

基于分布式高可用集群的网购系统优化

焦宇, 李民, 王欢, 余开朝

(昆明理工大学机电工程学院, 云南昆明 650500)

摘要: 当前,人们的生活离不开网购,传统的网购系统仍然存在单机容量较低、无法应对高并发及缓存不足等问题,而提高网购系统容量、抗并发能力并强化缓存可以有效提高数据查询效率。对传统单机模式下的网购系统进行优化重构,设计一种基于分布式集群的高可用框架,采用Nginx访问服务器实现集群的高可用并提升抗并发能力,通过Nginx lua调用Redis集群缓存的方式降低客户对服务器的访问压力并提高数据读取效率。最后进行压测实验,通过观察平均响应时间及最大吞吐量参数可知,相比传统系统,优化后的系统在1 000、1 500、2 000并发量下平均响应时间分别减少72ms、151ms、135ms,吞吐量分别提高624/s、1 274/s、1 062/s,进一步验证了优化后的系统在抗并发及数据提取性能上有明显提升。

关键词: 分布式部署;负载均衡;Redis集群;Nginx lua缓存;性能压力测试

DOI: 10.11907/rjdk.212017

开放科学(资源服务)标识码(OSID):



中图分类号: TP302

文献标识码: A

文章编号: 1672-7800(2022)006-0183-05

Optimization of Online Shopping System Based on Distributed High Availability Cluster

JIAO Yu, LI Min, WANG Huan, YU kai-chao

(School of Mechanical and Electrical Engineering, Kunming University of Technology, Kunming 650500, China)

Abstract: Nowadays, people's life is inseparable from online shopping. However, the traditional online shopping system still has some problems, such as low single machine capacity, unable to cope with high concurrency and insufficient cache. Therefore, improving the capacity of online shopping system, anti concurrency ability and strengthening cache can effectively improve the efficiency of query data. Therefore, optimize and reconstruct the online shopping system under the traditional stand-alone mode, designs a high availability framework based on distributed cluster, uses nginx access server to realize the high availability of the cluster and improve the anti concurrency ability, reduces the access pressure of customers to the server and improves the data reading efficiency by calling redis cluster cache through nginx Lua. Finally, the pressure measurement experiment is carried out. By observing the average response time and maximum throughput parameters, it shows that compared with the traditional system, the average response time of the optimized system under 1 000, 1 500 and 2 000 concurrency is reduced by 72ms, 151ms and 135ms respectively, and the throughput is improved by 624/s, 1 274/s and 1 062/s. It is further verified that the optimized system has obvious improvement in anti concurrency and data extraction performance.

Key Words: distributed deployment; load balancing; Redis cluster; Nginx Lua cache; performance stress test

0 引言

随着互联网技术的迅速发展,人们与网络的关系日益密切,互联网带给人们一种全新的生活方式,如网上购物、

滴滴打车、学习网课等都为人们的生活带来了全新的体验,而近些年的网上购物更是发展得如火如荼^[1]。网购系统的出现极大地改善了人们的购物体验,虽然不能完全替代线下购物,但已逐渐成为人们主流的购物方式。由此,对网购系统设计的要求随之提高,如遇到某些购物活动,大

收稿日期: 2021-08-07

基金项目: 云南省智能化自动化产业发展研究项目(YNDR2017G1C06)

作者简介: 焦宇(1997-),男,昆明理工大学机电工程学院硕士研究生,研究方向为企业信息化及系统工程;李民(1973-),女,博士,昆明理工大学机电工程学院副教授、硕士生导师,研究方向为质量管理;余开朝(1962-),男,昆明理工大学机电工程学院教授、硕士生导师,研究方向为生产及制造系统工程;王欢(1995-),女,昆明理工大学机电工程学院硕士研究生,研究方向为数据挖掘。本文通讯作者: 李民。

量客户进入系统进行购物,传统的单机部署式网购系统已经无法承受如此高的并发量,会发生提取数据缓慢甚至宕机的情况,影响人们的购物体验。因此,对网购系统进行分布式拓展及缓存优化具有重要意义。

