



基于开源软件的私有云计算平台构建^{*}

姜毅^{1,2}, 王伟军¹, 曹丽³, 刘凯¹, 陈桂强¹

(1. 华中师范大学信息管理学院 武汉 430079; 2. 浙江理工大学经济管理学院 杭州 310018;
3. 浙江理工大学信息学院 杭州 310018)

摘要:介绍了私有云的应用范围,讨论了自行开发、商业和开源解决方案各自的特点,分析了开源方案的主要优势,比较了主流的开源云计算项目 Eucalyptus、OpenStack、OpenNebula 和 Nimbus,重点研究了 OpenStack 架构。在此基础上,讨论了基于 OpenStack 的私有云计算平台构建方法,其主要过程包括基础环境配置、Nova、Glance、Swift、Keystone 和 Horizon 组件安装,虚拟机部署等,最后给出了该平台在云计算软件测试项目中的应用实例。

关键词:私有云;开源系统;OpenStack;KVM

doi: 10.3969/j.issn.1000-0801.2013.01.011

Construction of Private Cloud Computing Platform Based on Open Source Software

Jiang Yi^{1,2}, Wang Weijun¹, Cao Li³, Liu Kai¹, Chen Guiqiang¹

(1. School of Information Management, Central China Normal University, Wuhan 430079, China;
2. School of Economics and Management, Zhejiang Sci-Tech University, Hangzhou 310018, China;
3. School of Informatics and Electronics, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: This paper introduced the application range of the private cloud, discussed the characteristics of the self-developed and business and open source solutions, and analyzed the main advantage of the open source solution, then compared the mainstream projects of the open source cloud computing, such as Eucalyptus, OpenStack, OpenNebula and Nimbus, studied the structure of OpenStack mainly. Based on this, this paper presented the construction example of private cloud computing platform based on OpenStack, which major processes included configuration of base environment, components installation of Nova, Glance, Swift, Keystone and Horizon, deployment of virtual machines, and finally discussed the platform application examples in the cloud computing software testing.

Key words: private cloud, open source system, OpenStack, KVM

1 引言

云计算的应用越来越广泛,对相关问题的研究也逐渐

深入,不少公司和研究机构需要构建自己的私有云计算平台。商业解决方案存在价格高、授权模式复杂等问题,并且无法获得系统源代码,很多情况下并不适合采用。开源云

^{*} 湖北省自然科学基金资助项目(No.2011CDA116),芬兰科技创新基金资助项目(No.TEKES344/31/2011),浙江省自然科学基金资助项目(No.Y6110332)

计算技术具有成本低、灵活性高等优势,能够方便地获得源代码进行深入研究,因此很多机构都采用开源软件构造私有云计算平台。目前主要的开源云计算项目有 Eucalyptus、OpenStack、OpenNebula 和 Nimbus, 本文在对比这些项目的基础上,分析了 OpenStack 的架构,阐述了利用 OpenStack 构建私有云平台的方法。

2 私有云平台概述

2.1 私有云概念

从用户使用的角度来看,云计算平台可以分为公共云和私有云^[1]。公共云部署在互联网中,用户通过网络获得计算资源和存储资源,并按照“即用即付”(pay-as-you-go)的方式付费。私有云是企业或组织内部数据中心的资源,只提供给机构内部人员使用,不对外面向公众。私有云计算平台一般构建在机构的防火墙内部,相对于公共云,其安全性更高,服务质量更有保证。私有云能够充分利用现有的硬件资源和软件资源,可以有效地降低信息化成本。

2.2 私有云计算平台构建模式

私有云平台的构建可以采用 3 种模式:研究机构自行开发^[2],这种方式需要企业或研究机构有很强的技术实力,比较典型的是清华大学的透明计算平台;采用商业解决方案^[3],不少商业公司提供了较为成熟的私有云整体解决方案,比较典型的是 IBM 的“蓝云”计算平台;使用开源解决方案^[4],这种方案是利用开源系统构造企业或组织的私有

