

DOI:10.19651/j.cnki.emt.2106257

基于 OpenStack 的开源软件仓库设计与实现

王军阵 杨家辉 潘佳奇 丁光帅 许成朗
(中国卫星海上测控部 江阴 214400)

摘要: 针对远洋航天测量船逐步地部署自主可控设备后,在海上执行任务期间,卫星通信带宽资源有限,无法满足海上测控任务、科学研究和随行试验等对开源软件需求增大的问题,设计了一种基于云计算平台的开源操作系统软件仓库,Python 软件仓库和 git 分布式管理仓库。通过构建 OpenStack 平台,使用 apt-mirror 构建了 Debian/Ubuntu 操作系统软件仓库,使用 yum-utils 构建了 Fedora/CentOS 软件仓库,通过 pip2pi 设计了 Python 软件仓库,并使用 gitlab 搭建了 git 分布式管理服务,结果为各操作系统数据量均在 55 GB 以上,Python 数据量达 150 GB,gitlab 共享数据量也达到了 50 GB,海上离线模式传输速率大于 8 MB/s,在远航期间开源软件仓库得到了可观的访问量,其中 gitlab 达到了最高的 8 次/日的日均访问量。结果表明,通过模块化云平台搭建的开源软件仓库,操作简单,灵活性强,扩展性好,可满足测量船长期出海期间对开源软件的下载需求与协同工作。

关键词: 云计算;软件仓库;OpenStack

中图分类号: TP315 **文献标识码:** A **国家标准学科分类代码:** 520.4060

Design and implementation of open source software repository based on OpenStack

Wang Junzhen Yang Jiahui Pan Jiaqi Ding Guangshuai Xu Chenglang
(China Satellite Maritime Tracking and Control Department, Jiangyin 214400, China)

Abstract: After ocean space survey ship gradually deploys controllable equipment with independent property rights, in order to solve the problem that satellite communication bandwidth resources are limited during missions at sea, and cannot meet the increasing demand for open source software such as maritime measurement and control missions, scientific research and accompanying experiments, a kind of open source operating system software repository, Python software repository and git distributed management repository are designed based on the cloud computing platform. By building the OpenStack platform, the Debian/Ubuntu operating system software repository is built by using apt-mirror, the Fedora/CentOS software repository is also built by using yum-utils, the Python software repository is designed by pip2pi, and the git distributed management service is established using gitlab. The result is that the data volume of each operating system is more than 55 GB, the Python data volume is 150 GB, the gitlab shared data volume has also reached 50 GB, and the offline mode transmission rate is greater than 8 MB/s on the sea, during the voyage, the open source software repository has received considerable access, among them, gitlab has reached the highest average daily visit volume of 8 times per day. The results show that the open source software repository built through the modular cloud platform is simple to operate, flexible, and extensible, and can meet the requirements for downloading open source software and collaborative work during the long-term seagoing of the survey ship.

Keywords: cloud computing; software repository; OpenStack

0 引言

随着信息技术的快速发展,云计算、大数据和人工智能技术在各行业不断渗透,其中,开源软件起着关键的作用,特别是在科学研究和工程实践等领域^[1-7]。清华大学、中国

科技大学和上海交通大学等国内高校都构建有开源软件仓库镜像为科学研究提供必要的研究工具,而云计算技术的应用拓展和人工智能的蓬勃发展^[8-17],也促使国内华为、阿里和网易等企业开始构建开源软件仓库,供工程技术人员快速部署工程应用环境。

收稿日期:2021-04-01

远洋航天测量船在出海期间,任务、科研和随行试验等活动交替开展,对开源软件的需求逐年增大,测量船主要通过卫星通信实现网络数据的传输,但出海时间长,卫星通信带宽资源有限,软件需求量大,一直困扰着测量船上的科研和工程技术人员。特别是在完成设备升级改造,更多地采用自主可控设备之后,对开源软件的依赖程度更是越来越高,因此,非常有必要在测量船上构建开源软件仓库,来满足长期在海上执行任务期间对开源软件的需求。

高校和企业主要通过 RSYNC 技术实现开源软件的仓库构建,但是这种技术需要与部署在国外的主服务器仓库进行软件同步,同步速度慢,通常情况下只有几十 K,且国内的镜像仓库也没有启动 RSYNC 服务,不支持 RSYNC 技术^[18],所以在测量船上搭建开源软件仓库时,需要通过其他技术实现。

