

高并发 Web 航空票务秒杀系统的设计与实现

李军锋, 何明昕

(暨南大学 计算机科学系/信息技术研究所, 广东 广州 510632)

摘 要: 针对电子商务秒杀系统中出现的瞬时高并发访问的问题, 设计并实现了一个鲁棒的、可扩展的航空机票秒杀系统。根据高并发 Web 秒杀系统的特点, 从网站的系统架构优化入手, 从负载均衡、页面优化、缓存设计与优化、数据库优化等方面详细研究了高并发 Web 秒杀系统的解决方案。该方案可以有效解决高并发 Web 秒杀系统的高并发难题, 应用实践表明了该方案的可行性。

关键词: 秒杀; 高并发; 负载均衡; 系统架构; 缓存; 页面静态化

中图法分类号: TP393.02 **文献标识号:** A **文章编号:** 1000-7024 (2013) 03-0778-05

Design and implementation of web-based air ticket seckill system with high-concurrency

LI Jun-feng, HE Ming-xin

(Department of Computer Science, Jinan University, Institute of Information Technology, Guangzhou 510632, China)

Abstract: Concerning the problem of instantaneous and high-concurrency access in the e-commerce seckill system, a robust and extensible seckill system on the airline ticket sales is designed and implemented. Based on the feature of high-concurrency web seckill system and in consideration of the system architecture optimization of the website, the solution of the high-concurrency web seckill system is discussed in details from the aspects of load-balancing, page optimization, cache configuration and database optimization. The solution is put into practice in an airline company and it shows the strategy is applicable. Facts prove that the strategy can efficiently solve the problem of high-concurrency in the high-concurrency web seckill system.

Key words: seckill; high-concurrency; load balancing; system architecture; cache; URL rewrite

0 引 言

高并发 Web 应用在基于互联网的应用规模急剧扩大的环境下面临更大的并发访问压力。12306 铁路购票网站就是一个典型的高并发 Web 应用。今年春运期间, 该网站日均近 10 亿次访问, 同时在线人数高达 500 万, 如此高的负载与高并发量使网站访问拥堵, 频繁出现错误, 大量用户买不到票^[1]。2008 年北京奥运会售票系统也是一个高并发 Web 应用, 瞬时的高并发访问曾经使服务器不堪重负, 系统在刚开始使用的数小时内近于瘫痪^[2]。

秒杀应用是一类有代表性的高并发 Web 应用。其高并发性体现在短短数秒或几分钟内会产生大量的访问, 给服务器带来巨大压力, 导致服务器响应变慢、处理出错、宕机、甚至整个系统的瘫痪。对于 Web 秒杀应用, 除了要优

先考虑高并发处理能力之外, 还要考虑安全性、秒杀外挂、库存管理等问题。

经过文献搜索查找, 目前尚未发现公开发表的有关高并发 Web 秒杀应用的文章。本文从网站的系统架构入手, 提出了一个高并发 Web 航空票务秒杀系统的解决方案。首先在 Web 服务器端采用负载均衡技术来分担并发访问压力, 再运用缓存技术将常用页面缓存。其次进行页面优化, 例如页面静态化与页面压缩。此外进行数据库前端缓存设计和数据库 SQL 优化。该航空票务秒杀应用系统经过严格的压力测试后上线, 最近的一次秒杀活动承受了约 10 万人次的访问, 在应对高并发访问上取得了较好的效果。

1 秒杀系统的特点、目标及流程分析

本节首先对秒杀系统的特点进行分析, 然后阐述航空

收稿日期: 2012-06-07; 修订日期: 2012-08-12

基金项目: 国家科技部科研院所技术开发研究专项基金项目 (2012EG124225)

作者简介: 李军锋 (1987-), 男, 河北邯郸人, 硕士研究生, 研究方向为软件工程; 何明昕 (1963-), 男, 湖北荆州人, 博士, 副教授, 研究方向为软件工程、并行分布式网络计算及企业应用系统。E-mail: litjfly@126.com