1 相关研究

分布式计算技术最早由 OMG (Open Management Group) 组织于 1992 年提出,这一技术的出现很大程度上提高了分布式系统的开发效率。随着互联网与网络技术的迅速发展和广泛应用,Sun 公司和 Microsoft 公司分别推出了应用于 B/S 架构的 J2EE 开发应用平台和面向 B/S 应用的 .NET 开发应用平台^[2-3]。

相比于传统单机部署的企业级项目,分布式部署项目方式的出现能够极大地帮助企业提高系统的稳定性、扩展性和并发性。人们接触的架构通常是从简单到复杂、从单一到复合不断改进的过程。随着互联网技术的不断发展,可以在分布式架构中加入负载均衡算法、Redis 数据库等技术以实现项目的优化。李效利等^[4]通过分布式网络架构提高了监测平台的一体化管理;陈鹏炜^[5]运用 openresty 及其负载均衡策略设计了一个集群实名鉴权系统,相比传统系统提高了性能;李晓东^[6]采用 ssm 框架和 Nginx 负载均衡策略缓解了 Mysql 数据库的读取压力;孔祥真等^[7]为了解决网络服务器中高流量不稳定的问题,部署 Nginx 和 tomcat 服务器以提供一个高性能的服务器解决方案。这些学者在运用 Nginx 负载均衡时仅仅考虑了应对并发时的服务器压力问题,面对数据存储效率问题也只是简单地采用本地 Mysql 数据库,而面对大量人群访问数据时,其数据提取性能并没有得到有效改善。

在分布式集群中加入 Redis 缓存的方式同样可以提高系统性能。李彦辰等^[8]通过设计 Redis 集群的分布式搜索方法提高了连接分析性能;陈清^[9]为了应对大型机电设备数据交换时出现的问题,将 Redis 缓存运用其中,提高了数据查询速度;Li 等^[10]运用改进的一致性哈希算法进行过滤系统设计,提高了 Redis 集群的可用性等性能。这些研究通过 Redis 集群或者算法优化方式提高了数据提取性能,然而却忽略了大量人群访问服务器时的压力问题,当服务器因为压力过大发生宕机问题时,数据获取将变得十分困难。

针对以上不足,本文提出了一种新的分布式高可用集群,对之前单一的 Nginx 策略或者 Redis 集群进行改进,将 Nginx 负载均衡策略、Redis 哨兵集群相结合,并且加入最新的 Nginx lua 缓存技术,改善了之前在实现高可用时数据提取效率问题,以及在数据提取时所忽略的服务器压力过大问题,既保证了集群在高并发情况下的正常运作,又提升了数据提取性能。最后以传统本地部署的网购系统为背景,结合新建的集群进行试验验证,通过得到的参数证

明该集群可提高数据提取能力和抗并发能力。

2 关键技术分析

2.1 Nginx 与 Lua 缓存

OpenResty 是一个基于 Nginx 与 Lua 的高性能 Web 平台,其内部有精良的 Lua 库、第三方模块等依赖项。它可以用来搭建能够处理高并发、扩展性极高的动态 Web 应用、Web 服务和动态网关^[11-13]。Nginx 是一款高性能的 HTTP 服务器/反向代理服务器及电子邮件(IMAP/POP3)代理服务器,最高可以承受 5 万的并发量,而对 CPU、内存等资源消耗却非常低,运行十分稳定^[14-16]。此外,Nginx 在作为 Web 服务器、动态分离服务器,以及反向代理服务器时都发挥着重要作用。在反向代理方面,Nginx 实现的负载均衡算法主要有轮询、轮询权重、ip_hash、url_hash(第三方)、fail(第三方)等。Nginx lua 是基于 Nginx 协程机制的一种缓存方式,针对用户想要获取的内容,使用 lua 脚本的方式在 Nginx 上完成对应业务代码的处理逻辑,可以避免访问 Java 服务器。