云平台,比较典型的是使用 Eucalyptus 和 OpenStack 等开源系统。3 种构造方式的比较见表 1。

2.3 开源私有云计算平台优势

在自行开发、商业解决方案和开源解决方案 3 种私有云的构建模式中,开源方案具有如下多个优势。

(1)使用成本低

用户只需要购买硬件设备和网络设备,无需为开源软件的使用付费。即使用户需要开源软件开发商提供服务,其需要付出的费用也相对比较低廉。例如,Ubuntu 帮助用户构建基于 OpenStack 的私有云平台的服务费用是 9 000 美元,相关工作在 5 天内完成^[5]。

(2)开放源代码

开放源代码,有利于研究人员对系统内部的实现机制进行深入研究。开源软件使用 Apache 或其他授权协议,用户能够获得系统的源代码,并且可以通过社区、讨论组等多种形式,与开发人员和其他用户深入探讨各种技术问题。

(3)平台可扩展性强

由于现有的云计算技术尚在快速发展中,没有形成统一的技术标准和规范,一旦选择了一种商业解决方案,就很难转换到其他平台。而开源云平台在设计时一般会保持对主流云计算技术(如亚马逊 EC2、S3 等)的兼容性,同时提供转换工具,有效地避免了厂商锁定问题,增强了系统的可扩展性。

表 1 私有云计算平台构建模式对比

	自行开发	商业解决方案	开源解决方案
技术难度	高,需要开发完整的计算、存储等云平台功能模块	低,主要是学习云平台的管理方法	中,需要进行技术选型、部署、维护等工作
费用	中,前期需要投入较多开发费用,边际成本低	高,需要购买昂贵的整体解决方案,边际成本高	低,无需商业授权费用,边际成本低
见效时间	很慢,前期技术研究和系统开发时间很长	很快,直接使用成熟的平台解决方案	较快,需要解决云平台的部署和管理等问题
硬件平台	无限制,由开发人员自行确定	限制较多,往往需要购买指定厂商的硬件设备	无限制,可以使用低成本的 x86 服务器
开展研究难度	容易,拥有项目的源代码、完整的各类文档等	困难,无法获得商业公司的云平台源代码	比较容易,一般可获得全部的系统源代码,但文档可能不完整
服务支持	最好,有完整的开发、测试和管理等内部人员的服务支持	较好,提供较多的商业服务,但收费也比较高	一般,开源组织比较松散,一般服务支持也较少
可扩展性	较好,系统可按照需要自行扩展	差,一般只能使用一种解决方案,存在厂商锁定问题	好,只需要保证系统兼容性即可,能有效避免厂商锁定问题



3 开源云计算项目现状

3.1 主流开源云计算项目介绍

目前已经有多个云计算平台的开源实现,主要的开源云计算项目有 Eucalyptus、OpenStack、CloudStack 和 OpenNebula 等,基本介绍如下。

(1) Eucalyptus

Eucalyptus^[6]是加州大学圣芭芭拉分校计算机科学系 Rich Wolski 领导的项目组完成的开源软件项目,该项目开发和应用较早,目前占有最大的市场份额。其组件无需修改即可使用,可以运行在没做修改的 GNU Linux 内核上。Eucalyptus 提供了与 Amazon 云计算平台 EC2 和 S3 接口的兼容性,可以使客户应用方便地从私有云平台迁移到公共云平台。

(2) OpenStack

OpenStack^[7]是美国国家航空航天局(NASA)和 Rackspace 合作研发的开源云计算平台项目,整合了 NASA 的 Nebula Platform 和 Rackspace 的 Cloud Files Platform 技术。OpenStack 得到了多个商业公司和开源组织的支持,被认为是未来开源云计算平台最具有潜力的技术,从 Ubuntu 11.04 版本开始,已经把 OpenStack 作为内置集成的云计算技术。

(3) CloudStack

CloudStack^[8]起源于 2008 年成立的 VMOps 公司,2010 年改名为 Cloud.com,分为社区版本和商业版本。2011 年 7 月思杰系统(Citrix System)公司收购了 Cloud.com,把以前

维护的商业分支全部合并到了开源分支,并把该软件加入 Apache 软件基金会,成为了一个完全开源的项目。目前全球已经有多个大规模应用实例,如 Zynga、GoDaddy 等。