本文使用 OpenStack 构建了一个云计算平台,基于云计算平台,给出了 Debian10/Ubuntu18.04LTS, Fedora31/CentOS8 操作系统软件镜像仓库的构建方法,基于 Ubuntu18.04LTS 镜像实例,使用 pip2pi 构建了 Python 软件的 pip 仓库,基于 Ubuntu18.04LTS 镜像实例构建了 gitlab 分布式版本控制服务,并验证了每一种服务的功能,实现了海上离线条件下操作系统仓库的离线服务,Python 仓库离线服务和 git 分布式开源软件分享平台。

1 系统结构

基于云计算的虚拟化平台服务框架正在逐渐替换传统的信息技术服务模式,云计算已经成为信息技术和互联网技术发展的主流趋势,在开源云计算技术领域中,OpenStack 以其模块化程度高,可扩展性好,可以快速构造公有云和私有云平台而成为关注焦点。OpenStack 提供了一整套云计算平台的组件工具集,根据实际需求,可以便捷地以组件为单位实现云计算平台的迅速部署。本文以 6 个物理节点为基础,在各节点上安装 Ubuntu18.04LTS 操作系统,将控制节点、网络节点、计算节点、块存储节点和两个对象节点分别部署在 6 个物理节点之上,实现 OpenStack 云计算基础架构的构建,在 OpenStack 云计算平台上根据需求可以方便快速地搭建系统服务,针对测量船出海期间的开源软件需求,创建 Debian10、Ubuntu18.04LTS、Fedora31 和 CentOS8 实例用于构建对应操作系统的软件仓库镜像,然后,创建两个 Ubuntu18.04LTS 实例,其中一个采用 pip2pi 构建 Python 软件的 pip 仓库镜像,另一个使用 gitlab 构建 git 分布式开源软件管理共享平台,其组成结构如图 1 所示。

2 系统设计与功能验证

2.1 云计算平台的设计

OpenStack 云计算平台属于基础设施即服务(infrastructure as a service, IaaS), OpenStack 常被称作新

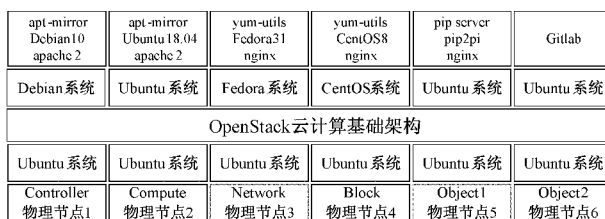


图 1 系统架构

一代的云操作系统,如同 Linux 一样,其旨在构建一个内核,构建面向全球开发者的云计算生态系统。OpenStack 当前包括的主要组件有计算服务 Nova、网络服务 Neutron、对象存储服务 Swift、块存储服务 Cinder、身份和访问管理服务 Keystone、仪表盘 Horizon、镜像服务 Glance 等,其中 Nova 主要用于计算资源生命周期管理,Neutron 提供云计算环境下的虚拟网络功能,Swift 用于永久类型的静态数据的长期存储,Cinder 用于管理计算实例所使用的块级存储,Keystone 提供了用户信息管理,为其他组件提供认证服务,Horizon 用以管理、控制 OpenStack 服务的网页控制面板,Glance 提供虚拟机镜像的发现、注册和获取服务。采用 OpenStack 的 train 稳定版本,通过规划内部通信地址和外部通信地址,OpenStack 各组件在 6 个节点的部署规划如图 2 所示。

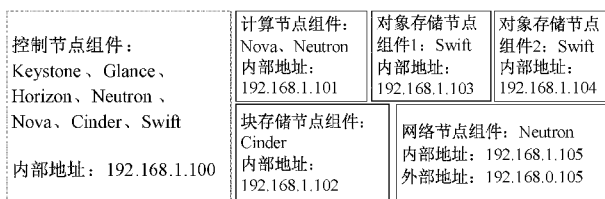


图 2 各节点的组件部署

在各物理节点上安装好 Ubuntu 操作系统之后,需要配置操作系统的网络地址,配置时钟同步,开启时钟同步服务,设置防火墙策略,安装必要的软件工具和配置相应的基础服务,根据图 2 的组件部署规划,在各物理节点上依次安装数据库、rabbitmq、Keystone、Glance、Neutron、Dashboard、Cinder 和 Swift 等相应的服务,具体构建流程如图 3 所示,成功安装之后通过 Horizon 访问管理员登陆界面的部分截图如图 4 所示。

