



一种基于工作流的云系统自动化部署架构

张 园,张云勇,房秉毅

(中国联合网络通信有限公司研究院 北京 100032)

摘 要:结合现有 OpenStack 主流开源部署工具现状,提出了一种基于工作流的云系统自动化部署架构,可按需求制定部署策略,从而实现云系统组件的定制化部署,对云系统的物理资源、软件资源、网络资源进行动态可视化部署管理,解决了节点部署过程中可靠性、可定制化、可扩展性问题。

关键词:云计算;OpenStack;工作流;自动化部署

doi: 10.3969/j.issn.1000-0801.2014.11.003

A Kind of OpenStack Auto-Deployment Architecture Based on Workflow

Zhang Yuan, Zhang Yunyong, Fang Binyi

(China Unicom Research Institute, Beijing 100032, China)

Abstract: Combined with the existing mainstream open-source deployment tools, a kind of OpenStack auto-deployment architecture based on Workflow was proposed. According to this architecture, the system can establish the deployment strategy to realize the cloud system components customized deployment and govern the physical resources, software resources and network resources of the cloud system, solved the node deployment process reliability, scalability and customizable problem.

Key words: cloud computing, OpenStack, Workflow, auto-deployment

1 引言

云计算的发展所带来的业务量、信息量,迫切需要规模化的数据中心提供平台支持,数据中心管理的 IT 资源是高度虚拟化的,与传统的数据中心相比,云数据中心需要对资源进行动态部署,从而节省成本。云计算追求更快、更高效、更灵活、功能更齐全的部署方案,快速部署就是一项重要的功能需求。在云环境中不但资源和应用规模的变化范围大而且动态性高,用户所需的服务主要采用按需部署的方式,即用户随时提交对资源和应用的请求,云环境管理程序负责分配资源、部署服务。另外,部署过程所支持的软件系统形式多样,系统结构各不相同,部署工具应能

适应被部署对象的变化^[1]。

OpenStack 作为一个开源的云计算管理平台,以其节约用户成本、提高运维效率以及更有利于企业创新等特点在各行各业都得到了广泛的部署和应用。然而,部署并且管理一个大规模的 OpenStack 集群是一件十分耗时且容易出错的工作,这主要是因为 OpenStack 相关组件的复杂依赖关系、众多配置参数以及成百上千台服务器的集群规模。因此,如何快速并且自动化地部署大规模的 OpenStack 集群成为一个值得深入研究的课题。

2 OpenStack 组成

OpenStack 是一个开源代码的云计算管理平台项目,

它本身是由多个子组件合作完成向用户提供计算、存储和网络等服务资源的任务。OpenStack 包含的子组件主要包括计算组件 Nova、镜像管理组件 Glance、虚拟网络管理组件 Neutron、块存储组件 Cinder、对象存储组件 Swift、身份认证组件 Keystone 和控制面板组件 Horizon 等,各个组件也可以单独向用户提供资源服务,各组件之间的关系如图 1 所示^[2]。

2.1 计算组 Nova

Nova 是 OpenStack 中最重要也是最复杂的子组件,它包含多个服务的集合,用于管理虚拟机实例的整个生命周期,其中包括虚拟机实例启动、暂停、重启、终止等操作。Nova 主要包含以下几个主要服务:

- nova-api 向用户提供了一个和云计算基础设施交互的接口;
- nova-compute 负责管理虚拟机实例的生命周期;
- nova-scheduler 根据适当的调度算法决定虚拟机实例运行的计算节点;
- 消息队列服务器用于 Nova 组件中各服务之间的消息通信;
- 数据库负责存储云平台的构建和运行时信息;
- Nova 组件还提供了通过代理访问虚拟机实例的控制台服务,这个服务由 nova-console、nova-xvncproxy、nova-spicehtml5proxy 和 nova-consoleauth 合作提供;
- Nova 还与其他子组件交互获取虚拟机实例运行时

需要的各种资源。

2.2 镜像管理组 Glance

Glance 作为 OpenStack 的镜像管理组件,主要为 OpenStack 提供虚拟机镜像的查找和存取服务。Glance 组件的构成包括 3 个主要部分:负责接收用户请求的 glance-api、负责与 Glance 组件的数据库交互的 glance-registry、负责实际存储虚拟机镜像文件后端实现的镜像存储,Glance 组件以插件的方式支持多种后端存储实现。

2.3 虚拟网络管理组 Neutron

Neutron 负责提供虚拟网络的管理服务,最早起源于 nova-network 服务,它提供一种强大的 API 定义其他服务的设备(如 Nova 的虚拟机实例)的网络连接和地址。为了描述实际的网络资源,Neutron 定义了 3 个抽象模型。