(4) OpenNebula

OpenNebula^[9]起源于 2005 年 Ignacio M Llorente 和 Rubén S Montero 创立的一个研究项目,于 2008 年 7 月发布了第一个公开版本。OpenNebula 是一个虚拟化企业数据中心和云基础设施建设和管理的行业开源解决方案,具有开放性、模块化和可扩展的架构,目前已成为欧盟云计算项目 RESERVOIR 中虚拟执行环境管理器 VEEM 的核心组件。

3.2 开源云计算项目对比

Eucalyptus、OpenStack、CloudStack 和 OpenNebula 这 4 个主流的开源云计算项目各有特点,表 2 从多个方面对这 4 个项目进行了对比,其中社区活跃程度的数据来源于蒋清野博客^[10]。

在主流的开源云计算项目中,OpenStack 目前已经获得英特尔、惠普、戴尔、Ubuntu、思科等多个重要公司和开源组织的支持,是目前最热门的开源云计算项目,因此本文选择 OpenStack 构造私有云平台。

4 OpenStack 平台架构

4.1 OpenStack 服务架构

OpenStack 中有 Compute、Storage、Image、Identity 和 Dashboard 5 个子项目,其服务架构如图 1 所示。

其中,Compute、Storage、Image 是核心的 3 个子项目,

表 2 主流开源云计算项目对比

	Eucalyptus	OpenStack	CloudStack	OpenNebula
发布时间	2008 年 5 月	2010 年 7 月	2010 年 5 月	2008 年 7 月
最新版本	3.1	Essex	3.0	3.6
授权协议	GPL 3.0	Apache 2.0	Apache 2.0	Apache 2.0
基本架构	主要包括 CLC、CC、NC、Walrus、SC 5 个组件	主要有 Compute、Storage、Image、Identity 和 Dashboard 5 个子项目	主要包括管理服务、云基础设施和网络三大部分	主要包括接口与 API、用户与组、主机、网络、存储、集群 6 个部分
虚拟化技术支持	Xen、KVM、VMWare	Xen、KVM、VMWare、LXC、QEMU、UML	Xen、KVM、VMWare	Xen、KVM、VMWare
用户界面	命令行、简单的浏览器用户界面	命令行、简单的浏览器用户界面	命令行、较好的浏览器用户界面	命令行、简单的浏览器用户界面
社区活跃程度	总人数最多,但活跃用户较少	总人数较多,活跃用户数最多	总人数最少,活跃用户数较少	总人数较少,活跃用户最少
公共云平台支持	兼容亚马逊公共云平台	兼容亚马逊公共云平台	兼容亚马逊公共云平台	兼容亚马逊公共云平台
官方网站	eucalyptus.com	openstack.org	cloudstack.org	opennebula.org

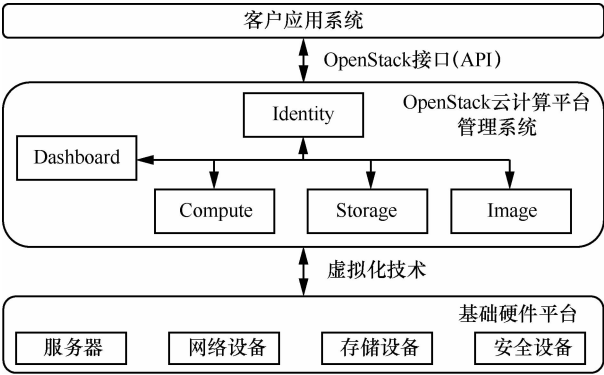


图 1 OpenStack 服务架构

提供计算、对象存储和镜像管理服务,其他的子项目包括提供身份认证的 Identity、基于 Web 的管理控制台 Dashboard 等。OpenStack 需要通过虚拟化技术使用基础硬件平台,客户应用系统可以通过 OpenStack 提供的接口(API)访问各个云服务实例。

4.2 OpenStack 主要实现

OpenStack 云计算平台管理系统目前有 Nova、Swift、Glance、Keystone 和 Horizon 5 个主要的具体实现。Nova 是 Compute 的具体实现,主要功能是计算资源管理和云平台控制服务管理,运行和管理虚拟服务器实例等;Swift 是 Object Storage 的具体实现,提供分布式云存储服务,是一个可扩展的对象存储系统;Glance 是 Image Service 的具体实现,提供虚拟机镜像的发现、注册、获取服务;Keystone 是 Identity 的具体实现,提供统一的身份认证服务;Horizon

