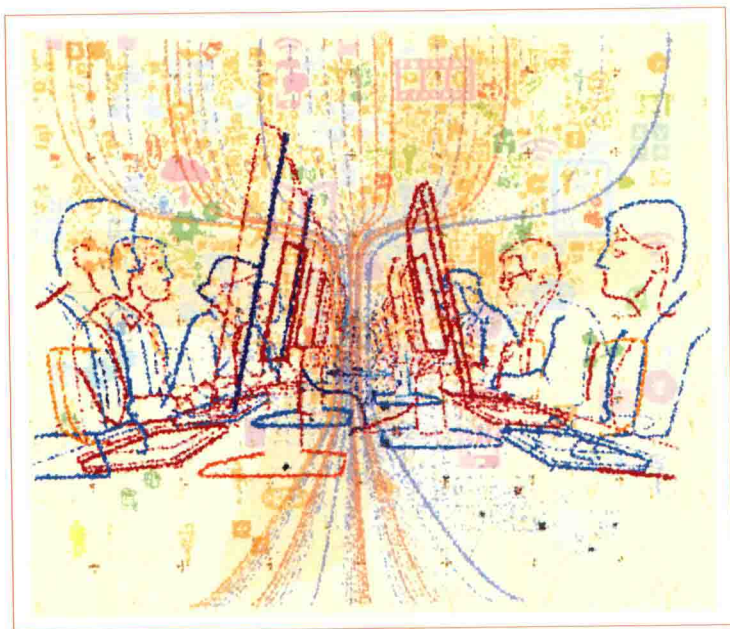




OpenStack

设计与实现 (第2版)

英特尔开源技术中心 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

OpenStack

设计与实现 (第2版)

英特尔开源技术中心 编著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书是一本介绍 OpenStack 设计与实现原理的书。本书内容以 Newton 版本为基础，覆盖了 OpenStack 的学习方法到设计与实现等各个方面内容，致力于帮助读者形成 OpenStack 及其各个主要组件与项目的拓扑。

本书语言通俗易懂，能够带领读者更为快速走入 OpenStack 的世界并作出自己的贡献。

本书适合希望能够参与 OpenStack 开发的读者，也适合对 OpenStack 茫然的初学者，以及有一定使用部署经验但是希望了解 OpenStack 实现原理的广大用户。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目(CIP)数据

OpenStack 设计与实现 / 英特尔开源技术中心编著. —2 版. —北京: 电子工业出版社, 2017.5
ISBN 978-7-121-31199-4

I. ①O… II. ①英… III. ①计算机网络 IV. ①TP393

中国版本图书馆 CIP 数据核字(2017)第 065892 号

责任编辑: 徐津平

印 刷: 北京京科印刷有限公司

装 订: 三河市华成印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×980 1/16 印张: 33 字数: 712 千字

版 次: 2017 年 5 月第 1 版

印 次: 2017 年 5 月第 1 次印刷

印 数: 3000 册 定价: 99.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: 010-51260888-819, faq@phei.com.cn。

推荐序 1

It's impressive to see just how far OpenStack has come since its founding. With customers increasingly recognizing OpenStack maturity and a thriving ecosystem of developers and partners supporting the solution, it's clear that OpenStack has cemented its role as a leader in open cloud Infrastructure-as-a-Service.

Intel and China have played important roles in that growth. Intel as a strong contributor across OpenStack and related open source technologies; China as a critical source of customers, technical leadership and development talent. Recognizing the strength of this partnership, Intel has invested in the China technology ecosystem through our team working in the Intel Open Source Technology Center in China. This talented team authored the first edition of this book and is now proud to offer a second edition, reflecting the latest trends and updates from the OpenStack Newton release, including new projects (Murano, Searchlight, Kolla), deployment tools, and monitoring improvements as well as technologies and solutions (Ceph, containers) which integrate well with OpenStack.

We look forward to working with the China OpenStack community on the continued success of OpenStack.