- 网络:一个二层的隔离,类似于 VLAN 在物理网络中的作用。
- 子网:一个 IPv4 或者 IPv6 的地址,与配置状态相关。
- 端口:单个设备连接到虚拟网络的连接点,描述网络的相关配置。

2.4 块存储组 Cinder

Cinder 是由 nova-volume 服务发展而来,负责为虚拟机实例提供可挂载的逻辑卷服务,以实现虚拟机数据的持久化存储。Cinder 主要由以下 3 个部分组成:

- 负责接收用户请求接口服务 cinder-api;
- 负责调度任务的 cinder-scheduler;

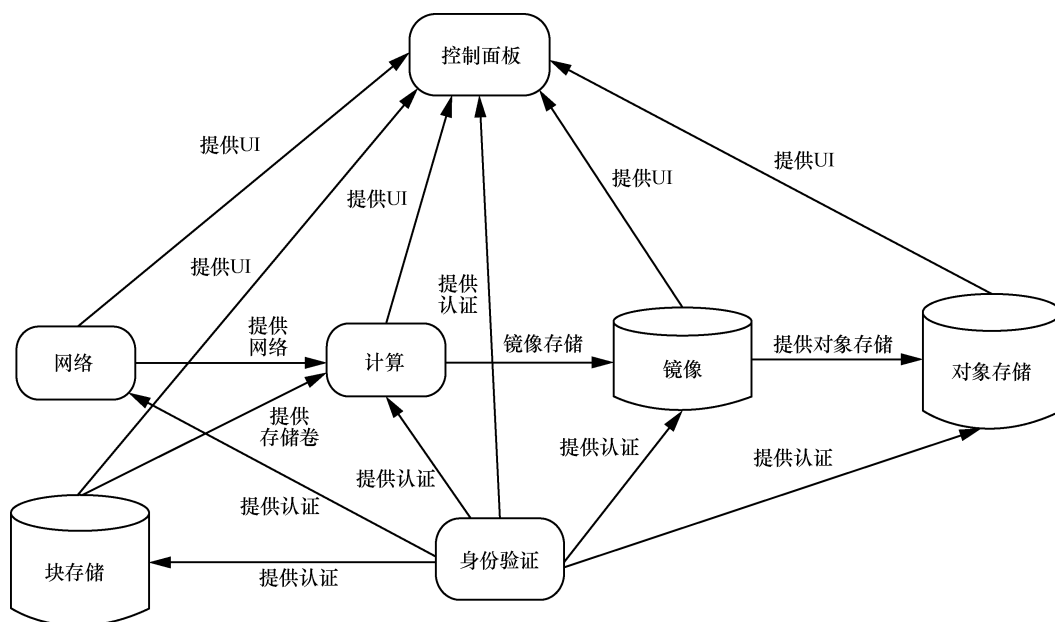


图 1 OpenStack 组件关系



- 若干个位于存储节点的 cinder-volume, cinder-volume 服务负责管理逻辑卷的存储空间, 存储节点的组合构成一个共享的存储资源池。

2.5 对象存储组 Swift

Swift 是 OpenStack 的对象存储组件, 可以作为 Glance 组件的后端存储虚拟机镜像文件。具有提供大规模的数据存储, 将每个对象复制多份分别存储在多个存储服务器之上以提供高可用和高扩展等特性。Swift 主要包括以下几个组成部分。

- ring 是 Swift 中最重要的服务, 它主要存储了对象与存储对象的物理位置之间的映射关系。
- proxy server 是用户访问 Swift 服务的接口。
- storage server 负责提供实际的存储服务。
- consistency server 负责解决存储服务的数据损坏和硬件故障引起的问题。

2.6 身份认证组 Keystone

Keystone 为 OpenStack 中的其他组件提供统一的身份验证和授权服务, 它作为一个整体集成了 identity、token、policy 和 catalog 等功能。Keystone 区分服务和管理两类 API, 负责处理用户请求和配置 identity、token、policy 和 catalog 服务信息。每一种 Keystone 功能支持可插拔的后端实现, 允许以不同方式实现特定的服务。

3 开源自动化部署工具

诺达咨询最新推出的《开源云及 OpenStack 应用分析报告 2014》显示, 潜在用户计划未来部署开源云方案中, 有 60% 的选择 OpenStack, 然而由于 OpenStack 组件的复杂依赖关系、众多配置参数, 各大开源服务公司纷纷推出自己的 OpenStack 部署工具。