是 Dashboard 的具体实现,提供通过 Web 用户界面管理云计算平台的方式。Nova、Glance、Keystone 的安装和使用都需要使用数据库,可以采用开源的 MySQL、PostgreSQL 或 SQLite 等数据库系统。

Nova 和 Swift 是核心的两类服务实现。Nova 组件是云平台的计算模块,提供多种服务,包括 API 服务(nova-api)、消息队列、计算服务(nova-compute)、网络控制(nova-network)、卷服务(nova-volume)和调度管理(nova-scheduler)等;Swift 组件提供云存储服务,适合存储大量、长期、需要备份的数据。Swift 通过“账户—容器—对象”(account-container-object)的形式提供对象存储服务,一个账户下可有多多个容器,每个容器中可存储多个对象。环服务是 Swift 最重要的组件,用于记录存储对象与物理位置间的映射关系。

5 基于 OpenStack 的私有云平台构建

5.1 总体说明

本文选择 OpenStack 来构造私有云平台实例,并采用 Ubuntu、KVM 等开源软件作为平台的基础操作系统和虚拟化工具。私有云平台使用的主要开源软件说明见表 3。

本文讨论的私有云平台实例主要应用在科研实验室环境中,服务器采用 x86 架构,硬件设备初始的投入成本较低,并可以根据需要动态扩展各种资源。私有云平台的初始硬件配置见表 4。

表 3 构造私有云平台的主要开源软件

名称	版本	说明	下载网站
Ubuntu	Server 12.04 LTS	开源 Linux 操作系统	www.ubuntu.com
KVM	1.0	开源虚拟化软件	linux-kvm.org
OpenStack	Essex	开源云平台管理软件	openstack.org
Python	2.7	OpenStack 二次开发语言	www.python.org
MySQL	5.5	开源关系型数据库	www.mysql.com

表 4 私有云平台的初始硬件配置

名称	内容	说明
CPU	Intel Xeon E3-1230	4 核 8 线程,支持 Intel VT 硬件虚拟化技术
内存	16 GB	需要时可扩展为 32 GB
硬盘	1 TB	需要时可扩展为 RAID 磁盘阵列
网卡	双吉比特以太网卡	分别连接内网和外网
服务器数量	2 个	一台控制节点,一台计算节点,可动态扩展



5.2 基础环境的配置

一个典型的私有云计算平台至少采用两台服务器,本文将其称为服务器 I 和服务器 II。服务器 I 上安装 Nova、Glance、Swift、Keystone 和 Horizon 全部组件,作为云平台的控制节点和存储节点;服务器 II 仅安装 Nova 组件,作为云平台的计算节点。在平台性能不能满足需要的情况下,可以动态扩展计算和存储等资源。每台服务器均配置 2 块吉比特以太网卡,第一块网卡 eth0 设置为外部可访问的 IP 地址,第二块网卡 eth1 设置为云平台各节点内部通信的 IP 地址。服务器 I 和服务器 II 的配置说明见表 5。

服务器 I 和服务器 II 首先需要安装基础操作系统(Base OS),本平台使用 Ubuntu Server 12.04 的 64 bit 版本。安装过程包括系统分区、设置初始用户、配置网卡的 IP 地址等过程,在选择系统组件时仅需要选择 Openssh-server。在 Ubuntu 安装完成后,需要使用 NTP (network time protocol)同步服务器时间,即在服务器 I 安装 NTP 服务器,其他服务通过该服务器进行同步。

由于 OpenStack 的 Nova 和 Glance 等组件需要使用关系数据库,本平台在基础操作系统中安装 MySQL 5.5 数据库,并使用下列 SQL 脚本构造 OpenStack 数据库:

```
CREATE DATABASE nova;
CREATE USER novadbadmin;
GRANT ALL PRIVILEGES ON nova.* TO 'novadbadmin'@'%';
SET PASSWORD FOR 'novadbadmin'@'%' = PASSWORD
('novasecret');

CREATE DATABASE glance;
CREATE USER glancedbadmin;
GRANT ALL PRIVILEGES ON glance.* TO 'glancedbadmin'@'%';
SET PASSWORD FOR 'glancedbadmin'@'%' =
PASSWORD('glancesecret');
```

```
CREATE DATABASE keystone;
CREATE USER keystonedbadmin;
GRANT ALL PRIVILEGES ON keystone.* TO 'keystoned
badmin'@'%' ;
SET PASSWORD FOR 'keystonedbadmin'@'%' = PASSWORD
('keystonesecret');
```

