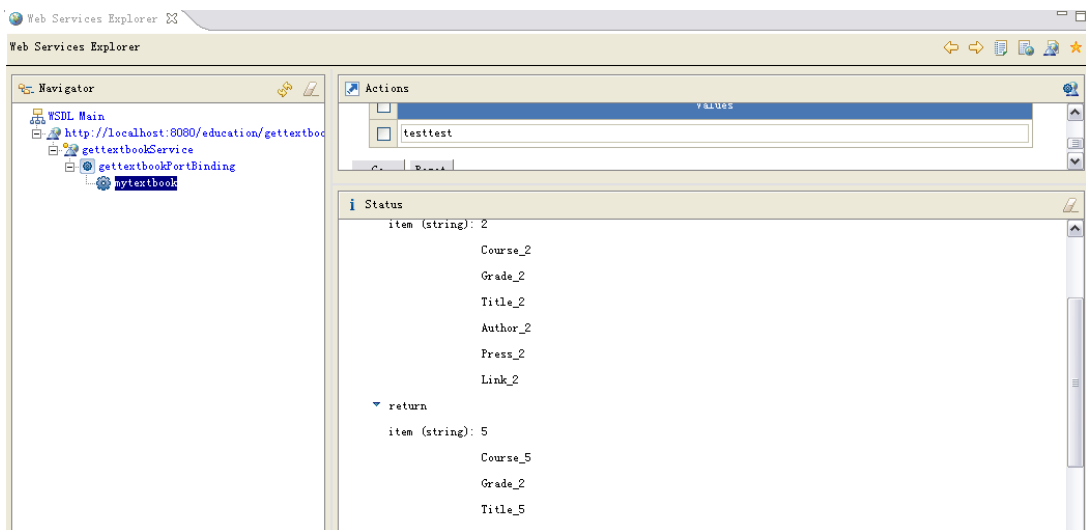


图 4.3 学生用户查找教材时的返回结果

## 4.2 虚拟机资源调度

云计算的一大特点就是虚拟机的调度和管理。通过灵活地调度虚拟机的开启与关闭，可以达到在高峰时段很好地应对网络压力、在空闲时段避免资源的过多浪费的目的。在本课题中，我们应用 `virtualbox` 虚拟机进行开发。相比于其它的虚拟机，`virtualbox` 有着其特有的几个优势[15]：

- 对主机资源占用少，在虚拟机上部署的系统运行速度快，安装文件也比其他虚拟机要小很多；
- `Virtualbox` 使用的是 XML 语言来描述每台虚拟机，在向其他电脑移植方面十分方便；
- 虚拟机和物理机之间可以通过共享文件夹的形式进行文件交流；
- 有很强的模块化设计，界定了明确的内部编程接口和客户端/服务器设计。
- 在选定了虚拟机开发环境后，接下来需要先后完成以下两项工作。

### 4.2.1 在虚拟机上部署网络服务器

我们需要在安装了 Linux 操作系统的虚拟机上部署我们开发完成的网络服务器。首先我们需要在虚拟机上安装 JDK，安装成功后在终端界面输入 `java -version` 命令后会出现如图 4.4 所示的信息表示安装成功。

```
java version "1.6.0_06"  
Java(TM) SE Runtime Environment (build 1.6.0_06-b02)  
Java HotSpot(TM) Client VM (build 10.0-b22, mixed mode, sharing)
```

图 4.4 表示 JDK 安装成功的信息

之后我们要在虚拟机操作系统上安装并配置好 `tomcat`。在 `tomcat` 安装完毕后，为了使以后系统启动后就配置好 `tomcat` 所需的环境，需要在 `/etc/profile` 文件中进行相应的设置：

```
CATALINA_BASE="/usr/local/jakarta-tomcat-5.0.28"  
CATALINA_HOME="/usr/local/jakarta-tomcat-5.0.28"  
PATH=$PATH:$CATALINA_HOME/bin  
export CATALINA_BASE CATALINA_HOME PATH
```

环境变量配置好后在浏览器中输入 `http://localhost:8080`，可以看到浏览器中出现如下页面，表示 `tomcat` 已经安装并配置成功。



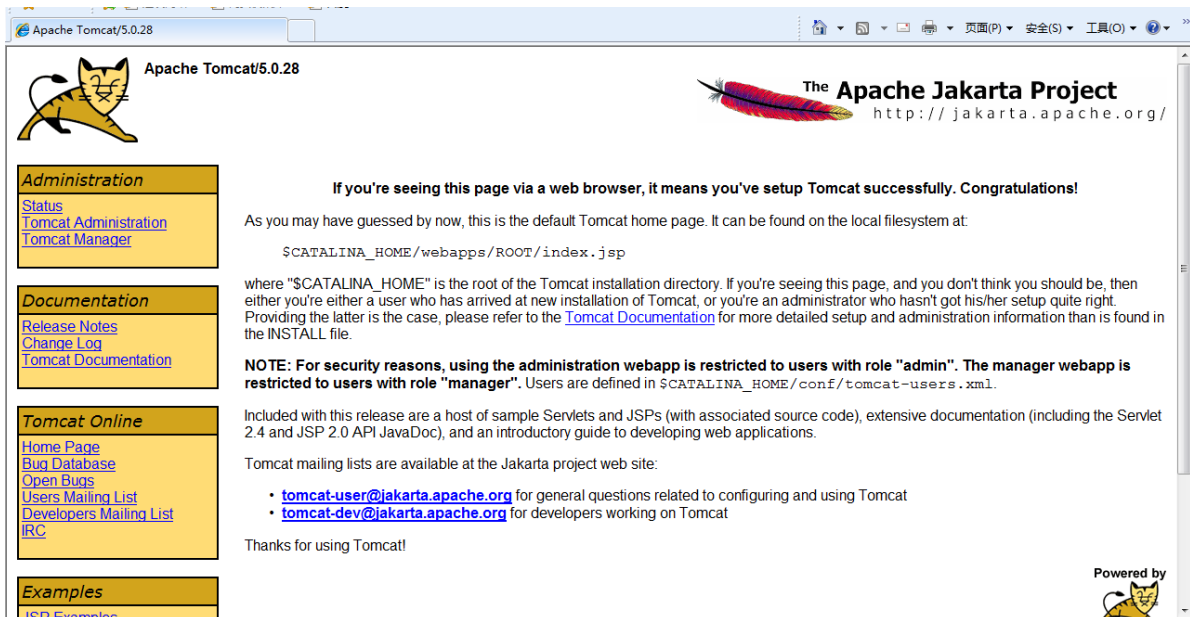


图 4.5 Tomcat 安装配置成功后的网页信息



图 4.6 AddNewMember 服务发布的 WSDL 文档

当 JDK 与 tomcat 均安装配置完成后，就可以将在 windows 下开发的网络服务器应用程序部署在 tomcat 上，以实现网络服务器在虚拟机上的运行与发布。将在 MyEclipse 中开发的 JAVA 文件（education 文件夹）拷贝到 tomcat 安装目录下的 webapps 目录下，之后重启 tomcat。再打开浏览器在地址栏中输入 http://localhost:8080/education/addmemberPort?wsdl，即可看到浏览器中出现如图 4.6 所示的 AddNewMember 服务的 WSDL 文档，表示网络服务器已经部署成功。

至此，网络服务器的功能已经能够完全部署在安装 Linux 系统的虚拟机上了。

#### 4.2.2 资源调度基础功能的实现

在网络服务器部署成功之后，需要实现虚拟机按访问量的多少进行灵活调度。由于实验室环境下资源和精力有限，本课题只实现了其中最典型、最基础的一部分功能。

首先按照上节的步骤配置好两台完全一样的 Linux 操作系统的虚拟机，安装部署的软件包括 JDK、MySQL 5.0、Tomcat 6.0 等。由于需要有区别地连接到两台不同的虚拟机上，所以需要特别设置一下虚拟机的网络连接属性，将其设置为 Use network address translation（NAT 网络）。在这种情况下虚拟机可以对外访问其他网络站点，但是外界不能直接连接到该虚拟机[16]。如图 4.7 所示。

在该图中 A1 和 A2 是物理机 A 的虚拟机，B1 是物理机 B 的虚拟机。当设置为“NAT 网络”属性时，A1 和 A2 可以单向访问主机 B、C，而主机 B、C 不能直接访问 A1 和 A2；A1、A2 和 A 可以通过内部网络分配的 IP 互访。

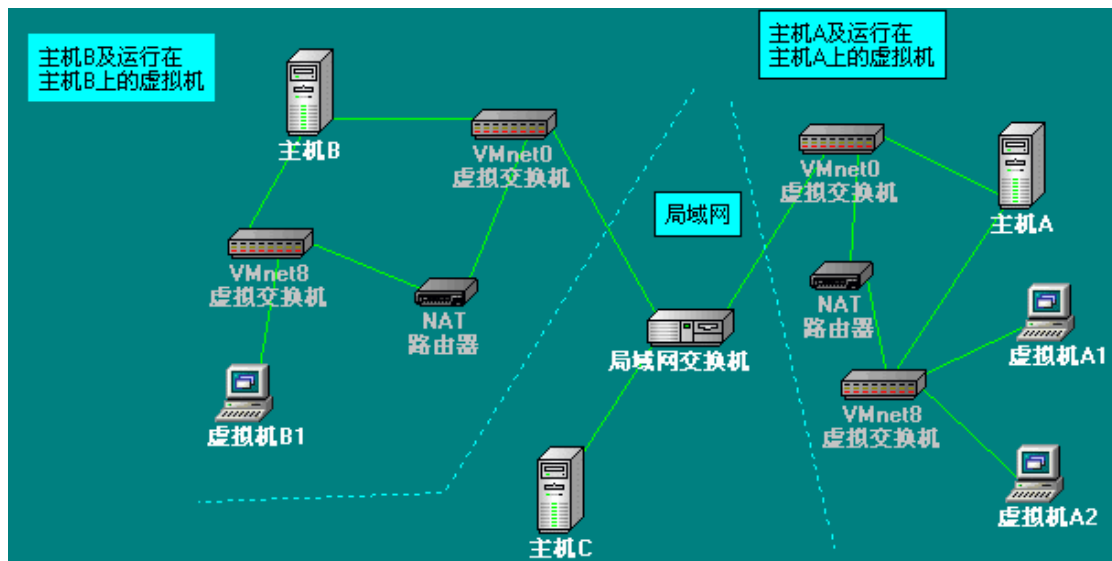


图 4.7 NAT 网络设置的虚拟机网络模型

所以需要在物理机上再独立运行一个 Web Service 模块，进行外部用户连接数的计数以及内部网络 ip 的分配，并且监控每台虚拟机正在服务的用户连接数量。当一台虚拟机的连接数量达到一定的阈值时，物理机将启动下一台虚拟机，并将接下来的连接任务向资源空闲最多的虚拟机进行分配。其功能架构如图 4.8 所示。