2.2 操作系统软件仓库设计

随着 Linux 技术的日渐成熟,Linux 系统在越来越多的领域得到了普及和应用,通常情况下,Linux 系统的更新及软件包的安装均是由系统内建的管理工具下载官方网站或国内镜像网站上的软件包来完成。操作系统的软件仓库(repository)是一个预先准备好的目录或一个网站,包含了软件包和索引文件。软件管理工具可以在仓库中自动地定位并获取正确的软件包,这样就不必手动搜索和安装新应用程序和升级补丁,只需要一个命令,就要以安装所需要的软件,并可解决软件安装过程中的依赖关系,还可以根据指

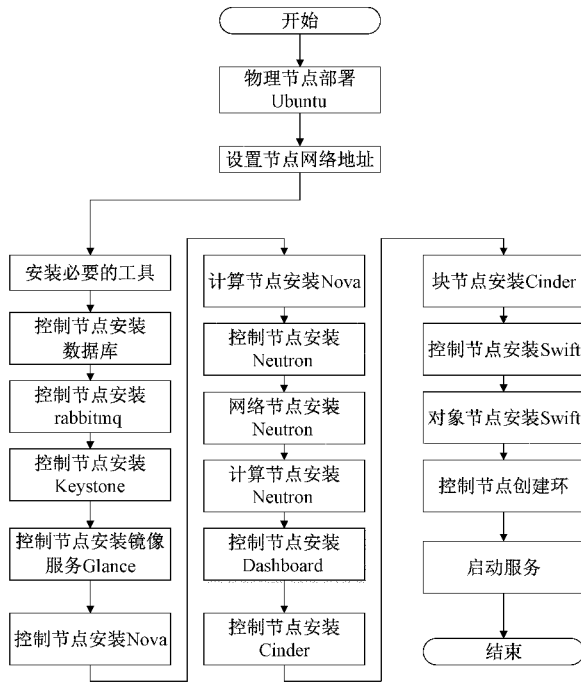


图 3 OpenStack 的软件配置流程



图 4 OpenStack 的 Web 界面

定搜索目录来查找安装新软件。在 Debian 和 Ubuntu 操作系统中,使用 apt 方式管理软件包,而在 Fedora 和 CentOS 中则使用 yum 方式管理软件包。

1) Debian/Ubuntu 系统仓库的设计

Debian/Ubuntu 系列操作系统的软件包管理,与国产麒麟操作系统的开源版本兼容,值得一提的是,基于 ARMv8 指令集的 64 位国产麒麟操作系统也开始使用 Debian 系列的软件包管理方式。在 OpenStack 中添加获取到的 Ubuntu/Debian 系统镜像文件,创建对应的实例,添加浮动地址用于外网访问,配置国内源用于快速下载仓库软件数据,在 Debian/Ubuntu 操作系统下安装 apt-mirror 软件镜像工具,配置 apt-mirror 配置文件,规划 apt

服务目录,下载服务相关软件,使用 apache2 通过 http 协议构建软件仓库离线服务,具体的构建流程如图 5 所示,构建成功之后,通过 Ubuntu 客户端测试的结果如图 6 所示。

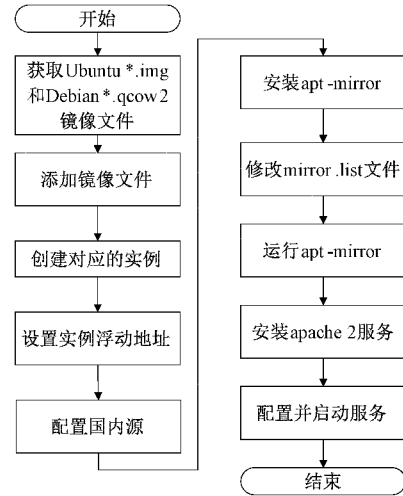


图 5 Debian/Ubuntu 操作系统仓库设计流程

```
joe@apt-test:/etc/apt$ sudo apt install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  ctags vim-doc vim-scripts
The following NEW packages will be installed:
  vim
0 upgraded, 1 newly installed, 0 to remove and 51 not upgraded.
Need to get 1,152 kB of archives.
After this operation, 2,852 kB of additional disk space will be used.
Get:1 http://192.168.0.202/ubuntu bionic-updates/main amd64 vim amd64
2 kB]
Fetched 1,152 kB in 0s (10.5 MB/s)
```