2.2 Redis 数据库

Redis 是一个开源的高性能键值对(key-value)数据库,根据官方测试得出 50 个并发执行 10 000 个请求,读的速度是 11 000 次/s,写的速度是 81 000 次/s^[17],且 Redis 可以提供多种键值数据类型如:字符串类型、哈希类型、列表类型等。Redis 是一个缓存的数据库中间件,可以将它设置为数据刷新到磁盘中的策略,可以支持当系统对一定数量 key 产生 set 操作变化时即刷新一次磁盘,也可以设置每个 1s 或 2s 轮询的方式刷新对应的磁盘。因此,Redis 磁盘具备了存储数据库的能力,但会丢失一定数量的数据,可知 Redis 也是一个易失性的数据存储。Redis 除读写高效外,还可以利用其搭建多节分布式集群从而提高数据读取效率,同时也极大地提高了系统性能。

3 分布式高可用集群搭建

目前,网购系统功能趋于稳定,对客户而言主要以订单功能、登录功能以及商品详情页模块为主,可以将其拆分放在不同的服务器中,降低在同一时间多客户访问时的访问压力;其次可以降低耦合度,有利于后期工程师对网购系统的功能维护及功能添加。

3.1 项目垂直拆分与水平拆分

项目垂直拆分是指按照项目的不同应用功能进行拆分,将系统不同的功能拆分到不同的服务器中,各功能之间独自运行,互不影响,同时提高系统抗并发能力,具体部署如图 1 所示。

项目水平拆分是指为了提高后期项目维护效果,按照业务对系统进行拆分。例如,本系统的业务代码层可以分

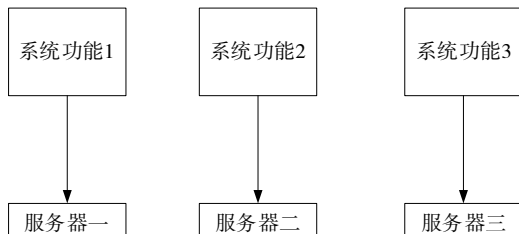


Fig. 1 Vertical split

图 1 垂直拆分

为 Controller 层、dao 层、service 层、dataobject 层、model 层、接口层,从而降低代码之间的耦合度,如图 2 所示。

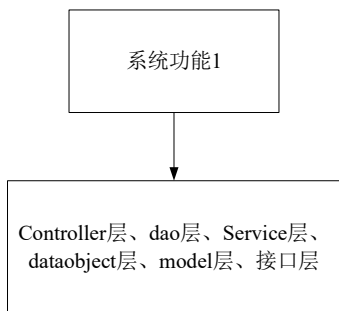


Fig. 2 Horizontal split

图 2 水平拆分

3.2 分布式高可用集群设计

系统优化主要是在高并发优化及数据提取效率方面,采用 Openresty 平台结合 Redis 数据库搭建集群框架。该网购系统采用前后端分离的方式进行设计,前端运用 html5 标准进行编写,因此首先将 Nginx 作为静态的 Web 服务器使用,然后将其作为动静分离的服务器,对应的 Nginx 作反向业务代理后,可以将对应的静态请求依旧路由在本地的 html 文件中,以静态资源请求的方式返回给前端,之后依赖反向代理服务,将动态请求返回到后端,以完成对应的动态请求代理,以 Ajax 请求的方式返回给前端固定的 Json 参数,实现动静分离的服务器使用。Nginx 采用轮询的负载均衡方式,将客户端的请求发送给 Tomcat 服务器。例如,用户在获取商品详情页并提取数据时,会将其请求发送到某一台 Tomcat 上,首先从 Redis 中提取数据,若 Redis 中无数据,则从 Mysql 数据库中进行提取,并且查出数据后将数据缓存到 Redis 中,这样用户在进行第二次查询时便可从 Redis 缓存中直接查询数据,提高了用户对数据的查询效率,如图 3 所示。