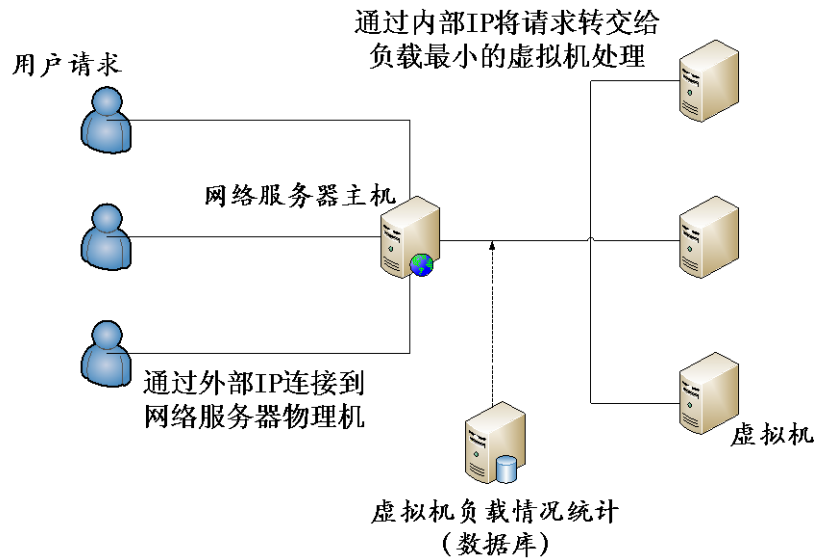


图 4.8 虚拟机调配部分的功能结构图

## 第5章 总结与展望

### 5.1 本文工作总结

随着云计算技术的日益成熟和广泛应用，渐渐发展出了满足不同行业特殊需求的垂直云计算架构。搭建垂直云计算平台，需要了解分析本行业的特殊需求和应用特点，有针对性地对云平台的功能进行优化和改良；同时利用云计算中分布式计算、虚拟化等技术，为用户提供比以往更好的服务。

本课题主要关注于教育行业，以设计和实现服务于教育行业的大型网络资源共享平台为目标，利用云计算、Web Service 网络服务等多项技术，通过实现云平台、网络服务器、客户端三方的交互，对基于垂直云平台的下一代教育行业的网络应用进行了创新性探索。本文从现已实现的功能出发，介绍了一下几个方面的工作情况：

1. 在实验室环境下搭建了三层网络结构，分别是云计算平台、网络服务器和客户端。针对教育行业特有的需求和特点提供服务，基本可以满足于中小学教育的整个过程；

2. 应用 Web Service 技术搭建了面向服务的架构，完成了 13 个独立的网络功能模块。每一个功能模块都是基于标准因特网协议的，且能够适应不同的开发平台和编程语言；

3. 在云平台端建立了分布式文件存储系统，可以满足大量数据存储的需求；同时使用虚拟化技术，通过虚拟机的调度实现计算资源更加高效的利用。

### 5.2 未来工作展望

由于该课题的工程性和应用性很强，在实验室的环境下受条件和人员精力所限尚难以真正与实际需求接轨、处理现实问题。目前虽然搭建了具有基本功能的网络平台 and 客户端，但是功能上距离投入应用还有很大的差距，组织架构亦不如实际大型系统成熟稳定。今后本课题的科研实践中需要进一步考虑以下几个方面的问题：

1. 进一步完善网络服务器所提供的服务，需要将服务架构扩展到教学过程的各个环节，如课堂考试、学生相互交流、课堂远程桌面交互等等，同时希望能够

让学生家长也参与到教学过程中，能够加强老师、家长、学生的三方互动；

2. 要加强网络服务器端的数据安全保障。现有的系统中并未对任何交互内容进行加密，数据库的安全性也不是很强。要最终面向实际的应用，需要加强数据在网络传输以及存储方面的安全性；

3. 客户端需要有更加良好的用户体验。为了适应信息时代发展的潮流，本工程中开发了基于 **Android** 系统的客户端，可以让学生在智能终端上进行操作。但是现有客户端只能实现基本的交互功能，需要进一步拓展功能、完善用户体验。

4. 云平台需要有更有效的虚拟机调度策略。现有的虚拟机调度算法还十分肤浅，只是具备了虚拟机管理的基本功能，尚无法实现高效的计算资源调度，需要在今后的工作中进一步加强这方面的研究。



## 插图索引

图 1.1	Web Service 的经典架构 .....	2
图 1.2	SOAP 典型的消息结构及 WSDL 对接口的描述 .....	3
图 1.3	Google 协作平台示例站点.....	4
图 1.4	微软公司开发的 Health Vault 医疗垂直云平台 .....	5
图 1.5	Google Health 平台已停止运营 .....	6
图 2.1	典型的学校管理体系模型.....	7
图 2.2	教育网垂直云计算平台架构.....	9
图 3.1	用户权限管理数据库逻辑结构 .....	12
图 3.2	网络服务各功能模块的组织结构 .....	15
图 3.3	学生用户可实现的功能组合 .....	16
图 3.4	Web Service 发布成功后访问所得网页 .....	22
图 3.5	用户登录功能的 WSDL 文档 .....	23
图 3.6	利用 Web Service Explorer 测试结果 .....	23
图 3.7	客户端程序用户登陆界面.....	24
图 3.8	用户登录后客户端主界面.....	25
图 3.9	教材查找服务显示结果 .....	25
图 3.10	对教材批注功能的演示 .....	26
图 4.1	网络服务器搭建成功后可访问的文件目录.....	28
图 4.2	分布式文件系统组织架构.....	29
图 4.3	学生用户查找教材时的返回结果 .....	29
图 4.4	表示 JDK 安装成功的信息 .....	30
图 4.5	Tomcat 安装配置成功后的网页信息 .....	31

图 4.6	AddNewMember 服务发布的 WSDL 文档 .....	31
图 4.7	NAT 网络设置的虚拟机网络模型 .....	32
图 4.8	虚拟机调配部分的功能结构图 .....	33



## 表格索引

表 3.1	表 user_account 的结构 .....	12
表 3.2	表 textbook_info 的结构 .....	13
表 3.3	表 shared_files 的结构 .....	13
表 3.4	表 homework_files 的结构.....	14
表 3.5	表 private_files 的结构 .....	14
表 3.6	用户账号和个人信息管理功能模块 .....	17
表 3.7	文件存储管理部分功能模块.....	19



## 参考文献

- [1] Kai Hwang, Jack Dongarra, Geoffrey C.Fox, *Distributed and Cloud Computing: From Parallel Processing to the Internet of things*, Morgan Kaufmann, 2011.
- [2] [http://en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing)
- [3] Liang-Jie Zhang, Carl K Chang, Ephraim Feig, Robert Grossman, Keynote Panel, *Business Cloud: Bringing The Power of SOA and Cloud Computing*, pp.xix, 2008 IEEE International Conference on Services Computing (SCC 2008), July 2008
- [4] [http://www.webopedia.com/TERM/V/vertical\\_cloud\\_computing.html](http://www.webopedia.com/TERM/V/vertical_cloud_computing.html)
- [5] Liang-Jie Zhang, Jia Zhang, Hong Cai, *Services Computing*, Springer and Tsinghua University Press, 2007, ISBN: 978-3-540-38281-2, July 2007
- [6] Nicholas. Carr, Yan Yu, *"IT is no longer important: the Internet great change of the high ground - cloud computing," The Big Switch: Rewining the World from Edison to Google*, CITIC Publishing House, October 2008 1-1
- [7] 刘露, 房秉毅, 张云勇, 基于云计算技术的智能云网络架构研究[J], 《电信网技术》, 2011. 5 (5) : 39-42
- [8] Y. Khmelevsky and V. Voytenko, "Cloud computing infrastructure prototype for university education and research," In the Proceedings of the 15th Western Canadian Conference on Computing Education (WCCCE 2010), May 7 - 8, 2010, Kelowna, British Columbia, Canada, ACM, Inc., New York, NY, USA, 2010.
- [9] 贺小华, 云计算在教育中的应用——以 Google 协作平台为例[J], 《软件导刊 (教育技术)》, 2009. 4 (9) : 71-74
- [10] <http://www.healthvault.com/>
- [11] <http://www.google.com/health>
- [12] 曹利娥, 新时期中学教育的新特点和新问题[J], 《唐都学刊》, 2011. 17 (2) : 222-224
- [13] Shuai Zhang, Shufen Zhang, Xuebin Chen, Xiuzhen Huo, *Cloud Computing Research and Development Trend*, second international conference on future networks, 2010.
- [14] <http://www.gootry.com/wangzhuan/article/100509000236/21>

- [15] S. E. Kern, D. Jaron , *Healthcare technology, economics and policy: an evolving balance*,  
IEEE Engineering in Medicine and Biology Magazine, vol. 22, pp. 16-19, 2003
- [16] 倪晚成, 刘连臣, 吴澄, Web 服务组合方法综述。《计算机工程》: 2008, 34(4): 79-81.

