开源云OpenStack技术指南

唐　宏　秦润锋　范均伦　编著

科学出版社

开源云 OpenStack 技术指南

唐　宏　秦润锋　范均伦　编著

科学出版社
北京

内 容 简 介

本书系统介绍开源云 OpenStack 项目，内容包括：OpenStack 总体介绍，Nova、Glance、Swift、Quantum、Keystone 等 OpenStack 项目的技术基础以及相应的实践部署和管理操作。本书理论结合实践，旨在帮助读者了解和掌握 OpenStack 的基础知识，并能够实践部署、管理 OpenStack，为 OpenStack 项目的研究、开发、运维、应用打下坚实基础。

本书适合开源云 OpenStack 用户，尤其是 OpenStack 初学者阅读；也可供 IaaS 云计算领域架构师、研发人员、运维工程师、技术工程师，高等院校计算机等相关专业广大师生，以及对 OpenStack 技术感兴趣的读者阅读和参考。

图书在版编目(CIP)数据

开源云 OpenStack 技术指南 / 唐宏，秦润锋，范均伦编著. —北京：科学出版社，2013

ISBN 978-7-03-037510-0

Ⅰ.①开… Ⅱ.①唐… ②秦… ③范… Ⅲ.①计算机网络-指南 Ⅳ.①TP393-62

中国版本图书馆 CIP 数据核字(2013)第 103614 号

责任编辑：裴 育 / 责任校对：钟 洋
责任印制：张 倩 / 封面设计：蓝正设计

科 学 出 版 社 出版
北京东黄城根北街 16 号
邮政编码：100717
http://www.sciencep.com
骏 主 印 刷 厂 印刷
科学出版社发行 各地新华书店经销

*

2013 年 6 月第 一 版 开本：B5(720×1000)
2013 年 6 月第一次印刷 印张：20 1/2
字数：399 000

定价：78.00 元

(如有印装质量问题，我社负责调换)

前　言

近年来，开源 IaaS 云计算领域风起云涌，随着相关技术的成熟，开源项目已经成为 IaaS 的重要组成部分，其中开源云 OpenStack 项目是目前最受业界关注的焦点，备受众多企业以及社区用户的关注，并积极投身其中。2012 年 OpenStack 成立基金会，吸引了全球多家主流软硬件设备厂商以及主流云服务提供商。在中国，OpenStack 同样受到众多企业和用户的青睐。但到目前为止，国内尚未有系统讲解 OpenStack 的中文书籍或资料，OpenStack 初学者受困于前期的入门过程，难以快速熟悉和理解 OpenStack。本书希望能够让众多初学者迅速了解 OpenStack，为将来应用与开发 OpenStack 打下扎实的基础。此外，本书对于其他 OpenStack 高级用户也有一定程度的参考价值。

全书共 11 章，内容由两大部分组成，分别是 OpenStack 技术详解篇和 OpenStack 实战安装篇。其中，OpenStack 技术详解篇着重于 OpenStack 各项目的技术原理分析，内容包括：OpenStack 技术概要，OpenStack 项目（Nova、Quantum、Keystone、Swift）的系统详解；OpenStack 实战安装篇则着重于 OpenStack 各项目的实践应用，内容包括：OpenStack 项目（Nova、Glance、Swift 及相关项目）的部署、管理、常用命令。

本书按照先理论后实践的顺序进行编写，理论联系实际，内容翔实全面，力求使读者对 OpenStack 有完整、正确和深入的认识。在阅读方面，各章节内容之间相对独立，读者可以按照先后顺序通读，也可以单独选择有兴趣的章节进行阅读。

OpenStack 是一个快速发展的开源云平台，到目前已经更新到 Folsom 版本，而新版本 Grizzly 也已于 2013 年 4 月发布。因此，基于 Diablo 版本进行编写的本书在技术方面可能稍有滞后，但在 OpenStack 基本原理和实践认识方面依然有较为系统的描述和介绍，具有一定的借鉴作用和参考价值。此外，由于作者经验有限，书中难免存在不妥之处，敬请读者批评指正。

