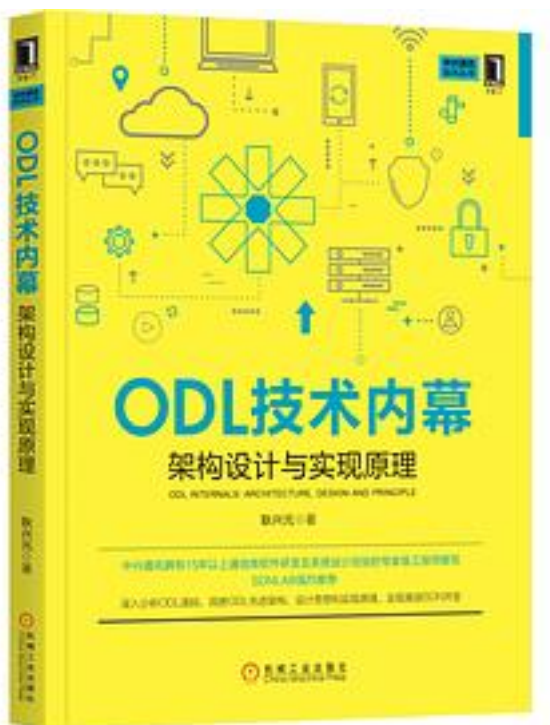


ODL技术内幕：架构设计与实现原理



[ODL技术内幕：架构设计与实现原理 下载链接1](#)

著者:耿兴元 著

出版者:机械工业出版社

出版时间:2019-9-3

装帧:平装

isbn:9787111635093

这是一本从源代码层面深入剖析ODL的著作，旨在帮助读者在透彻理解ODL的先进架构、设计思想和实现原理后，能更有高效地进行SDN开发。

作者是资深的ODL专家，是SDN领域的布道者，有在通讯类软件研发和系统设计领域有超过15年的经验对ODL及其源码有深入的研究和理解。ODL架构演进极快，核心模块和接口变动频繁，ODL子项目众多，功能和接口碎片化严重，学习门槛很高，面对数百万行的源代码更是无从下手。

作者根据自己的经验，对ODL的核心功能及其源代码（氟版本和氖版本）进行反复提炼

、抽丝剥茧，不仅让读者理解ODL的系统架构、设计思想、实现原理，而其能让读者领略ODL实现源码中的优秀代码和设计模式，最终实现让读者更高效地使用SDN的目的，掌握SDL的精髓。

全书13章，分为三个部分：

第一部分 基础环境篇（第1-2章）

主要介绍了ODL的核心概念、架构、设计目标、编译构建环境的搭建、源码阅读的方法，以及ODL社区对众多子项目的管理实践。

第二部分 核心架构篇（第3-10章）

从源代码的角度详细分析了ODL的基本对象、数据树、MD-SAL DataStore、MD-SAL RPC、MD-SAL Notification、MD-SAL Mount、MD-SAL Cluster Service的工作机制与实现原理；

第三部分 公共组件篇（第11-13章）

从源代码角度详细分析了ODL的AAA、RESTCONF、Blueprint等公共组件的设计、实现与扩展。

作者介绍:

耿兴元

资深ODL技术专家，目前就职于中兴通讯，是操作系统及支撑平台的软件专家级工程师，在通讯类软件研发及系统设计领域已有超过15年的工作经验。2015~2017年期间，负责基于ODL的商用SDN控制器平台的设计和研发管理工作。