## 致 谢

在论文即将完成之际，首先要感谢我的导师曹军威老师在我的毕业设计过程中耐心的帮助和细致的指导。从上学期的寒假开始，曹老师就带领着我一步步走上了科研工作的道路。作为一个刚刚接触科研工作的同学，我在整个毕业设计的过程中犯了不少的错误，也多次走入误区而不知该如何处理。曹老师对我的指导和点评使我打开了思路，让我一次次战胜困难、向着正确的方向前进。在此，我希望对我科研道路的领路人、敬爱的曹军威老师致以最衷心的感谢！

感谢同组的陈伟师兄在工程实践中的细致指导。他的科研能力、工作经验以及团队协作能力让我看到了自身的不足，也让我看到了未来前进的方向。他对我整个毕设全程细致的指点让我少走了许多弯路。

感谢同组的万宇鑫师兄的无私帮助，他对于云计算技术的理解、对于实际工程开发的经验多次给我很大的启发，解决了我在项目研究中的很多关键性问题。

感谢同组的陈硕同学的合作与启发，平时与他的交流和讨论往往能对彼此的困惑有所启发，同时也能使思路更加开阔。

回首一学期的科研经历，我深深感受到了自己在科研能力、知识储备方面的巨大成长。再次感谢曹老师、陈伟师兄、万宇鑫师兄，以及许多帮助过我的老师和同学，谢谢你们！



## 声 明

本人郑重声明：所提交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：\_\_\_\_\_ 日 期：\_\_\_\_\_





## 附录 A 书面翻译

### 多服务器优化配置实现云计算利润最大化

#### 摘要

随着云计算的日益流行，了解云计算所产生的经济效益变得十分重要。为了达到利润的最大化，服务供应商需要了解服务费用和经营成本，同时要了解这些开销是如何由网络应用及多服务器系统的配置特征所决定的。我们对如何优化配置多服务器以实现云计算利润最大化的问题进行了研究。我们的价格模型考虑了以下因素：服务量、应用环境负载、多服务器系统的配置、服务等级协议、消费者的满意程度、服务质量、低质量服务的处罚、租赁成本、能耗成本、服务供应商的毛收入及利润等。我们的方法是将多服务器系统看做一个  $M/M/m$  的排队模型，从而可以将优化问题用公式表示并分析解决。我们考虑了两个服务器速度和能耗模型（分别是速度闲置模型和速度恒定模型），并推导出了一个新的服务请求的等待时间的概率密度函数。我们还计算出了服务请求的预期服务费用，并得出了单位时间预期的净收入，求出了最优服务器规模和最优服务器速度的数值结果。

#### 关键词

云计算，多服务器系统，价格模型，利润，排队模型，响应时间，服务器配置，服务费用，服务等级协议，等待时间。

## 1. 引言

云计算正在迅速成为整合计算资源和计算服务的一个可行且高效的方法[9]。通过资源和服务的集中管理，云计算提供互联网上的托管式服务，根据用户需求

为消费者提供共享的硬件、软件、数据库、信息和各类资源的访问业务。在计算资源管理和计算服务供给方面，云计算能够提供经济效益最高和能源使用率最大的方法。云计算通过采用根据使用情况付费的价格模型，将信息技术变成了普通的商品和公用事业[3,5,15]。然而，云计算永远也不会免费[8]，所以了解云计算中的经济效益变得至关重要。

一个有吸引力的云计算环境是由基础设施供应商、服务供应商和消费者组成的三层结构[12]。这三个部分也被称为集群结点、集群管理者和在集群计算系统上的消费者[18]，或资源供应商、服务供应商以及网格计算系统的客户端[16]。基础设施供应商负责维护基本的软硬件设施；服务供应商向基础设施供应商租用资源、搭建合适的多服务器系统，并向用户提供多样的服务；用户向服务供应商提交服务请求，根据特定的服务等级协议从供应商处得到预期的结果，然后根据服务的数量和质量付款。服务供应商可以为不同的应用领域搭建不同的多服务器系统，于是不同性质的服务请求可以发送到不同的系统。每个多服务器系统包含多台服务器，这样一个系统可以专门服务于一种类型的服务请求和应用。多服务器系统的配置有两个基本特征，一个是多服务器系统的规模（服务器的数量），另一个是多服务器系统的运行速度（服务器的执行速度）。

和所有的商业一样，云计算中服务供应商的价格模型基于两个组成元素，也就是收益和成本。对于服务供应商来说，收益（毛收入）是通过向用户提供服务收取的费用，而成本则是付给基础设施供应商的租用成本加上公用事业成本。云计算的价格模型包含多个因素，例如服务量（服务需求）、应用环境负载、多服务器系统的配置（规模及速度）、服务等级协议、用户的满意程度（预期的服务时间）、服务质量（任务等待时间和任务响应时间）、低质量服务的处罚、租赁费用、能耗成本，以及服务供应商的毛收入和利润等，其中利润（也就是净收入）是毛收入与成本的差值。为了使利润最大化，服务供应商需要了解服务费用和经营成本，另外需要特别地了解他们是如何由应用和多服务器系统配置的特点所决定的。

一个服务请求的服务费用由两个因素决定，即预期服务时长和实际服务时长。预期服务时长（预期的服务时间）是一个应用在一台标准服务器上、以基准速度或标准速度运行的执行时间。一旦基准速度确定后，服务的预期时长就取决于服务本身及服务需求（服务量），以执行指令数来衡量。服务的预期时间越长（越短），服务费用越高（越低）。实际服务时长（实际的服务时间）是应用的实际执行时间，其长短依赖于多服务器系统的规模、服务器的速度（可能比基准速度高

些或低些), 以及多服务器系统的工作负荷。我们要注意到实际服务时间是一个随机变量, 在多服务器系统建立以后它将由任务等待时间确定。

在服务等级协议中, 有很多不同的服务性能指标[2]。在本文中我们考虑的性能指标为任务响应时间(或周转时间), 即要完成一个任务所需的时间, 包括任务等待时间和任务执行时间。服务等级协议是完成服务的承诺时间, 等于一个常数与服务预期时间的乘积。如果服务的实际时间长度(或者说服务请求完成的时刻)在服务等级协议的范围内, 该服务将会全额收费。然而如果实际服务时间长度超出服务等级协议, 该服务的收费将会减少。服务的实际时间越长(越短), 服务费用减少将越多(越少)。换句话说, 服务供应商会因违背服务等级协议而受到处罚。如果实际服务时间超过了一定的限度(该限度与服务请求有关), 这次服务将会完全免费。我们注意到一个服务请求的费用是一个随机变量, 我们会关心其期望值。

服务供应商的成本包括两个方面, 即租用成本和公用事业成本。租用费用与多服务器系统的规模即服务器的数量成正比; 公用事业成本实际上就是能耗成本, 由多服务器系统的规模和速度决定。速度越快(越慢), 公用事业成本越高(越低)。要计算能耗成本, 我们需要建立一定的服务器速度和功耗的模型。

为了增加经营收入, 服务供应商可以用许多高速服务器建立和配置多服务器系统。由于实际服务时间(任务响应时间)包括任务等待时间和任务执行时间, 而服务器越多等待时间越短、服务器越快等待时间和执行时间越短。因此, 一个功能强大的多服务器系统可以减少违背服务等级协议的处罚并增加实际收益。然而服务器越多(多服务器系统越大), 给基础设施供应商的设备租赁费用和基础功耗费用就会越大。与此同时, 提高服务器的速度也会增加能耗成本。这些增加的成本可能会抵消掉处罚减少带来的收益。因此, 对于一个特定负载、任务到达率和平均任务执行需求确定的应用环境, 服务供应商需要确定一个最优的多服务器配置(即多服务器系统的规模和速度), 以使得预期利润最大化。

在本文中, 我们对云计算环境中通过优化多服务器配置达到利润最大的问题进行研究。我们的方法是多服务器系统看做一个  $M/M/m$  的排队模型, 这样最优化问题就可以用公式来表达和分析解决。我们考虑两个服务器的速度和功耗模型, 分别是速度闲置模型和速度恒定模型。我们的主要贡献如下。我们推导出了一个新到的服务请求等待时间的概率密度函数, 其结果本身十分重要, 是我们讨论的基础。我们还计算出了一个服务请求的预期服务费用。在这些结果的基础上, 我们得到了单位时间内的预期净收入, 并计算出了服务器规模和速度的最优

化数值。据我们所知，现在还没有类似的文献报告。

一个相关的研究是以用户为中心、以市场为基础、公用事业驱动的资源管理和任务调度，该技术已在集群计算系统[7,17,18]和网格计算系统[4,10,16]中得到应用。通过利用经济机制，例如拍卖等手段来竞标共享的计算资源，用户可以指定任务的价值（效用、收益），即完成任务得到的回报（价格、利润）。效用函数可用来衡量任务的价值和重要性、用户对拖延的忍耐程度及对服务质量的敏感度，该函数为以市场为基础的招标、谈判和准入控制提供支持。服务供应商利用经济手段提供面向服务的实用计算，以利润最大化为目的进行资源分配和任务调度。不同于传统的以系统为中心的性能优化（例如使平均任务响应时间最短），在这类计算经济中主要关心以用户为中心的性能优化，也就是说传递到用户的总效用（用户感受到的总价值）最大化。