5.3 OpenStack 各组件的安装配置

OpenStack 一般按照 Keystone、Glance、Nova、Swift 和 Dashboard 顺序进行安装和配置,服务器 I 依次安装以上全部组件,服务器 II 只需要安装 Nova 组件。

(1) 安装和配置 Keystone 组件

该组件为 OpenStack 提供认证、授权、用户管理和角色管理等服务。在使用 apt-get 命令安装并连接数据库成功后,需要创建租户(tenant)、用户(user)和角色(role),关键代码如下:

```
keystone tenant-create-name tenant_name
keystone user-create-name user_name -pass user_password
-email user_email
keystone role-create-name role_name
然后将租户、用户和角色关联起来,关键代码如下:
keystone user-role-add-user $USER_ID-role $ROLE_ID-
tenant_id $TENANT_ID
```

需要注意的是,\$USER_ID、\$ROLE_ID 和 \$TENANT_ID 必须使用 OpenStack 分配的 ID 号码,该号码可以使用 keystone user-list、keystone tenant-list 和 keystone role-list 命令查询。

接下来是创建服务目录和端点(endpoint)信息,关键代码如下:

```
keystone service-create-name service_name -type service_
type -description 'Description of the service'
```

表 5 服务器 I 和服务器 II 的配置说明

项目	服务器 I	服务器 II
功能组件	Nova、Glance、Swift、Keystone、Horizon	Nova
网络接口	eth0 为外部访问接口,eth1 为内部通信接口	eth0 为外部访问接口,eth1 为内部通信接口
IP 配置	eth0:122.204.144.220	eth0:122.204.144.214
	eth1:192.168.3.1	eth1:192.168.3.2
主机名	Server1	Server2
DNS 服务器	218.104.111.114, 202.114.32.1	218.104.111.114, 202.114.32.1

Binary	Host	Zone	Status	State	Updated_At
nova-consoleauth	server1	nova	enabled	:-)	2012-07-10 10:13:23
nova-compute	server1	nova	enabled	:-)	2012-07-10 10:13:27
nova-scheduler	server1	nova	enabled	:-)	2012-07-10 10:13:23
nova-network	server1	nova	enabled	:-)	2012-07-10 10:07:50
nova-cert	server1	nova	enabled	:-)	2012-07-10 10:13:23

图2 Nova 服务状态

keystone endpoint-create -region region_name-service_id service_id -publicurl public_url-adminurl admin_url-internalurl internal_url

全部配置成功后,Keystone 组件安装完成。

(2)安装和配置 Glance 组件

Glance 提供虚拟机镜像管理服务,其安装配置比较简单,首先使用 apt-get 命令下载安装,然后配置租/用户和密码信息、连接 MySQL 数据库并配置环境变量,主要是修改/etc/glance/目录下的 glance-api-paste.ini、glance-registry-paste.ini 和 glance-api.conf 3 个文件。在配置成功 Glance 服务后,通过使用 Keystone 的授权机制,可以将虚拟机镜像上传到私有云计算平台中。

(3)安装和配置 Nova 组件

Nova 组件是计算模块,提供多种相关服务。首先使用 apt-get 命令安装,然后修改/etc/nova/nova.conf 配置信息,创建物理卷和卷组信息,改变/etc/nova 目录和/etc/nova/nova.conf 文件的权限,并通过/etc/nova/api-paste.ini 配置租/用户和密码信息,连接 MySQL 数据库。在这些操作完成后,需要为虚拟机实例设置待分配的 IP 地址段,关键代码如下:

```
sudo nova-manage network create private -fixed_range_v4=192.168.4.32/27- num_networks =1 -bridge =br100-bridge-interface=eth1-network_size=32
```