在ODL领域有多年的研究和实践经验，曾与SDNLAB一起创建了开源项目Jaguar(基于ODL的Kubernetes网络解决方案)，是该开源

目录: 前言

第一部分 基础环境篇

第1章 阅读源代码前的准备 2

1.1 ODL项目介绍 2

1.1.1 ODL框架之争 3

1.1.2 SAL的演进 3

1.1.3 ODL的子项目及分类 4

1.1.4 ODL项目的管理 6

1.2 搭建ODL编译构建环境 6

1.2.1 安装JDK 6

1.2.2 安装及配置Maven 8

1.3 阅读和调试ODL源代码 9

1.3.1 ODL项目源码下载 9

1.3.2 IntelliJ IDEA安装 10

1.3.3 IntelliJ IDEA调试ODL的项目源码 11

1.4 ODL设计目标 12

1.5 ODL总体架构 13

1.6 本章小结 15

第2章 ODL项目管理设计详解 16

2.1 问题的提出	16
2.2 解决思路	17
2.3 实现详解	20
2.3.1 基础parent设计	20
2.3.2 模块构建	23
2.3.3 feature组织	24
2.3.4 版本打包	25
2.4 项目模板	26
2.4.1 项目目录布局设计	26
2.4.2 ODL模板项目	27
2.5 本章小结	28
第二部分 核心原理篇	
第3章 ODL基本对象的设计与实现	30
3.1 QName	30
3.1.1 QName定义	30
3.1.2 QName对象比较	36
3.1.3 QName对象创建	37
3.2 YangInstanceIdentifier	38
3.2.1 Path接口定义	38
3.2.2 YangInstanceIdentifier的类定义	39
3.2.3 YangInstanceIdentifier的比较	42
3.2.4 InstanceIdentifier类	44
3.3 NormalizedNode	44
3.3.1 NormalizedNode类的定义	45
3.3.2 NormalizedNode实例的创建	48
3.4 本章小结	49
第4章 数据树的设计与实现	50
4.1 基本概念	50
4.1.1 配置树与状态树	51
4.1.2 标识与定位	51
4.1.3 快照与MVCC	52
4.2 数据树的设计与实现	52
4.2.1 Tree结构的设计	52
4.2.2 DataTree相关接口定义	55
4.2.3 DataTree的创建	57
4.3 数据树的读写过程	59
4.3.1 快照实现原理	61
4.3.2 数据校验的实现	61
4.4 MVCC机制与实现	63
4.4.1 版本号变更规则	63
4.4.2 并发控制	65
4.5 本章小结	67
第5章 MD-SAL DataStore接口设计	68
5.1 基本概念	69
5.1.1 事务和事务链	70
5.1.2 数据分片	70
5.1.3 三阶段提交	71
5.2 DataStore SPI设计	72
5.2.1 DOMStore	73
5.2.2 DOMStoreThreePhase-CommitCohort	75
5.2.3 DOMStoreTreePublisher	76
5.3 DataStore DOM API设计	77
5.3.1 DOMDataBroker	77
5.3.2 DOMDataTreeSharding-Service	78

5.3.3 DOMDataTreeChange-Service	80
5.4 DataStore Binding API设计	82
5.4.1 Binding基本对象接口	82
5.4.2 DataBroker	84
5.4.3 DataTreeChangeService	87
5.5 本章小结	87
第6章 MD-SAL DataStore的实现原理	88
6.1 概述	89
6.1.1 背景知识	89
6.1.2 实现原理	91
6.2 Raft算法及其实现	92
6.2.1 Raft算法介绍	93
6.2.2 RaftActor设计与实现	98
6.3 DataStore后端实现详解	106
6.3.1 Shard的实现	106
6.3.2 ShardManager	110
6.3.3 ShardStrategy及实现	112
6.4 DataStore前端实现详解	113
6.4.1 DOMStore的实现	113
6.4.2 DOMDataBroker的实现	121
6.4.3 事务链实现	124
6.5 Binding DataBroker的实现	125
6.5.1 Adapter设计	125
6.5.2 BindingDOMDataBroker-Adapter的初始化	126
6.6 本章小结	130
第7章 MD-SAL RPC的设计与实现	131
7.1 一个实例	131
7.1.1 RPC的YANG模型定义	131
7.1.2 RPC的生成接口	133
7.1.3 RPC的实现与调用	135
7.2 RPC机制的总体设计	136
7.2.1 Binding接口设计	136
7.2.2 DOM接口设计	137
7.2.3 总体实现流程	139
7.3 RPC机制实现详解	141
7.3.1 DOMBroker实现详解	141
7.3.2 BindingBroker实现详解	144
7.4 Remote RPC实现详解	149
7.4.1 Gossip协议的实现	150
7.4.2 远程RPC注册及调用	152
7.4.3 Actor设计实现总结	154
7.5 本章小结	155
第8章 MD-SAL Notification的设计与实现	156
8.1 一个实例	156
8.1.1 YANG模型定义	156
8.1.2 生成的接口	157
8.1.3 消息发布	157
8.1.4 消息订阅	158
8.2 MD-SAL Notification接口设计	158
8.2.1 DOM接口	159
8.2.2 Binding接口	160
8.3 MD-SAL Notification实现剖析	161
8.3.1 DOM层实现详解	161
8.3.2 Binding适配实现	169