本文的其余部分安排如下。在第 2 部分我们将介绍多服务器系统的排队模型，第 3 部分我们提出服务器速度和功耗模型，第 4 部分我们推导出新到达的服务请求等待时间的概率密度函数。在第 5 部分我们定义一个服务费用函数，并计算出一个服务请求的预期费用，第 6 部分我们得到一个单位时间内预期的净收益。在第 7 部分我们将展示如何计算出服务器大小和速度的最优化数值，之后我们将在第 8 部分进行本文的总结。

## 2. 多服务器模型

在本文中，我们将用  $P[e]$  来表示事件  $e$  的发生概率。对于随机变量  $x$ ，我们用  $f_x(t)$  来表示  $x$  的概率密度函数（缩写为 *pdf*），用  $F_x(t)$  来表示  $x$  的分布函数（缩写为 *cdf*），用  $\bar{x}$  来表示  $x$  的数学期望。

云计算的服务供应商利用多服务器系统服务于用户的需求，而该系统由基础设施供应商搭建和维护、由服务供应商租用。多服务器系统的架构细节可以是相当灵活的，例如刀片式服务器和刀片中心的每个服务器都是一个刀片服务器[13]，传统服务器的集群中每个服务器都是一个正常的处理器[7,17,18]，多核服务器处理器中每个服务器都是一个单核[14]。我们将刀片、处理器、核简称为服务器。用户（服务供应商的顾客）向服务供应商提交服务请求（应用和任务），然后服务供应商在多处理器系统上进行服务（运行应用程序和执行任务）。

假设一个多服务器系统  $S$  有  $m$  个相同的处理器，在本文中，一个多服务器系

统被看做一个  $M/M/m$  的排队模型，具体阐述如下。假设一个到达率为  $\lambda$  的服务请求泊松流，即到达时间间隔独立同分布（缩写为 *i.i.d*）于均值为  $1/\lambda$  的指数随机变量。当  $m$  个处理器均处于繁忙状态时，多服务器系统  $S$  可以维护一个无限容量的任务等待队列，并采用先到先得（缩写为 *FCFS*）的排队准则。任务的执行需求量（以待执行的指令数来衡量）独立同分布于均值为  $\bar{r}$  的指数随机变量  $r$ ，且  $S$  系统的  $m$  个服务器（例如刀片、处理器、核）有完全相同的处理速度  $s$ （以单位时间执行的指令数来衡量）。因此在系统  $S$  的处理器上的任务执行时间独立同分布于均值为  $\bar{x} = \bar{r}/s$  的指数随机变量  $x = r/s$ 。

令  $\mu = 1/\bar{x} = s/\bar{r}$  表示平均服务速率，即系统  $S$  中单个服务器一个单位时间内可以完成的服务请求。则服务器利用率为

$$\rho = \frac{\lambda}{m\mu} = \frac{\lambda\bar{x}}{m} = \frac{\lambda}{m} \cdot \frac{\bar{r}}{s}$$

也就是  $S$  中一个服务器繁忙时间的平均百分比。令  $p_k$  表示在系统  $S$  的  $M/M/m$  排队系统中有  $k$  个服务请求（等待或处理中）的概率，那么我们有（[11]，第 102 页）

$$p_k = \begin{cases} p_0 \frac{(m\rho)^k}{k!}, & k \leq m; \\ p_0 \frac{m^m \rho^k}{m!}, & k \geq m; \end{cases}$$

其中

$$p_0 = \left( \sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \frac{(m\rho)^m}{m!} \cdot \frac{1}{1-\rho} \right)^{-1}$$

排队（由于所有服务器繁忙，新提交的服务请求需要等待）的概率为

$$P_q = \sum_{k=m}^{\infty} p_k = \frac{p_m}{1-\rho} = p_0 \frac{(m\rho)^m}{m!} \cdot \frac{1}{1-\rho}$$

系统  $S$  中的平均服务请求（等待或处理中）的数量为

$$\bar{N} = \sum_{k=0}^{\infty} k p_k = m\rho + \frac{\rho}{1-\rho} P_q$$

应用 Little 的结果，我们得到任务的平均响应时间为

$$\bar{T} = \frac{\bar{N}}{\lambda} = \bar{x} + \frac{P_q}{m(1-\rho)} \bar{x} = \bar{x} \left( 1 + \frac{P_q}{m(1-\rho)} \right) = \bar{x} \left( 1 + \frac{P_m}{m(1-\rho)^2} \right)$$

服务请求的平均等待时间为

$$\bar{W} = \bar{T} - \bar{x} = \frac{P_m}{m(1-\rho)^2} \bar{x}$$

等待时间是顾客反感的主要原因。服务供应商应该通过提供足够的服务器和/或提高服务器速度使等待时间保持在一个较低水平。而当等待时间超出一定限制后服务供应商则应愿意对顾客进行赔偿。

### 3. 功耗模型

在数字 CMOS 电路中的功率损耗和电路延迟可以被简单的方程准确地模型化，即使对于复杂的微处理器电路也是如此。CMOS 电路具有动态、静态和短路功耗，然而在一个设计良好的电路中动态功耗  $p$ （电源开关的损耗）是最主要的功耗。动态功耗可近似表示为  $P = aCV^2f$ ，其中  $a$  活性因子， $C$  是负载电容， $V$  是供电电压， $f$  是时钟频率[6]。由于  $s \propto f$ ，其中  $s$  是服务器速度；另外  $f \propto V^\Phi$ ，其中  $0 < \Phi \leq 1$ [19]，这就表示  $V \propto f^{1/\Phi}$ 。我们知道耗电量  $P \propto f^\alpha$  且  $P \propto s^\alpha$ ，其中  $\alpha = 1 + 2/\Phi \geq 3$ 。为简化讨论，我们假设速度为  $s$  的服务器分配的电量  $s^\alpha$ 。

我们将考虑两类服务器速度和功耗模型。在速度闲置模型中，服务器在没有执行任务时速度为零。由于速度为  $s$  时分配的电量  $s^\alpha$ ，所以单个服务器一个单位时间所消耗的平均能量为

$$\rho s^\alpha = \frac{\lambda}{m} r s^{\alpha-1}$$

我们注意到当服务器闲置时公式中的服务器速度为零。一个单位时间内  $m$  个服务器的系统  $S$  的平均耗电量，即多服务器系统  $S$  的电量供给为

$$P = m \rho s^\alpha = \lambda \bar{r} s^{\alpha-1}$$

其中  $m\rho = \lambda\bar{x}$  是  $S$  中处于繁忙状态的服务器平均数量。由于服务器在闲置时仍然会消耗一些能量  $P^*$ （假设一台闲置的服务器消耗一定的基础电量  $P^*$ ，其中包括静态能量损耗、短路能量损耗，以及其它的泄露及浪费的能量[1]），我们将  $P^*$  算在  $P$  中，即

$$P = m(\rho s^\alpha + P^*) = \lambda\bar{r}s^{\alpha-1} + mP^*$$

注意到当  $P^* = 0$  时，上述的  $P$  与  $m$  无关。

在速度恒定模型中，所有的服务器都会以速度  $s$  运行，即使当前没有执行任何任务。我们再次使用  $P$  来代表分配到多服务器系统  $S$  上的电量。由于运行速度为  $s$ ，分配电量为  $s^\alpha$ ，所以分配到多服务器系统  $S$  上的电量为  $P = m(s^\alpha + P^*)$ 。

## 4. 等待时间分布

令  $W$  表示一个新的服务请求到达多服务器系统后的等待时间。在本节我们将得到  $W$  的概率密度函数  $f_w(t)$ 。为此，我们根据新服务请求到达时排队系统中任务数量的不同，来对  $W$  进行分情况讨论。令  $W_k$  表示当一个新的任务到达  $M/M/m$  排队系统后的排队时间，假设此前排队系统中共有  $k$  个任务。

我们定义单位冲激函数  $u_z(t)$  如下：

$$u_z(t) = \begin{cases} z, & 0 \leq t \leq \frac{1}{z}; \\ 0, & t > \frac{1}{z}; \end{cases}$$

函数  $u_z(t)$  有如下特性

$$\int_0^\infty u_z(t) dt = 1$$

例如，当  $u_z(t)$  是一个随机变量的概率密度函数时，其数学期望为

$$\int_0^\infty t u_z(t) dt = z \int_0^{1/z} t dt = \frac{1}{2z}$$

令  $z \rightarrow \infty$  并定义

$$u(t) = \lim_{z \rightarrow \infty} u_z(t)$$

显然任何概率密度函数为  $u(t)$  的随机变量的数学期望均为 0。

下面的定理给出了新到达服务请求的等待时间的概率密度函数。

**定理 1** 新到达服务请求的等待时间  $W$  的概率密度函数为

$$f_W(t) = (1 - P_q)u(t) + m\mu p_m e^{-(1-\rho)m\mu t},$$

其中  $P_q = p_m / (1 - \rho)$  且  $p_m = p_0(m\rho)^m / m!$ 。

**证明：** 如果新的服务请求到达时，队列系统中有  $k < m$  个任务，那么该服务请求的等待时间  $W_k = 0$ 。 $W_k$  的概率密度函数可表示为

$$f_{W_k}(t) = u(t)$$

对于所有的  $0 \leq k \leq m-1$  成立。进一步，我们得到

$$\overline{W_k} = \lim_{z \rightarrow \infty} \frac{1}{2z} = 0$$

对于所有  $0 \leq k \leq m-1$  成立。

如果当新的服务请求到达时，队列系统中共有  $k \geq m$  个任务，则该服务请求必须等待一段时间直到一台服务器可用为止。注意到由于指数分布的无记忆性，所以任务执行的剩余时间总是一个与之前相同的随机变量，也就是最初的任务执行时间  $x$ ，其概率密度函数为

$$f_x(t) = \mu e^{-\mu t}$$

不论这个任务已经运行了多长时间，令  $x_1, x_2, \dots, x_m$  分别为新任务到达时  $m$  个任务的剩余时间，那么我们有  $f_{x_j}(t) = \mu e^{-\mu t}$ ，对于所有  $1 \leq j \leq m$  均成立。

