

一种麒麟系统下基于 Django 的网络性能管理系统设计与实现

李朝阳 周维贵 张小锋 熊宗炬

(中国西昌卫星发射中心 四川 西昌 615000)

摘要 通过对网络性能管理系统在网络运维中的重要作用和国产麒麟操作系统在各关键领域得到广泛应用的事实进行分析阐述的基础上,该文基于 Django 框架设计集数据采集、存储、分析、显示及管理于一体的综合网络性能管理系统。结合 Pysnmp、Django、Highcharts 等开源工具包,重点阐述设计中所包含的各关键模块的具体实现技术细节,并给出相应的应用实例效果图。实践表明,该设计实现方法稳妥可靠,具有进一步推广应用的价值。

关键词 Django 网络性能 SNMP Kylin Python

中图分类号 TP311.11

文献标志码 A

DOI:10.3969/j.issn.1000-386x.2024.03.020

DESIGN AND IMPLEMENTATION OF A NETWORK PERFORMANCE MANAGEMENT SYSTEM BASED ON DJANGO UNDER KYLIN OPERATING SYSTEM

Li Zhaoyang Zhou Weigui Zhang Xiaofeng Xiong Zongju

(China Xichang Satellite Launch Center, Xichang 615000, Sichuan, China)

Abstract Based on the analysis of the important role of network performance management system in network maintenance and the fact that domestic Kylin operating system has been widely used in various key fields, this paper designed an integrated network performance management system with data acquisition, storage, analysis, display and management functions based on Django framework. Combining with open source toolkits such as Pysnmp, Django, Highcharts, this paper focused on the technical details of the key modules in the design, and gave the corresponding application examples. The practice shows that the design method is reliable and has the value of popularization and application.

Keywords Django Network performance SNMP Kylin Python

0 引言

网络性能管理作为日常网络运维的重要组成部分,是确保网络稳定可靠运行的必要手段,也是网络配置策略制定调整的主要依据,更是评判网络健康状况良好与否的重点标准。因此,对信息化建设水平较高的大中型企业而言,构建一套配置灵活、运行高效、用户友好的网络性能管理系统尤为重要。本文旨在设计一种国产麒麟平台下基于 Django 框架的企业综合网络性能管理系统,以实现管理用户对域内网络设备、服务器及应用系统相关性能指标参数的实时监测、存储、分析、可视化及历史数据追溯的功能。

1 技术途径

1.1 Django 框架

Django^[1]是利用 Python 开发的基于 MVC 设计模式的开源 Web 开发框架。它将业务逻辑层、前端视图层及数据模型层以高内聚低耦合的方式实现,且内嵌有轻量级 Web 服务器,使架构于其上的 Web 开发更加迅捷、简便、高效。随着 Python 语言受青睐程度与日俱增,Django 亦逐渐受到开发者的重视和青睐。Django 的模型(Model)、模板(Template)和视图(View)架构和传统的 MVC 框架类似,其中 Model 与 MVC 中的

Model 相同, Templates 与 View 之和相当于 MVC 中的 View, Django 中的 URLconf 类似于 MVC 模型中 Control 的概念。基于上述设计理念的 Django 主要包括如下核心组件: (1) 关系对象映射(ORM), 优点在于可灵活选择数据存储工具而无须修改代码进行再次适配。(2) 视图模板, 大量模板系统极大降低 Web 前端维护成本。(3) 后台管理系统, 实现对网站数据的自动化灵活配置管理。(4) UrlConf, 基于正则表达式的控制部件使得相应资源的定位查询更加简洁高效。

1.2 Kylin 系统

中标麒麟是 2010 年 10 月确定的核高基项目中的重点基础软件项目。目前已经广泛应用于交通、金融、能源、政府及军队等行业领域的军民两用的, 具有强化 Linux 内核的, 基于细粒度自主访问控制的, 从内核到应用提供全方位安全保护的国产操作系统。Qt, Java、Python 是麒麟系统上经常采用的应用程序开发语言。本文旨在阐述在国产麒麟平台下基于 Django 框架开发用于网络性能管理的 Web 系统。