票务秒杀系统的设计目标,最后对其流程进行分析。

1.1 秒杀系统的特点

秒杀系统与普通的高并发票务系统相比,既有相同之处,又有其自身的特点。以12306订票网站为例,该网站的前端Web和后台数据库查询都面临高并发的问題,因为订票用户不仅大量访问前台页面,更要详细地查询车次、座位、票价等车票信息,这无疑增加了数据库的并发访问,因而对于12306而言,前端和后端都是高并发瓶颈。

与12306不同,秒杀应用的瓶颈主要集中在前端Web访问部分,后台数据库的负担相对较轻。因为秒杀活动开展的时候流量主要集中在前端,尤其是大量用户同时点击按钮的瞬间以及前后几分钟内,并发量达到最大。而涉及数据库的操作主要在于秒杀航线及座位的查询以及结果的写入,完全可以在产生秒杀成功者名单的时候再将这些数据一次写入数据库。此外由于秒杀提供的奖品座位数比较少,查询操作也只限于这几个座位,完全可以利用缓存技术,将这些座位数据写入数据库前端的应用服务器中,减少数据库访问次数。

1.2 秒杀系统的设计目标

秒杀系统的设计目标主要有:①满足高并发访问。这是秒杀系统设计的主要目标与首要解决的问题。②可扩展性。由于秒杀活动的逐渐升温,参与秒杀的人越来越多,

这意味着秒杀系统需要不断投入更高速度的带宽和更高性能的硬件资源,因此在架构设计之初就要考虑到扩展的需要。③高可靠性。高可靠性是指网站具有稳定性,系统结构合理,产生故障时能够较快地排查与恢复。例如在一台服务器宕机后可以快速地切换到另一台备份服务器上,不使服务中断过长的时间。④安全性设计。是指结构和功能上的设计充分考虑到安全。要能控制非法入侵和防止黑客攻击,例如DoS、XSS攻击等。⁵⁵秒杀应用的特殊要求。例如:秒杀外挂的防止。秒杀外挂是一种原理类似游戏外挂的软件,利用这款软件可以不断刷新页面,从而比一般用户更快地秒杀到商品;这就严重损害了秒杀活动的公平性。

1.3 航空票务秒杀流程

如图1所示,航空票务秒杀流程为:①用户登陆网站,查看航线信息,Web服务器返回未来航线的信息。②查看库存数,服务器返回当前的库存。③倒计时变为0后,用户点击“秒杀”按钮,首先验证该用户是否登陆。如果未登陆,返回“未登陆,登陆后才能进行秒杀”并跳转至登陆页面;已登陆则进行下一步。④首先判断用户是否秒杀成功。要秒杀的座位库存数设为N,有用户访问则进行N-1操作,直到N变为0。此时用户再访问,则返回信息“秒杀失败”。如果秒杀成功,库存进行减1操作,查询系统库存返回秒杀座位数,提示“秒杀成功”,进行下一步。如果

