

# 图书馆数据中心运维中开源软件的应用

郑海山<sup>1</sup> 林俊伟<sup>2</sup>

<sup>1</sup>(厦门大学信息与网络中心 厦门 361005)

<sup>2</sup>(厦门大学图书馆 厦门 361005)

**摘要:**【目的】解决高校图书馆数据中心在服务器数量膨胀下的自动化运维问题。【应用背景】高校图书馆数据中心承担的任务越来越重,服务器数量急剧攀升。图书馆数据中心内有各种虚拟化平台和传统的服务器,单纯的人工管理已无法适应。【方法】查找互联网上的开源软件;部署测试并找到最佳实施方法;研究开源软件存储的数据结构进行二次开发,消除多个开源软件之间的信息孤岛。【结果】解决服务器和虚拟机的位置管理、网络互联结构、自动化部署、IP 分配、监控、备份、集中式日志管理等一系列运维要点,并最终展示给读者一个目前服务器状态的页面。【结论】应用开源软件后图书馆数据中心对内管理清晰化、规范化、自动化,对外透明化。

**关键词:** 数据中心 自动化运维 开源软件

**分类号:** TP393

## 1 引言

随着高校信息化的兴起,当前高校图书馆数据中心服务器数量不断膨胀,数据中心运维层面需要关注的越来越多。厦门大学图书馆数据中心目前共有 105 台各式服务器和交换机等设备以及 72 台虚拟机。这些服务器内运行着多种类型的服务,分配了多达 2 个 C 类的 IP 地址。在数据中心建立初期,因为服务器和业务较少,数据中心没有配备管理软件。随着服务器数量的增加,服务器经常性错误数量也在攀升,在根据 IP 地址或 MAC 地址查找服务器位置上花费了较多时间。而且业务故障时工程师无法得到通知,只有等工作人员或者读者报告上来后才得知,无法在第一时间处理业务故障。传统的人工管理模式已经无法适应越来越膨胀的服务器数量,工程师疲于奔命、忙于救火的状态已经难以扩展新的业务。

为解决以上问题,笔者开始零星地引入开源软件。随着开源软件的引入,多个开源软件构成了信息孤岛,工程师不得不在各个开源软件管理界面之间切

换,信息无法互通,因此急需一个能聚合起多个开源软件的运维平台。

本文介绍厦门大学图书馆数据中心在选择开源软件中的思考,分享 RackTables、Nagios、Ganglia、SmokePing、Bacula、logstash、Puppet 等多个开源软件的实施方法和使用要点以及为消除多个开源软件信息孤岛而进行的二次开发工作。

## 2 数据中心运维平台的需求和技术思路

国内对于数据中心的管理一般有采购商用软件、完全自主开发<sup>[1]</sup>和使用开源软件<sup>[2]</sup>等三种模式。在资金充裕的条件下,采购商业软件是一个一劳永逸的做法,但商业软件也存在弊端:商业软件内部的数据格式不对外公开,有些甚至对其加密,使得无法进行二次开发,若要定制开发,则开发周期很长,成本较高。同时,商业软件过于注重界面展示,不提供文本文件型配置或命令行等功能,自动化程度低。

另有一种模式是完全使用虚拟化技术<sup>[3]</sup>,使用虚拟化平台自带的控制台。然而此方案只对新建的数据

收稿日期: 2014-02-18

收修改稿日期: 2014-04-18

中心有效。厦门大学图书馆数据中心不仅有虚拟化平台，还有老旧服务器和无法虚拟化迁移的服务器，有些设备诸如磁带库也无法纳入虚拟化平台管理。

基于以上原因，为了更加自由的控制，在厦门大学图书馆数据中心运维中大量采用开源软件。开源软件的弊端是学习成本较高，实施难度大。由于开源软件一般都较为灵活，实现同一个需求有多种方法，需要不停测试适合的实施方法。