图 6 Ubuntu18.04LTS 客户端测试

2) Fedora/CentOS 系仓库的设计

Fedora/CentOS 系列操作系统的软件包管理,与国产麒麟操作系统的商业版本兼容,都使用了 RedHat 公司的软件包管理方式,由于云计算技术的持续升温,2019 年 RedHat 公司已经被 IBM 收购。在 OpenStack 中添加 Fedora/CentOS 系统镜像文件,创建对应的实例,添加浮动地址,配置国内源等操作之后,在 Fedora/CentOS 操作系统中安装 yum-utils 和 createrepo 软件镜像下载和配置工具,规划服务目录,同步镜像资源,下载相关软件,创建和配置本地镜像,使用 nginx 通过 http 协议构建软件仓库离线服务,构建成功之后,具体的构建流程如图 7 所示,通过 CentOS8 客户端测试的结果如图 8 所示。

2.3 Python 软件仓库设计

Python 是一种解释型脚本语言,广泛应用于 Web 服务开发、科学计算、人工智能等领域。pip 是 Python 的一种包管理工具,可以实现对 Python 软件包的查找、下载和安装等功能。而 pip2pi 可以快速完成服务器端的 pip 服务构建,在 OpenStack 下创建 Ubuntu18.04LTS 实例等操作之后,在操作系统中安装 Python3,并通过 pip 工具安装 pip2pi 等必要软件包,通过 pip2tgz 下载指定 Python 软件包到服务器端,并使用 dir2pi 构建 pip 服务列表,最后通过

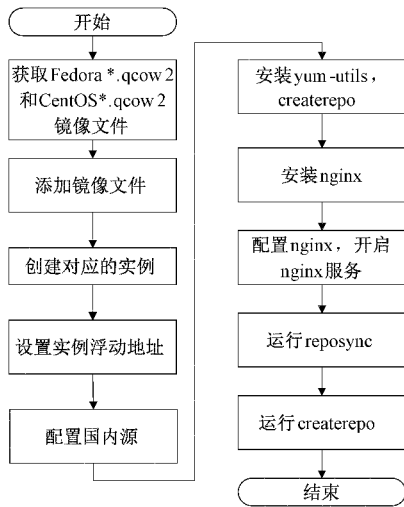


图 7 Fedora/CentOS 操作系统仓库设计流程

```

Downloading Packages:
kernel-devel-4.18.0-80.11.2.el8_0.x86_64.rpm                23 MB/s
-----
Total                                                       23 MB/s
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
Installing : kernel-devel-4.18.0-80.11.2.el8_0.x86_64
Running scriptlet: kernel-devel-4.18.0-80.11.2.el8_0.x86_64
Verifying  : kernel-devel-4.18.0-80.11.2.el8_0.x86_64

Installed:
kernel-devel-4.18.0-80.11.2.el8_0.x86_64
  
```

图 8 CentOS8 客户端测试

nginx 提供的 http 协议构建 pip 软件仓库离线服务,构建流程如图 9 所示。构建成功之后,客户端通过配置文件将 index-url 指定为构建好的服务器 IP 地址,测试结果如图 10 所示。

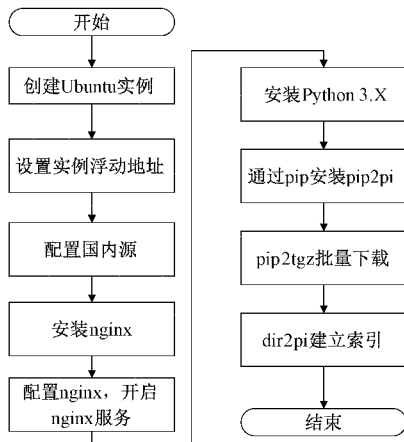


图 9 Python 软件仓库设计流程

2.4 gitlab 服务的设计

Git 是一种流行的版本控制系统,广泛应用于软件开发和共享的协同工作,gitlab 是一个利用 Ruby on Rails 开发的开源应用服务,用于实现一个自托管的 git 项目仓库,可通过 Web 界面进行访问公开或私有项目,它拥有与

```

joe@joe-VirtualBox:~$ pip3 install numpy
Collecting numpy
  Downloading http://192.168.0.207/simple/numpy/numpy-1.18.1-x1_x86_64.whl (20.1MB)
    100% |#####| 20.2MB 84.1MB/s
Installing collected packages: numpy
Successfully installed numpy-1.18.1
  