在导出环境变量并重启 Nova 服务后,可以使用 sudo nova-manage service list 查看 Nova 服务状态,如图 2 所示。

(4)安装和配置 Swift 组件

该组件提供云存储服务,其配置过程比较复杂。首先使用 apt-get 命令安装,然后设置 Swift 存储后端(backend)。存储后端即可以使用已经存在的分区/卷,也可以创建回送文件(loopback file)系统。本平台在服务器 I 分区时已经给 Swift 预留了约 500 GB 的磁盘空间,因此直接使用该分区。使用 XFS 对该分区进行格式化,然后挂载该分区,并根据需要创建一些节点。

然后配置文件分布工具 Rsync,Rsync 能够维护对象副本,需要被 Swift 的多个服务使用。再配置 Swift 组件信息,需要修改 paste.deploy 和/etc/swift/swift.conf 等文件。下一步

是配置 Swift 的代理服务,代理服务提供看守功能,用来响应客户请求,并负责其他组件间的相互通信过程,配置时需要修改/etc/swift/proxyserver.conf 文件。

接下来需要配置账号服务、容器服务和对象服务,主要是修改/etc/swift 目录下的 account-server.conf、container-server.conf、object-server.conf 3 个配置文件信息。最后配置 Swift 的环服务,所有的配置完成后,即可使用私有云存储服务。

(5)安装 Dashboard 组件

该组件提供 Web 形式的管理界面,其安装配置比较简单,直接使用 apt-get 命令安装,然后在浏览器中输入服务器 I 的 IP 地址即可访问 Dashboard 服务。

5.4 配置虚拟化环境

Nova 组件本身不提供虚拟化的能力,而是通过 libvirt API 与虚拟机进行交互,支持 Xen、KVM、VMware、UML (user mode Linux)、LXC 和 QEMU 等虚拟化技术。这些技术中,Xen 和 KVM 是最主流的开源虚拟化项目。Xen 使用半虚拟化技术,能够在不支持硬件虚拟化的服务器上运行,其架构比较复杂。KVM 是轻量级的虚拟化管理程序模块,目前已经被集成到 Linux 内核中,只支持全虚拟化技术,需要服务器支持硬件虚拟化技术。本文讨论的私有云计算平台服务器的 CPU 为 Intel Xeon E3-1230,支持 Intel VT 硬件虚拟化技术,因此选择 KVM 虚拟化技术。

首先安装 KVM,这一过程比较简单^[11]。然后制作操作系统镜像文件,这些镜像既可以从网络中下载,也可以自行制作。由于不同的应用环境需要默认安装的应用软件各不相同,所以一般需要自行制作镜像。以 Ubuntu 为例,自行制作镜像首先需要创建 img 文件,关键代码如下:

```
kvm-img create -f qcow2 server.img 10G
```

然后利用 KVM 虚拟光驱加载操作系统 ISO 文件,完成虚拟机中操作系统的安装。在镜像制作完成后,通过 Glance 命令上传到私有云平台中,关键代码如下:

```
glance add name = " <Image name > " is_public =true container_format=ovf disk_format=qcow2 < <filename>.img
```

在镜像上传成功后,就可以在私有云计算平台中创建虚拟机实例。创建过程主要步骤包括创建授权、加载镜像、



连接和使用镜像等。

5.5 管理 OpenStack 服务

在需要的服务配置完成后,可以通过 Dashboard 对 OpenStack 的各项服务进行在线监控。OpenStack 服务状态的 Web 监控如图 3 所示。

Services			
		Filter	Filter
Name	Service	Host	Enabled
nova	compute	122.204.144.220	Enabled
glance	image	122.204.144.220	Enabled
volume	volume	122.204.144.220	Enabled
ec2	ec2	122.204.144.220	Enabled
swift	object-store	122.204.144.220	Enabled
keystone	identity(native backend)	122.204.144.220	Enabled
Displaying 6 items			

图 3 OpenStack 服务状态的 Web 监控 (Dashboard)

Dashboard 同时提供了对 OpenStack 各服务在线管理的功能,主要包括管理虚拟机实例、管理操作系统镜像、管理默认硬件配置文件、管理用户、管理卷等,其图形化操作界面比命令行模式更加友好。