选择开源软件的原则是选择使用可远程访问的数据库后端的开源软件，选择开发活跃度高、使用者众多、文档齐全、同时可部署在多种操作系统上的开源软件。测试过程为：下载多个相同类型的开源软件，安装、测试、比较，了解每个开源软件的用途、原理和应用场景。选定开源软件后，对选定的开源软件的配置文件和数据保存方式进行研究。找出开源软件尚未实现的功能和不足，对其进行二次开发。

数据中心运维的是首先是实现对设备的位置管理，对设备运行状态的监控，对服务器的数据进行备

份，对服务器的日志进行集中管理。

机房和设备的物理位置是人工配置的，这个配置是无法有效检查的。笔者拟使用脚本查找交换机目前运行的所有 IP 地址的网络端口对应关系，对配置进行正确性检查。开源软件均有 Web 管理界面，通过所有服务器都有的唯一的 IP 地址和机器名进行内在关联，编写一个统一的管理员控制台实现门户的功能，同时实现开源软件没有实现的诸如多个开源软件配置冲突和空余 IP 地址等功能。脚本可使用 Python、Perl 定时运行。门户控制台使用 PHP、MySQL 开发，网站前端 CSS 框架使用 JQuery Mobile，应用响应式布局方便移动客户端访问。

### 3 数据中心运维平台实现细节

#### 3.1 平台框架及功能

整个运维平台部署多个开源软件，几个自定义脚本，一个给管理员使用的统一的 Web 控制台和一个对外展示数据中心状态的站点。部署框架如图 1 所示：

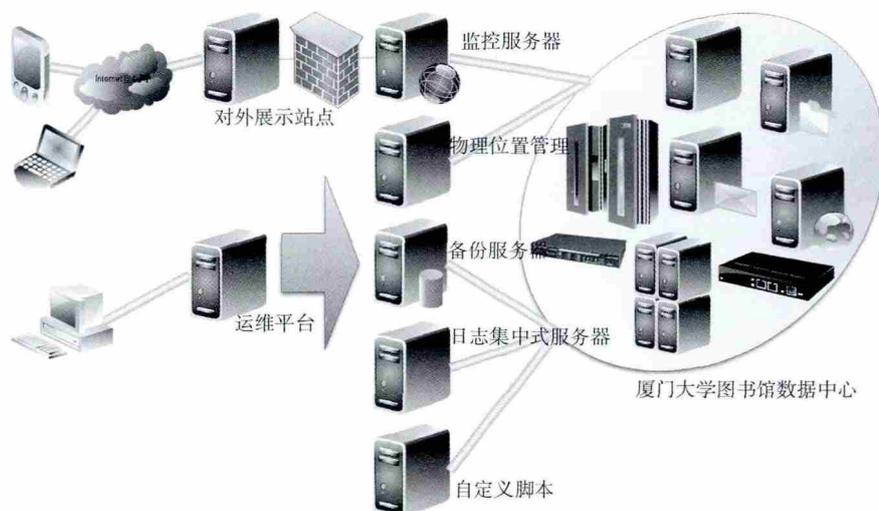


图 1 系统部署的框架

运维平台实现的功能主要有以下三部分，分别为：

#### (1) 基于开源软件实现的功能

将服务器、交换机、机柜、机房关系图形化，可视化。管理包括机房、机柜、设备和虚拟机、网线连接情况、IPv4/IPv6 地址占用、IP 负载均衡、VLAN 等属性，并可以打印报表统计机器数量、IP 空余状况、设备保修状况。

对服务器的可用性、服务器的性能指标、两个机房之间的网络质量情况进行监控。

对服务器数据进行备份，对日志进行集中式管理。

#### (2) 自定义脚本和管理员控制台

① 列出业务网段内的所有 IP 地址，显示是否在用，为新机器的 IP 地址分配提供参考。

② 列出 IP 地址对应的机器名、MAC 地址、DNS 地址、机器厂商、连接到交换机的端口位置,方便查找过滤。点击 RackTables、Nagios、Bacula、VMware 等相应的图标可跳转到对应服务器的 Web 管理系统界面。

