

# B/S 模式在线考试系统性能优化及实现<sup>①</sup>

闫中威<sup>1,2</sup>, 孙大崑<sup>2</sup>

<sup>1</sup>(中国科学院大学, 北京 100049)

<sup>2</sup>(中国科学院 沈阳计算技术研究所, 沈阳 110168)

**摘要:** 在线考试相比传统考试方式有着很多的优点, 例如在线考试的成本更低, 在线考试试题的即时更新性, 在线考试可以更快给出应试人员的答题情况的统计分析, 还可以给不同的应试人员随机出题以防止抄袭等. 本文系统的分析了在线考试系统的架构以及如何就系统性能方面进行优化, 着重分析了基于 Ajax 的前端优化, 分布式缓存机制等优化手段.

**关键词:** 在线考试; 分布式缓存; Ajax 局部刷新; 高并发

## Performance Optimization and Implementation of Online Examination System Based on B/S Mode

YAN Zhong-Wei<sup>1,2</sup>, SUN Da-Wei<sup>2</sup>

<sup>1</sup>(University of Chinese Academy of Sciences, Beijing 100049, China)

<sup>2</sup>(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

**Abstract:** Online examination compared to traditional examination methods has many advantages, such as lower cost, the real-time updates of the examination papers, online examination can offer statistical analysis of examination result in time, it also can generate different examination questions to prevent plagiarism. This paper systematically analyzes the architecture of online examination system and how to optimize system performance, such as front-end optimization based on Ajax, distributed caching mechanism and so on.

**Key words:** online examination; distributed cache; Ajax partial page refresh; high concurrency

### 1 引言

在我国传统的考试方式一般由人工命题, 考生参加考试, 人工阅卷, 成绩评估和分析试卷, 这五大步骤组成<sup>[1]</sup>. 这种传统的考试方式不仅成本高, 并且给老师和学生都带来了很大麻烦, 还有就是传统人工阅卷方式很容易出差错, 在一定程度上影响到考试的公平性.

随着计算机技术和互联网的迅速发展, 建立基于 Web 浏览器的在线考试系统成为了一种发展的趋势<sup>[2][3]</sup>. B/S 模式即是浏览器/服务器模(Browser/Server), 它以 HTTP 为传输协议, 客户端通过浏览器访问 WEB 服务器和与其相连接的后台数据库. 采用 B/S 模式的在线考试系统, 用户只需要使用浏览器登录就可进行在线考试, 成本低廉. 在线考试系统除了可以进行远程考试, 还被广泛应用于校园网内的学生自学与考核. 通过对题库大量题目的练习, 可以提高学生

的学习和应试能力. 它不受时间和空间的限制, 学生可以在任何闲暇时间进行练习和自我考查.

### 2 在线考试系统的结构分析

在线考试系统主要面向的用户有两类, 一类是教师角色, 另一类是考生角色. 因此根据使用用户的不同, 如图 1 所示, 在线考试系统从功能上可以分为两大部分: 教师管理子系统和学生考试子系统.



图 1 在线考试系统两大子系统

<sup>①</sup> 收稿时间:2016-01-20;收到修改稿时间:2016-03-17 [doi:10.15888/j.cnki.csa.005371]