3.3 Redis 集群设计

单机部署项目通常部署的是单机版的 Redis,其弊端在于对应 Redis 的单点问题瓶颈难以处理,若对应单机节点的 Redis 中断,则所有的业务操作都会消失,且单机版有容量上限,无法满足高效存储要求。采取 sentinel 的哨兵模式可以很好地解决该问题:引入 Redis sentinel 哨兵机制,假设有两台 Redis 服务器,将 Redis2 作为 Redis1 的从机,Redis 支持主从同步模式,Redis1 可以将数据同步给 Redis2。理想情况下,当网购系统服务器探测到 Redis1 出

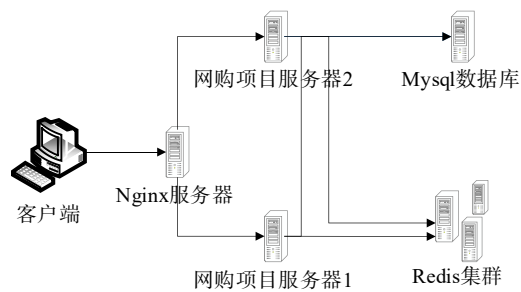


Fig. 3 Distributed high availability cluster

图 3 分布式高可用集群

问题时,可以自动切换到 Redis2。然而探测 Redis1 是一个繁杂的过程,因为对应的分布式环境十分复杂,无法明确 Redis1 是否为宕机状态,并且它需要知道切换到哪一台备机上。因此,引入哨兵机制,它与 Redis1、Redis2 都建立了长连接,并且是一个心跳机制,Redis sentinel 清楚地知道 Redis1 和 Redis2 是处于哪种状态,当项目启动时无需感知 Redis1 和 Redis2,只需询问 Redis sentinel 即可知道需要连接哪一台服务器。Redis sentinel 是一个单点机器,因此可以完全确定 Redis1 和 Redis2 哪一台是主,它可以将 Redis1 指定为 master,将 Redis2 指定为 slave,一旦确定 Redis1 位于 master 后,项目服务器就会连接 Redis1 做一个 get、set 操作,当 Redis1 产生异常后,心跳机制就会被破坏掉,Redis sentinel 将立刻做一次切换,将 Redis2 指定为 master,Redis1 指定为 slave,此时 Redis sentinel 会通知项目服务器产生了变化,所部属项目重新触发询问流程,会将 get、set 操作改变到 Redis2 去。哨兵机制很好地解决了当用户提取数据时,一台 Redis 服务器出现问题时,用户无法获取信息的问题。本文采取的是开启 2 台 Redis 主机、两台 Redis 从机的方式实现 Redis 集群,如图 4 所示。

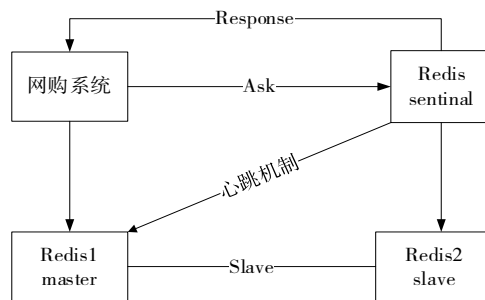


Fig. 4 Redis cluster

图 4 Redis 集群

3.4 Nginx Lua 缓存强化数据缓存性能

尽管采用了 Redis 集群部署,但是面对大量访问涌入时,服务器压力仍然较大,若将 Nginx 与 Redis 集群联系起来,则对数据缓存性能有所提升。例如,当用户发起访问商品详情页请求时,Nginx 收到请求后不会直接链路到后端部署的两台项目服务器,它直接连到 Redis 从机上,进行只读不写的操作,同时分担了 Redis 主机的负载,若 Redis 中没有指定数据,就回源到项目服务器上,项目服务器也

对 Redis 中的数据进行判断,若还是没有,则回源到 MySQL 数据库中进行数据提取,并且放入 Redis 中,则下一次客户将对请求发送到 Nginx 上时就可以直接通过 Redis 进行“读”的操作。将 Lua 缓存 Redis 主从搭配方式相结合,可以从容应对大流量的数据,提高数据提取效率。