③ 列出所有与 RackTables 配置不匹配的设备,为管理员验证 RackTables 配置提供数据支持。列出所有 IP 地址和机器名不相同的机器,方便管理员检查各个系统配置。

④ 列出所有未安装 VMware Tools 的虚拟机,督促管理员安装。列出所有没有备份的服务器,督促管理员备份。

⑤ 统计所有不同类型的服务器数量。

⑥ 编辑和修改对外展示网站的备注。

PHP 语言开发的统一的 Web 控制台通过各个服务器唯一的 IP 地址和机器名匹配整合所有后台数据。网站效果如图 2 所示:

IP 段	IP 和交换机接口	所有虚拟机	未安装 VMware Tools	RackTables 不匹配	
有备份服务器	未监控服务器	统计			
Q 搜索字段					
IP 地址	机器名	MAC 地址	DNS	MAC 厂商	系统
1 210.34.4.1	switch1	38:22:d6:2d:2d:17	router4.xmu.edu.cn	H3C Technologies Co., Limited	RT
2 210.34.4.2					
3 210.34.4.3	ebook	00:06:5b:f6:b2:c2		Dell Computer Corp.	RT
4 210.34.4.4					
5 210.34.4.5	cnki	00:1a:64:24:3a:a2		IBM Corp	
6 210.34.4.6	DSpace	00:06:5b:f6:b2:c1	dspace.xmu.edu.cn	Dell Computer Corp.	RT
7 210.34.4.7	natsserver	00:50:56:a3:58:39		VMware, Inc.	RT
8 210.34.4.8	ilas2backup	78:2b:cb:1d:a2:34		Dell Inc	RT

图 2 聚合仪表盘式网站

### (3) 对外展示站点

通过监控服务器收集的数据有选择地将部分重要

服务器状态显示出来,对外展示数据中心的服务器健康状态。显示效果如图 3 所示:

服务器、业务名称	状态	备注	咨询电话
1 图书馆主页	✓		2184973
2 座位查看系统	✗	由于硬盘损坏,目前服务器正在维修中,预计在2014/02/09 18:00:00恢复。	2184973
3 DSpace	✓		2184973
4 馆际互借	✓	为了给读者提供更优质的服务,本服务器预计在周三下午 2014/02/12 21:00:00 - 2014/02/12 22:00:00之间维护,维护期间本系统无法使用。	2185190
5 论文提交系统	✓		2184973

图 3 对外展示的站点

### 3.2 开源软件数据保存结构分析

开源软件保存的数据可分为配置数据和运行数据两种,一般保存在文本文件或者数据库内,二次开发时需要先分析这些数据。为了能关联其他开源软件,所有服务器在各个开源软件内应有相同的 IP 地址和机器名。

在运维平台用到的几个开源软件中, RackTables 配置信息和运行数据全部保存在 MySQL 内。可使用 RackTables 的 Object 表生成 Nagios 的配置信息,所有配置了 IP 地址的设备按格式写入/etc/nagios3 即可对设备实施监控。

默认安装模式下, Nagios 配置文件在/etc/nagios3 目录内,运行数据保存在/var/cache/nagios3 和/var/lib/nagios3/spool/checkresult 内。通过安装 NDOUtils 插件,可将所有配置数据和运行数据存入 MySQL 内,判断是否加入监控可通过查询 Nagios\_hosts 表获得。

Bacula 配置文件保存在/etc/bacula 内, Bacula 会自动将配置信息存入 MySQL,同时所有的备份文件名也保存在 MySQL 内,备份的文件数据存放在自定义的目录内。判断服务器是否有备份可通过查询 Bacula 的 Client 表获得。

logstash 服务端把所有获得的日志保存在 Redis 键值型数据库内,配置文件分散在各个客户端。笔者在配置中通过 type 字符串字段区分不同的客户端, type 字段由 IP 地址和服务说明字符串拼接而成。