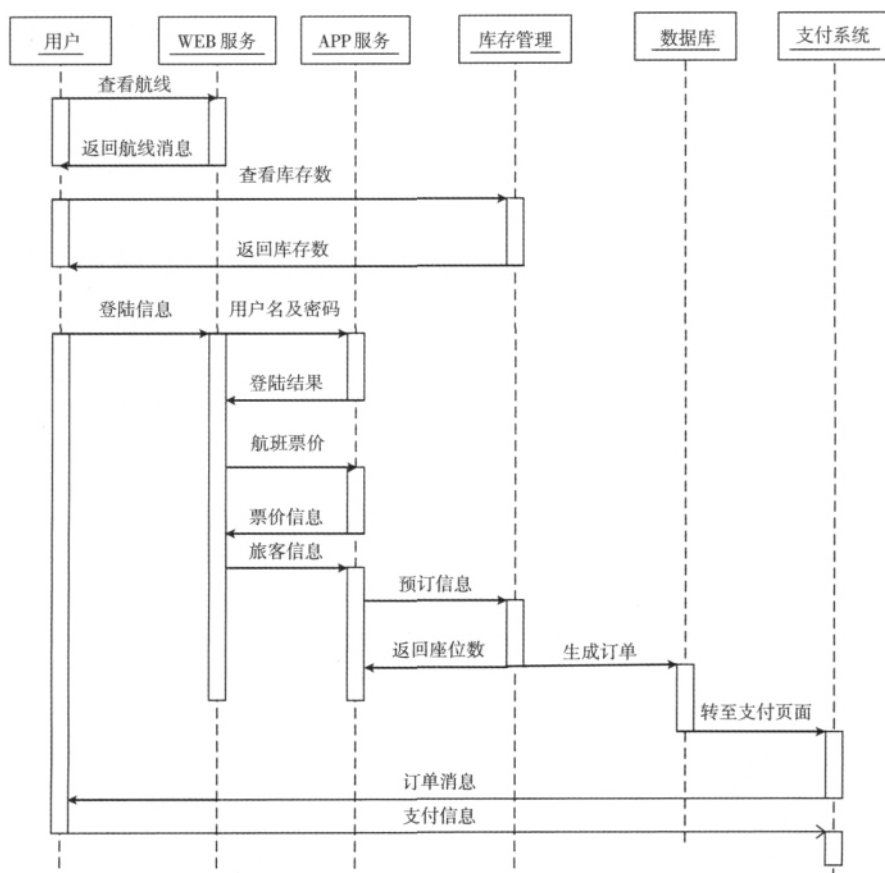


图1 航空票务秒杀流程

库存为 0 则提示信息“可秒杀座位数为 0”，返回秒杀页面。

⑤秒杀成功后，将用户的个人信息、秒杀信息等写入数据库。

⑥用户进入支付页面，选择支付方式进行支付，如果支付成功，则返回“支付成功”。系统在进入支付页面后开始计时，如果 20 分钟内未完成支付，则库存数加 1，N 加 1，数据库执行回滚操作，系统返回初始秒杀页面。此时其它用户仍可以点击“秒杀”按钮进行秒杀。

2 秒杀系统的架构与负载均衡

要解决高并发问题，最关键的是系统架构设计。本节先介绍航空票务秒杀系统的功能模块及瓶颈，然后介绍负载均衡技术并给出了一个应对高并发的架构解决方案。

2.1 航空票务秒杀的功能及瓶颈

如图 2 所示，航空票务秒杀系统的主要功能包括登陆模块、秒杀活动设置管理模块、航班票价信息获取模块、秒杀航线展示模块、用户秒杀模块、支付模块。

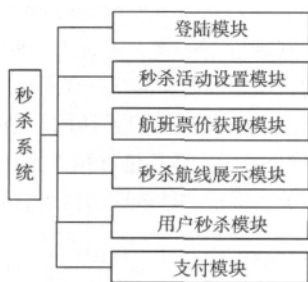


图 2 航空票务秒杀的功能模块

其中登陆和支付模块分别用于用户登陆和支付。秒杀活动设置模块由系统管理员提前设置待秒杀航线、座位数、价格、开始时间及支付方式等信息。票价获取模块可以自动获取航班及相关票价。航线展示模块用于展示秒杀的航线及相关信息。用户秒杀模块包括用户点击秒杀按钮，输入旅客信息和产生秒杀结果的过程。

在功能模块中，登陆一般不会成为并发瓶颈。系统规定只有登陆的用户才可参与秒杀，用户多会在秒杀开始前登陆并密切关注倒计时。由于登陆的时间分散，通常不会产生高并发现象。支付模块也不会造成网络瓶颈，因秒杀奖品数量不多，只有秒杀成功的少数用户进入支付环节，对系统压力不大。票价展示模块也不会形成瓶颈。秒杀航线展示模块会有不小的访问压力，本方案采用静态页面技术来解决。最容易发生拥堵的是用户秒杀模块，在用户点击“秒杀”的一瞬间以至几分钟时间内，产生巨大的并发访问流量，本方案采用负载均衡技术以及缓存技术来缓解并发压力。