显然  $y = \min\{x_1, x_2, \dots, x_m\}$  是从现在到下一个任务完成的时间间隔。由于

$$P[y \geq t] = \prod_{j=1}^m P[x_j \geq t] = \prod_{j=1}^m e^{-\mu t} = e^{-m\mu t}$$

我们得到

$$F_y(t) = P[y \leq t] = 1 - P[y \geq t] = 1 - e^{-m\mu t}$$

以及

$$f_y(t) = m\mu e^{-m\mu t}$$



与此同时， $y$  也是一个指数分布的随机变量，其均值为  $1/m\mu$ 。从现在到下一个任务完成的时间间隔总是同样的随机变量  $y$ ，即  $m$  的最小值是独立同指数分布的随机变量，其概率密度函数为  $f_y(t) = m\mu e^{-m\mu t}$ 。

我们注意到由于有多台服务器，任务  $t$  不必等到所有之前的任务都完成后才开始。实际上，任务  $t$  的等待时间  $W_k$ （任务  $t$  到达时排队系统中已有  $k \geq m$  个任务）可表示为  $W_k = y_1 + y_2 + \dots + y_{k-m+1}$ ，其中  $y_1, y_2, \dots, y_{k-m+1}$  独立同指数分布且概率密度函数为  $f_y(t) = m\mu e^{-m\mu t}$ 。该结论的原因是在  $k-m+1$  个任务完成以后，任务  $t$  已经在等待队伍的队首且有一个服务器可用，之后该任务就会被调度执行。众所周知  $y_1 + y_2 + \dots + y_{k-m+1}$  服从 Erlang 分布，其概率密度函数为

$$\frac{m\mu(m\mu t)^{k-1}}{(k-1)!} e^{-m\mu t}$$

因此，我们得到  $W_k$  的概率密度函数为

$$f_{W_k}(t) = \frac{m\mu(m\mu t)^{k-m}}{(k-m)!} e^{-m\mu t}$$

对于所有的  $k \geq m$  均成立。我们注意到  $\bar{y} = 1/m\mu$ ，于是

$$\bar{W}_k = (k-m+1)\bar{y} = \frac{k-m+1}{m\mu} = (k-m+1)\frac{\bar{x}}{m}$$

综上所述，我们得到一个请求服务的等待时间  $W$  的概率密度函数如下：

$$\begin{aligned} f_W(t) &= \sum_{k=0}^{\infty} p_k f_{W_k}(t) \\ &= \left( \sum_{k=0}^{m-1} p_k \right) u(t) + \sum_{k=m}^{\infty} p_k \frac{m\mu(m\mu t)^{k-m}}{(k-m)!} e^{-m\mu t} \\ &= (1-P_q)u(t) + \sum_{k=m}^{\infty} p_0 \frac{m^m \rho^k}{m!} \cdot \frac{m\mu(m\mu t)^{k-m}}{(k-m)!} e^{-m\mu t} \\ &= (1-P_q)u(t) + p_0 \frac{m^m \rho^m}{m!} m\mu e^{-m\mu t} \sum_{k=m}^{\infty} \frac{\rho^{k-m} (m\mu t)^{k-m}}{(k-m)!} \\ &= (1-P_q)u(t) + p_0 \frac{(m\rho)^m}{m!} m\mu e^{-m\mu t} \sum_{k=0}^{\infty} \frac{(\rho m\mu t)^k}{k!} \\ &= (1-P_q)u(t) + p_m m\mu e^{-m\mu t} e^{\rho m\mu t} \\ &= (1-P_q)u(t) + m\mu p_m e^{-(1-\rho)m\mu t} \end{aligned}$$

至此定理证明完毕。

在此基础上，我们得到预期等待时间

$$\begin{aligned}
\bar{W} &= \sum_{k=m}^{\infty} p_k \bar{W}_k \\
&= \sum_{k=m}^{\infty} p_k (k-m+1) \frac{\bar{x}}{m} \\
&= \frac{\bar{x}}{m} \left( \sum_{k=m}^{\infty} k p_k - (m-1) \sum_{k=m}^{\infty} p_k \right) \\
&= \frac{\bar{x}}{m} \left( p_0 \frac{m^m}{m!} \sum_{k=m}^{\infty} k \rho^k - (m-1) P_q \right) \\
&= \frac{\bar{x}}{m} \left( p_0 \frac{m^m}{m!} \cdot \frac{\rho^m}{1-\rho} \left( m + \frac{\rho}{1-\rho} \right) - (m-1) P_q \right) \\
&= \frac{\bar{x}}{m} \left( \frac{P_m}{1-\rho} \left( m + \frac{\rho}{1-\rho} \right) - (m-1) P_q \right) \\
&= \frac{\bar{x}}{m} \left( P_q \left( m + \frac{\rho}{1-\rho} \right) - (m-1) P_q \right) \\
&= \frac{\bar{x}}{m} P_q \left( 1 + \frac{\rho}{1-\rho} \right) = \frac{P_q}{1-\rho} \cdot \frac{\bar{x}}{m} = \bar{T} - \bar{x}
\end{aligned}$$

上面的结果与我们之前在第 2 部分对  $\bar{W}$  的讨论是一致的。

## 5. 服务费用

如果所有的服务器都有一个固定的速度  $s$ ，那么一个执行需求量为  $r$  的服务请求的执行时间就是  $x = r/s$ ，该服务请求的响应时间为

$$T = W + x = W + \frac{r}{s}$$

(注：上述方程说明  $T$  的概率密度函数是  $W$  和  $x$  的概率密度函数的简单卷积。然而为了避免复杂的数学计算，我们不从这个角度进行深究。事实上我们只会用到  $W$  的概率密度函数。) 在云计算中，响应时间  $T$  与服务供应商向用户收取的服务费用有关。

令  $s_0$  表示一个服务器的基准速度。当一个服务请求的执行需求量为  $r$ 、响应时间为  $T$  时，我们定义服务费用函数为

$$C(r, T) = \begin{cases} ar, & 0 \leq T \leq \frac{c}{s_0} r; \\ ar - d \left( T - \frac{c}{s_0} r \right), & \frac{c}{s_0} r < T \leq \left( \frac{a}{d} + \frac{c}{s_0} \right) r; \\ 0, & T > \left( \frac{a}{d} + \frac{c}{s_0} \right) r. \end{cases}$$

以上函数的定义有以下合理性。

- 如果处理一个服务请求的响应时间  $T$  不长于  $(c/s_0)r = c(r/s_0)$ （当该任务以速度  $s_0$  执行时，常数  $c$  与任务执行时间的乘积），那么服务供应商就认为该请求成功地以高质量的服务完成，并向用户收取与执行需求量  $r$ （服务量）成正比的费用  $ar$ 。常数  $c$  是表征服务等级协议的参数。
- 如果处理服务请求的响应时间  $T$  长于  $(c/s_0)r$  但是短于  $(a/d + c/s_0)r$ ，那么服务供应商就会认为该请求以低质量的服务完成，向顾客收取的费用应该随着  $T$  的增加而线性减少。参数  $d$  表征违背服务等级协议后受到的处罚程度。
- 如果处理服务请求的响应时间  $T$  长于  $(a/d + c/s_0)r$ ，那么服务供应商就会认为该请求等待时间过长，则该服务不会收取任何费用。

我们注意到任务响应时间  $T$  是与服务器的速度为  $s_0$ （基准速度或参考速度）时任务的执行时间进行对比。而实际的服务器运行速度  $s$  可能低于或高于  $s_0$ ，具体将由服务供应商根据工作负荷（ $\lambda$  和  $\bar{r}$ ）、系统参数（例如  $m$ 、 $\alpha$  和  $P^*$ ）以及服务费用函数（ $a$ 、 $c$  和  $d$ ）来决定。于是下面将根据这种情况定义最大化的净利润。

在前文关于任务等待时间的结果上进一步讨论，我们注意到服务费用函数可以用参数  $r$  和  $W$  进行等价描述为

$$C(r, W) = \begin{cases} ar, & 0 \leq W \leq \left( \frac{c}{s_0} - \frac{1}{s} \right) r; \\ \left( a + \frac{cd}{s_0} - \frac{d}{s} \right) r - dW, & \left( \frac{c}{s_0} - \frac{1}{s} \right) r < W \leq \left( \frac{a}{d} + \frac{c}{s_0} - \frac{1}{s} \right) r; \\ 0, & W > \left( \frac{a}{d} + \frac{c}{s_0} - \frac{1}{s} \right) r. \end{cases}$$

接下来的定理将给出一个服务请求的预期收费。

**定理 2** 一个服务请求的预期收费为

$$C = ar \left[ 1 - \frac{P_q}{\left( (ms - \lambda \bar{r})(c/s_0 - 1/s) + 1 \right) \left( (ms - \lambda \bar{r})(a/d + c/s_0 - 1/s) + 1 \right)} \right]$$