### 3.3 RackTables 配置和数据库结构

RackTables 管理设备的属性和属性的字典都是可以自定义的。虽然 RackTables 也可以添加端口的二层物理地址,然而由于添加工作量较大,而且可在笔者二次开发的程序中自发现管理,所以在实际使用中,未对该项数据进行添加。

RackTables 带有标签功能,标签的作用是在查看设备时可对设备进行过滤。在实际应用中,笔者定义了三个一级标签,分别是“归属部门”、“负责人”、“所属业务”,在一级标签下建立二级、三级标签,通过标签,可以很方便地过滤所有的设备。

每一台 RackTables 内的设备信息均保存在 Object 表中,设备的所有端口保存在 Port 表中。Port 表的 object\_id 字段与 Object 表外键关联,Port 表的 name 字段是端口名,在 Linux 下一般为 eth0, Windows 下为“本

地连接”。服务器物理端口与交换机物理端口连接的信息保存在 Link 表中, Link 表只有 porta 和 portb 字段,记录了网线两端端口的 Link 表 id 字段值。IPv4Allocation 表有 object\_id、ip、name 等字段。ip 字段为 IP 地址, name 字段为 Port 的端口名,通过下面 SQL 语句可以查询到对于特定的服务器 object\_id = 1 连接到的交换机的端口信息:

```
SELECT ob.id, ob.name, po.id, po.name, porta, portb
FROM Object ob
LEFT JOIN Port po ON po.object_id = ob.id
LEFT JOIN Link li ON li.porta = po.id OR li.portb = po.id
WHERE ob.id = 7
```

而对于虚拟机则较为复杂。由于虚拟机没有物理端口,所以需要通过 EntityLink 表的 child\_entity\_id 和 parent\_entity\_id 字段得到虚拟机的宿主集群。在集群的所有子机器内寻找所有物理的虚拟化宿主服务器,再通过物理的虚拟化宿主服务器的端口来判断连接到交换机的端口信息。

在集群内的所有子机器寻找物理的虚拟化宿主服务器的步骤是:通过 Object 表和 EntityLink 表可以得到其所有的服务器,包括虚拟机和实体机。实体机和虚拟机的区别在于 Hypervisor 字段是否为 Yes。前面提到 RackTables 的设备属性都是可自定义的,设备的属性不保存在 Object 表内,而是通过 Attribute、AttributeMap 和 AttributeValue 等表组成。Attribute 表定义了系统内置属性和自定义属性的类型和名字, AttributeValue 表包括 object\_id、attr\_id、3 个类型的 value 字段。而 Hypervisor 字段结果值的定义存放在 Dictionary 字典表内。所以查找一个 object\_id 为 8 的机器是否为虚拟机的 SQL 语句为:

```
SELECT ob.id, ob.name, di.dict_value #dict_value 为 Yes 则为
实体机器
FROM Object ob
LEFT JOIN AttributeValue av ON av.object_id = ob.id
LEFT JOIN Dictionary di ON di.dict_key = av.uint_value
WHERE ob.id = 8 AND av.attr_id = 26 #26 为系统默认
Hypervisor 的 ID
```

### 3.4 监控配置

在 Nagios 监控系统<sup>[4]</sup>配置方面,没有引入 Web 配置系统,采用文本配置更为灵活。为了可以跟随版本升级应用到一些新的配置参数,笔者独立出一个目录来配置所有主机、服务、联系人。由于配置文件

最终保存在 Git 源代码版本库中,所以对配置文件按服务器的用途进行区分,每个用途一个配置文件,每个管理员负责的多台服务器分开配置文件,采取这种策略后,版本控制冲突减少,文件的频繁变化也被隔离开来。