目　录

OpenStack 实战安装篇

OpenStack
技术详解篇

第 1 章　OpenStack 技术概要

1.1　OpenStack 总体概况

随着 IaaS 云计算技术的快速发展，开源技术业已成为业界不可忽视的重要力量。目前，开源技术已经深入影响整个云计算产业结构和商业模式，并且以其独特的低成本性、开放性、灵活性以及创新性优势获得越来越多 IT 企业的青睐，逐步成为云计算的基础解决方案。开源 IaaS 云计算技术具有传统商业软件不可比拟的优势：首先，开源创造云平台部署的灵活性，云服务提供商以及 IT 企业不仅可有效避免单一供应商云平台产品的套牢以及高成本风险，而且通过开源组件的灵活开发与部署，可快速响应云时代下不断变化的云业务需求；其次，开源引入创新，开源环境中的云业务开发与部署更易于达到高度的敏捷和出色的创新，相比于传统商业产品的业务开发流程，开源技术无论在业务创新还是响应时间上都具备很大的优势；再次，开源降低云计算基础设施建设的成本，企业采用开源方案，大大减少前期软件开发与后期系统运维的成本；最后，开源提升质量，开源技术源代码的开放性将其软件缺陷暴露无遗，开源社区成千上万的开发者不仅及时修补 BUG 以保证系统的稳定性，而且积极追踪业界最新云服务技术，并在最短的时间内实现相关功能，在功能和性能上保证系统平台的最优。目前，业界处于领先地位并且社区活跃度较高的开源 IaaS 云计算项目包括：OpenStack、Eucalyptus、OpenNebula、AbiCloud 等，其中尤以 OpenStack 在全球关注度最高且技术发展最快。

OpenStack 是由美国国家航空航天局（NASA）与美国云服务提供商 Rackspace 公司于 2010 年 9 月共同发起并推动的一个非盈利性的开源云计算项目。OpenStack 基于全开源的理念，目的是推动实现云计算开放性、无边界性的内涵与实质，为云计算管理平台领域带来更加开放、自由与灵活的构建方法，致力于提供规模化、灵活扩展易部署且功能丰富的全开源模式平台，协助运营商、企业、ISP/CP、科研机构等实现公共云和私有云服务。OpenStack 项目在短短一年里发展迅速，截至 2011 年底已经发布了四个正式版本 Austin、Bexar、Cactus 和 Diablo，并于 2012 年 4 月发布了第五个版本 Essex。目前，OpenStack Diablo 版本主要由五个子项目组成：Nova（弹性计算服务）、Swift（对象存储服务）、Glance（虚拟机磁盘镜像服务）、Keystone（安全统一认证服务）和 Horizon（平台自服务界面），并且全部五个子项目基于宽松的 Apache 2.0 License 发布，这意味着云服务商以及企业均可免费自由使用其源代码，并可基于 OpenStack 的基础架构开发满足自身业务需求的