Hillarie Prestopine

Intel Vice President and Director, Intel Open Source Technology Center

推荐序 2

Since its introduction in 2010, OpenStack has gained tremendous momentum with thousands of developers and a couple of thousand deployments. Today, OpenStack is considered the de-facto open source cloud Infrastructure-as-a-Service provisioning software for companies including Baidu, the European Organization for Nuclear Research (CERN), HP, Huawei, IBM, Intel, and Walmart, as well as many more Fortune 500 companies.

Intel strongly believes investments and engagements in open source help foster a strong ecosystem. The Open Source Technology Center at Intel is proud to be a leader in open source software across a wide range of technologies and market segments, including enterprise Linux and related technologies such as virtualization, data center and cloud software; embedded market segments and client Linux programs. Intel is one of the top 10 OpenStack contributors, with a focus on Nova, Horizon, Ironic, Ceilometer, Swift, and Cinder.

Increasingly established companies and new ventures are considering OpenStack to achieve savings through pooled compute, storage, and network resource utilization. China represents a vast talent resource pool, from young engineers groomed in universities to professionals in growing industries. The need for an OpenStack developer resource in Mandarin can help engineers learn about the cloud and OpenStack while speeding their ability to contribute to and influence OpenStack.

It is with immense joy that we present this book to the Chinese community. I hope you find it compelling and its content helps increase China's influence in OpenStack development, easing adoption and offering cost savings for new economic endeavors.

Imad Sousou

Intel Vice President and General Manager, Open Source Technology Center
Platinum Board Member, OpenStack Board of Directors

前言

至此落笔之际，OpenStack 问世几近 7 年，7 年的时间，对很多项目来说已经足够走过一个创建发展到没落的轮回，而对于 OpenStack，7 年的时间仍然远远不够让我们看到它最终能够达到的高度。

从哲学的辩证角度：今天的必然正是由之前一系列的偶然所决定的。2010 年的一个偶然，OpenStack 由 RackSpace 和美国国家航空航天局合作发布，于是随后的时间里无数公司与个人偶然初识 OpenStack 并深陷其中，而正是这些偶然相联合，从而决定了会有今天这样一本书，会有现在写下的这些话。那么，当您偶然地拿起这本书，偶然地看到这段话，您是否会问自己：这样的偶然又会导致什么样的必然？

如果您依然决定继续这次的偶然之旅，还请您问自己一个问题：我在强迫自己学习 OpenStack 么？很希望您能回答不是，但希望与现实往往都有段不小的距离，因为很多时候，我们都是因为各种原因而强迫自己去喜欢的。或许，针对这个问题，最让人愉悦的回答是“说实话，我学习的热情从来都没有低落过。Just for Fun.”

其次，在您继续之前，面对 OpenStack 这样一个新生事物，让人最为惴惴不安的问题或许便是：我该如何更快更好的适应这个全新的世界？人工智能与机器学习领域里研究的一个很重要的问题是“为什么我们小时候有人牵一匹马告诉我们那是马，于是之后我们看到其他的马就知道那是马了？”。针对这个问题的一个结论是：我们头脑里形成了一个生物关系的拓扑，我们所认知的各种生物都会放进这个拓扑的结构里，而我们随着年纪不断成长的过程就是形成并完善各种各样或树形或环形等拓扑的过程，并以此来认知我们所面对的各种新事物。

由此可见，或许我们认知 OpenStack 最快也最为自然的方式就是努力在脑海里形成它的拓扑，并不断的进行细化。比如作为一个云计算的平台它包括了哪些功能分别对应哪些项目，各个项目又实现了哪些服务以及功能，这些功能又是以什么样的方式实现的，等等，对于我们感兴趣的项目或服务又可以更为细致的去勾勒它其中的脉络。就好似我们头脑里形成的有关一个城市的地图，它有哪些区，区里又有哪些标志建筑以及街道，对于我们熟悉的地方可以将它的周围进行放大细化，甚至于一个微不足道的角落。

而对于这个拓扑细化的过程能够起到有益辅助的是概念空间的勾勒。站在架构设计的角度，软件从需求进到架构出的全过程中，勾勒描绘概念空间是很重要的一个中间过程。这个阶段会形成所需要引入的各种新概念，比如操作系统中的进程、虚拟内存、系统调用等等，它们就类似一个拓扑中的标志建筑，而我们去认知研究这个软件的时候，描绘这个概念空间也就不可避免成为重中之重。