Ganglia<sup>[5]</sup>用于监控集群的 CPU、内存、磁盘空间和网络。Nagios 关注的是个体服务器,如果服务器集群内服务器数量达到上百台,单个服务器故障时有发生,使用个体的服务器错误来衡量整个集群的状态是不合适的。而 Ganglia 关注的是整个集群对外提供的 CPU 数、内存数,如果整个集群参数在正常范围内,部分服务器有问题 Ganglia 会认为是正常的。

SmokePing<sup>[6]</sup>用于监控多个数据中心之间和数据中心与互联网其他数据中心之间的网络互联质量情况,从监控图上的延时与丢包能分辨出数据中心机房的网络是否稳定。

### 3.5 备份、日志集中式管理、自动化部署

在厦门大学图书馆数据中心,笔者使用了服务器级别和数据层面两种备份策略。服务器级别的备份在虚拟机中可通过建立虚拟机快照和使用虚拟化平台自带的备份策略。数据层面的备份,于春生<sup>[7]</sup>对 Bacula、Amanda、BackupPC 等多个开源备份软件进行了比较,为了备份软件客户端程序可同时运行在 Windows 和 Linux 服务器上并为了保护被备份的数据不被备份管理员随意恢复,笔者选择 Bacula。

数据中心日志没有集中的管理,系统出现错误后需要进入各个系统查看,非常消耗时间,同时集中式日志作为安全审计的依据,在系统和网络管理中有着重要的意义<sup>[8]</sup>。由于日志集中式管理软件需要同时支持多种类型的日志导入方式,所以笔者选择 logstash。所有需要用来分析定位问题的日志都被传输到日志服务器,同时笔者和开发人员、服务器维护人员约定,任何需要保密的信息不能写入日志,不能被传输到 logstash 里。

在对数据中心管理维护中,必须确保每个服务器都应用了安全补丁;时间通过 NTP 服务器同步;软件安装和更新来源使用校内开源软件镜像服务器;特定的机器安装 Ganglia、Bacula、logstash。为了确保以上这些要点,引入自动化部署系统。在虚拟机层面,建立了多个模板,这些模板用于快速克

隆部署。而实体机器和克隆后的虚拟机,笔者采用 Puppet 管理<sup>[9]</sup>,引入 Puppet 后极大地减少了部署工程师的工作量。

### 3.6 收集虚拟机信息的方法

使用 VMware 自带的 API 和 Psphere 库收集虚拟机的信息并存入数据库。Python 代码如下:

```
client = Client("vCenter IP", "UserID", "Password")
//连接 VMware vCenter Server
cr_list = ComputeResource.all(client)
for cr in cr_list:
    ccr = ComputeResource.get(client, name=cr.name)
    for host in ccr.host:
        for vm in host.vm:
            ip = ""
            if hasattr(vm.summary.guest, 'ipAddress'):
                ip = vm.summary.guest.ipAddress
            //如果没有安装 VMware Tools 无法得到 IP 地址,直接引用会出错
            save_to_db(host.name, vm.name, ip, vm._mo_ref.value,
                vm.summary.guest.toolsStatus)
            //把虚拟机的宿主机 IP 地址,虚拟机名,虚拟机 IP 地址,虚拟机 ID, VMware Tools 状态存入数据库
```

构建 VMware Console 的链接具体实现是:链接格式为 <https://vmware.example.com:9443/vsphere-client/vmrc/vmrc.jsp?vm=urn:vmomi:VirtualMachine:vm-10X-X:16C304D3-27FC-4363-983F-FA6934CA9E7B>,其中 vm-10XX 替换为 vm.\_mo\_ref.value 即可,链接最后面的 UUID 为常量。