3.1 RDO/Devstack

RDO 是 RedHat 的 OpenStack 开源项目, 包含核心的 OpenStack 组件——Nova、Glance、Keystone、Cinder、Quantum、Swift 和 Horizon, 还有用于云应用协调的孵化项目 Heat 以及用于资源监视和计量的 Ceilometer。利用 RedHat 开发的安装工具 Packstack, 可以实现非常简便轻松的安装过程。Packstack 是一个命令行工具, 支持 Fedora、centOS、RHEL, 使用 Puppet 模块支持 OpenStack 快速部署在现有的服务器, 提供的交互是部署选项, 通过命令行或者通过一个包含预先配置的文本文件实现 OpenStack 裸机和组件安装。

DevStack 是一套 shell 脚本, 可以用来快速搭建 OpenStack 的运行和开发环境, 利用 DevStack 脚本安装 git 源码, 然后下载 DevStack 代码到本地, 运行 stack.sh 脚本依次设定 MySQL、RabbitMQ、OpenStack Dashboard 和 Keystone 的密码, 密码输入后, stack.sh 脚本会自动开始安装必要的软件包和库并下载最新的 OpenStack 及其组件代码, 整个过程自动完成无需干预。

3.2 SUSE Cloud

SUSE Cloud^[3] 是 SUSE 为企业提供无缝部署开放标准的私有云功能。利用 Crowbar 软件框架, SUSE Cloud 管理服务器提供了集成化、可重复的安装框架及流程, 实现了关键云服务的高可用性配置和部署, 支持快速部署和轻松管理基于 OpenStack 的云基础设施, 主要实现 OpenStack、Ceph 等安装部署。通过易于使用的图形化界面, 支持物理服务器节点管理、组管理、网络配置、裸机部署、OpenStack、Ceph 组件部署及配置文件参数编辑功能。

3.3 MAAS+Juju

MAAS+Juju 是 Canonical 推出的部署工具, 可以用在 OpenStack 的部署, 类似 Puppet、Chef 的部署工具。MAAS 用于对云环境下物理服务器的管理, 包含服务器的自动扫描、节点管理、IP 池管理、网络管理、zone 管理、用户管理、镜像管理、裸机部署等。Juju 是用于部署应用, 如 MySQL 集群、Hadoop 等, Juju 不仅支持虚拟化环境, 同样也可以在物理机上面部署 Juju, 同时也兼容 OpenStack、EC2 的所有 API、Ubuntu MAAS、local Dev 和 fully abstracted 等。

3.4 Fuel

Fuel 是 Mirantis 公司推出的一个用于快速部署 OpenStack 云的工具, 提供 GUI 和 CLI 的部署方式, 采用 Cobbler 和 Puppet 开源工具实现多云系统环境管理、节点管理、角色管理、裸机部署、网络管理、配置管理、关键参数设置、健康检查、日志管理等, 可进行 OpenStack 和 Ceph 快速部署, 支持 HA 和多节点模式以及包含 OpenStack dashboard 用户名密码、vCenter IP 用户名、密码、管理器类型、调度方式、存储及系统日志相关设置。

3.5 Chef/Puppet Labs

Chef/Puppet Labs 是开源的系统配置管理工具, 基于 C/S 的部署架构, 为实现数据中心自动化管理而设计的配置管理软件, 可实现包括升级软件包、管理配置文件、系统服务、Cron 任务以及添加新的配置、修复错误等, 在设计上

可同时管理上千台服务器。penstack-ci 社区的 puppet-openstack 项目用于完成 OpenStack 服务的自动化部署和管理,目前已经在 stackforge 中托管并通过 OpenStack 的 Gerrit 系统来管理代码;此外,Cisco、RedHat、Mirantis 等多家公司的 OpenStack 发行版或部署工具中均使用到了 puppet-openstack。

3.6 小结

由于 OpenStack 组件多,逻辑关系复杂,手动安装繁杂而且容易出错,很多公司都投入人力进行开源,使安装、配置变得更加简单易用。然而,目前没有一种开源的 OpenStack 部署软件能应用到实际生产环境中,还处于搭建开发和测试环境阶段。其中 Devstack/RDO 部署过程的可定制性和灵活性不是太好,无法按需增加组件和功能,无法实现一键部署、可扩展、可添加任意计算节点,且组件的版本无法控制,正如 DevStack 官方所强调的,DevStack 不适合用在生产环境。SUSE Cloud/Fuel 实现了 Web 界面管理功能,裸机安装、节点管理、OpenStack 组件部署更加方便,但目前支持 SUSE linux 操作系统,Fuel 支持 Ubuntu,且无 OpenStack 组件管理软件仓库,不能对定制化的组件进行部署。Chef/Puppet labs 作为配置管理工具,需二次开发实现集群管理、OpenStack 组件安装,由于 OpenStack 组件间逻辑关系复杂,Chef/Puppet 只能对单个对象(服务器)操作,无法实现多组件一键部署。

