

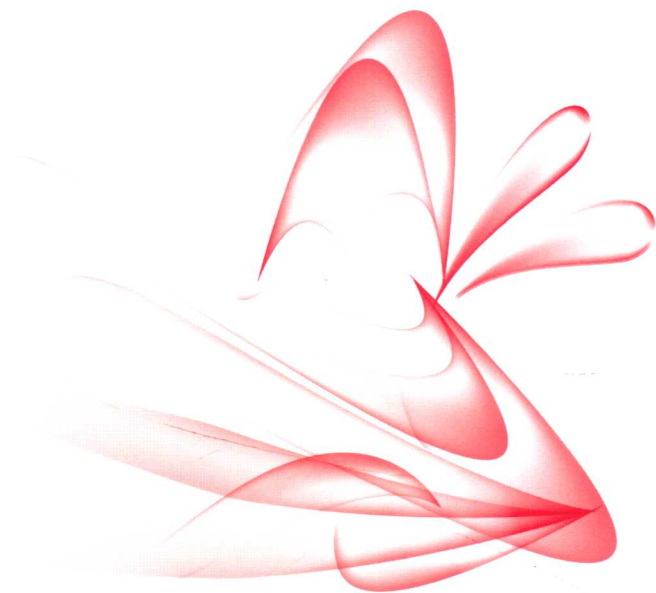
首部Hadoop YARN专著，资深Hadoop技术专家根据最新版本撰写，ChinaHadoop和51CTO等专业技术社区联袂推荐！

从应用角度系统讲解YARN的基本库和组件用法、应用程序设计方法、YARN上流行的各种计算框架，以及多个类YARN的开源资源管理系统。

从源代码角度深入分析YARN的设计理念与基本架构、各个组件的实现原理，以及各种计算框架的实现细节。



技术丛书



Hadoop Internals: in-depth study of YARN

Hadoop技术内幕

深入解析YARN架构设计与实现原理

董西成◎著



机械工业出版社
China Machine Press



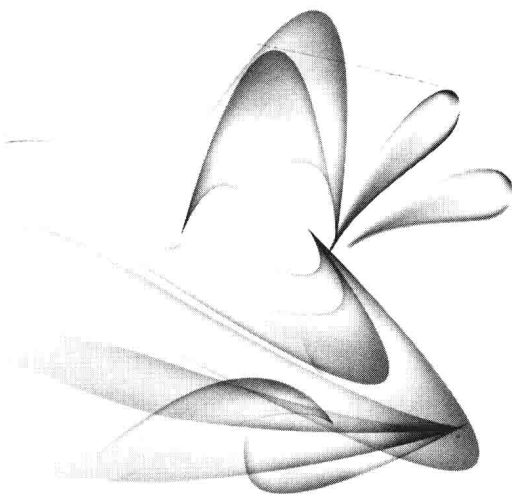
技术丛书

Hadoop Internals: in-depth study of YARN

Hadoop技术内幕

深入解析YARN架构设计与实现原理

董西成◎著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Hadoop 技术内幕: 深入解析 YARN 架构设计与实现原理 / 董西成著. — 北京: 机械工业出版社, 2013.12 (2014.1 重印)
(大数据技术丛书)

ISBN 978-7-111-44534-0

I. H… II. 董… III. 数据处理软件 IV. TP274

中国版本图书馆 CIP 数据核字 (2013) 第 252913 号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书是“Hadoop 技术内幕”系列的第 3 本书, 前面两本分别对 Common、HDFS 和 MapReduce 进行了深入分析和讲解, 赢得了极好的口碑, Hadoop 领域几乎人手一册, 本书则对 YARN 展开了深入的探讨, 是首部关于 YARN 的专著。仍然由资深 Hadoop 技术专家董西成执笔, 根据最新的 Hadoop 2.0 版本撰写, 权威社区 ChinaHadoop 鼎力推荐。

本书从应用角度系统讲解了 YARN 的基本库和组件用法、应用程序设计方法、YARN 上流行的各种计算框架 (MapReduce、Tez、Storm、Spark), 以及多个类 YARN 的开源资源管理系统 (Corona 和 Mesos); 从源代码角度深入分析 YARN 的设计理念与基本架构、各个组件的实现原理, 以及各种计算框架的实现细节。