云服务平台。

目前，大多数云计算管理平台通常都是围绕一个中心组件进行架构，通过服务器集群节点来部署和控制虚拟服务器，往往更加关注中心控制节点的功能性应用，使得其分布式部署能力以及资源可扩展性存在缺陷。在一个相对较小的规模中，可模拟类似 Amazon Web(AWS)服务与应用的 IaaS 云服务环境，但是这种集中式架构的性能瓶颈决定了无法实现大规模的运营，无法满足大量服务同时并发的应用需求。开源云 OpenStack 项目基于分布式模块化架构实现系统规模化、层次化部署，支持跨数据中心管理，支持单数据中心内部集群式管理，支持外部资源弹性自动伸缩，支持系统内部模块的横向扩展，在网络部署方面支持纵向层次化管理，为公共云和私有云客户提供规模化、可扩展、弹性以及简单易部署的云计算解决方案。目前，开源云 OpenStack 社区吸引了全球超过 150 家主流硬件设备厂商、IT 服务商、主流云服务提供商以及运营商的参与和大力支持，并且越来越深入地提供基于 OpenStack 技术的 IaaS 云服务。OpenStack 与其他开源云平台项目的最大区别在于其全开源理念，它的系统内核、外围模块以及特色高级功能等源代码全部在社区公开，任何感兴趣的个人或者专业云服务企业均可以自由参与 OpenStack 各个子项目的开发以及平台技术标准的制定。OpenStack 社区成员涉及业界各个领域的主流厂商，包括芯片厂商 Intel 和 AMD，IT 设备厂商 HP、Dell、NEC 等，网络厂商 Cisco、Nicira、Brocade、H3C、F5、Radware 等，操作系统发行商 Canonical、Suse，电信运营商 AT&T、NTT 和 Korea Telecom；此外，还包括其他 IT 应用服务商和系统架构优化服务商，共同致力于推动基于 OpenStack 的 IaaS 架构标准化以及技术应用与创新。OpenStack 社区几乎整合了 IaaS 云平台解决方案的整个产业链，并且在技术层面上全面体现云平台系统弹性以及规模性要求、底层软硬件的兼容性要求以及上层服务平台的能力开放性要求等，其主要目的是推进基于开放体系的云计算平台架构，改变目前云计算孤岛式发展现状，并有效地避免单一厂商产品所带来的 lock-in 风险。

1.2 OpenStack 社区总体组织架构

目前，OpenStack 社区管理组织架构可归纳为“2+1”模式，包括两个技术管理机构 PTL(Project Technical Lead)/PPB(Project Policy Board)，以及一个顾问机构 AB(Advisory Board)。其中，PTL 主要负责每个子项目(Nova/Swift/Glance/Keystone/Horizon)内部的技术发展决策，PTL 同时也是 PPB 成员；PPB 主要负责整体 OpenStack 社区的技术发展路线以及各个子项目之间的互通合作；AB 倾向于 OpenStack 的营销市场合作，其成员主要由 OpenStack 的商业合作伙伴组成，基于市场需求为 PPB 和 PTL 提供技术决策咨询建议。Rackspace 公司作为

OpenStack 项目的发起者，PTL 和 PPB 管理成员主要来自于 Rackspace 公司，由其主导并控制着社区的技术发展路径。

1. PTL

全球开源云 OpenStack 项目采取较为“民主”的管理模式，倡导每个子项目的开发者成员共同决定 OpenStack 技术以及标准的发展演进方向。社区每个子项目的所有成员每半年投票选举一名该项目的技术负责人，即 PTL，主要协调和解决在项目发展过程中的决策争议，社区中每名贡献者都具有选举权和被选举权，PTL 一般是由社区内代码贡献最多、影响力最大、信任度最高的成员担任，任期半年，在每届 Design Summit 之前的一个月选举产生。PTL 的主要职责包括：

- 组织日常邮件组的技术讨论，解答技术疑问；
- 征求社区所有成员的项目发展建议，平衡社区内部技术争议；
- 负责源代码审核与验证，决策新版本发布；
- 负责与其他子项目以及 PPB 的沟通合作。

OpenStack 目前由五个子项目组成，其中每个子项目选举一名技术负责人，本届的 PTL 成员主要来自于 Rackspace 公司和 Nebula 公司。

2. PPB

PPB 是 OpenStack 社区的最高决策机构，原则上并不干涉每个独立子项目的技术发展路径，其主要目标是提升整体 OpenStack 项目的合作效率以及拓展各个子项目之间的兼容互通能力。PPB 的主要职责包括：

- 定义 OpenStack 的使命和愿景，决策 OpenStack 技术与商业发展的路线；
- 协调与解决所有 OpenStack 子项目之间的合作与争议；
- 负责 OpenStack 社区新增项目的孵化进程，保证新增项目与整体 OpenStack 项目的统一性与兼容性；
- 负责 OpenStack 项目 API 定义的一致性。