8.4 本章小结	171
第9章 MD-SAL Mount机制与NETCONF	172
9.1 Mount服务接口设计	172
9.1.1 DOM接口	173
9.1.2 Binding接口	174
9.2 Mount机制的实现	175
9.2.1 DOM接口实现	176
9.2.2 NETCONF南向插件的实现	178
9.3 本章小结	186
第10章 MD-SAL Cluster Service	187
10.1 EntityOwnershipService	187
10.1.1 基本概念	187
10.1.2 接口设计	188
10.1.3 实现说明	192
10.2 ClusterSingletonService	195
10.2.1 接口设计	195
10.2.2 实现说明	196
10.3 本章小结	198
第三部分 公共组件篇	
第11章 AAA	200
11.1 Shiro框架介绍	201
11.1.1 Shiro是什么	201
11.1.2 Shiro的架构	202
11.1.3 Shiro核心处理流程	204
11.2 AAA实现原理	210
11.2.1 Shiro配置优化	210
11.2.2 Realm的8个实现	212
11.2.3 Filter的实现	214
11.2.4 加解密服务	216
11.2.5 数字证书管理	218
11.3 本章小结	219
第12章 RESTCONF	220
12.1 RFC 8040解读	220
12.1.1 操作	221
12.1.2 消息	222
12.1.3 资源	223
12.2 RESTCONF的实现	226
12.2.1 Jersey框架简介	226
12.2.2 RESTCONF资源接口定义	228
12.2.3 Wrapper设计模式	231
12.2.4 初始化过程	233
12.2.5 客户端访问	235
12.3 本章小结	236
第13章 Blueprint及其扩展	237
13.1 Blueprint	238
13.1.1 基础知识	238
13.1.2 运行原理	240
13.1.3 命名空间扩展	241
13.2 Blueprint的使用	244
13.3 本章小结	247
• • • • •	(收起)

标签

好书，值得一读

计算机

SDN

ODL

分布式

评论

没什么启发好吧 真的很一般很一般

SDN网络的出现，可以说是网络世界的第二次工业革命，其中ODL作为目前两种高可用开源SDN控制器之一，担当了这次网络革命中的主要推力，但是ODL在国内的中文文档几乎没有，这本书的出现确实填补了这一空白。

ODL项目庞大，拥有几十个子项目，之前学习起来总感觉到无从入手，非常的吃力。但是这本书将ODL讲述的很细致。

作为SDN的从业者，学习过耿老师的ODL课程，也开发过一些应用，但当要深入了解ODL的源码时总觉得高深莫测，代码繁多，这本书让我一览用到的各种核心技术，包括集群方面的知识，对我们这些要进一步了解和研究ODL的人来说无疑是非常重要的和及时，谢谢耿老师。

目录安排合理，内容充实，适合初学者学习，由浅入深，内容很实用。

ODL开源控制器架构的确复杂，自己学习确实很难入门，官网的资料也是零零散散，耿老师的这本书为SDN爱好者和相关从业者带来了福音

如果你是一个纯粹的网络从业者，那么加入ODL大家庭是你不二的选择。

ODL作为SDN应用最广泛的软件平台，几乎成为了各个场景的默认选择，项目经历了多个版本的迭代也逐渐趋于成熟和稳定，各个子项目也是百花齐放。但是对广大有志于投身网络变革大潮的从业者而言，这门技术依然具有很高的门槛，学习这本书，可以很好的帮助大家越过这个门槛。

作为一个网络工程师，面对SDN这个历史性节点，想越过ODL不太现实。但是想了解、学习ODL，确实又一时无法入门，难得其法。个人感觉可以通过这本书，窥得一些门道进而登堂入室。

对于我这种小白，急需上进的新手，内容是足够了
但是感觉在进阶问题上略有不足，缺乏足够落地的场景实验
希望作者再接再厉，出一版有实际场景应用的进阶书籍

一直在从事SDN方面的工作，最近工作需要开始接触ODL控制器，庞大的项目系统让我眼花缭乱，希望通过这本书慢慢学习ODL的基础框架，并应用到工作中！

我从事NFV领域，OPNFV的版本在多次迭代后现在支持的控制器只剩下ODL了，在NFV中网络控制器的地位非常重要。随着ODL的版本迭代与发展项目也变的越来越大资料也越来越多，耿老师的书讲解ODL的技术架构与原理为深入学习ODL提供了极大的帮助。

ODL在架构设计上的先进性和灵活性，ODL成了SDN领域最具影响力的开源平台。ODL不仅仅是一个控制器还是一个优秀的模型驱动架构实现，学习ODL还能够提升编程技

术还有助于提升架构设计能力。

[ODL技术内幕：架构设计与实现原理_下载链接1_](#)

书评

[ODL技术内幕：架构设计与实现原理_下载链接1_](#)