教师管理子系统的功能结构如图 2 所示。

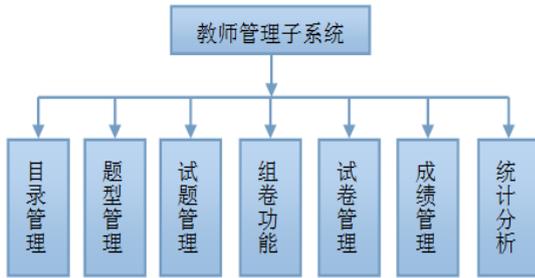


图 2 教师管理子系统

学生考试子系统的功能结构如图 3 所示。

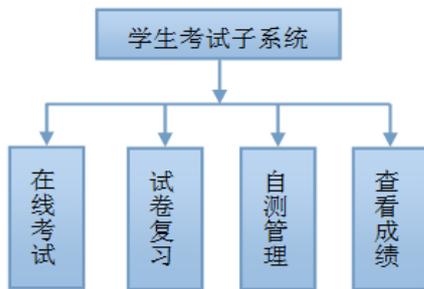


图 3 学生考试子系统

### 3 如何对在线考试系统性能进行优化

基于 B/S 模式的在线考试系统的的确是有着客观的优点的，但是系统在性能方面的表现对系统有着相当大的影响，甚至可以说系统性能的好坏直接影响了在线考试系统能否胜任的了大型的在线考试。

在传统纸质考试方式时，学生获取试卷试题是很快速的，等待监考老师发下试卷之后可以很轻松随意的查看任意一道试题，而在在线考试系统却不具备这一特点，由于带宽的限制，在线考试方式一般不会显示全部试题，而是一部分一部分的展示，这样以来就产生了一个问题，即增加了用户等待时间。这一点直接影响了上节中学生考试子系统中在线考试功能的好坏。

#### 3.1 系统要能够支持高并发以及快速响应

在组织大型的考试时，就特别需要在线考试系统提供安全可靠的性能，不能出现用户页面卡死以及登陆不了系统等不良情况，所以这就要求了系统要具有支持高并发的能力。

考试一般都是限时的，因此在线考试系统在与考生的交互过程中必须做到快速响应，考生在获取试题

时不能产生较长时间的延迟，在考试中时间是非常宝贵的，较慢的响应速读势必会影响到考生，也在一定程度上影响了考试的公平性，因此必须对在线考试进行性能方面的优化，从而能避免出现页面卡顿，服务器响应较慢等影响考生答题时间的情况。

## 4 基于 Ajax 局部刷新技术

为了实现页面与考生快速交互，Ajax 局部刷新技术无疑是一个很好的选择。Ajax 这个概念的最早提出者 Jesse James Garrett[4]。Ajax 全称为“Asynchronous JavaScript and XML”（异步 JavaScript 和 XML），是指一种创建交互式网页应用的网页开发技术。

### 4.1 传统 Web 请求方式

传统的 web 应用允许用户填写表单(form)，当提交表单时就向 web 服务器发送一个请求。服务器接收并处理传来的表单，然后返回一个新的网页(如图 4 所示)。这个做法浪费了许多带宽，因为在前后两个页面中的大部分 HTML 代码往往是相同的。由于每次应用的交互都需要向服务器发送请求，应用的响应时间就依赖于服务器的响应时间，这导致了用户界面的响应比本地应用慢得多[5]。

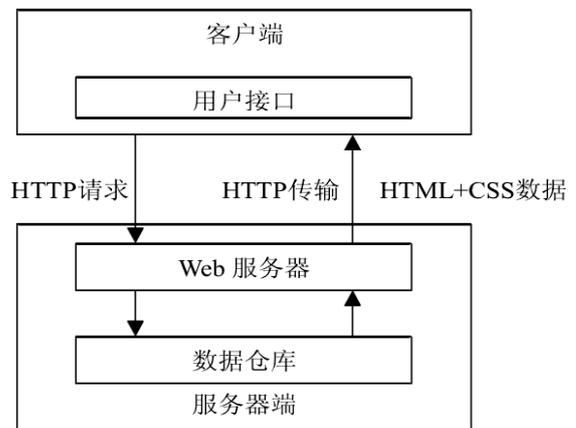


图 4 Web 工作原理图

### 4.2 基于 Ajax 的 Web 请求方式

与传统方式不同，Ajax 技术可以仅向服务器发送并取回必需的数据，它相当于在客户端和服务器之间加了一个中间层，即 Ajax 引擎。并不是所有的用户请求都提交给服务器，一些数据的验证和处理由 Ajax 自己来做，只有确实需要从服务器读取新数据时，才

由客户端通过 JavaScript 调用 Ajax 引擎向服务器端发出 Http 请求,但它并不等待请求的响应,用户可以继续浏览或交互.当服务端的数据以 XML 形式返回时,Ajax 引擎接收数据,并指定 JavaScript 函数来完成相应的处理或页面的更新,而不是刷新整个页面,从而实现用户操作与服务器响应的异步化.如图 5 所示.