由于 PPB 是 OpenStack 社区的最高决策机构，其成员构成有严格规定。目前，规定 PPB 委员会由 14 名成员(5+5+4)组成：

- 五名子项目的 PTL，由社区全体成员选举产生，任期半年；
- 五名常委，由社区全体成员选举产生，任期一年，其中两名常委在一年中的第一次 Design Summit 选举产生，另外三名在第二次 Design Summit 选举产生，且常委与 PTL 不能兼任；
- 其他四名由 Rackspace 公司指定，其中一人被任命为 PPB 主席，主要承担 PPB 章程制定、整体 OpenStack 社区事务的管理决策工作。

3. AB

OpenStack 项目成立一年多以来，吸引业界诸多关注，其中不乏商业公司与机构投入大量资源支持 OpenStack 的发展，包括 IT 设备厂商推出基于 OpenStack 软件的商业云 IDC、云服务提供商的 OpenStack 私有云以及云服务中间商的集成化 OpenStack 解决方案等，而 AB 顾问理事会主要由 OpenStack 商业合作伙伴组成。AB 顾问理事会对于云计算市场的嗅觉更加灵敏，经验更为丰富，对 OpenStack 社区的技术发展具有更前沿的指导价值。AB 顾问理事会的主要职责包括：

- 为 OpenStack 技术发展路线提供建议与指引；
- 为 OpenStack 组织架构提供建议；
- 为 OpenStack 品牌商标的推广提供建议；
- 推进 OpenStack 的市场应用。

发展初期，AB 顾问理事会的成员仍由 Rackspace 公司指定；待 OpenStack 发展成熟之后，AB 顾问理事会根据申请者的 OpenStack 商业推广承诺自行确定其成员资格。

目前，从 OpenStack 社区的组织架构、成员选定方式以及成员组成来看，Rackspace 公司作为云服务提供商无疑控制并主导着整个 OpenStack 社区，这直接导致外界对 OpenStack 未来发展前景的疑虑，也招致内部盟员对 OpenStack 能否坚持“公平开放”理念的担忧。Rackspace 公司已经意识到目前社区发展过程中所面临的严峻形势，因此在 2011 年秋季 Design Summit 上宣布有意在 2012 年成立 OpenStack 基金会。该基金会接手 OpenStack 所有的知识产权和商标权，监管并主导整个项目的技术发展路线，形成厂商中立的局面，保证未来平台以及标准的开放愿景，实现 OpenStack 社区生态链的可持续发展。OpenStack 基金会计划得到所有盟员的支持，Rackspace 公司就基金会的组织架构以及运作方式广泛征求社区所有盟员的意见，并于 2012 年 9 月 19 日宣布基金会正式成立。

1.3 OpenStack 总体系统架构

OpenStack 系统按需灵活地提供 IT 基础设施资源，具备规模化、弹性扩展、自动调度、安全可靠以及应用快速部署等特点。目前，OpenStack 正式发布的 Diablo 版本由五个组件构成：Nova/Swift/Glance/Keystone/Horizon，其总体系统架构如图 1-3-1 所示。