3.5 集群启动策略

本文将网购系统及 Nginx、Redis 集群部署到多台虚拟机 Linux 系统中,开启访问权限和防火墙,运用 Xshell 文件实现本地 Windows 系统与 Linux 系统的通信连接,从而可以进行对项目的运行操作,如图 5 所示。

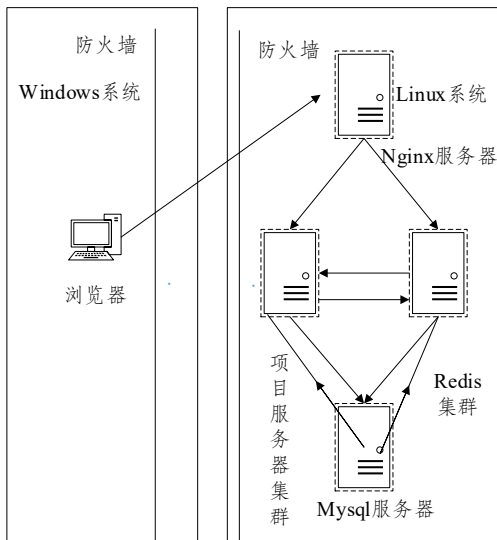


Fig. 5 Project server cluster deployment

图5 项目服务器集群部署

4 项目运行与压测实验

4.1 项目运行实验

将本地的网购项目打成 jar 包,分别传输至 2 台 Vmware 虚拟机中。将两台服务器静态 IP 地址分别设置为 192.168.157.138 和 192.168.157.139,对应端口号都设置为 8 090 端口;再将 Openresty 部署到第三台虚拟机中,设置静态地址为 192.168.157.140,点开 nginx.conf 文件完成监听端口号、域名 ip、负载均衡轮询方式等一系列设置。Redis 集群采用 2 主 2 从的方式,将其部署到 4 台虚拟机中,分别设置好静态 IP 地址,并且开启端口分别为 6 500、6 501、6 502 和 6 503,通过测试,集群可以正常启动。再进入 Openresty 服务器中,新建 itemredis.lua 文件,在其中设置 lua 调用 Redis 的脚本,同时修改 nginx.conf 文件,使其运行时可以启动相应的脚本。部署完所有项目后开启项目进行压力测试。

本次测试主要验证项目优化后的抗并发能力及数据提取能力,采用 jmeter 压测工具进行测试。实验测试环境为:i7 10 代/16G 内存、Windows10 系统、JDK1.8、Redis4.0、Tomcat8.5、Mysql5.7。

4.2 压测实验结果

Jmeter 是基于 Java 的压力测试工具,可以用来测试服务器在不同压力下的强度与性能^[18-19]。使用该测试工具,分别设置测试用户在 1 000、1 500、2 000 时的并发量;启动线程时间为 10s,循环次数 15 次,进行压力测试。具体通过观察数据提取的平均值、中位数、90% 线位、错误率以及吞吐量对比出原始网购系统与优化后网购系统的性能。图 6—图 8 分别表示在 1 000、1 500、2 000 并发量时的压测结果。