2 体系结构设计

2.1 总体结构设计

整个网络性能管理系统采用 python 语言在国产麒麟平台下开发完成, 其总体结构及信息流程如图 1 所示^[2]。该系统由数据中心模块、Web 接口模块、管理接口模块、网络基础设施四部分构成。其特点为: 国产平台下实现数据主动推送的集实时性能数据采集、存储、分析、管理于一体的即插即用的高性能 Web 服务管理系统。

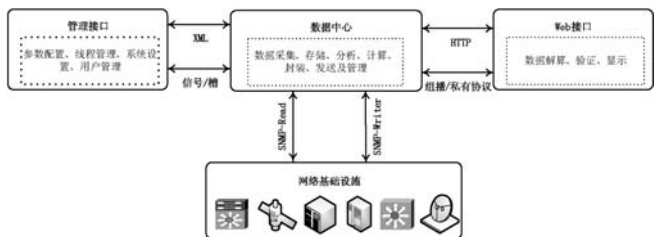


图 1 系统总体结构示意图

数据中心模块作为整个系统的核心, 集中实现性能数据的周期性采集、存储及分析等功能, 即整个 B/S 系统的服务端。Web 接口模块重点在 Django 框架下实现, 用于数据可视化分析及报表生成等。Web 接口模块获取数据包括三种方式, 采用何种方式, 可根据需求灵活处置。(1) 对于网络规模较大且存在大屏投放的应用场景, 采用以插件形式接收并解算数据中心组

播而来的性能数据, 以此驱动各类图表, 实现性能数据的主动推送功能^[3]。(2) 对于历史数据分析而言, 采用在 Django 框架下实现对 MySQL 数据库中历史数据的读取、分析对比操作。(3) 对于故障排除应用场景而言, 可采用在 Django 框架下实现对网络设备的直接数据采集。

2.2 数据中心模块设计

数据中心模块及其内部信息流程如图 2 所示^[4], 其核心部件是控制单元和若干任务子模块。该设计旨在实现整个系统的可扩展性, 即只需添加或更换图中数据传感器子模块便可完成整个系统功能的丰富和扩展。本文对其中关键子模块简要阐述如下: (1) 控制单元, 基于 APScheduler 开发的多任务定时调度程序, 用于对线程池中的任务进行细粒度精确控制。(2) 任务子模块, 主要作用是执行具体的数据采集、封装、发送、存储等事务的低级别线程、交由控制单元完全控制。(3) 共享库函数: 对诸多任务子模块实现过程中共性的部分进行抽象而得来的库函数, 主要包括私有协议包、Multicast、SharedLib 及 TimeService。(4) 数据传感器: 每个任务中实现数据获取的相应子模块, 譬如对网络设备可基于 Pysnmp 包来实现其相应数据传感器^[5]。

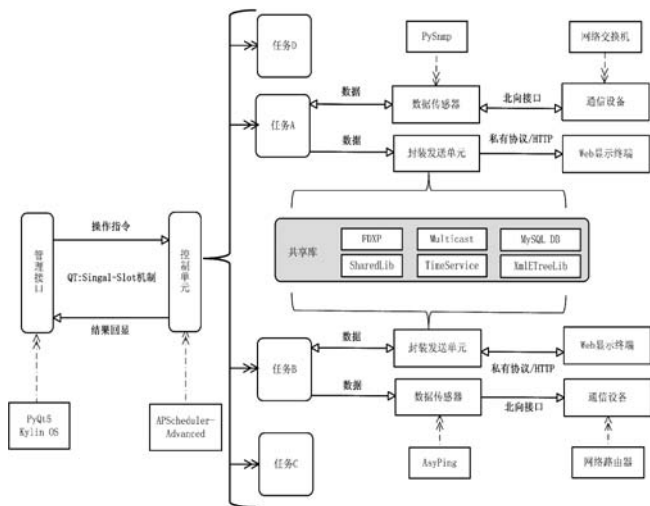


图 2 数据中心模块结构示意图

2.3 Web 接口设计

基于 Django 框架下的 Web 接口设计及其细节如图 3 所示。在相应数据源的基础上, 依照设备管理、拓扑管理、流量管理等性能监测类别, 分层分块实现 Django 框架下的 MTV 结构 (Model-Template-Control)。对于 Web 前端而言, 主要借助相应 jquery 库及 Django 模板特性设计与整个网络性能管理系统功能相对应的数据可视化及分析模板^[6]。