2.2 秒杀系统的负载均衡方案

为了应对高并发访问，采用负载均衡技术构建网站系统架构是有效的办法。负载均衡的好处是：①将负载分担

于多台服务器上，解决了高并发难题；②提高了可扩展性；③提高了安全性。用户见到的是负载均衡设备，服务器对用户是透明的^[3]。

根据负载种类与逻辑分层，负载均衡分为：①前端 WEB 服务器的负载均衡；②逻辑层应用服务器负载均衡；③后端的数据库集群、文件存储系统的负载均衡。WEB 服务器、App 服务器主要分担的是网络流量；后端多利用集群技术来分担数据操作。

负载均衡的实现方法有多种，最常用的有以下几种^[4]：①DNS 负载均衡；②反向代理负载均衡；③基于 NAT 的负载均衡；④软件负载均衡；⑤硬件负载均衡。DNS 方式简单、易实现，但是不能获知各台服务器的差异。反向代理方式用反向代理服务器来做均衡器，缺点是流量过大时服务器本身容易成为瓶颈^[4]。NAT 负载均衡利用网络地址转换技术来实现。软件方式通过在服务器上安装各种软件来实现负载均衡，如企业级 PCL-SIS、开源的 LVS 等，成本低，但易受操作系统影响，扩展性和稳定性较差，如 LVS 仅限于 Linux 服务器。硬件负载均衡是把软件负载均衡的功能用硬件来实现。硬件负载均衡设备，又称为负载均衡器，在 OSI 七层网络模型中作用于 L4/L7 层，本质上是一个 L4/L7 层上的交换机，因而兼有 L4/L7 层智能流量管理功能，一般通过 NAT 技术来实现。基本可以实现各种负载均衡算法。常用的有 Foundry、F5 BIG-IP、Radware、Citrix NetScaler 等。

本方案采用硬件方式，用 F5 负载均衡器实现负载均衡。F5 性能稳定、技术成熟，应用于新浪、雅虎、百度、搜狐、猫扑等大型网站。本方案使用 F5 负载均衡器的优点包括：①便于进行操作和维护，管理简单。②易于部署和扩展，当需要扩展时，只需要简单配置和增加新的链路和服务即可。③充分利用 F5 自带的各项安全应用：开启 F5 负载均衡器的 DoS 与 XSS 防护功能，能有效防止 DoS 攻击以及 XSS 跨站脚本攻击，大大提高系统整体安全性能。④使用 F5 的智能流量管理功能。⑤开启了 F5 负载均衡器自带的 SSL 加速功能，可快速处理用户发送的 HTTPS 请求，解除了各台服务器的 SSL 加解密负担，从而减轻了服务器的负荷。⑥开启并使用了 F5 负载均衡器的缓存功能。

2.3 航空票务秒杀系统的架构

航空票务秒杀系统的系统架构如图 3 所示。根据三层逻辑结构，将 Web 服务器与应用服务器、数据库服务器三层分离。在 Web 服务器群的前端采用 F5 负载均衡器进行流量的负载均衡，并在 F5 负载均衡器中进行缓存设置。系统采用两条链路单独部署，WEB 服务器分别与两台应用服务器相连，最后连接库存服务器和数据库服务器。应用服务器与数据库服务器之间有一台 Memcached 缓存服务器。本系统的 Web 服务器采用 Nginx 1.0.4，应用服务器采用 Websphere6.0，数据库服务器为 Oracle 10 g。