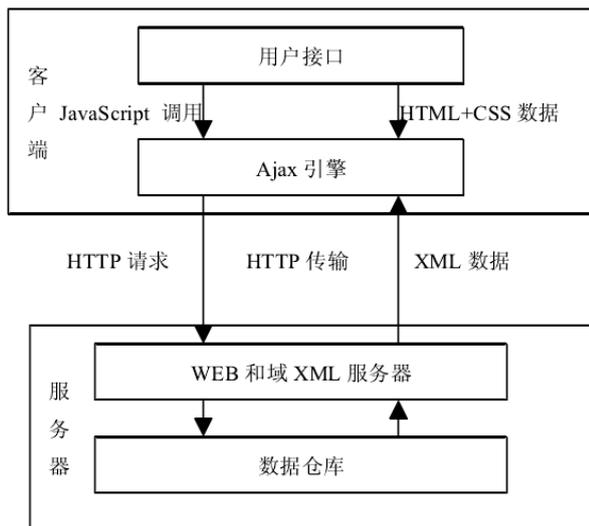


图 5 Ajax 工作原理图

Ajax 的这种工作方式,在浏览器和服务器之间使用异步数据传输进行 HTTP 请求,这样子不用整个页面的变化,只需要更新的模块发生变化,这样网页从服务器请求少量的信息,减轻了网页的负担,提高了网页的响应速度,整个页面也不会出现刷新,改善了用户体验<sup>[6]</sup>.在第二节中提到的教师管理子系统中的试题管理模块中添加试题功能以及学生在线考试获取试题功能都用到了这种局部刷新技术.

#### 4.3 Ajax 局部刷新技术的实现

实现 Ajax 局部刷新技术大体上需要一下六个步骤:(1)创建 XMLHttpRequest 对象,也就是创建一个异步调用对象;(2)创建一个新的 HTTP 请求,并指定该 HTTP 请求的方法、URL 及验证信息;(3)设置响应 HTTP 请求状态变化的函数;(4)发送 HTTP 请求;(5)获取异步调用返回的数据;(6)使用 JavaScript 和 DOM 实现局部刷新.

将 Ajax 技术应用的在线考试系统中,将会使得网页的响应速度得到很大的提高,为考生节省了宝贵的时间,同时也在一定程度上减轻了服务器的压力.

## 5 Redis 分布式缓存技术

Redis 全称是 Remote Dictionary Server(远程字典服务器),是一个由 Salvatore Sanfilippo 写的 Key-value 存储系统<sup>[7]</sup>,属于 NoSQL 系列产品之一.Redis 提供了一些丰富的数据结构,如 String、Lists、Set, 这些数据类型都支持 push/pop、add/remove 取交集并集和差集及更丰富的操作,而且这些操作都是原子性的.为了保证效率,数据都是缓存在内存中,读写性能极高.同时,Redis 可以实现数据持久化<sup>[8][9]</sup>.

### 5.1 Redis 用作数据缓存服务器

由于 Redis 的这种将常用数据都缓存在内存中的特性,所以 Redis 特别适合为在线考试系统提供分布式缓存服务.将 Redis 部署在 web 应用服务器和关系型数据库之间,web 应用服务器在需要获取关系型数据库数据时先从 Redis 中取,如果 Redis 中存在该数据,就省去了访问关系型数据库的时间;如果 Redis 中不存在该数据,再到关系型数据库中获取,并且同时将取出的数据缓存到 Redis 中.示意图如图 6 所示.