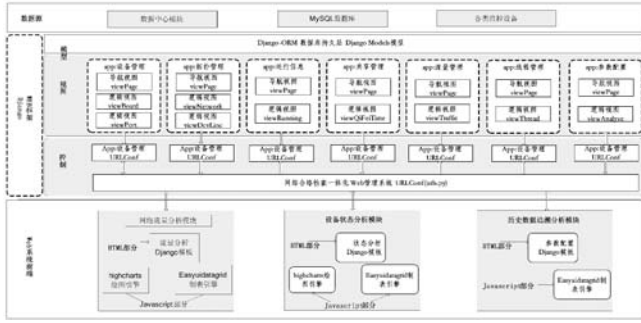


图3 Web接口设计示意图

3 系统实现方法

3.1 数据中心模块实现

将图2所示的模式设计映射到实际网络环境中,即产生如图4所示的实例示意图,亦可根据性能管理对象类别及数量的不同适当增加或减少相应任务。各个任务子模块之间的根本区别仅仅在于数据采集方式的不同,其余诸如组播、私有协议封装、数据库存储等共性特点在该系统中已经被封装为相应共享库函数,因此该系统的二次开发及功能扩展极易实现。

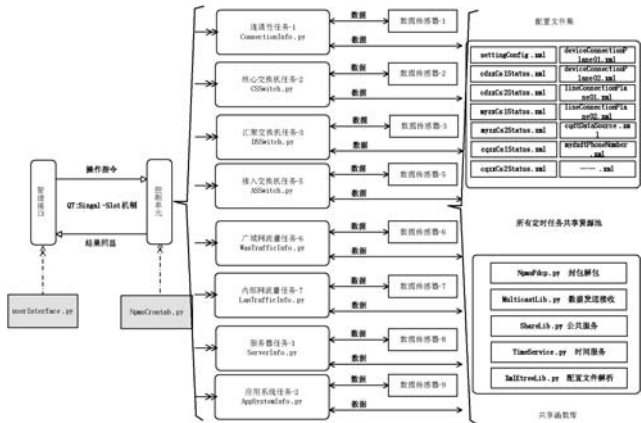


图4 数据中心模块实现实例

本文着重就控制单位子模块及网络流量采集任务子模块的实现进行阐述,其余子模块实现类似。(1)控制单元,基于开源工具包 APScheduler 实现对若干网络性能数据管理任务的周期性定时调度,亦包括各任务的重启、暂停及执行消息回显等。其中, APScheduler 是基于 Quartz 的一个 Python 定时任务框架,支持基于日期、固定时间间隔以及 crontab 类型的细粒度任务调度的 Python 开发库,且可以持久化任务。(2)网络流量采集,基于 Pysnmp^[7] 开源库实现相应网络设备接口上 SystemUpTime、InBytes、OutBytes 三个参数的实时采集,并在此基础上计算出某时刻的网络流量并以时间序列方式存入 MySQL 数据库,或以组播方式发往全网路由可达的客户端。