本书的组织形式

本书的内容组织正是为了尽一切能力帮助读者能够形成有关 OpenStack 以及各个重要项目与功能比较细致的拓扑。首先是前四章，这几章的内容希望能够帮助您对 OpenStack 有个全面的认识和了解，从而形成对 OpenStack 整体的拓扑。

第 1 章主要介绍了 OpenStack 的成长史以及它的体系结构和社区现状。

第 2 章详尽的介绍了 OpenStack 开发的基础流程以及如何去分析 OpenStack 的源码。

第 3 章介绍了 OpenStack 的底层基石——虚拟化技术。大多数 OpenStack 的使用者和开发者并不了解虚拟化的一些细节，有了这一章的介绍，我们能够对 OpenStack 有一个更好的认识。

第 4 章将 OpenStack 众多项目中所使用到的通用技术加以介绍，有了这一章，我们理解各个具体项目的设计与实现时，可以少去很多的阻碍。

然后第 5~14 章的内容对 OpenStack 主要组件及项目的实现进行介绍。按照认识的发展规律，通过前面几章的介绍我们已经对 OpenStack 有了全局的认识和了解，接下来就可以以兴趣或工作需要为导向，寻找一个组件或项目，对其实现进行深入的钻研和分析。这些章节的内容也是希望能够尽量帮助您形成对相应项目的比较细致的拓扑，并不求对所有实现细节的详尽分析。

第 5 章讨论计算组件也就是 Nova 项目。Nova 为我们实现了 OpenStack 这个虚拟机世界的抽象，控制着一个个虚拟机的状态变迁与生老病死，管理着他们的资源分配。

第 6 章讨论存储相关的四个项目：Swift, Cinder, Glance 以及 Ceph。他们共同为这个虚拟机世界的主体——虚拟机提供了安身之本，负责为每个虚拟机本身的镜像以及它所产生的各种数据提供一个家，尽可能的去做到“居者有其屋”。

第 7 章讨论网络组件也就是 Neutron 项目。没有网络，任何虚拟机都将只是这个虚拟机世界中的孤岛，不知道自己生存的价值。

安全是每个软件无法回避的问题，第 8 章便针对安全问题进行讨论，包括 Keystone 项目以及可信计算池的相关内容。

第 9 章的内容有关计量与监控的项目 Ceilometer，计量与监控是公有云运营的一个重要环节。

第 10 章的内容与物理机管理有关，Ironic 项目被应用于 OpenStack 中的裸机管理和部署。

第 11 章介绍了 OpenStack 的控制面板。提供一个简洁方便、用户友好的控制界面给最终的用户和开发者对 OpenStack 尤为重要。

随着容器技术的发展，容器与云基础架构的结合受到越来越多的关注，第 12 章便讨论了 OpenStack 对容器的支持。

第 13 章的内容与部署有关，但是这里讨论的并不是如何部署的详细步骤与过程，而只是与部署有关的几个主要项目。

第 14 章介绍了几个新兴的项目，包括 Searchlight 与 Watcher 等。

感谢

作为英特尔的开源技术中心，参与 OpenStack 的开发与推广是再为自然不过的事情。除了为 OpenStack 的完善与稳定贡献更多的思考和代码，我们也希望能通过这本书让更多的人更快捷的融入 OpenStack 的大家庭。

如果没有 Imad Sousou（英特尔软件与服务事业部副总裁兼开源技术中心总经理）、Mauri Whalen（英特尔软件与服务事业部副总裁兼开源技术中心核心系统研发总监）、Hillarie Prestopine（英特尔软件与服务事业部副总裁兼开源技术中心云和网络系统研发总监）、David L Brown（英特尔开源技术中心云计算核心研发总监）、练丽萍（英特尔开源技术中心网络和存储研发总监）、Malini K Bhandaru（英特尔开源技术中心云计算主任工程师）、冯晓焰（英特尔开源技术中心中国安卓研发总监）、李少凡（英特尔开源技术中心虚拟化研发总监）、陈绪（英特尔开源技术中心中国云计算战略总监）的支持，这本书不可能完成，谨在此感谢他们对本书编写过程中的关怀与帮助。