4 基于工作流的自动化部署架构

目前没有一种开源部署软件能很好地用于生产环节中,基于目前主流开源 OpenStack 部署软件的优缺点,提出一种云环境下数据中心的自动化部署的新模型。通过基于自动部署工作流来完成对数据中心物理资源的自动部署问题,通过自定义数据模型解决数据中心部署服务定制化、动态化、可靠性问题,自定义的数据模型是将服务器、存储等硬件资源和逻辑概念定义到自动部署 workflow 数据库中。自动部署 workflow 引擎是实现资源自动化部署的核心机制,将相应服务器资源、存储资源、网络资源、软件资源、安装策略等整合到一个集中、强健的数据库中,用于调用和触发自动部署过程中的工作流,从而形成集中和强健的自动部署工作流。

4.1 设计原则

云计算自动化部署架构设计需要满足云环境下的部署任务业务驱动原则,以简化业务流程为初衷,设计出的

云计算自动化部署模型需应用到各个规模的数据中心,需符合一定的标准化原则,且平台架构需要为各个等级的 IT 人员所使用(从操作员到维护人员,到数据中心领导层),需要达到易使用、易操作的效果。云环境下的自动化部署目标如下。

- 实现管理规范:以流程为导向,建立一系列规范化的管理流程,实现可靠、安全、高效的运行要求。
- 工作信息化:实现信息化管理,提高部署工作的质量,使人员从大量的手工式模式中解放出来。
- 规避风险,降低运维成本:提高预警和监管能力,规避运行风险,减少不必要的疏忽带来的损失。
- 实现 IT 与业务系统的全面结合:可有效提高 IT 对业务系统价值的影响,整体上提高了业务运营质量。
- 使系统能够自己定位故障并解决,提高业务的执行效率^[4]。

4.2 总体架构

云系统自动化部署系统整体架构可分成 5 层:物理资源层、执行层、接口层、业务层和展现层。物理资源层是数据中心需云化的物理资源,主要有物理服务器、网络和存储;执行层提供自动化部署功能,完成裸机安装、配置执行、软件安装、云系统组件安装等;接口层支持业务层各种请求;业务层提供展现层业务逻辑,并利用 workflow 引擎实现对部署服务的流程定义;展现层主要实现物理资源管理、软件资源管理、模板管理、策略管理、策略/服务定义、资源调度、自动化部署和监控管理等功能。云系统自动化部署系统总体架构如图 2 所示。

在图 2 中,在云系统自动化部署系统中以 workflow 引擎为核心,整合各种物理资源和软件资源,按需求制定部署策略,可以有处理部署流程的依赖关系,增强部署服务可用性、部署服务的可靠性,从而实现云系统组件的定制化部署,对云系统的物理资源、软件资源、网络资源进行动态可视化部署管理^[5]。

4.3 工作流引擎

工作流引擎利用任务路由和任务分配功能将 workflow 活动中的相关资源整合起来,实现自动化控制,利用 workflow 引擎来创建部署流程,在定义流程的过程中分配设定资源分配模式,实现引擎和外部应用的触发和交互^[6]。自动化部署 workflow 引擎系统模型如图 3 所示。