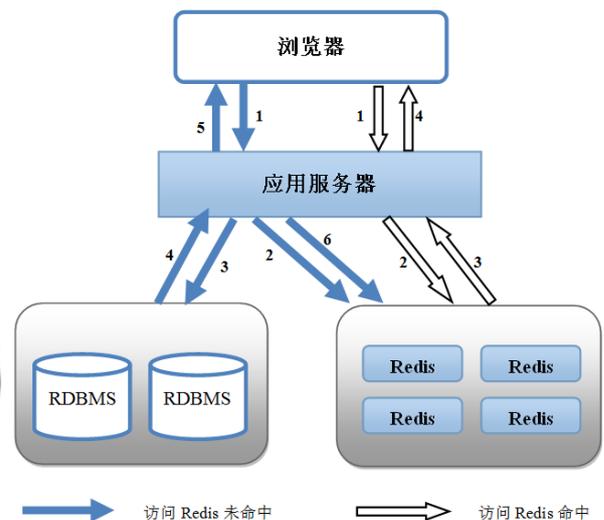


图 6 Redis 缓存原理图

在图 6 中,访问未命中的情况:1.浏览器发起数据请求;2.应用服务器去 Redis 获取数据,未命中;3.去 RDBMS 去获取数据;4.应用服务器获得数据;5.将数据发给浏览器;6.把数据缓存到 Redis 中.访问命中的情况:1.浏览器发起数据请求;2.应用服务器去 Redis 获取数据,命中;3.应用服务器从 Redis 中取出数据;4.将数据发给浏览器.

### 5.2 分布式缓存机制

如果只有一台 Redis 缓存服务器,那么当该服务器出问题后将会影响到整个系统,并且一台缓存服务器很可能遇到性能上的瓶颈,因此利用多台缓存服务器构建一个分布式缓存系统是十分有必要的。

以由 3 个缓存节点为例,其构成的分布式缓存系统如图 7 所示,当浏览器向应用服务器发起读写数据的请求时,web 应用服务器调用应用程序库中相应的分布式算法来确定去哪个缓存节点上面去获取或者写入数据。

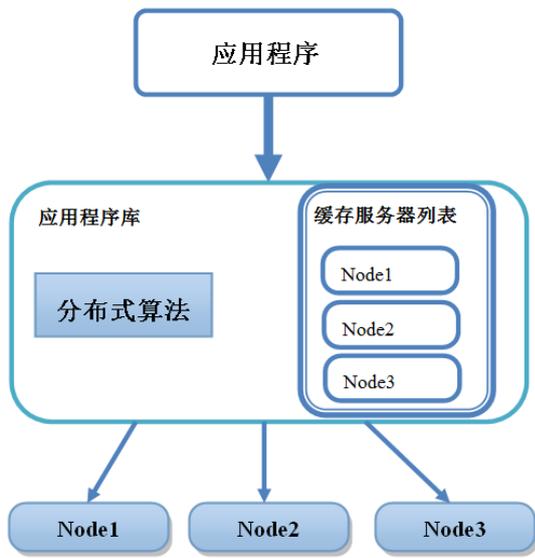


图 7 分布式缓存系统

常用的分布式算法是先对 key 进行哈希,然后对节点个数进行取余运算,进而得到该 key 所对应的节点,即  $Hash(key) \% N$ ,其中  $N$  为节点个数.这个算法的优点是:在  $N$  个节点性能相同时,大部分情况下这  $N$  个节点之间能实现良好的负载均衡,不会出现其中某些节点压力很大,而另一些压力很小的情况.但是这种分布式算法也有一个很致命的缺点,就是其不适合做动态的节点扩充和缩减,当节点数  $N$  增加或者减少时,例如  $Hash(key) \% (N+1)$  所得到的结果与  $Hash(key) \% N$  将会不同,这就会对所有节点产生影响,如此一来所有节点上的缓存数据不得不重新创建,这势必会给系统带来很大的不良影响。

为了应对上述算法的不良影响,就有人发明了一致性哈希算法(Consistent Hashing),一致性哈希算法在 1997 年由麻省理工学院的 Karger 等人在解决分布