也要感谢本书的编辑孙学瑛老师，从选题到最后的定稿，整个过程中，都给予我们无私的帮助和指导。

然后要感谢参与第一版与第二版各章内容编写的各位同事，他们是王庆、丁建峰、任桥伟、陆连浩、翟纲、徐贺杰、程盈心、李晓燕、臧锐、贺永立、郭瑞景、乔立勇、陈巍、杜永丰、杨林、张磊、冯少合、金运通、魏刚、田双太、汪亚雷、谭霖、辛晓慧，为了本书的顺利完成，他们付出了很多努力。他们不仅为英特尔开源技术中心做出了很多的贡献，而且长期活跃在中国的云计算技术生态系统中。

最后感谢所有对 OpenStack 抱有兴趣或从事 OpenStack 工作的人，没有你们的源码与大量技术资料，本书便会成为无源之水。

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- **提交勘误：**您对书中内容的修改意见可在【提交勘误】处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **与作者交流：**在页面下方【读者评论】处留下您的疑问或观点，与作者和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/31199>



作者简介

英特尔开源技术中心（Intel Open Source Technology Center -- OTC）是英特尔公司内专职进行开源软件开发的团队，负责在系统软件领域进行与英特尔®平台相关的技术开发和创新，并在 Linux 操作系统内核，Android/Chrome 操作系统，云计算和虚拟化技术，HTML5 Web Runtime，图形驱动及多媒体软件以及系统软件的优化等方面积累了业界领先的技术能力。同时依托强大的技术团队，OTC 积极参与开源社区对开源软件的推广普及活动，并对国内及国际上主流开源操作系统厂商提供有力支持。



目 录

第 1 章 初识 OpenStack	1
1.1 从虚拟化到 OpenStack	1
1.1.1 虚拟化	1
1.1.2 云计算	2
1.1.3 OpenStack	4
1.2 OpenStack 基金会以及管理模式	7
1.2.1 董事会	8
1.2.2 技术委员会	9
1.2.3 用户委员会	10
1.3 OpenStack 体系结构	11
1.4 OpenStack 项目发展流程	18
1.4.1 新项目	18
1.4.2 孵化项目、集成项目和核心项目	19
1.4.3 大帐篷 (Big Tent)	20
1.5 OpenStack 社区	21
1.5.1 邮件列表	21
1.5.2 IRC 和项目例会	22
1.5.3 Summit 和 Meetup	23
1.5.4 其他社交平台	25
1.6 其他开源项目	25
1.7 OpenStack 的技术发展趋势	30
第 2 章 OpenStack 开发基础	33
2.1 相关开发资源	33
2.1.1 OpenStack 社区	33
2.1.2 OpenStack 文档	33
2.1.3 OpenStack 书籍	34
2.1.4 其他网络资源	35
2.2 OpenStack 开发的技术基础	35

2.3	部署开发环境	36
2.3.1	Git	37
2.3.2	Devstack	38
2.4	浏览 OpenStack 源代码	42
2.4.1	浏览代码的工具	43
2.4.2	分析源码如何入手	44
2.5	OpenStack 代码质量保证体系	48
2.5.1	编码规范	50
2.5.2	代码评审 Gerrit	53
2.5.3	单元测试 Tox	58
2.5.4	持续集成 Jenkins	61
2.6	如何贡献	66
2.6.1	文档	67
2.6.2	修补 bug	67
2.6.3	增加 feature	69
2.6.4	review	72
2.6.5	调试	73
第 3 章	虚拟化	75
3.1	概述	75
3.1.1	虚拟化实现方式	77
3.1.2	虚拟化现状和未来	79
3.2	高层管理工具	87
3.2.1	XenAPI	88
3.2.2	Libvirt	89
3.3	OpenStack 相关实现	98
3.3.1	Libvirt 驱动	98
3.3.2	XenAPI 驱动	100
第 4 章	OpenStack 通用技术	102
4.1	消息总线	102
4.2	SQLAlchemy 和数据库	107
4.3	RESTful API 和 WSGI	111
4.4	Eventlet	120