6 开源私有云平台运行实例

6.1 云计算软件测试平台说明

在笔者参与的研究项目中,目前已经利用 OpenStack、KVM 等开源软件构造了私有云计算平台,该平台用于进行云计算软件测试项目。用户通过浏览器登录和使用云测试项目管理系统,完成申请与执行测试项目、配置虚拟测试环境(用术语“硬件矩阵”和“软件矩阵”表示)、分配和使用虚拟测试服务器等操作,系统的业务流程如图 4 所示。

测试用户可选择合适的测试环境,申请需要的 CPU、内存和磁盘等硬件资源和应用服务器、数据库等软件资源,通过云平台实现快速部署。用户对云测试平台计算、存储和镜像等资源的调用需要使用 Keystone 组件的认证服务,需要长期保存和使用的测试数据需要使用 Swift 组件的云存储服务。云测试平台提供了 Java、.net 和 PHP 3 种开发技术的云测试环境,3 种云测试环境的详细说明见表 6。

6.2 云测试执行实例

笔者使用了私有云平台的虚拟测试服务器进行了某信息采集软件客户端的功能测试,硬件配置选择 1CPU、512 MB 内存和 2 GB 硬盘空间,软件环境选择了 Java 的 Web 软件测试环境。创建测试项目的系统界面如图 5 所示。

虚拟测试服务器的创建分为自动创建和手工创建两

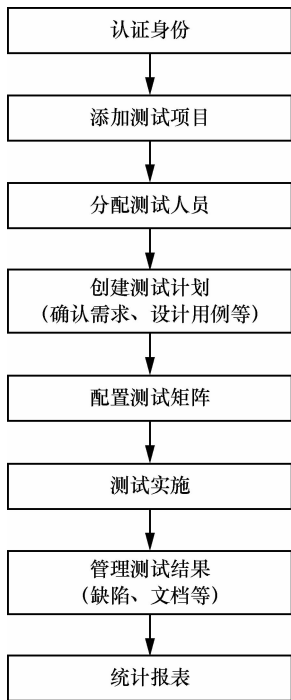


图 4 云计算软件测试项目管理系统业务流程

表 6 3 种云测试环境说明

名称	Java 测试环境	.net 测试环境	PHP 测试环境
操作系统	Ubuntu Server 12.04	Windows Server 2003	Ubuntu Server 12.04
开发语言	Servlet 3.0/JSP 2.2	.net 3.5	PHP 5
Web 服务器	Tomcat 7	IIS 7	Apache 2.2
数据库	MySQL 5.5	SQLServer 2008	MySQL 5.5

添加新项目

项目名称: 同信息采集客户端测试项目

项目经理: [输入框]

公司名称: [输入框]

测试类别: ☒ 单元测试 ☐ 压力测试 ☐ U测试 ☒ 功能测试

实施时间: 2012-09-03

开始时间: 2012-09-04

结束时间: 2012-09-05

硬件矩阵配置: 小型(1 CPU/512MB RAM/2 GB HDD) [新建]

软件矩阵配置: ☒ Java Web ☐ PHP/LAMP ☐ .NET [新建]

测试组件: ☒ AutoRunner ☐ TestCenter ☐ Visual Unit ☐ WebBench ☐ LoadRunner [新建]

[添加] [关闭]

图 5 创建测试项目的系统界面

种方式。如果测试矩阵变量的默认取值可以满足用户需要,用户配置完参数后系统通过调用 OpenStack 的 API 自动开启虚拟测试服务器。如果默认取值不符合用户要求,

用户可以填写新建矩阵需求表,由系统管理员根据实际情况进行手工添加。

用户一般通过远程连接对虚拟服务器进行管理。Linux 平台使用 OpenSSH 的命令行界面和 VNC 的图形界面, Windows Server 平台使用 Windows 系统中的“远程桌面”工具。用户远程连接进入虚拟测试服务器后,就可以配置和使用该虚拟服务器,其使用方式和使用普通物理服务器一致。