3.2 Web 接口模块实现

本节着重阐述图5所示的Web系统在 Django 框架下的资源调度器 UrlConf、应用 App 实现、前端显示模板 (Templates) 的具体实现过程。

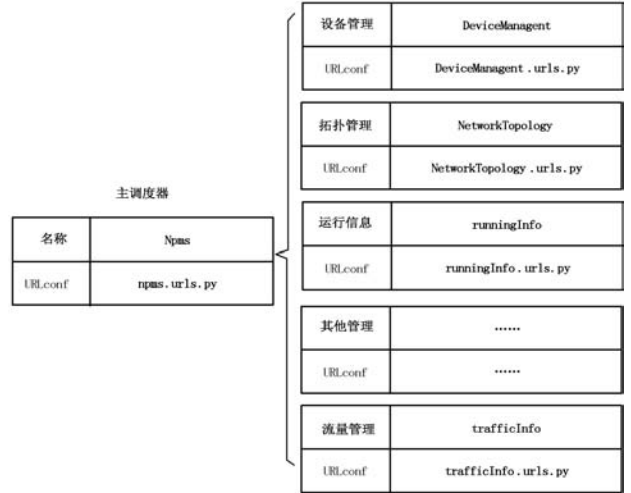


图5 系统调度器设计实现示意图

(1)资源调度器 UrlConf 实现。为实现复用性强,配置灵活的管理系统,本文设计如图5所示的资源调度管理模式。即一个一级调度器和若干与功能 App 相对应的子调度器,一级调度器控制和管理所有子调度器。

(2)应用 App 实现。在 Django 框架内部,应用 App 作为整个网络性能管理系统服务端逻辑处理的基本单元,可根据系统功能设置将系统分解为若干应用 App。对每个应用 App 而言,主要是在模型 (Model) 基础上实现两类视图 (View),譬如对于运行信息 App 而言,包括导航视图 ViewPage 和逻辑视图 ViewRunning,它们是数据存储模块和 Web 应用前端程序之间信息交互的纽带^[8]。由于篇幅原因,此处对具体代码实现不作说明,可参考 Django 官方文档。

(3)前端显示模板 (Templates),以网络流量可视化为例,本文旨在借助 Django 的模板技术,实现一个 HTML 页面绘制若干个流量显示页面。基于此,文章设计如图6所示的针对网络流量可视化的系统前端模板实现示意图,主要借助 Django 模板技术、Highcharts、Easyui、Jquery 等工具实现。

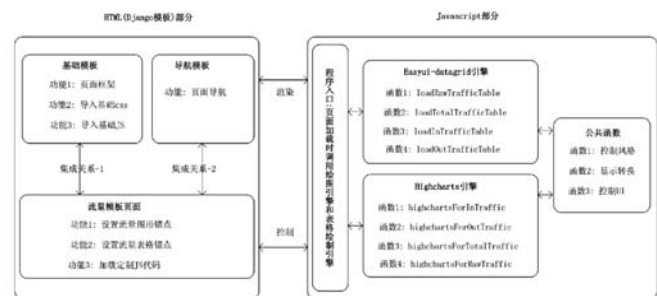


图6 前端显示模板实例实现示意图

4 应用实践

本文所设计实现的网络性能监控系统是在国产麒麟操作系统下借助 Python、Django、jQuery、Highcharts 等多个开源工具包开发实现的,具有网络拓扑管理、流量监测、设备状态监测等多项功能,图7是其进行网络流量监测的效果图。鉴于篇幅原因,其余不再赘述。

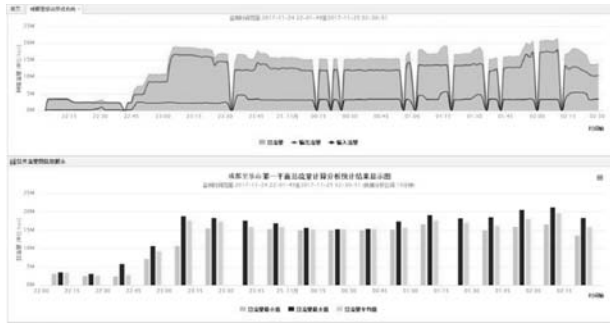


图7 系统应用实例示意图

5 结语

本文在对网络性能管理在网络运维中的积极作用及国产麒麟操作系统在政府、交通、金融、军队等关键部门大规模部署应用背景进行阐述分析的基础上,设计出一种基于 Django 框架的集数据采集、存储、封装、分析、显示及管理于一体的高性能网络性能管理系统,并对整个系统实现的关键细节结合具体实现技术进行了详细说明,给出了该系统运行的具体实例。应用实践表明,文章所述方法可靠可行,具有一定的应用推广价值。

参 考 文 献

[1] Django Software Foundation. Django Documentation [EB/OL]. (2021-01-01). <https://www.djangoproject.com/>.