式 Cache 中提出的<sup>[10]</sup>.考虑到分布式系统每个节点都有可能失效,并且有时根据需要新的节点很可能动态的增加进来,如何保证当系统的节点数目发生变化时仍然能够对外提供良好的服务,如果某台服务器失效或者动态增加节点,对于整个系统来说几乎所有的缓存都会失效,如果不采用合适的算法来保证一致性,那么在上述情况发生时整个缓存系统中所有节点都会受到影响,一致性哈希算法很好的解决了这个问题. Consistent Hashing 算法基本原理如图 8 所示.首先求出 Redis 服务器(节点)的哈希值,并将其配置到  $0 \sim 2^{32}$  的圆(continuum)上.然后用同样的方法求出存储数据的键的哈希值,并映射到圆上.然后从数据映射到的位置开始顺时针查找,将数据保存到找到的第一个服务器上.如果超过  $2^{32}$  仍找不到服务器,就会保存到第一台 Redis 服务器。

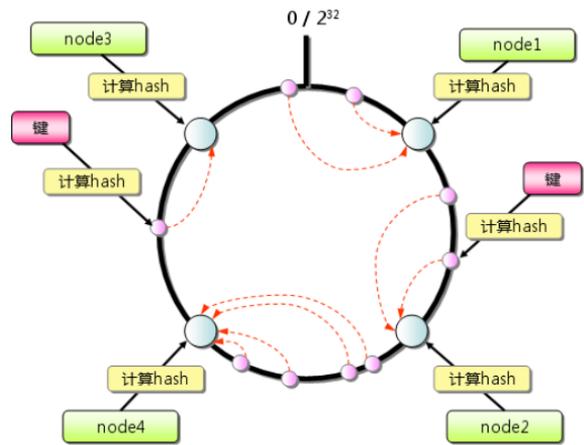


图 8 一致性哈希算法基本原理

这时,如果动态的增减节点数将不会对整个系统产生影响,影响的只是某一个节点.例如,现新增一个节点 node5, node5 经过 Hash 以后,其值是介于 node2 与 4 之间,那么我们仅需将 node4 中哈希值小于 node5 哈希值的一部分数据迁移到 node5 中即可,对其他节点不会产生影响;减少节点个数时,只需将该节点上的数据全部迁移到它的下一个节点即可,也同样不会对其他节点产生影响.这样以来,动态的增加和减少节点对整个分布式缓存系统产生的影响将会降低,最大限度地抑制了键的重新分布。

但是,一致性哈希算法也有着两个很明显的缺点: 1) 该算法将不能在保持负载均衡,结点较少时易导致数据分配不均,例如,在上图中若 Node1 和 Node2 之间的间距很小,而 Node2 和 Node4 之间的间距很大,这样 Node4 的压力将会比 Node2 的压力大得多; 2) 无

法适应结点的异质性,即不能根据结点能力调整负载。为了抑制这种不均衡的现象,可以运用虚拟节点(virtual node)的思想,即为每个物理节点(服务器)在 continuum 上分配多个节点,并且根据物理节点能力来决定分配的虚拟节点个数,这样就能在一定程度上抑制分布不均匀,并且能最大限度地减小服务器增减时的缓存重新分布,原理图如图 9 所示。