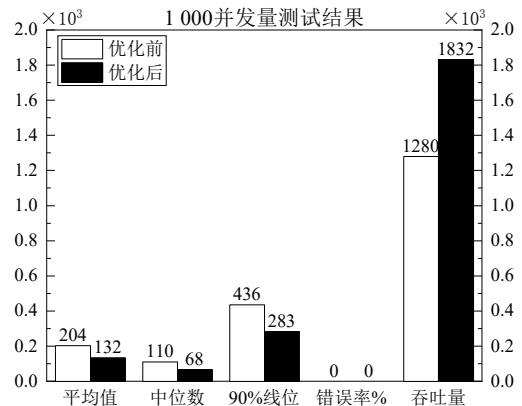


Fig. 6 1 000 concurrency comparison results

图6 1 000 并发量对比结果

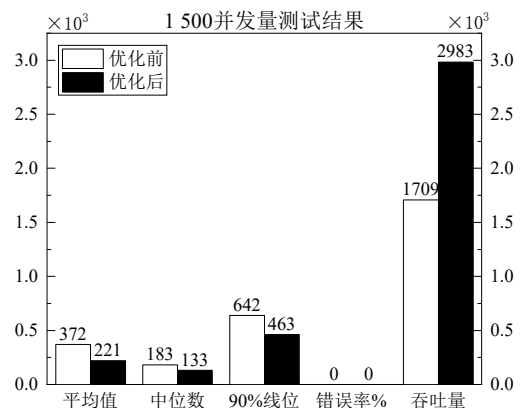


Fig. 7 1 500 concurrency comparison results

图7 1 500 并发量对比结果

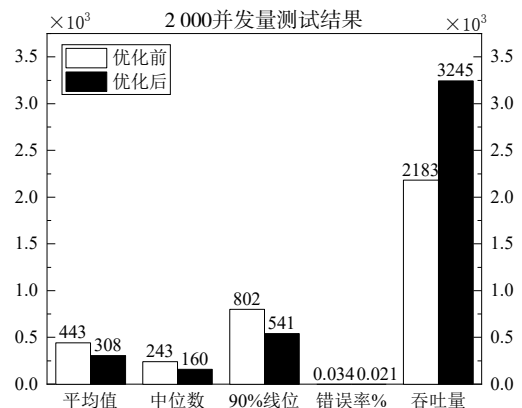


Fig. 8 2 000 concurrency comparison results

图8 2 000 并发量对比结果

测试结果表明,与传统本地部署的网购系统相比,优化后的网购系统在吞吐量、数据响应时间上有明显性能提升,在并发量为1 000的情况下,吞吐量相比增加624/s、平均响应时间减少72ms;并发量为1 500的情况下,吞吐量相比增加1 274/s、平均响应时间减少151ms;并发量为2 000的情况下,吞吐量相比增加1 062/s、平均响应时间减少135ms。错误率方面,并发量为1 000、1 500时错误率都为0,在并发量为2 000时,优化后系统的错误率仍然低于之前的错误率。

5 结语

本文针对传统本地部署的网购系统进行了项目优化及重构,通过将项目部署到虚拟机服务器中,并且运用Nginx、Redis集群及Lua缓存等技术,构建出一个分布式高可用的集群以支撑该系统。通过压力测试验证所建集群在控制错误率的情况下,该系统的并发能力及数据处理效率都有一定提高,可以更好应对高并发情况下产生的问题。然而,本文并没有考虑如何应对类似“双十一”活动的瞬时并发问题,这有待进一步研究解决。

参考文献:

- [1] DONG X W, ZHAO Y Y, ZHANG Y M. Design and implementation of e-commerce seckill system [J]. Information Technology and Informatization, 2020, 45(9): 40-42.
董晓玮,赵月英,张一鸣. 电商秒杀系统的设计与实现[J]. 信息技术与信息化, 2020, 45(9): 40-42.
- [2] TANG H, JIANG C Y, TIAN J F. Research on distributed database technology and development trend [J]. Information and Communication Technology and Policy, 2020, 46(11): 86-90.
唐浩,姜春宇,田稼丰. 分布式数据库技术及发展趋势研究[J]. 信息技术与政策, 2020, 46(11): 86-90.
- [3] XU G Y. Research and implementation of online mall based on Dubbo distributed architecture [D]. Shenyang: Shenyang Normal University, 2019.
徐光耀. 基于Dubbo分布式架构网上商城的研究与实现[D]. 沈阳: 沈阳师范大学, 2019.
- [4] LI X L, WANG Z X, YU G. Application of cluster distributed network architecture in short wave monitoring integrated platform [J]. Digital Communication World, 2021, 17(6): 55-56, 99.
李效利,王志欣,于滢. 集群分布式网络架构在短波监测一体化平台中的应用[J]. 数字通信世界, 2021, 17(6): 55-56, 99.
- [5] CHEN P W. Design and implementation of real name authentication system based on load balancing and caching technology [D]. Chengdu: University of Electronic Science and Technology of China, 2017.
陈鹏炜. 基于负载均衡与缓存技术的实名鉴权系统设计与实现[D]. 成都: 电子科技大学, 2017.
- [6] LI X D. Research and development of social systems supporting high concurrency [D]. Beijing: Beijing University of Technology, 2020.
李晓东. 支持高并发的社交系统的研究与开发[D]. 北京: 北京工业大学, 2020.
- [7] KONG X Z, ZHANG D, LI Z Y. Application of Linux load balancing cluster technology in network server [J]. Software Guide, 2016, 15(12): 144-147.
孔祥真,张丁,李忠远. Linux负载均衡集群技术在网络服务器中的应用[J]. 软件导刊, 2016, 15(12): 144-147.
- [8] LI Y C, AI Q Z, WANG S F. Research on Redis based distributed search engine [J]. Software Guide, 2018, 17(3): 201-204.
李彦辰,艾庆忠,王少非. 基于Redis的分布式搜索引擎研究[J]. 软件导刊, 2018, 17(3): 201-204.
- [9] CHEN Q. Networking architecture of large electromechanical equipment in coal mine based on Redis [J]. Industrial and Mining Automation, 2020, 46(10): 109-113.
陈清. 基于Redis的煤矿大型机电设备联网架构[J]. 工矿自动化, 2020, 46(10): 109-113.
- [10] LUO B Z, ZHU W J, LI P, et al. Cluster-based distributed dynamic cuckoo filter system for Redis [J]. International Journal of Parallel, Emergent and Distributed Systems, 2020, 35(3): 340-353.
- [11] YUN Z, YONG Q H, GUO S. Research on the construction of high available load balancing cluster based on LVS [C]// Dalian: 2nd Workshop on Advanced Research and Technology in Industry Applications, 2016.
- [12] LIU Q Y, LU J C, FENG Y Q. Application of openresty in bidirectional CA system [J]. Cable TV Technology, 2019, 16(6): 91-93.
刘巧艳,陆嘉程,冯燕强. OpenResty在双向CA系统中的应用[J]. 有线电视技术, 2019, 16(6): 91-93.
- [13] LI Y J. Research on distributed operation mode of Internet of Things [J]. Journal of Physics: Conference Series, 2021, 1827: 012129.
- [14] XUE J, LI Z Z, WANG Y L. Development of load balancing technology [J]. Small Microcomputer System, 2003, 24(12): 2100-2103.
薛军,李增智,王云岚. 负载均衡技术的发展[J]. 小型微型计算机系统, 2003, 24(12): 2100-2103.
- [15] MA Y L. Research on Nginx load balancing technology [D]. Chongqing: Chongqing University of Posts and Telecommunications, 2016.
马原龙. Nginx负载均衡技术研究[D]. 重庆: 重庆邮电大学, 2016.
- [16] WANG L P. Research and improvement of load balancing technology based on Nginx server cluster [D]. Ji'nan: Shandong University, 2015.
王丽萍. 基于Nginx服务器集群负载均衡技术的研究与改进[D]. 济南: 山东大学, 2015.
- [17] JIN Y. Research on content management system based on cache technology [D]. Changchun: Jilin University, 2014.
靳莹. 基于缓存技术的内容管理系统研究[D]. 长春: 吉林大学, 2014.
- [18] CHEN G, XIE J J, ZHOU X J. Design and implementation of automated testing for large management information systems [J]. Frontier of Data and Computing Development, 2021, 3(2): 77-85.
陈果,谢建军,周小军. 面向大型管理信息系统的自动化测试设计与实现[J]. 数据与计算发展前沿, 2021, 3(2): 77-85.
- [19] MEMON P. Scrutinizing automated load testing via Blazemeter, Apache Jmeter and HP Loadrunner [J]. International Journal of Advanced Studies in Computer Science and Engineering, 2018, 7(3): 35-42.

(责任编辑:孙娟)