4.5	OpenStack 通用库 Oslo	121
4.5.1	Cliff.....	122
4.5.2	oslo.config.....	125
4.5.3	oslo.db.....	128
4.5.4	oslo.i18n.....	131
4.5.5	oslo.messaging.....	132
4.5.6	stevedore	139
4.5.7	TaskFlow.....	142
4.5.8	cookiecutter.....	149
4.5.9	oslo.policy.....	150
4.5.10	oslo.rootwrap	151
4.5.11	oslo.test	154
4.5.12	oslo.versionedobjects.....	156
第 5 章	计算.....	160
5.1	Nova 体系结构.....	161
5.2	Nova API	166
5.2.1	Nova v2.1 API.....	167
5.2.2	Nova API 实现.....	168
5.3	Rolling Upgrade.....	178
5.3.1	Rolling Upgrade 实现	179
5.4	Scheduler	186
5.4.1	调度器	187
5.4.2	Resource Tracker	191
5.4.3	调度流程	193
5.5	典型工作流程	195
5.5.1	创建虚拟机	195
5.5.2	冷迁移与 Resize.....	196
5.5.3	热迁移	198
5.5.4	挂起和恢复	200
5.5.5	Rebuild 和 Evacuate	200
第 6 章	存储.....	201
6.1	Swift.....	201

6.1.1	Swift 体系结构	201
6.1.2	Ring	209
6.1.3	Swift API	218
6.1.4	认证	226
6.1.5	对象管理与操作	228
6.1.6	数据一致性	231
6.2	Cinder	234
6.2.1	Cinder 体系结构	234
6.2.2	Cinder API	239
6.2.3	cinder-scheduler	241
6.2.4	cinder-volume	243
6.2.5	cinder-backup	248
6.3	Glance	249
6.3.1	Glance 体系结构	249
6.3.2	Glance API	252
6.4	Ceph	257
6.4.1	Ceph 体系结构	259
6.4.2	RADOS	261
6.4.3	Ceph 块设备	281
6.4.4	Ceph FS	285
6.4.5	Ceph 与 OpenStack	286
第 7 章 网络		289
7.1	Neutron 体系结构	289
7.1.1	Linux 虚拟网络	290
7.1.2	Neutron 网络抽象	294
7.1.3	Neutron 架构	295
7.1.4	Neutron 源码结构	297
7.2	Neutron API	299
7.2.1	neutron-server	300
7.3	ML2 Plugin	301
7.4	Port Binding 扩展	308
7.5	Open vSwitch Agent	317
7.6	Service Plugin	324
7.6.1	Firewall	325

7.6.2	LoadBalance	326
7.7	Neutron 热点话题	329
7.7.1	DVR.....	329
7.7.2	SDN	329
7.7.3	NFV/SRIOV	330
7.7.4	OVS 和 DPDK.....	333
第 8 章 安全.....		335
8.1	OpenStack 安全概述.....	335
8.2	Keystone	336
8.2.1	Keystone 体系结构	336
8.2.2	Keystone 启动过程	343
8.2.3	用户认证及令牌获取.....	346
8.2.4	签名证书生成.....	349
8.2.5	Keystone 高阶应用	352
8.3	可信计算池	355
8.3.1	体系结构	355
8.3.2	Intel TXT 与 TBoot	356
8.3.3	可信认证与 OpenAttestation 项目	358
8.3.4	TrustedFilter.....	362
8.3.5	部署	364
第 9 章 计量与监控.....		366
9.1	Ceilometer.....	367
9.1.1	体系结构	367
9.1.2	Pipeline	370
9.1.3	Polling Agent 与 Pollster 插件	372
9.1.4	Notification Agent 与 Notification Listeners 插件	373
9.1.5	Collector 与 Dispatcher 插件.....	373
9.1.6	Storage/DB.....	374
9.1.7	API Server.....	374
9.1.8	部署与使用	375
9.1.9	插件的开发	386
9.2	Aodh	396