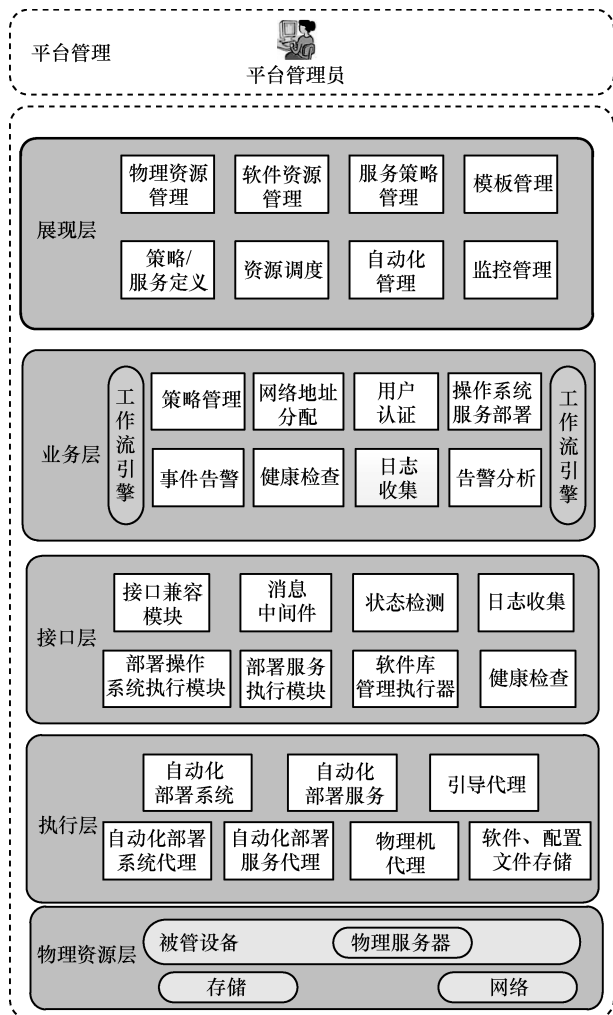


图2 云系统自动化部署系统架构

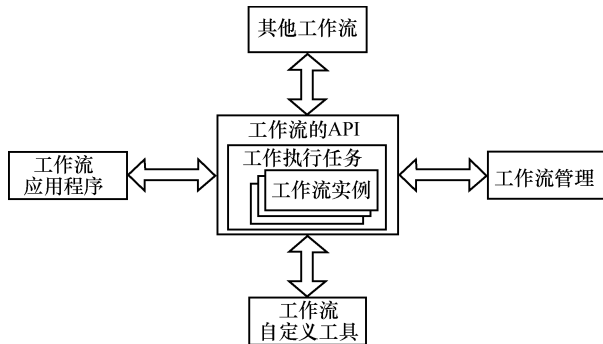


图3 自动化部署 workflow 系统模型

在云环境下的数据中心中,资源部署是指将原始的资源变成可以使用资源的过程,主要是通过安装和配置来实现。这种资源包括硬件资源(服务器)、软件资源(中间件或数据库等)以及网络资源和存储资源,将这些组件都定义到数据模型中,自动部署 workflow。数据中心要实现资源的快速自动部署,首先要规划系统资源的部署过程,接下来确定流程先后的关系、各个步骤的操作,然后将手动过程脚本化,定制为自动化流程,最后调试自动化的资源部署。将硬件、软件、逻辑概念定义到数据模型中,通过系统提供商提供的管理工具,将这些资源标识到组件资产中^[7]。由于云数据中心存在流程控制复杂的特点,系统采用了基于 workflow 来对自动部署系统的部署流程进行控制,将各种脚本整合到系统数据库中,使系统功能更加完善,形成一个强健和可重复使用的 workflow。组件部署 workflow 模型如图4所示。

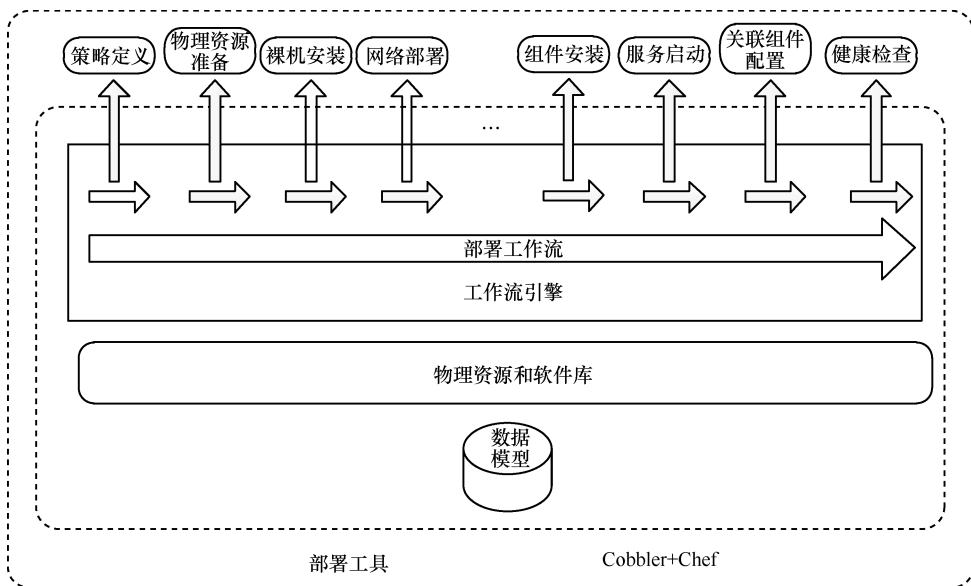


图4 组件部署 workflow 模型

4.4 系统设计

基于工作流引擎的自动化部署系统分布式框架中,云系统部署管理门户分为展现层、业务层和接口层,主要利用工作流引擎和消息中间件完成物理服务器管理、软件资源管理、网络管理、策略管理、配置管理、工作流引擎以及执行服务器管理等功能;执行层的部署执行模块完成具体部署动作,每个集群中部署一个部署执行服务器,主要运行 Cobbler 和 Chef 服务,利用 Cobbler 完成对集群内物理服务器的自动发现、裸机安装操作, Chef 与目标机代理进行通信,完成组件下发、组件安装以及配置管理功能,实现对云系统组件的安装及配置工作,详细设计如图 5 所示。