其中  $P_q = p_m / (1 - \rho)$ ,  $p_m = p_0(m\rho)^m / m!$ 。

**证明:** 由于  $W$  是一个随机变量, 而当  $r$  固定时  $C(r, W)$  可以看做是  $W$  的函数, 也是一个随机变量。于是当执行需求量为  $r$  时, 服务请求的预期收费为

$$\begin{aligned} C(r) &= \overline{C(r, W)} \\ &= \int_0^\infty f_W(t) C(r, t) dt \\ &= \int_0^{(a/d + c/s_0 - 1/s)r} f_W(t) C(r, t) dt \\ &= \int_0^{(a/d + c/s_0 - 1/s)r} \left( (1 - P_q)u(t) + m\mu p_m e^{-(1-\rho)m\mu t} \right) C(r, t) dt \\ &= \int_0^{(a/d + c/s_0 - 1/s)r} (1 - P_q)u(t) C(r, t) dt \\ &\quad + \int_0^{(a/d + c/s_0 - 1/s)r} m\mu p_m e^{-(1-\rho)m\mu t} C(r, t) dt \\ &\quad + \int_{(c/s_0 - 1/s)r}^{(a/d + c/s_0 - 1/s)r} m\mu p_m e^{-(1-\rho)m\mu t} C(r, t) dt \\ &= (1 - P_q)ar + \int_0^{(c/s_0 - 1/s)r} m\mu p_m e^{-(1-\rho)m\mu t} C(r, t) dt \\ &= (1 - P_q)ar + \int_0^{(c/s_0 - 1/s)r} m\mu p_m e^{-(1-\rho)m\mu t} ar dt \\ &= (1 - P_q)ar + m\mu p_m ar \int_0^{(c/s_0 - 1/s)r} e^{-(1-\rho)m\mu t} dt \\ &\quad + \int_{(c/s_0 - 1/s)r}^{(a/d + c/s_0 - 1/s)r} m\mu p_m e^{-(1-\rho)m\mu t} \left( \left( a + \frac{cd}{s_0} - \frac{d}{s} \right) r - dt \right) dt \\ &\quad + m\mu p_m \int_{(c/s_0 - 1/s)r}^{(a/d + c/s_0 - 1/s)r} e^{-(1-\rho)m\mu t} \left( \left( a + \frac{cd}{s_0} - \frac{d}{s} \right) r - dt \right) dt \\ &= (1 - P_q)ar + m\mu p_m ar \int_0^{(c/s_0 - 1/s)r} e^{-(1-\rho)m\mu t} dt \\ &\quad + m\mu p_m \left( a + \frac{cd}{s_0} - \frac{d}{s} \right) r \int_{(c/s_0 - 1/s)r}^{(a/d + c/s_0 - 1/s)r} e^{-(1-\rho)m\mu t} dt \\ &\quad - dm\mu p_m \int_{(c/s_0 - 1/s)r}^{(a/d + c/s_0 - 1/s)r} t e^{-(1-\rho)m\mu t} dt \end{aligned}$$

继续计算，我们注意到

$$\int e^{bt} dt = \frac{e^{bt}}{b},$$

以及

$$\int te^{bt} = \frac{1}{b} \left( t - \frac{1}{b} \right) e^{bt}$$

因此，我们有

$$\begin{aligned} \int_0^{(c/s_0-1/s)r} e^{-(1-\rho)m\mu t} dt &= -\frac{e^{-(1-\rho)m\mu t}}{(1-\rho)m\mu} \Big|_0^{(c/s_0-1/s)r} \\ &= \frac{1 - e^{-(1-\rho)m\mu(c/s_0-1/s)r}}{(1-\rho)m\mu} \end{aligned}$$

以及

$$\begin{aligned} \int_{(c/s_0-1/s)r}^{(a/d+c/s_0-1/s)r} e^{-(1-\rho)m\mu t} dt &= -\frac{e^{-(1-\rho)m\mu t}}{(1-\rho)m\mu} \Big|_{(c/s_0-1/s)r}^{(a/d+c/s_0-1/s)r} \\ &= \frac{e^{-(1-\rho)m\mu(c/s_0-1/s)r} - e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r}}{(1-\rho)m\mu} \end{aligned}$$

同时有

$$\begin{aligned} &\int_{(c/s_0-1/s)r}^{(a/d+c/s_0-1/s)r} te^{-(1-\rho)m\mu t} dt \\ &= -\frac{1}{(1-\rho)m\mu} \left( t + \frac{1}{(1-\rho)m\mu} \right) e^{-(1-\rho)m\mu t} \Big|_{(c/s_0-1/s)r}^{(a/d+c/s_0-1/s)r} \\ &= \frac{1}{(1-\rho)m\mu} \left( \left( \frac{c}{s_0} - \frac{1}{s} \right) r + \frac{1}{(1-\rho)m\mu} \right) e^{-(1-\rho)m\mu(c/s_0-1/s)r} \\ &\quad - \frac{1}{(1-\rho)m\mu} \left( \left( \frac{a}{d} + \frac{c}{s_0} - \frac{1}{s} \right) r + \frac{1}{(1-\rho)m\mu} \right) e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r} \end{aligned}$$

基于以上的结果，我们得到

$$\begin{aligned} &C(r) \\ &= (1 - P_q)ar + m\mu p_m ar \frac{1 - e^{-(1-\rho)m\mu(c/s_0-1/s)r}}{(1-\rho)m\mu} \end{aligned}$$

$$\begin{aligned}
& +m\mu p_m \left( a + \frac{cd}{s_0} - \frac{d}{s} \right) r \frac{e^{-(1-\rho)m\mu(c/s_0-1/s)r} - e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r}}{(1-\rho)m\mu} \\
& -dm\mu p_m \frac{1}{(1-\rho)m\mu} \left( \left( \frac{c}{s_0} - \frac{1}{s} \right) r + \frac{1}{(1-\rho)m\mu} \right) e^{-(1-\rho)m\mu(c/s_0-1/s)r} \\
& +dm\mu p_m \frac{1}{(1-\rho)m\mu} \left( \left( \frac{a}{d} + \frac{c}{s_0} - \frac{1}{s} \right) r + \frac{1}{(1-\rho)m\mu} \right) e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r} \\
= & (1-P_q)ar + \frac{ap_m}{1-\rho} (r - re^{-(1-\rho)m\mu(c/s_0-1/s)r}) \\
& + \frac{P_m}{1-\rho} \left( a + \frac{cd}{s_0} - \frac{d}{s} \right) \left( re^{-(1-\rho)m\mu(c/s_0-1/s)r} - re^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r} \right) \\
& - \frac{dp_m}{1-\rho} \left( \left( \frac{c}{s_0} - \frac{1}{s} \right) re^{-(1-\rho)m\mu(c/s_0-1/s)r} + \frac{1}{(1-\rho)m\mu} e^{-(1-\rho)m\mu(c/s_0-1/s)r} \right) \\
& + \frac{dp_m}{1-\rho} \left( \left( \frac{a}{d} + \frac{c}{s_0} - \frac{1}{s} \right) re^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r} - \frac{1}{(1-\rho)m\mu} e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r} \right) \\
= & (1-P_q)ar + aP_q (r - re^{-(1-\rho)m\mu(c/s_0-1/s)r}) \\
& + P_q \left( a + \frac{cd}{s_0} - \frac{d}{s} \right) \left( re^{-(1-\rho)m\mu(c/s_0-1/s)r} - re^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r} \right) \\
& - dP_q \left( \left( \frac{c}{s_0} - \frac{1}{s} \right) re^{-(1-\rho)m\mu(c/s_0-1/s)r} + \frac{1}{(1-\rho)m\mu} e^{-(1-\rho)m\mu(c/s_0-1/s)r} \right) \\
& + dP_q \left( \left( \frac{a}{d} + \frac{c}{s_0} - \frac{1}{s} \right) re^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r} - \frac{1}{(1-\rho)m\mu} e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r} \right) \\
= & ar - \frac{dP_q}{(1-\rho)m\mu} \left( e^{-(1-\rho)m\mu(c/s_0-1/s)r} - e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r} \right)
\end{aligned}$$

由于  $r$  是一个随机变量，而  $C(r)$  是  $r$  的函数，也是一个随机变量。令任务执行需求量  $r$  的概率密度函数为

$$f_r(z) = \frac{1}{r} e^{-z/\bar{r}}$$

则对于一个服务请求的预期收费为

$$\begin{aligned}
& C \\
&= \overline{C(r)} \\
&= \int_0^\infty f_r(z)C(z)dz \\
&= \int_0^\infty \frac{1}{r} e^{-z/\bar{r}} C(z)dz \\
&= \frac{1}{r} \int_0^\infty e^{-z/\bar{r}} \left( az - \frac{dP_q}{(1-\rho)m\mu} \left( e^{-(1-\rho)m\mu(c/s_0-1/s)z} - e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)z} \right) \right) dz \\
&= \frac{a}{r} \int_0^\infty z e^{-z/\bar{r}} dz \\
&\quad - \frac{dP_q}{r(1-\rho)m\mu} \left( \int_0^\infty e^{-(1-\rho)m\mu(c/s_0-1/s)+1/\bar{r}} z dz - \int_0^\infty e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)+1/\bar{r}} z dz \right)
\end{aligned}$$

由于

$$\int_0^\infty z e^{-bz} dz = -\frac{1}{b} \left( z + \frac{1}{b} \right) e^{-bz} \Big|_0^\infty = \frac{1}{b^2}$$