7 结束语

利用开源软件构造私有云计算平台具有成本低、部署快、无特殊硬件限制等多个优势,将在教学、科研和商业中得到更加广泛的应用。OpenStack 是最值得关注的开源云计算项目,但是目前其配置过程比较复杂。接下来的研究中,笔者将编写自动化安装脚本,以简化 OpenStack 的部署和应用难度。同时深入研究 OpenStack 各组件的 API,根据研究需要对 OpenStack 进行二次开发。

参考文献

- Berkeley. Above the clouds: a berkeley view of cloud computing. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>, 2011
- 陈康, 郑纬民. 云计算:系统实例与研究现状. 软件学报, 2009, 20(5):1337~1348
- 张帆, 李磊, 杨成胡等. 基于 Eucalyptus 构建私有云计算平台. 电信科学, 2011(11):57~61
- 秦润锋, 樊勇兵, 唐宏等. 开源云计算管理平台技术在电信运营商私有云建设中的应用研究. 电信科学, 2011(10):24~29
- Ubuntu. A working private cloud for just \$9,000. <http://www.ubuntu.com/cloud>, 2012
- Eucalyptus Systems. Eucalyptus cloud computing software. <http://www.eucalyptus.com/>, 2012
- OpenStack. Open source software for building private and public clouds.<http://openstack.org/>,2012
- CloudStack. Open source cloud computing with cloud stack. <http://www.cloudstack.org/>,2012
- OpenNebula. The cloud data center management solution. <http://opennebula.org/>,2012
- 蒋清野. CY12-Q2 OpenStack, OpenNebula, Eucalyptus, CloudStack 社区活跃度比较.<http://www.qyjoho.net/?s=活跃度&x=0&y=0>, 2012
- 秦学东. 开源虚拟化—KVM 的构建. 现代图书情报技术, 2011(11):89~92
- 【作者简介】 姜毅,男,浙江理工大学讲师,华中师范大学博士研究生,主要研究方向为云计算环境下的软件测试、个性化信息服务。王伟军,男,博士,华中师范大学教授、博士生导师,主要研究方向为信息资源管理、电子商务、知识管理。曹丽,女,浙江理工大学讲师、博士研究生,主要研究方向为模式识别、视觉检测。刘凯,男,华中师范大学博士研究生,主要研究方向为云计算环境下的软件测试。陈桂强,男,华中师范大学本科在读,主要研究方向为 Web 开发技术。

(收稿日期:2012-09-10)

(上接第 67 页)

- Li C, Lu S T, Liu J. Adaptive double threshold energy sensing algorithm based on compressed sensing over wideband frequency. Proceedings of IEEE the 3rd International Conference on Communications and Mobile Computing, Qingdao, China, 2011: 230~233
- Wang Y, Pandharipande A, Polo L Y, *et al.* Distributed compressive wide-band spectrum sensing. Proceedings of 2009 Information Theory and Application Workshop, San Diego, CA, USA, 2009: 178~183
- Hong S. Direct spectrum sensing from compressed measurements. Proceedings of Military Communications Conference, San Jose, California, USA, 2010:1187~1192
- Wei D, Milenkovic O. Subspace pursuit for compressive sensing signal reconstruction. IEEE Transactions on Information Theory, 2009, 55(5): 2230~2249
- Tropp J A, Wakin M B, Duarte M F, *et al.* Random sampling for analog-to-information conversion of wideband signals. Proceedings of IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software, Richardson, TX, USA, 2006: 119~122
- 刘兆霆, 何劲, 刘中. 基于压缩感知的高分辨率频率估计. 信号处理, 2009, 25(8): 1252~1256
- Lu Y, Guo W B, Wang X, *et al.* Distributed streaming compressive spectrum sensing for wide-band cognitive radio networks. Proceedings of 2011 IEEE the 73rd Vehicular Technology Conference, Budapest, Hungary, 2011: 1~5
- 【作者简介】 章坚武,男,博士后,杭州电子科技大学教授,主要研究方向为无线通信与网络信号处理等。池景秀,女,杭州电子科技大学硕士研究生,主要研究方向为基于 CS 的认知无线网络频谱感知和多用户资源分配。许晓荣,男,博士,杭州电子科技大学讲师,主要研究方向为认知无线网络中的资源分配与资源共享等。

(收稿日期:2011-09-28)