整个系统的对外交互是通过门户系统来完成的,用户执行一个部署软件的任务来触发系统进行一系列的动作,

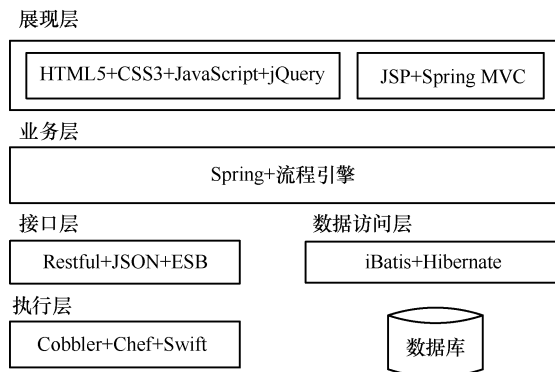


图 5 自动化部署系统设计

最终将用户请求的任务完成。在进行部署之前,用户需要对软件进行标准化描述与配置。并且,为了支持软件的自动化部署,软件提供者必须随软件提供每个组件的自动化安装包。部署的整体流程如图 6 所示。

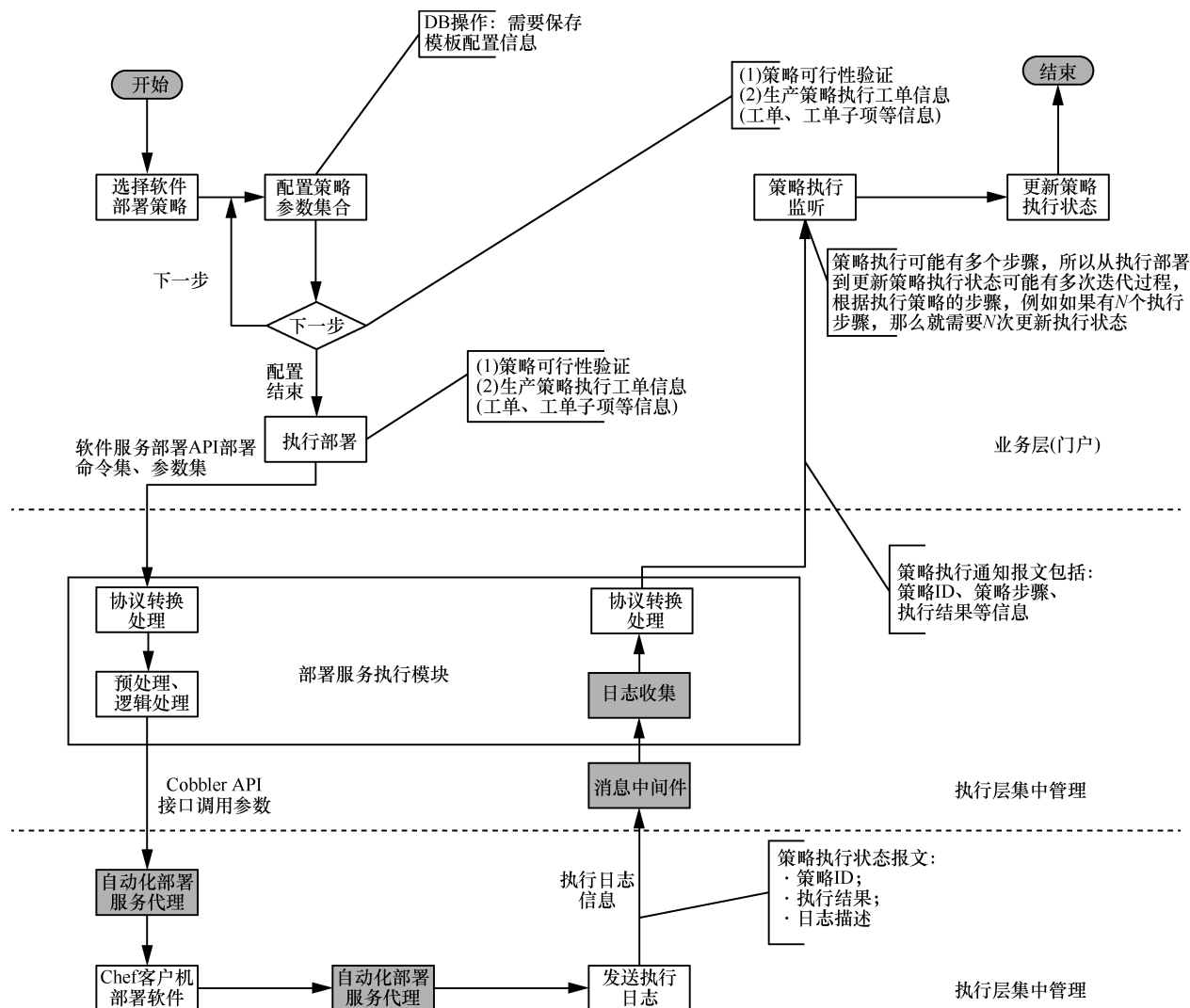


图 6 云组件部署关键流程



流程具体描述如下:

(1)软件部署用户需要准备待部署的软件,保证含有正确的自动安装包,定义相关的部署流程、配置策略参数等;

(2)选择相关部署策略,提交部署任务;

(3)对部署策略进行相关协议转换,预处理及逻辑处理后,将任务发送到工作流引擎;

(4)工作流引擎对部署任务流程化,运行安装定义的部署流程,并启动部署监控;

(5)根据部署流程,调用 Cobbler 模块,并从 Swift 获取操作系统文件,对目标机进行操作系统安装;

(6)根据部署策略配置参数,调用 Chef 模块,对目标机进行系统配置;

(7)从 Swift 获取安装包,调用 Chef 模块,执行云组件安装包部署,并进行参数配置;