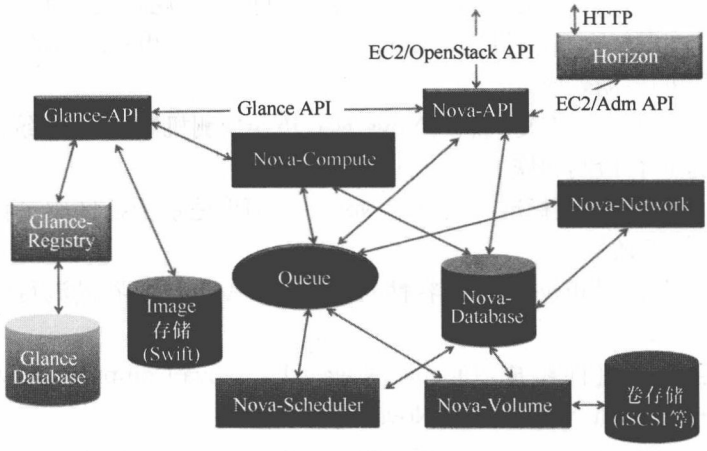

图 1-3-1　OpenStack 总体系统架构

在 OpenStack 系统架构中，Nova 实现弹性计算服务，其中 Nova 的相关模块以分布式的方式进行异步调用与通信，以 Nova-Scheduler 为业务调度核心实现计算资源在不同条件下的自动分配，Nova-Scheduler 实现底层虚拟化平台的兼容以及虚拟机的生命周期管理，Nova-Network 实现虚拟网络、虚拟防火墙以及相关网络配置等功能，Nova-Volume 主要实现用户虚拟机的块存储挂载功能；Swift 实现对象存储功能，可创建大量的、可扩展的对象存储软件，主要用于商用的集群服务器上，能够存储 TB 级甚至 PB 级的数据；Glance 主要实现平台服务目录镜像模板的定义与生成；Keystone 提供整体平台的统一认证服务；Horizon 在 Django 框架下开发，可以完成简单的镜像管理、虚拟机实例管理、Volume 管理等，目前功能比较简单，需要进一步完善。以下简要描述 OpenStack 三个关键组件——Nova、Swift 和 Glance 的系统架构与实现方式。

1.3.1　Nova 系统架构概况

Nova 是 OpenStack 云平台的计算模块，与 Amazon EC2 类似，为用户提供 IaaS 服务。Nova 自身并不提供虚拟化底层软件，如 VMWare ESX、Xen、KVM 或者 Hyper-V 等 Hypervisor，却提供了这些虚拟化软件的驱动，通过 Web API 与 Hypervisor 进行通信，从而在用户层面实现虚拟机实例的管理功能。Nova 运行在 Linux 系统之上，官方建议 Ubuntu。Nova 的安装是一个系统集成和系统配置的过程，其自身系统已经包括了其他一些开源软件，如对于虚拟化的底层管理是通过 Libvirt 来完成。Nova 设计的基本原则遵循以下六点：

• Component based architecture(模块化结构)：Nova 模块化架构，实现功能的便捷增加和规模的弹性扩展。

• Highly available(高可用性):对过载计算单元实现智能弹性扩容。

• Fault-tolerant(容错机制):Nova 系统中的进程是相互隔离的,以避免由进程错误而引起的雪崩效应。

• Recoverable(可恢复机制):Nova 具有错误侦测机制,对于系统出现的错误可以很便捷地进行检测和恢复。

• Open standards(标准开放性):Nova 的 API 是基于社区驱动而形成的标准,具有完全的开放性。

• API compatibility(标准兼容性):Nova 的 API 与业界最流行的系统相兼容,如 Amazon EC2。

Nova 由七个组件组成:Queue、Nova-API、Nova-Compute、Nova-Volume、Nova-Database、Nova-Network 和 Nova-Scheduler。

• Queue:负责 OpenStack 系统组件之间信息传递,类似于全局系统的信息交换器,为守护进程传递消息。当前采用 RabbitMQ 实现,但是理论上可以是 Python AMQPLIB 支持的任何 AMQP(高级消息队列协议)消息队列。

• Nova-API:提供 OpenStack API 和 EC2 API 的接口,主要为 Cloud Controller提供前端 Web Service 服务,初始化绝大多数部署活动(如运行实例),以及实施一些策略(绝大多数的配额检查)。

• Nova-Compute:主要是一个创建和终止虚拟机实例的 Worker 守护进程。其工作原理是从队列中接收行为,然后在更新数据库的状态时,执行一系列的系统命令完成任务。

• Nova-Volume:管理映射到虚拟机实例的存储卷的创建、接入和取消等生命周期管理。

• Nova-Database:存储 OpenStack 实例创建和运行过程中的各种状态,包括可用的实例类型、在用的实例、可用的网络和项目等。理论上可以支持 SQL-Alchemy的任何数据库,目前 Diablo 版本支持的数据库包括 MySQL、PostgreSQL、Sqlite3,其中 Sqlite3 仅用于测试和开发工作。