9.2.1	体系结构	396
9.2.2	部署与使用	398
9.2.3	插件的开发	402
9.3	Gnocchi.....	408
9.3.1	体系结构	409
9.3.2	部署与使用	412
9.4	Panko	414
第 10 章 物理机管理.....		415
10.1	IroniC 体系结构.....	415
10.1.1	IroniC Driver.....	419
10.1.2	IroniC API.....	423
10.1.3	IroniC Conductor	424
10.1.4	IroniC-python-agent.....	425
10.1.5	ironic-inspector	426
10.2	IroniC 中的网络管理.....	426
10.2.1	物理交换机管理.....	426
10.2.2	多租户网络的支持.....	427
10.3	IroniC 节点的注册和启动	428
第 11 章 控制面板.....		432
11.1	Horizon 体系结构.....	432
11.1.1	Horizon 与 Django.....	432
11.1.2	Horizon 网站布局.....	435
11.1.3	Horzion 源码结构.....	437
11.2	Horizon 部署	439
11.3	页面渲染流程.....	441
第 12 章 容器.....		455
12.1	容器技术	455
12.1.1	容器的原理	455
12.1.2	常见的容器集群管理工具.....	456
12.2	容器与 OpenStack.....	460
12.2.1	nova-docker/heat-docker.....	461

12.2.2	Magnum	461
12.2.3	Murano	469
12.2.4	Kolla	472
12.2.5	Solum	472
12.2.6	Kuryr	474
12.2.7	容器技术与 OpenStack 的展望	476
第 13 章 部署		477
13.1	配置管理工具	478
13.2	OpenStack 部署项目	480
13.2.1	Bifrost	481
13.2.2	Kolla	483
13.2.3	TripleO	490
13.2.4	Fuel	493
第 14 章 新兴项目		495
14.1	Searchlight	495
14.1.1	Searchlight 体系结构	495
14.1.2	plugin 的开发	497
14.2	Watcher	502
14.2.1	Watcher 使用	503
14.2.2	Watcher 体系结构	505
14.2.3	strategy 的开发	507

初识 OpenStack

如果你尚未与 OpenStack 亲密接触过，那么希望这里的内容可以成为你初识 OpenStack 的见证。如果你已经是 OpenStack 达人，那么就选个安静的早晨，抑或下午，一起缅怀与 OpenStack 一起走过的青葱岁月吧。

1.1 从虚拟化到 OpenStack

至此落笔之际，OpenStack 已经成长了 4 年多，云计算被提出了 20 多年，虚拟化则发展了 50 多年，风雨颇多，感慨颇多，谨以这些许年来的点滴之事为献。

1.1.1 虚拟化

1. 1959 年

6 月，一个并不属于万物萌芽的月份。在 1959 年国际信息处理大会上，Christopher Strachey（克里斯托弗）亲手为虚拟化埋下了种子，他在名为《大型高速计算机中的时间共享》的报告中，提出了“虚拟化”的概念，从此拉开了虚拟化发展的帷幕。

2. 20 世纪 60 年代

虚拟化在这期间，由概念孕育到雏形，并得到了进一步的发展。1964 年，一种名为 CP-40 的新型操作系统首次实现了虚拟内存和虚拟机。随后，IBM 推出了 TSS (Time Sharing System，分时共享系统)，允许多个用户远程共享同一高性能计算设备的使用时间，这也被认为是最为原始的虚拟化技术。

3. 20 世纪 70 年代

1972 年 IBM 发布了用于创建灵活大型主机的虚拟机技术，可以根据用户动态的应用需求来调整和支配资源，使昂贵的大型机资源得到尽可能的充分利用。虚拟化由此进入了大型机时代。

这一时期的 IBM System 370 系列通过一种叫虚拟机监控器 (Virtual Machine Monitor，VMM) 的程序在物理硬件之上生成许多可以运行独立操作系统软件的虚拟机实例，从而使虚拟机开始流行起来。