(8)对部署流程进行监控,安装完成后进行健康检查,测试通过后反馈部署流程结束。

4.5 部署方案

图7中,云系统管理门户负责物理服务器管理、软件资源管理、网络管理、策略管理、配置管理、工作流引擎等功能;每个集群中控制单元主要通过开源软件 Cobbler 和 Chef 实现对集群内物理服务器的自动发现、裸机安装、组件下载、组件安装以及配置管理功能。

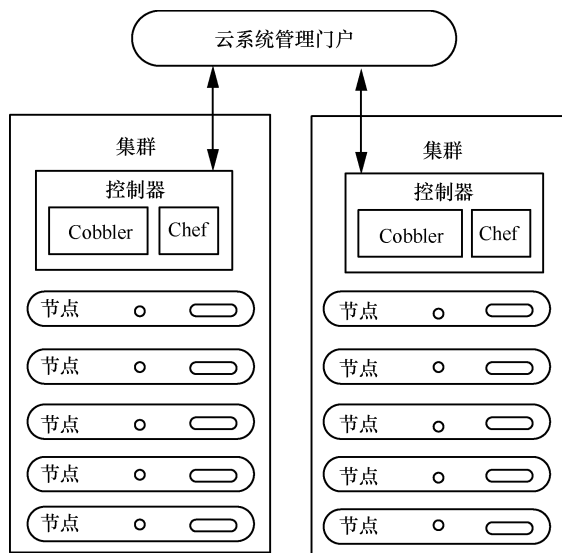


图7 部署架构

4.6 性能测试

在实验室环境的服务器集群构建过程中^[8],使用基于工作流的自动化部署系统来安装 OpenStack 云系统,

部署服务器和目标节点都是 HP380,硬件配置:2个 Xeon 2.4 GHz CPU,内存 8 GB,IDE 硬盘,100 Mbit/s 网卡。网络物理连接是:10个节点分为2组,每组5个节点连接到一个 100 Mbit/s 交换机上,部署服务器连到 100 Mbit/s 交换机。

部署需要的总时间是 35 min。测试中发现部署服务器的 CPU 利用率很低,始终小于 10%;系统性能的瓶颈是部署服务器网卡速率。从图8也可以看出,部署过程中网络流量平均为 80 Mbit/s,达到了饱和。