```

图 10 pip 客户端测试

github 类似的功能,能够浏览源代码,管理缺陷和注释,可以管理团队对仓库的访问,它非常易于浏览提交过的版本并提供一个文件历史库,团队成员可以利用内置的简单聊天程序 (Wall) 进行交流。在 OpenStack 下创建 Ubuntu18.04LTS 实例等操作之后,安装 gitlab 必须的软件 curl、openssh-server、ca-certificates、postfix,安装 gitlab-ce_12.3.6-cc.0_amd64.dcb 软件包,修改/etc 目录下的 gitlab.rb 文件完成服务端配置,设置 HostIP 为指定 IP 地址,并将对应汉化内容复制到 gitlab-rails 目录下,执行 gitlibctl reconfigure 和 gitlibctl start,可以实现服务部署,操作流程如图 11 所示,成功安装的界面如图 12 所示,图 13 所示为基本的 clone 操作。

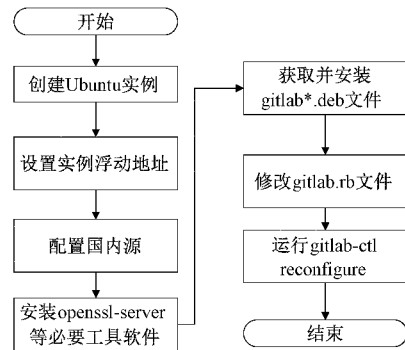


图 11 gitlab 服务搭建流程



图 12 gitlab 项目提交成功界面

```

Cloning into 'l3dd3snull'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.
joe@debian:~/workspace/gitteft/pull$ ls
l3dd3snull
joe@debian:~/workspace/gitteft/pull$ cd l3dd3snull/
joe@debian:~/workspace/gitteft/pull/l3dd3snull$ ls
Makefile  README.txt  snull.c  snull.h

```

图 13 clone 成功的界面

通过在码头数据同步,在海上执行任务的试运行,对具体仓库的数据量、下载速度和日均访问量进行了统计,如表 1 所示,操作系统仓库数据量最大的是 Ubuntu,最小的是 CentOS,这是由于 Ubuntu 是由软件数量较多的 Debian 衍生而来,多用于桌面系统,应用软件丰富,而 CentOS 以稳定著称,软件版本较旧,版本更新缓慢,在服务器端深受欢迎,其应用软件较少,Fedora 主要用于 RedHat 对新软件和新技术的测试探索,其软件也很丰富,通常应用于桌面系统,但其系统稳定性较差。第 2 列给出了仓库软件的下载速度,Debian 和 Ubuntu 的下载速度远慢于 Fedora 和 CentOS,这是由于在设计之初 Debian/Ubuntu 采用的是 apache2 服务,Fedora/CentOS 采用的是 nginx,Python 仓库也采用了 nginx,通常情况下 nginx 服务的性能优于 apache2,gitlab 速度不理想的主要原因是其上存储的文件多为文本文件。从日均访问次数可以看出,访问最多的是 gitlab 平台,主要因为其不仅可以下载软件,还可以上传、分享和协作开发,是一个可以互动的平台,接下来是 Python,考虑人工智能等项目的不断升温导致 Python 在项目当中的应用越来越广泛,然后是 Ubuntu 和 CentOS,Ubuntu 在开发应用中较多,特别是近几年,基于 Ubuntu 的终端设备开始普及应用,而 CentOS 主要是由于一些老旧服务设备仍然在用,船内服务器领域的模拟需要用它开展测试,目前 Debian 和 Fedora 在船上的使用量并不大。从表 1 中可以看出操作系统和 Python 的数据量远小于官方仓库,是因为没有完全同步服务端数据,比如在操作系统中没有同步软件的源代码,在 Python 中只根据需要下载了部分数据包,后续可根据需要调整同步数据量大小,另外,可以将 apache2 服务调整为 nginx 服务或增加 SSD 硬盘用于改善下载速度,如果有其他的比如 ArchLinux 和 Manjaro 操作系统等需求,可以逐步扩展。