### 3.7 收集实体机和虚拟机网络位置的方法

使用 Python 脚本,通过交换机的 SNMP 协议读取交换机每个端口的 VLAN、MAC 地址;使用 Perl 代码调用 Linux 命令行 arp-scan 得到 MAC 地址对应的厂商来区分服务器是实体机或者虚拟机;通过三层交换机查找路由表,得到 IP 地址对应的 MAC 地址;通过 arp-scan 扫描特定的 IP 地址段查找 IP 是否在使用;通过命令行 nslookup 得到 IP 地址对应的 DNS 地址。以 H3C S5800 交换机为例,Python 代码如下:

```
def get_ip_mac(self): #获得 IP 地址和 MAC 地址对应关系
    oid = netsnmp.VarList(netsnmp.Varbind
        ('iso.3.6.1.2.1.3.1.1.2.116.1.1'))
    netsnmp.snmpwalk(oid, Version = 2, DestHost = 'IP 地址',
        Community = '密码')
    IpMac.objects.all().delete() #临时性清空表
    for var in oid:
```

```
mac = ":".join((leading0(hex(ord(var.val[i]))[2:4])) for i in
range(0, 6)) #得到 MAC 地址, 做了格式转换
IpMac.objects.create(ip=var.tag[28:], mac=mac) #加入 IP 和
MAC 对应表
```

获取端口索引和端口名对应关系的 SNMP OID 为 'iso.3.6.1.2.1.31.1.1.1', 获取 MAC 地址和端口索引对应关系的 SNMP OID 为 'iso.3.6.1.2.1.17.7.1.2.2.1.2', 通过以上类似的代码, 获得所有的对应关系。以上收集的信息保存在表 1 内。

表 1 运行时服务器信息数据

字段名	字段类型	长度	说明
IPAddress	字符型	255	IP 地址
MACAddress	字符型	255	MAC 地址
SwitchName	字符型	255	交换机名称
SwitchPort	字符型	255	所连接在交换机上的端口
VMHost- IPAddress	字符型	255	虚拟机宿主物理服务器 IP
VMName	字符型	255	虚拟机名称
VMMoRefID	字符型	255	虚拟机的 MoRef ID
VMwareTools Status	字符型	255	虚拟机是否安装 VMware Tools

#### 4 结果分析

基于以上实施的多个开源软件和二次开发的聚合网站, 对厦门大学图书馆数据中心进行了有效透明的管理: 无需登录多个系统, 即可定位到故障服务器; 可在短时间内统计所有服务器数量; 可在一个页面获知所有服务器的备份状态; 可在一个界面分析所有服务器日志; 可实时对外公布服务器问题和预计的修复时间。

为了验证节省的时间, 做了简单的测试。测试选择过去一年信息技术部最常遇到的业务请求, 测试结果如表 2 所示:

表 2 业务处理时间比较

业务	原先持续时间	目前持续时间
根据 IP 地址查找服务器位置 (随机提供 5 个 IP 地址, 5 个 MAC 地址, 取平均值)	23 分钟	1 秒
服务器故障到工程师发现时间	1 小时	9 分钟
统计所有服务器数量	19 分钟	1 秒

从表 2 可以看出, 引入的多个开源软件和二次开发的聚合网站大大减少了信息技术部馆员的工作量, 提升了读者的满意度。

#### 5 结 语

厦门大学图书馆数据中心在商业软件、自主开发和开源软件中选择了开源软件, 通过二次开发把所有开源软件聚合在一起, 对数据中心进行了有效透明的管理。读者只需访问对外展示站点即可获知数据中心的服务器和业务运行状况, 可根据状态自行诊断是服务器故障或者自身计算机问题, 减少了前台接听电话的数量。由于开源软件可选择性较多, 多个开源软件功能重叠, 笔者今后还将持续寻找适合的开源软件加入本平台。