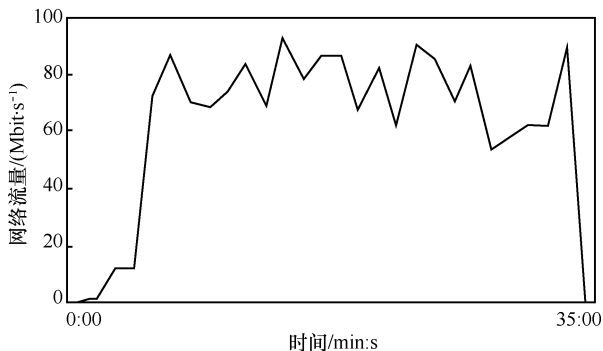


图8 部署时的网络流量

5 结束语

随着云计算的发展,云环境下的数据中心规模不断增大,极大增加了计算部署的工作量。目前,云环境下数据中心的自动部署需要解决的难题有部署速度慢、自动化程度低、部署服务器故障和部署服务器瓶颈等。系统部署速度会影响整体的工作效率,自动化程度可以减小出错率和降低系统的工作效率,部署服务器的故障会导致节点安装中断,部署服务器的瓶颈会影响物理资源的部署速度^[9]。云数据中心环境部署包括了硬件资源(服务器)、软件资源(中间件或数据库等)、网络资源和存储资源,其中有虚拟化设备和不能被虚拟化的设备。本文研究了 OpenStack 主流的开源部署工具,如 SUSE Cloud、MAAS+Juju、Devstack、RDO、Fuel 等,对其优劣进行分析,从软件整合和服务集成角度提出了一种基于工作流的云系统自动化部署架构,以工作流为核心,整合各种物理资源和软件资源,按需求制定部署策略,可以有处理部署流程的依赖关系,增强部署服务可用性、部署服务的可靠性,从而实现云系统组件的定制化部署,对云系统的物理资源、软件资源、网络资源进行动态可视化部署管理,解决了节点部署过程中可靠性、可定制化、可扩展性问题。

参考文献

- 1 谢涛.云环境下数据中心自动部署系统.华中科技大学硕士学位论文,2013
- 2 高贵升.基于 OpenStack 的计算云的研究与实现.成都理工大学硕士学位论文,2012
- 3 南俊杰,俞岳.OpenStack Heat 与企业应用部署.互联网天地, 2013(10)
- 4 卢坤.面向云计算的工作流系统研究与设计.北京邮电大学硕士学位论文,2013
- 5 陶聪,李巍,李云春.一种支持自动部署的脚本语言的设计与实现.计算机应用研究,2005(11)
- 6 卢坤.面向云计算的工作流系统研究与设计.北京邮电大学硕士学位论文,2013
- 7 戴琪.云计算的自动化部署服务在银行 power 系统上的应用研究.上海交通大学硕士学位论文,2012
- 8 董小社,孙发龙,李纪云等.基于映像的集群部署系统设计及实现.计算机工程,2005(24)
- 9 曾令华.基于 Puppet 的自动化配置系统设计与实现.华南理工大学硕士学位论文,2012

(上接第 13 页)

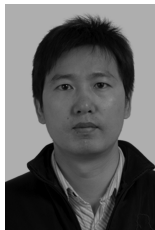
- 3 房秉毅,张云勇,程莹等.云计算国内外发展现状分析.电信科学,2010(S1)
- 4 Shaikh F B, Haider S. Security threats in cloud computing. Proceedings of Internet Technology and Secured Transactions (ICITST), Abu Dhabi, 2011
- 5 吴吉义,傅建庆,平玲娣等.一种对等结构的云存储系统研究.电子学报,2011(5)
- 6 刘国奇,刘慧,高宇等.一种基于效用的云计算动态资源计费策略.吉林大学学报(工学版),2013(6)
- 7 黄卫东,于瑞强.云计算环境下基于 Petri 网的流程计费模型.北京邮电大学学报,2013(1)
- 8 吴俊,徐溟.公有云服务计费模式比较研究.电信科学,2012,28(1):127~132
- 9 Wiki. http://en.wikipedia.org/wiki/Agentless_data_collection,2014

[作者简介]



陈豪,男,现就职于中国联合网络通信有限公司研究院,主要研究方向为平台架构、云计算安全、物联网安全等,参与 ITU、GSMA 物联网国际标准制定,申请专利 10 余项。

[作者简介]



张园,男,中国联合网络通信有限公司研究院工程师,主要从事云计算、行业应用、运营商新业务的研究。



张云勇,男,博士后,中国联合网络通信有限公司研究院副院长、高级工程师,主要研究方向为下一代开放网络、固定移动融合核心网、移动互联网及业务、云计算。



房秉毅,男,博士,中国联合网络通信有限公司研究院高级工程师,主要从事云计算、核心网新技术的研究。

(收稿日期:2014-10-16)



陶冶,男,现就职于中国联合网络通信有限公司研究院,主要研究方向为云计算安全、SDN 安全等。



张尼,男,博士后,中国联合网络通信有限公司研究院云计算实验室高级工程师,主要从事移动互联网安全、云计算安全、信息安全等领域研发工作。



张云勇,男,博士后,中国联合网络通信有限公司研究院副院长、高级工程师,主要研究方向为下一代开放网络、固定移动融合核心网、移动互联网及业务、云计算。

(收稿日期:2014-09-15)