表 1 软件仓库运行统计

系统服务	数据大小/ GB	下载速度/ (MB·s ⁻¹)	日均访问 次数
Debian	85.7	8.3	2
Ubuntu	131.0	10.5	5
Fedora	89.4	75.2	1
CentOS	55.2	22.3	4
Python	150.3	80.3	6
gitlab	50.0	15.0	8

3 结 论

本文通过构建 OpenStack 云计算平台,搭建了操作系统镜像软件仓库,Python 软件仓库和 gitlab 软件共享平台,实现了测量船海上执行任务期间,操作系统软件和 Python 软件的快速安装,通过构建的 gitlab 服务实现了软件等资源的分布式共享和协同开发。随着测量船自主可控设备和软件的逐步部署,开源软件的需求会随之增大,根据海上任务需求,今后,在此云平台基础之上,可进一步扩展、更新和升级系统功能,实现更加完善的离线软件仓库和协同开发共享平台。

参考文献

- [1] 李文娟. 改进粒子群优化算法的云计算任务调度策略[J]. 国外电子测量技术, 2020, 39(10): 63-67.
- [2] 王金海, 国海丁, 王慧泉, 等. 互联网+下的云健康监护系统设计[J]. 仪器仪表学报, 2016, 37(S1): 88-93.
- [3] 宋元, 周丹媛, 石文昌. 增强 OpenStack Swift 云存储系统安全功能的方法[J]. 吉林大学学报(工.学版), 2021, 51(1): 314-322.
- [4] 刘国成, 吴丹. 基于 OpenStack 的中小企业服务云平台架构研究[J]. 吉林大学学报(信息科学版), 2020, 38(6): 709-713.
- [5] 白翼铭, 燕玮, 许凤凯, 等. HTTP Slow DoS 攻击机理分析及针对 OpenStack 云平台的防御策略与架构研究[J]. 信息技术与网络安全, 2021, 40(1): 21-25.
- [6] MOHAMMED B, MOYO S, MAIYAMA K, et al. Technical report on deploying a highly secured OpenStack cloud infrastructure using BradStack as a case study [M]. Cloud Computing Modelling and Simulation Research Group School of Electrical Engineering and Computer Science, University of Bradford, UK, October 30, 2017.
- [7] HAN X. Scaling OpenStack clouds using peer-to-peer technologies[D]. Department of Computer Science and Engineering, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden, 2016.
- [8] 郑泽丹. OpenStack 分布式虚拟路由及带宽分配策略研究[D]. 广州: 华南理工大学, 2019.
- [9] 林初晏, 朱智强, 孙磊, 等. OpenStack 云平台自适应认证系统负载均衡策略研究[J]. 计算机应用与软件, 2019, 36(9): 302-309.
- [10] 张可颖, 彭丽苹, 吕晓丹, 等. 开源云上的 Kubernetes 弹性调度[J]. 计算机技术与发展, 2019, 29(2): 109-114.
- [11] 杨皓森, 胡晓勤, 黄传波. 面向 OpenStack/Ceph 的虚拟机备份系统研究[J]. 计算机系统应用, 2018, 27(11): 96-102.

- [12] 周勇. 新时期云计算环境下的计算机网络安全技术研究[J]. 产业与科技论坛, 2018, 17(10): 94-95.
- [13] 王建明. 云计算环境下对 Web 数据挖掘技术的研究[J]. 现代信息科技, 2019, 3(5): 100-101, 104.
- [14] 庞双龙, 陈晓丹, 曾德生, 等. 云计算环境下基于 SDN 的数据中心网络架构研究[J]. 电子技术, 2020, 49(8): 39-41.
- [15] 王权. 基于云平台的大数据信息安全保护策略分析[J]. 数字通信世界, 2019(9): 93-94.
- [16] 马学森, 谈杰, 陈树友, 等. 云计算多目标任务调度的优化粒子群算法研究[J]. 电子测量与仪器学报, 2020, 34(8): 138-148.
- [17] 段辉. 基于云计算平台的大型图书馆馆藏书籍快速检索研究[J]. 电子测量技术, 2020, 43(14): 50-54.
- [18] 陈梁, 王瑞胡. 基于 YUM 本地源的 Linux 环境快速配置方法[J]. 红河学院学报, 2015, 13(5): 47-50.

作者简介

王军阵, 硕士, 工程师, 主要研究方向为云计算、大数据和操作系统等。

E-mail: qwjzh@126.com