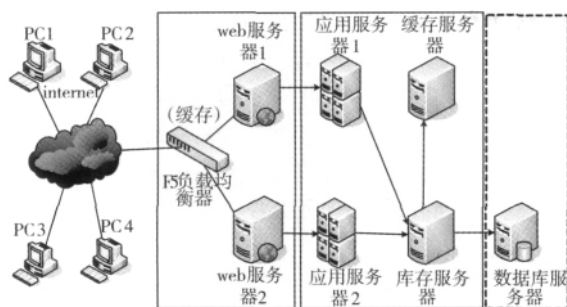


图3 秒杀系统的拓扑结构

3 秒杀系统的优化方案

本节主要从航空票务秒杀系统的页面优化、缓存设置、数据库优化3个方面给出应对高并发的系统优化方案。

3.1 秒杀系统的页面优化方案

对于高并发高负载网站, 页面质量是影响网站运行速度的一个重要因素。首先要考虑的是页面大小。页面设计得过大, 加载页面就会耗费大量网络带宽, 因此页面设计应尽可能简洁。这个看起来很简单的策略, 事实上可大大减少网络流量, 有效地缓解高并发带来的压力。另外, 现在广泛采用动态网页, 这些网页中的动态元素的加载, 也会大量增加服务器的负担。采用页面静态化技术^[5]与页面压缩技术, 可以解决上述问题。

页面静态化 (URL rewrite) 是提升处理速度, 应对高并发的有效途径之一。页面静态化是指将指含有大量动态元素的动态网页, 如jsp、asp、php等, 转换为html和html等静态页面。静态页面由于不用加载动态元素, 其访问速度要比动态页面快得多, 页面静态化也因而成了提升网站速度的常用技术。

航空票务秒杀应用系统的页面主要有秒杀航线展示页面、秒杀页面、支付页面、登陆页面等。其中秒杀页面seckill.jsp是最常访问的页面之一。本方案在WEB服务器上部署了页面静态化服务。设置该服务每隔5秒钟执行一次, 访问应用服务器中的seckill.jsp, 并将其转换为静态页面seckill.html, 并保存在本地的秒杀静态文件目录中。当用户需要访问seckill.jsp时, 就不需要访问应用服务器, 访问到的就是Web服务器中已经静态化的seckill.html。这样就大大提高了加载和访问速度。

另外本方案采用页面压缩技术, 系统开启Apache的Gzip压缩^[6], 将页面压缩之后再传送。网页压缩技术有效地减少了页面尺寸, 经测试压缩后的页面大小平均为5K, 仅为之前平均25K的五分之一, 明显节省带宽, 缓解了高并发的压力。

3.2 秒杀系统的缓存方案

缓存技术是应对高并发高负载的另一个重要手段。网站开发中的缓存技术是指将用户要访问的数据存放在服务

器内存中, 当用户要访问这些数据时就不用再访问磁盘或数据库; 而数据库询问和磁盘读写是系统最耗时的工作。这就大大提升了访问速度, 降低了各台服务器的负担, 减轻了高并发所带来的拥堵。根据缓存的位置^[7]与缓存的内容不同, Web缓存主要分为: ①浏览器和客户端缓存; ②前端页面缓存; ③后端数据库的数据缓存。浏览器端和客户端缓存是指将数据或页面缓存在用户的客户端, 当用户需要再次调用时无需访问服务器, 直接访问本地缓存即可。设计高并发网站时, 应尽量将缓存放在客户端而非服务器端。

前端页面缓存在系统前端对Web服务器上的页面进行缓存, 实现方案有: ①直接在Web服务器中开辟一小块内存空间, 用来作缓存^[8], 这是以牺牲Web服务器的内存为代价。现在常用的服务器如Apache等, 都有自己的缓存模块。②在反向代理服务器上实现缓存功能。常用的反向代理缓存服务器有nginx服务器、squid等。但由于反向代理服务器兼有负载均衡的功能, 网络流量都要经过代理服务器, 所以很容易造成服务器单个节点的故障。③使用专门的服务器或其它负载均衡器。在Web服务器集群的前端增加一台服务器做缓存服务器, 其上安装squid等缓存软件^[9]。由于负载均衡器位于Web服务器的前端, 在均衡流量的同时, 也可以使用负载均衡器的缓存功能。

与前端页面缓存类似, 数据库缓存的主要技术有: ①直接与应用服务器内存中开辟一块区域, 用来作数据缓存。②在应用服务器与数据库服务器之间增加服务器作为缓存服务器。这样当应用服务器需要访问数据库时, 则先查询缓存服务器, 如果命中则访问缓存服务器, 如果未命中才连接数据库服务器。