以及

$$\int_0^\infty e^{-bz} dz = -\frac{e^{-bz}}{b} \Big|_0^\infty = \frac{1}{b}$$

我们得到

$$\begin{aligned}
C &= \frac{1}{r} \left( ar^{-2} - \frac{dP_q}{(1-\rho)m\mu} \left( \frac{1}{(1-\rho)m\mu(c/s_0-1/s)+1/\bar{r}} - \frac{1}{(1-\rho)m\mu(a/d+c/s_0-1/s)+1/\bar{r}} \right) \right) \\
&= ar^{-2} - \frac{dP_q}{(1-\rho)m\mu} \left( \frac{1}{\bar{r}(1-\rho)m\mu(c/s_0-1/s)+1} - \frac{1}{\bar{r}(1-\rho)m\mu(a/d+c/s_0-1/s)+1} \right) \\
&= ar^{-2} - \frac{dP_q}{(1-\rho)m\mu} \cdot \frac{\bar{r}(1-\rho)m\mu(a/d)}{(\bar{r}(1-\rho)m\mu(c/s_0-1/s)+1)(\bar{r}(1-\rho)m\mu(a/d+c/s_0-1/s)+1)} \\
&= ar^{-2} - \frac{\bar{a}rP_q}{(\bar{r}(1-\rho)m\mu(c/s_0-1/s)+1)(\bar{r}(1-\rho)m\mu(a/d+c/s_0-1/s)+1)} \\
&= ar^{-2} \left( 1 - \frac{P_q}{((ms-\lambda\bar{r})(c/s_0-1/s)+1)((ms-\lambda\bar{r})(a/d+c/s_0-1/s)+1)} \right)
\end{aligned}$$

故此定理得到证明。

在图 1 中，我们考虑一个执行需求量为  $r$  的服务请求的预期费用，即

$$C(r) = ar - \frac{dP_q}{(1-\rho)m\mu} \left( e^{-(1-\rho)m\mu(c/s_0-1/s)r} - e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r} \right)$$

我们假设  $\bar{r} = 10$  亿条指令， $m = 7$  台服务器， $s_0 = 10$  亿条指令/秒， $s = 10$  亿条指令/秒， $a = 10$  分/十亿条指令（注：本文中的货币单位“分”可能和美元不完全相同，但是会与之成线性比例）， $c = 3$  分/秒， $d = 1$  分/秒。对于  $\lambda = 6.15, 6.35, 6.55, 6.75, 6.95$  个服务请求/秒，我们展示了对于  $0 \leq r \leq 3$ （单位为十亿）的  $C(r)$ 。可以看出服务费用函数对于  $\lambda$  是递减的，这是因为等待时间和迟到处罚随着  $\lambda$  的增加而增加。同时可以看出服务费用相对  $r$  是递增函数，也就是说大的服务请求量会比小的请求量带来更多的利润。

在图 2 中，我们用于图 1 相同的参数进一步展示了  $C(r)/ar$ 。由于对于执行需求量为  $r$  的服务请求来说， $ar$  是理想的（最大的）服务费用，所以  $C(r)/ar$  可看做归一化的服务费用。对于  $\lambda = 6.15, 6.35, 6.55, 6.75, 6.95$  个服务请求/秒，我们展示了对于  $0 \leq r \leq 3$  的  $C(r)/ar$ 。可以看出归一化的服务费用是  $\lambda$  的减函数，因为等待时间和迟到处罚随着  $\lambda$  的增加而增加。同时可以看出归一化服务费用相对  $r$  是递增函数，因为随着服务需求量  $r$  的增加，由等待时间带来的服务费损失的百分比会减少。换句话说，对于大的服务需求量更有可能获利，而对于小的服务需求量则更可能无法收费。可以证明当  $r$  趋近于 0 时，归一化服务费用为

$$\lim_{r \rightarrow 0} \frac{C(r)}{ar} = 1 - P_q$$

其中  $P_q$  随着  $\lambda$  的增加而增加（ $1 - P_q$  会减少）。同时可以证明当  $r$  趋近于无穷时，归一化服务费用为

$$\lim_{r \rightarrow \infty} \frac{C(r)}{ar} = 1$$

对于所有的  $\lambda$  均成立。

## 6. 净经营收益

由于在稳定的  $M/M/m$  队列系统中，每个单位时间处理的服务请求数量为  $\lambda$ ，所以每个单位时间的预期服务费用为  $\lambda C$ ，其实也就是服务供应商的预期毛收入。假设单位时间一个服务器的租赁费用为  $\beta$ ，同时假设每瓦特的耗电成本为  $\gamma$ 。则



服务供应商的成本就是基础设施租用费和耗电费之和，也就是  $\beta m + \gamma P$ 。那么服务供应商单位时间的预期净经营收益（净利润）为

$$G = \lambda C - (\beta m + \gamma P)$$

也就是毛收入减去成本。对于速度闲置模型，上述方程转化为

$$G = \lambda C - \left( \beta m + \gamma \left( \lambda \bar{r} s^{\alpha-1} + m P^* \right) \right)$$

对于速度恒定模型，方程为

$$G = \lambda C - \left( \beta m + \gamma m \left( s^\alpha + P^* \right) \right)$$

在图 3 和图 4 中，我们应用图 1 和图 2 相同的数据，分别就两种能耗模型展示了单位时间毛收入  $\lambda C$  和净收入  $G$  随  $\lambda$  变化的函数。此外，我们假设  $P^* = 2$  瓦特， $\alpha = 3$  分/秒， $\beta = 2$  分/秒， $\gamma = 0.5$  分/瓦特。对于  $0 \leq \lambda \leq 7$ ，我们展示了  $\lambda C$  和  $G$  的变化。基础设施租用费为  $\beta m = 14$  分/秒，速度闲置模型能耗费用为  $0.5\lambda + 7$  分/秒，恒定速度模型能耗费用为  $10.5$  分/秒。我们观察到  $\lambda C$  和  $G$  都随着  $\lambda$  近似线性增长，而在某一点后急速下降。换句话说，服务请求越多，就会带来越多的毛收入和净利润；然而当单位时间服务请求达到一定数量时，过多的等待时间会增加迟到处罚，则毛收入为零时净利润会成为负值。

有两种情况会导致经营收入为负值。第一种情况是没有足够的业务（服务请求）。在这种情况下，服务供应商应该考虑减少服务器数量  $m$  和/或服务器速度  $s$ ，这样基础设施租用费和耗电费用都会减少。第二种情况是过多的业务（服务请求）。在这种情况下，服务供应商应该考虑增加服务器数量和/或服务器速度，以减少等待时间、增加经营收入。然而增加服务器数量和/或服务器速度同样也会增加基础设施租用和耗电的费用。因此我们面临着如何优化服务器规模和/或服务器速度使得利润最大化。

## 7. 利润最大化

为了用公式表达并分析解决我们的最优化问题，我们需要  $C$  的解析表达式。为此，我们要用到以下解析近似公式，

$$\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} \approx e^{m\rho}$$

当  $m$  不是很小且  $\rho$  不是很大时，上式有很高的准确度[14]。我们还需要用到  $m!$  的 Stirling 近似，也就是

$$m! \approx \sqrt{2\pi m} \left(\frac{m}{e}\right)^m$$

因此，我们得到以下  $p_0$  的解析近似表达式，

$$p_0 \approx \left( e^{m\rho} + \frac{(e\rho)^m}{\sqrt{2\pi m}} \cdot \frac{1}{1-\rho} \right)^{-1}$$

以及下面  $p_m$  的解析近似表达式，

$$p_m \approx \frac{\frac{(e\rho)^m}{\sqrt{2\pi m}}}{e^{m\rho} + \frac{(e\rho)^m}{\sqrt{2\pi m}} \cdot \frac{1}{1-\rho}}$$

即

$$p_m \approx \frac{1-\rho}{\sqrt{2\pi m}(1-\rho)(e^\rho / e\rho)^m + 1}$$

同时  $P_q$  的解析近似表达式如下

$$P_q \approx \frac{1}{\sqrt{2\pi m}(1-\rho)(e^\rho / e\rho)^m + 1}$$

通过应用以上  $P_q$  的解析表达式，我们得到一个服务请求的预期服务费用的解析近似为

$$C \approx a\bar{r} - \frac{a\bar{r}}{\left(\sqrt{2\pi m}(1-\rho)(e^\rho / e\rho)^m + 1\right)} \times \frac{1}{\left(\left(ms - \lambda\bar{r}\right)(c/s_0 - 1/s) + 1\right)\left(\left(ms - \lambda\bar{r}\right)(a/d + c/s_0 - 1/s) + 1\right)}$$

为了方便，我们将  $C$  改记为

$$C = a\bar{r} \left( 1 - \frac{1}{D_1 D_2 D_3} \right)$$

其中

$$\begin{aligned} D_1 &= \sqrt{2\pi m}(1-\rho)(e^\rho / e\rho)^m + 1 \\ D_2 &= (ms - \lambda\bar{r})(c/s_0 - 1/s) + 1 \\ D_3 &= (ms - \lambda\bar{r})(a/d + c/s_0 - 1/s) + 1 \end{aligned}$$

我们本节的讨论正是基于上面得到的  $C$  的解析表达式。

## 7.1 最优规模

当  $\lambda$ 、 $\bar{r}$ 、 $s$ 、 $P^*$ 、 $\alpha$ 、 $\beta$ 、 $\gamma$ 、 $a$ 、 $c$  和  $d$  都确定了以后，我们第一个问题就是找到  $m$  使  $G$  最大。为了使  $G$  达到最大值，我们需要在速度闲置模型下找到  $m$  满足

$$\frac{\partial G}{\partial m} = \lambda \frac{\partial C}{\partial m} - (\beta + \gamma P^*) = 0$$

在速度恒定模型下找到  $m$  满足

$$\frac{\partial G}{\partial m} = \lambda \frac{\partial C}{\partial m} - (\beta + \gamma(s^\alpha + P^*)) = 0$$

其中

$$\frac{\partial C}{\partial m} = \frac{a\bar{r}}{(D_1 D_2 D_3)^2} \left( D_2 D_3 \frac{\partial D_1}{\partial m} + D_1 D_3 \frac{\partial D_2}{\partial m} + D_1 D_2 \frac{\partial D_3}{\partial m} \right)$$