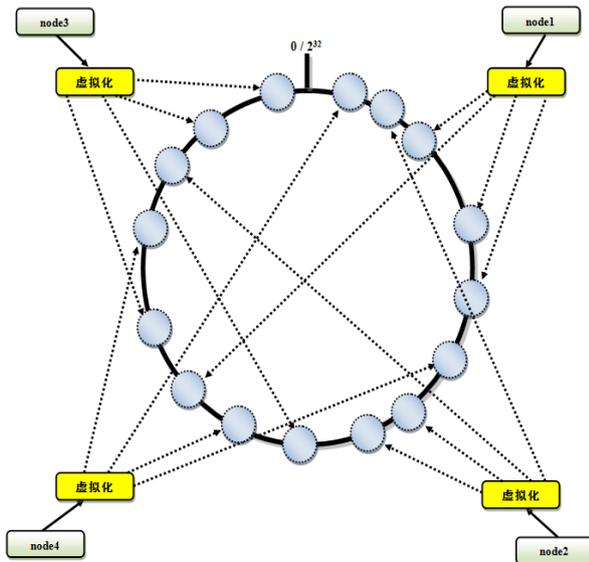


图 9 一致性哈希算法虚拟化基本原理

在下图中对每个物理节点进行了虚拟化,每个物理节点对应 4 个虚拟节点,这样就能在最大程度上实现负载的均衡。

### 5.3 性能验证

在线考试系统中引入分布式缓存系统究竟能在性能上带来多大的提升,为此我们对分布式缓存系统在性能提升上进行了实验验证。

#### 1) 在线考试系统实验环境配置

Web 服务器	8CPU, 32G 内存
数据库	Oracle10g
操作系统	CentOS 6.5

#### 2) 分布式系统实验环境配置

Redis 服务器	2 台 Pentium 4, 4G 内存
操作系统	CentOS 6.5

每个物理缓存节点虚拟化为 4 个虚拟节点。

我们用 LoadRunner 仿真 100-2000 名 vuser 登录平台进行考试的情景,分别测试了不同并发的用户数量时启用分布式缓存和不启用缓存时的系统性能表现,即获取试题事务的平均响应时间。结果如图 10,图中横轴代表并发用户数量,纵轴代表请求试题事务

的平均响应时间,单位为秒;从图 10 中我们可以很清楚的看出不启用分布式缓存和启用分布式缓存之间的性能差距,在并发量较小时缓存系统的作用不是很明显,但随着并发量的增加缓存系统的作用就显得非常大了,原因就是当并发量很大时,数据库压力会很大,这就导致了很事务都要排队等候,而启用缓存之后,很多请求数据是很快地直接从 Redis 服务器上获取的,因此响应速度很快,同时也减轻数据库的压力。

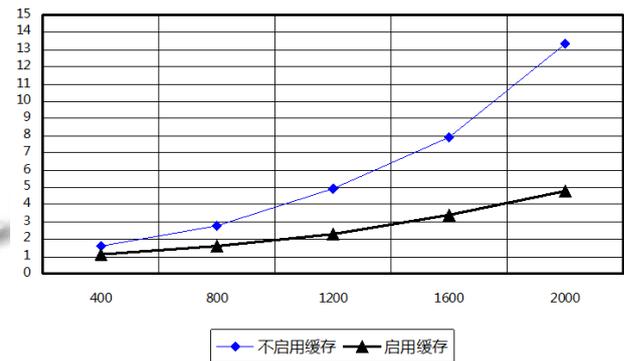


图 10 试验结果

## 6 结语

本文对基于 B/S 模式的在线考试系统关键技术进行了研究,重点研究了通过前端优化和构建分布式缓存系统来提高在线考试系统的响应速度,为考生提供更好的体验。

### 参考文献

- 余胜泉,王耀武.网络课程的设计与开发.教育技术通讯,2000,(10):34-37
- 马莉.基于 InItmet 远程课程考试系统的设计.计算机工程与应用,2001,22
- 崔梦天,胥学跃.通用多媒体在线考试系统的研究与实现.情报杂志,2006,(8):59-61
- Garrett JJ. Ajax: A New Approach to Web Applications. <http://www.adaptivepath.com/publications/essays/archives/000385.php>, 2007.
- Robert D. Hof in San Mateo, Calif. Ajax: How to weave a faster web. Business Week, 2005
- Crane D, Pascarello E, James D. Ajax in action. Manning Publications Co., 2005
- 王心妍.Memcached 和 Redis 在高速缓存方面的应用.无线互联科技,2012,(9):8-9
- 范凯.NoSQL 数据库综述.程序员,2010,(6):76-78
- Xiang P, Hou R, Zhou Z. Cache and consistency in nosql. 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT). IEEE. 2010, 6. 117-120.
- Darger D, Lehman E, Leighton T, Levine M, Lewin D, Panigrahy R. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web. ACM Symposium on Theory of Computing. 1997. 654-663.