考虑到本方案的网络拓扑结构, 本方案直接利用F5服务器专门提供的缓存功能, 将秒杀页面、秒杀航线展示页面airline.html等经过处理的常用静态化页面放进缓存。在高并发的环境下, 这样可以有效地降低前台Web服务器集群的访问压力, 同时也可大大提高访问效率。

此外, 为了减小数据库访问压力, 本方案在数据库与应用服务器之间配置了一台Memcached服务器。将用户常用的查询结果存在Memcached缓存中, 当用户再次访问时, 先检测缓存中是否有该查询结果, 如果命中则直接从缓存中读取, 未命中才访问数据库, 这样也可减少数据库服务器的访问负担。将来在系统需要进行缓存扩展时可以增加Memcached服务器, 再进一步可以实现Memcached的分布式缓存系统。

3.3 秒杀系统的数据库优化

数据库访问的高并发一直是高并发Web应用的一个难题, 尤其是有大量数据操作的应用。解决方案主要有: 数据库集群负载均衡、数据库读写分离、数据库缓存优化、数据库查询优化等。根据第一节中介绍的秒杀系统的特点, 本方案不采用数据库集群的负载均衡技术, 只采用Mem-

cached 缓存服务器来缓存数据。

数据库查询操作的效率是影响并发响应能力的另一个重要因素。文献 [10] 分析了 SQL 语句执行的过程, 并提出了 ORACLE 数据库中 SQL 优化的若干原则, 例如: ①合理使用索引, 能够起到优化查询的作用。②将具有最大限制的条件放在 where 字句的最后面, 以减少子句的访问次数。③使用联合查询时, 对 from 后面各个表应选择记录条数最少的表作为基础表, 因为 oracle 是按从右往左的顺序处理各个表名的。本方案在开发设计过程中, 按照这些原则^[10]对代码中调用的 SQL 语句进行了严格的书写, 提高了数据库的访问效率。

4 测试与应用

本方案分别对单台 Web 服务器和服务器集群进行压力测试, 选取并发量最高的秒杀页面 seckill.jsp 作为测试页面。经测试, 单台 Nginx 服务器在 2.5 万并发时 CPU 利用率已接近 90%, 由此估计服务器可以承受的并发量在 2 万到 3 万之间。采用两台同样的 Nginx 服务器集群负载均衡后, 在 5 万并发的情况下, 各台服务器的 CPU 平均利用率达 90%, 继续增加并发量会引起宕机, 所以集群的最高并发量约 5 万。可见采用两台 Web 服务器的集群负载均衡后, 系统所能承受的并发量大约是之前的两倍。另外, 响应时间和 TPS (transaction per second) 即每秒处理的事务数是衡量系统吞吐量的重要指标。经测试, 在相同并发下服务器集群的 TPS 约为单台服务器 TPS 的两倍, 响应时间约为单台的 1/2。可见, 采用两台 Web 服务器的负载均衡后吞吐量约为单台 Web 服务器的两倍。另外 F5 对请求的分配, 在 1000 个并发时两台 Web 服务器分配的 session 数分别为 487、513; 2000 个并发时为 946、1054; 5000 个时为 2483、2517; 这些数据表明经过 F5 的调度, 负载是接近均衡分配的。

系统经过多方面严格的压力测试后上线应用。在最新的一次秒杀活动中, 该系统承受了近 10 万人次的在线访问, 峰值并发访问量约为 3.8 万人次, 此时监测表明 Web 服务器的 CPU 利用率约为 85%, 而应用服务器的 CPU 利用率约为 50%, 库存服务器 50%, 数据库服务器仅为 35%, 系统各部分运行正常, 这也验证了秒杀应用的主要瓶颈是在网络前端而非数据库端。