为了继续计算，我们将  $D_1$  改写为

$$D_1 = \sqrt{2\pi m}(1-\rho)R + 1$$

其中

$$R = (e^\rho / e\rho)^m$$

我们注意到

$$\ln R = m \ln(e^\rho / e\rho) = m(\rho - \ln \rho - 1)$$

因为

$$\frac{\partial \rho}{\partial m} = -\frac{\lambda \bar{r}}{m^2 s} = -\frac{\rho}{m}$$

我们得到

$$\frac{1}{R} \frac{\partial R}{\partial m} = (\rho - \ln \rho - 1) + m \left( 1 - \frac{1}{\rho} \right) \frac{\partial \rho}{\partial m} = -\ln \rho$$

同时

$$\frac{\partial R}{\partial m} = -R \ln \rho$$

现在，我们有

$$\begin{aligned} \frac{\partial D_1}{\partial m} &= \sqrt{2\pi} \left( \frac{1}{2\sqrt{m}} (1-\rho) R + \sqrt{m} \left( -\frac{\partial \rho}{\partial m} \right) R + \sqrt{m} (1-\rho) \frac{\partial R}{\partial m} \right) \\ &= \sqrt{2\pi} \left( \frac{1}{2\sqrt{m}} (1-\rho) R + \sqrt{m} \frac{\rho}{m} R - \sqrt{m} (1-\rho) R \ln \rho \right) \\ &= \sqrt{2\pi} \left( \frac{1}{2\sqrt{m}} (1-\rho) R + \frac{1}{\sqrt{m}} \rho R - \sqrt{m} (1-\rho) (\ln \rho) R \right) \\ &= \sqrt{2\pi} \left( \frac{1}{2\sqrt{m}} (1+\rho) R - \sqrt{m} (1-\rho) (\ln \rho) R \right) \end{aligned}$$

此外，我们还得到了

$$\frac{\partial D_2}{\partial m} = cs / s_0 - 1$$

以及

$$\frac{\partial D_3}{\partial m} = as / d + cs / s_0 - 1$$

在图 5 和图 6 中，我们应用图 1 到图 4 的参数，分别就两种能耗模型展示了单位时间内净收入  $G$  随  $m$  和  $\lambda$  的函数变化。对于  $\lambda = 2.9, 3.9, 4.9, 5.9, 6.9$ ，我们展示了当  $m$  足够大使得  $\rho < 1$  时  $G$  的变化。我们注意到  $m$  存在一个使  $G$  取到最大值的最优选择。应用我们的分析结果，我们可以找到满足  $\partial G / \partial m = 0$  的  $m$ 。对于速度闲置模型， $m$  的最优值为 3.51509, 4.61045, 5.69288, 6.76612, 7.8323；对于速度恒定模型， $m$  的最优值为 3.47641, 4.56648, 5.64430, 6.71340, 7.77584。

## 7.2 最优速度

当  $\lambda$ 、 $\bar{r}$ 、 $s$ 、 $P^*$ 、 $\alpha$ 、 $\beta$ 、 $\gamma$ 、 $a$ 、 $c$  和  $d$  都确定了以后，我们的第二个问题是找到使  $G$  最大的速度  $s$ 。为了使  $G$  取到最大值，在速度闲置模型中  $s$  应满足

$$\frac{\partial G}{\partial s} = \lambda \frac{\partial C}{\partial s} - \gamma \lambda \bar{r} (\alpha - 1) s^{\alpha-2} = 0$$

在速度恒定模型中  $s$  应满足

$$\frac{\partial G}{\partial s} = \lambda \frac{\partial C}{\partial s} - \gamma m \alpha s^{\alpha-1} = 0$$

其中

$$\frac{\partial C}{\partial s} = \frac{a \bar{r}}{(D_1 D_2 D_3)^2} \left( D_2 D_3 \frac{\partial D_1}{\partial s} + D_1 D_3 \frac{\partial D_2}{\partial s} + D_1 D_2 \frac{\partial D_3}{\partial s} \right)$$

与上个小节的计算过程类似，我们有

$$\frac{\partial \rho}{\partial s} = -\frac{\lambda \bar{r}}{m s^s} = -\frac{\rho}{s}$$

以及

$$\frac{1}{R} \frac{\partial R}{\partial s} = m \left( 1 - \frac{1}{\rho} \right) \frac{\partial \rho}{\partial s} = \frac{m}{s} (1 - \rho)$$

还有

$$\frac{\partial R}{\partial s} = \frac{m}{s} (1 - \rho) R$$

现在，我们得到了

$$\begin{aligned} \frac{\partial D_1}{\partial s} &= \sqrt{2\pi m} \left( \left( -\frac{\partial \rho}{\partial s} \right) R + (1 - \rho) \frac{\partial R}{\partial s} \right) \\ &= \sqrt{2\pi m} \left( \frac{\rho}{s} R + \frac{m}{s} (1 - \rho)^2 R \right) \\ &= \sqrt{2\pi m} \left( \rho + m(1 - \rho)^2 \right) \frac{R}{s} \end{aligned}$$

此外，我们还有

$$\begin{aligned}\frac{\partial D_2}{\partial s} &= m\left(\frac{c}{s_0} - \frac{1}{s}\right) + (ms - \lambda\bar{r})\left(-\frac{1}{s^2}\right) \\ &= \frac{mc}{s_0} - \frac{2m}{s} + \frac{\lambda\bar{r}}{s^2}\end{aligned}$$

以及

$$\begin{aligned}\frac{\partial D_3}{\partial s} &= m\left(\frac{a}{d} + \frac{c}{s_0} - \frac{1}{s}\right) + (ms - \lambda\bar{r})\left(-\frac{1}{s^2}\right) \\ &= m\left(\frac{a}{d} + \frac{c}{s_0}\right) - \frac{2m}{s} + \frac{\lambda\bar{r}}{s^2}\end{aligned}$$

在图 7 和图 8 中，我们应用图 1 到 6 的参数，分别就两种能耗模型展示了单位时间内净收入  $G$  随  $s$  和  $\lambda$  的函数变化。对于  $\lambda = 2.9, 3.9, 4.9, 5.9, 6.9$ ，我们展示了当  $s$  足够大使得  $\rho < 1$  时  $G$  的变化。我们注意到  $s$  存在一个使  $G$  取到最大值的最优选择。应用我们的分析结果，我们可以找到满足  $\partial G / \partial s = 0$  的  $s$ 。对于速度闲置模型， $s$  的最优值为 0.64142, 0.77247, 0.90393, 1.03669, 1.17074；对于速度恒定模型， $s$  的最优值为 0.59600, 0.73131, 0.86641, 1.00207, 1.13843。

### 7.3 最优规模与速度

当  $\lambda$ 、 $\bar{r}$ 、 $P^*$ 、 $\alpha$ 、 $\beta$ 、 $\gamma$ 、 $a$ 、 $c$  和  $d$  都确定了以后，我们的第三个问题是找到使  $G$  最大的  $m$  和  $s$ 。为了使  $G$  取得最大值，我们需要分别找到满足  $\partial G / \partial m = 0$  的  $m$  和满足  $\partial G / \partial s = 0$  的  $s$ ，其中  $\partial G / \partial m$  和  $\partial G / \partial s$  在以上两个小节都已经推导出来了。

在图 9 和图 10 中，我们应用图 1 到 8 的参数，分别就两种能耗模型展示了当  $\lambda = 6.9$  时单位时间内净收入  $G$  随  $s$  和  $m$  的函数变化。当  $m = 4, 5, 6, 7, 8$  时，我们展示了当  $s$  足够大使得  $\rho < 1$  时  $G$  的变化。应用我们的分析结果，我们可以找到满足  $\partial G / \partial m = 0$  和  $\partial G / \partial s = 0$  的  $m$  和  $s$ 。在速度闲置模型中，理论最优值是  $m = 5.2997$ 、 $s = 1.49733$ ，此时应用 C 的解析近似得到  $G$  的最大值为  $G = 44.27222$ 。在实际操作中， $m$  可以是 5 或者 6。当  $m = 5$  时， $s$  的最优解为 1.57769，此时  $G$  的最大值为  $G = 44.23815$ ；当  $m = 6$  时， $s$  的最优解为 1.34063，此时  $G$  的最大值为  $G = 43.85816$ 。因此，实际操作中最优的设定为  $m = 5$ 、 $s = 1.57769$ ，应用 C 的解析近似可以得到单位时间最大净利润  $G = 44.30783$ 。在速度恒定模型中，理论最优值

是  $m = 5.45539$ 、 $s = 1.42536$ ，此时应用解析近似得到  $G$  的最大值为  $G = 43.22597$ 。在实际操作中， $m$  可以是 5 或者 6。当  $m = 5$  时， $s$  的最优解为 1.54338，此时  $G$  的最大值为  $G = 43.13990$ ；当  $m = 6$  时， $s$  的最优解为 1.30750，此时  $G$  的最大值为  $G = 42.93951$ 。因此，实际操作中最优的设定为  $m = 5$ 、 $s = 1.54338$ ，应用 C 的解析近似可以得到单位时间最大净利润  $G = 43.21301$ 。

## 8. 总结

我们提出了云计算中考虑多种因素的价格模型，其中包括服务的需求量  $r$ 、应用环境的负载  $\lambda$ 、多服务器系统的配置（ $m$  和  $s$ ）、服务等级协议  $c$ 、用户满意程度（ $r$  和  $s_0$ ）、服务质量（ $W$  和  $T$ ）、低质量服务的处罚  $d$ 、租用费用（ $\beta$  和  $m$ ）、耗电费用（ $\alpha$ 、 $\gamma$ 、 $P^*$  和  $P$ ），以及服务供应商的收益和利润  $a$ 。通过应用  $M/M/m$  排队系统，我们阐述并解决了云计算中多服务器最优化配置达到利益最大化的问题。我们的讨论可以很容易地扩展到其它的服务收费功能，我们的方法也可以被应用到其它价格模型中。