• Nova-Network:从队列中接收网络任务并执行相关操作以实现虚拟网络的创建与管理,如创建桥接接口或改变 iptables 规则,为服务器提供完善且安全的虚拟网络,提供虚拟机之间的访问以及服务器与外网的访问通道。

• Nova-Scheduler:为 OpenStack 云平台所创建的虚拟机实例部署选择最合适的计算节点。

Nova 是基于 shared-nothing 以及消息传递模式的架构,其所有组件均可运行在多个服务器之上。Cloud Controller 通过 AMQP 与其他组件进行通信,在组件之间的通信过程中,为了避免消息阻塞,Nova 采用异步调用机制,该机制在组件请求消息得到响应之后出发消息处理,类似于 TCP 的三次握手机制。为了实现同一

组件多个备份的 shared-nothing 架构属性，Nova 将所有云系统状态数据均保存在一个分布式云存储中，系统状态的更新都将被写入该存储系统中，组件通过 Web Service 将系统状态请求发送给存储系统，存储系统将请求相关内容复制给请求组件，在某些情况下，读取结果会在较短时间内缓存在控制器中。

1.3.2　Swift 系统架构概况

Swift 是一个冗余、可扩展的分布式对象存储系统，提供类似 Amazon S3 云存储服务，其存储容量可扩展到 PB 级，以满足大容量存储业务的需求。Swift 的设计架构为全分布式，避免单一主控节点所造成的 I/O 性能瓶颈，并且通过标准 REST 接口实现外部系统的调用。Swift 系统架构由四个组件和一个关键模块构成，四个组件为 Proxy Server、Object Server、Container Server 和 Account Server，一个关键模块为 RING。RING 模块分布存在于四个组件中，决定数据的分配与一致性维护，主要实现实体名称与所在的物理磁盘位置之间的一对一映射，包括 zone、device、partition 和 replica 之间的映射关系，当其他组件需要针对相应对象、容器或账户执行任何操作时，需要通过 RING 模块确定其在集群中的位置；同时，RING 模块实现分区多份复制能力，默认情况下复制三份，每份复制在不同的 zone 中，保证数据的安全性；另外，RING 模块也承担集群故障检测以及最优化数据分配功能，实现对象在集群中所有设备中的均匀分配。Swift 系统架构如图 1-3-2 所示。

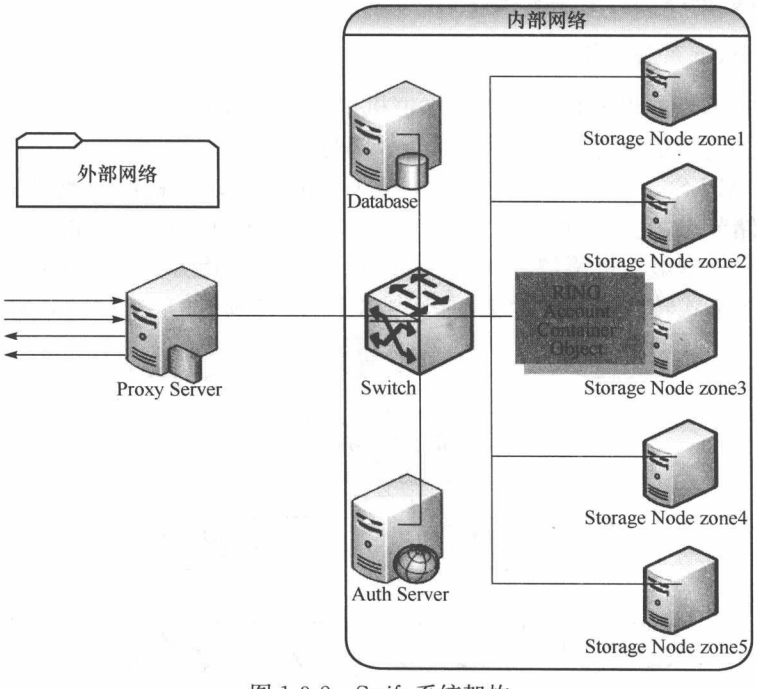

图 1-3-2　Swift 系统架构