[2] Ilya Etingof. Pysnmp Documentation [EB/OL]. (2021-01-01). <https://github.com/etingof/pysnmp>.

[3] Joe Kuan. Learning Highcharts 4 [M]. Birmingham UK: Packt Publishing, 2015: 321 - 328.

[4] 曹丹阳,霍然,孙凌,等. 卫星可见光波段观测模拟与分析系统设计与开发[J]. 软件, 2018, 39(3): 1 - 7.

[5] 常佳宁,李阳齐. 基于 Django 的个人博客系统设计开发[J]. 中国科技信息, 2021(2): 75 - 77.

[6] 牛作东,李捍东. 基于 Python 与 flask 工具搭建可高效开发的实用型 MVC 框架[J]. 计算机应用与软件, 2019, 36(7): 21 - 25.

[7] 林刚,文全刚,傅晓阳,等. 解析 MIB 文件的 API 设计与应

用[J]. 计算机应用与软件, 2018, 35(2): 141 - 144.

[8] 赵鹏,褚剑等. 基于开源硬件平台的分布式网络性能测量系统研究[J]. 软件, 2015, 36(8): 150 - 154.

(上接第 129 页)

[12] Yu S, Gu G F, Barnawi A, et al. Malware propagation in large-scale networks [J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(1): 170 - 179.

[13] Wang X M, He Z, Zhang L. A pulse immunization model for inhibiting malware propagation in mobile wireless sensor networks [J]. Chinese Journal of Electronics, 2014, 23(4): 810 - 815.

[14] 赵金皓,曹奇英,沈士根. 受到恶意程序攻击的 MWSNs 节点状态时空动力学分析[J]. 计算机应用与软件, 2018, 35(6): 122 - 128.

[15] Shen S G, Zhou H P, Feng S, et al. SNIRD: Disclosing rules of malware spread in heterogeneous wireless sensor networks [J]. IEEE Access, 2019, 7: 92881 - 92892.

[16] Shen S G, Zhou H P, Feng S, et al. HSIRD: A model for characterizing dynamics of malware diffusion in heterogeneous WSNs [J]. Journal of Network and Computer Applications, 2019, 146: 102420.

[17] 沈士根,周海平,黄龙军,等. 基于扩展传染病模型的异质传感网恶意程序传播建模与分析 [J]. 传感技术学报, 2019, 32(6): 923 - 930.

[18] Lalropuia K C, Gupta V. A Bayesian game model and network availability model for small cells under denial of service (DoS) attack in 5G wireless communication network [J]. Wireless Networks, 2020, 26(1): 557 - 572.

[19] 沈士根,范恩,胡珂立,等. 面向恶意程序传播的传感网可靠度评估 [J]. 电子学报, 2018, 46(1): 75 - 81.

[20] 苏玉泽,孟相如,康巧燕,等. 基于半马尔可夫过程的虚拟网络生存性模型 [J]. 科学技术与工程, 2019, 19(24): 260 - 267.

[21] Chen H S, Zhuang H Y, Shan Z G, et al. A novel SMP-based survivability evaluation metric and approach in wireless sensor network [J]. Computer Science and Information Systems, 2019, 16(3): 733 - 751.

[22] Shakya R K, Rana K, Gaurav A, et al. Stability analysis of epidemic modeling based on spatial correlation for wireless sensor networks [J]. Wireless Personal Communications, 2019, 108(3): 1363 - 1377.

[23] 罗宁. 基于因果关联攻击场景重构的方法研究 [D]. 武汉: 华中科技大学, 2005.

[24] 经小川,胡昌振,谭惠民. 基于关联序列分析的协同攻击检测方法研究 [J]. 武汉理工大学学报, 2004, 26(6): 78 - 81.

[25] 宋明秋,李艳博. 考虑攻击相关性的蠕虫传播模型 [J]. 运筹与管理, 2020, 29(1): 79 - 85.