#### 参考文献:

- [1] 李灿全. 基于 ITIL 的应用级运维支撑平台研究与实现 [D]. 上海: 上海交通大学, 2013. (Li Canquan. Research and Implementation of Application-level Maintenance Platform Based on ITIL[D]. Shanghai: Shanghai Jiaotong University, 2013.)
- [2] 王广凤. 开源软件与专有软件的竞争 [D]. 沈阳: 辽宁大学, 2008. (Wang Guangfeng. Competition Between Open Source and Proprietary Software[D]. Shenyang: Liaoning University, 2008.)
- [3] 朱海涛. VMware 虚拟化系统在高校图书馆中的构建与应用 [J]. 现代图书情报技术, 2012(1): 68-72. (Zhu Haitao. Construction and Application of Virtualization System Using VMware in University Library[J]. New Technology of Library and Information Service, 2012(1): 68-72.)
- [4] 董波, 沈青, 肖德宝. 云计算集群服务器系统监控方法的研究 [J]. 计算机工程与科学, 2012, 34(10): 68-72. (Dong Bo, Shen Qing, Xiao Debao. Research on Monitoring Method of Cloud Computing Cluster Server System[J]. Computer Engineering & Science, 2012, 34(10): 68-72.)
- [5] 何丽萍, 刘立程. 改进的基于 Ganglia 的网格监控系统 [J]. 广东工业大学学报, 2006, 23(1): 85-89. (He Liping, Liu Licheng. Improved Grid Monitor System Based on Ganglia[J]. Journal of Guangdong University of Technology, 2006, 23(1): 85-89.)
- [6] 使用 SmokePing 监控 IDC 机房网络质量情况 [EB/OL]. [2014-01-05]. <http://dl528888.blog.51cto.com/2382721/1266598>. (Using SmokePing to Monitor the Network Quality in IDC[EB/OL].[2014-01-05].<http://dl528888.blog.51cto.com/2382721/1266598>.)
- [7] 于春生. 企业级开源备份软件在高校数据中心中的应用 [J]. 北京联合大学学报: 自然科学版, 2012, 26(2): 11-14.

(Yu Chunsheng. Application of Enterprise-class Open Source Backup Software in the Data Center of University[J]. Journal of Beijing Union University: Natural Sciences, 2012, 26(2): 11-14.)

- [8] 李传勇, 沈辉. 网络设备运行日志集中管理经验介绍[J]. 金融科技时代, 2013(7): 85-87. (Li Chuanyong, Shen Hui. The Experience of Log Centralized Management of Network Devices[J]. Financial Technology Time, 2013(7): 85-87.)
- [9] 曾令华. 基于 Puppet 的自动化配置系统设计及实现[D]. 广州: 华南理工大学, 2012. (Zeng Linghua. Design and Implementation of Automated Configuration System Based

on Puppet[D]. Guangzhou: South China University of Technology, 2012.)

### 作者贡献声明:

郑海山: 研究命题的提出、设计;

郑海山, 林俊伟: 研究过程的实施;

郑海山: 论文起草;

郑海山, 林俊伟: 最终版本修订。

(通讯作者: 郑海山 E-mail: haishion@xmu.edu.cn)

## Application of Open Source Software in Operation and Maintenance in the Data Center of Library

Zheng Haishan<sup>1</sup> Lin Junwei<sup>2</sup>

<sup>1</sup>(Information & Network Center, Xiamen University, Xiamen 361005, China)

<sup>2</sup>(Xiamen University Library, Xiamen 361005, China)

**Abstract:** [Objective] This paper aims to address problems of automatic operation and maintenance in university library's data center where a large number of servers work. [Context] The data center of university library is dealing with more and more tasks, which leads to the increasing number of servers. With both the virtualization platforms and traditional servers in the data center, manual work can no longer manage them all. [Methods] Search for suitable open source software; design, deploy, and test the best practices; study on the data structure of open source software for secondary development, and integrate multiple open source software. [Results] Successfully solve the problems which include: location management of servers and virtual machines, network interconnection structure, automatic deployment, IP allocation, monitoring, backup, centralized log management etc., and ultimately manage to demonstrate the readers a current server status page. [Conclusions] With the application of open source software, the library's data center reaches the goal of clearness, standardization, automation and external transparency.

**Keywords:** Data center Automatic operation Open source software