5 结束语

本文设计并实现了一个高并发 Web 航空票务秒杀应用系统。经文献搜索并无公开发表的有关高并发 Web 秒杀应用的文章, 本文首先分析高并发 Web 秒杀应用的特点并根据其特点提出了一个成功的解决方案。本方案综合运用负载均衡、页面优化、缓存优化、数据库优化等技术, 构造了一个鲁棒的高并发 Web 秒杀应用系统。该系统经过多方面严格的压力测试后上线运行。与其它高并发 Web 应用相比

该系统具有高可靠性、易扩展性、安全性等特点, 并有效地解决了 Web 秒杀系统的瞬时高并发难题, 对构建 Web 秒杀应用或类似的高并发 Web 应用有一定的参考意义。另外, 秒杀外挂是高并发 Web 秒杀应用的一个特有的问题。目前淘宝等网站都采取了一些措施来禁止秒杀外挂, 但尚未从根本上解决问题。如何有效阻止秒杀外挂, 有待进一步研究。

参考文献:

- [1] 12306 网站 1 天点击 10 亿次频瘫痪: 最牛电商商诊 [EB/OL]. [2012-01-10] <http://www.yuewe.cn/portal.php?mod=view&aid=74461>.
- [2] 奥运订票人数超预期票务系统“开工”就“瘫痪” [EB/OL]. [2007-10-30]. <http://2008.qq.com/a/20071030/000108.htm>.
- [3] QIN Wei, HE Xinhua. Research of web tickets platform based on high availability load balancing [J]. Computer Engineering and Design, 2009, 30 (23): 5321-5324 (in Chinese). [覃伟, 何新华. 基于高可用性负载均衡的 Web 票务平台研究 [J]. 计算机工程与设计, 2009, 30 (23): 5321-5324.]
- [4] XUE Zhi. Research and design of performance tuning and high availability in e-commerce web platform [D]. Shanghai: Shanghai Jiao Tong University, 2007: 41-45 (in Chinese). [薛质. 电子商务平台的性能优化和高可靠性研究与实现 [D]. 上海: 上海交通大学, 2007: 41-45.]
- [5] ZHU Lianzhang, TIAN Chao. Performance testing and analysis of improving web application performance method [J]. Computer Engineering and Design, 2008, 29 (7): 1817-1823 (in Chinese). [朱连章, 田超. 改进 Web 应用性能方法及性能测试分析 [J]. 计算机工程与设计, 2008, 29 (7): 1817-1823.]
- [6] HUANG Libin, KOU Yanan. Design of content filtering firewall based on semantics dependency [J]. Computer Engineering and Design, 2011, 32 (5): 1597-1608 (in Chinese). [黄利斌, 寇雅楠. 基于依存句法的网页内容防火墙设计 [J]. 计算机工程与设计, 2011, 32 (5): 1597-1608.]
- [7] WANG Yu, HOU Zhengfeng. The application of cache technology to on-line examination system [J]. Natural Science Journal of Harbin Normal University, 2011, 27 (2): 59-63 (in Chinese). [王瑜, 侯整风. 缓存技术在在线考试系统中的应用 [J]. 哈尔滨师范大学自然科学学报, 2011, 27 (2): 59-63.]
- [8] ZHAO Yuwei. The application and research of cache technology based on WWW [D]. Wuhan: Wuhan University of Technology, 2006: 16-17 (in Chinese). [赵玉伟. WWW 中缓存机制的应用研究 [D]. 武汉: 武汉理工大学, 2006: 16-17.]
- [9] XIA Chunzhong. Design and implementation of lightweight enterprise content management system [J]. Computer Engineering and Design, 2007, 28 (17): 4233-4240 (in Chinese). [夏纯中. 轻量级企业内容管理系统的设计与实现 [J]. 计算机工程与设计, 2007, 28 (17): 4233-4240.]
- [10] GUO Min. Optimization principles of SQL in ORACLE database [J]. Computer Systems and Applications, 2010, 19 (4): 165-173 (in Chinese). [郭岷. ORACLE 数据库 SQL 优化原则 [J]. 计算机系统应用, 2010, 19 (4): 165-173.]