全书共四部分 13 章: 第一部分 (第 1~2 章) 主要介绍了如何获取、阅读和调试 Hadoop 的源代码, 以及 YARN 的设计思想、基本架构和工作流程; 第二部分 (第 3~7 章) 结合源代码详细剖析和讲解了 YARN 的第三方开源库、底层通信库、服务库、事件库的基本使用和实现细节, 详细讲解了 YARN 的应用程序设计方法, 深入讲解和分析了 ResourceManager、资源调度器、NodeManager 等组件的实现细节; 第三篇 (第 8~10 章) 则对离线计算框架 MapReduce、DAG 计算框架 Tez、实时计算框架 Storm 和内存计算框架 Spark 进行了详细的讲解; 第四部分 (第 11~13 章) 首先对 Facebook Corona 和 Apache Mesos 进行了深入讲解, 然后对 YARN 的发展趋势进行了展望。附录部分收录了 YARN 安装指南、YARN 配置参数以及 Hadoop Shell 命令等非常有用的资料。



机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 孙海亮 罗词亮

藁城市京瑞印刷有限公司印刷

2014 年 1 月第 1 版第 2 次印刷

186mm × 240 mm · 24.75 印张

标准书号: ISBN 978-7-111-44534-0

定 价: 69.00 元

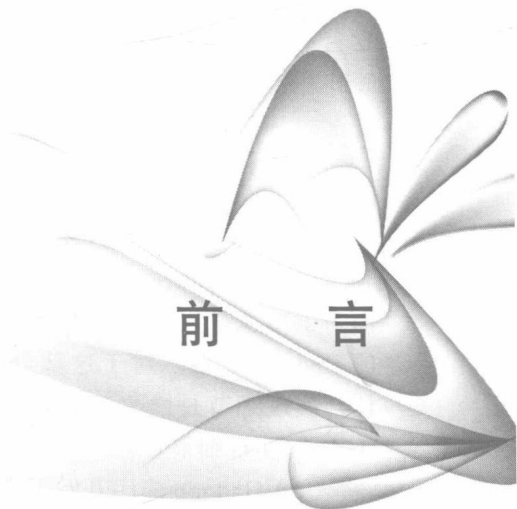
凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsjj@hzbook.com



前言

为什么要写这本书

在互联网巨头的带动下，开源软件 Hadoop 的应用变得越来越广泛，目前互联网、金融、银行、零售等行业均在使用或者尝试使用 Hadoop。IDC 对未来几年中国的预测中就专门提到了大数据，其认为未来几年，会有越来越多的企业级用户试水大数据平台和应用，而这之中，Hadoop 将成为最耀眼的“明星”。

尽管 Hadoop 整个生态系统是开源的，但由于它包含的软件种类过多，且版本升级过快，大部分公司，尤其是一些中小型公司，难以在有限的时间内快速掌握 Hadoop 蕴含的价值。此外，Hadoop 自身版本的多样化也给很多研发人员带来了很大的学习负担，尽管当前市面上已有很多参考书籍，但遗憾的是，能够深入剖析 Hadoop 内部实现细节的书籍少之又少，而本书则尝试弥补这一缺憾。本书是笔者继《Hadoop 技术内幕：深入解析 MapReduce 架构设计与实现原理》之后的又一本剖析 Hadoop 内幕的书籍。

本书介绍的 YARN (Yet Another Resource Negotiator) 系统是 Hadoop 2.0 新增加的一个子项目 (与 Common、MapReduce 和 HDFS 三个分支并列)，它的引入使得分布式计算系统进入平台化时代，即各种计算框架可以运行在一个集群中，由资源管理系统进行统一管理和调度，它们共享整个集群中的资源进而提高资源利用率。

本书以 Hadoop 2.0 为基础，从基本概念、程序设计和内部实现等方面深入剖析了 Hadoop YARN。本书重点分析了 YARN 的核心实现以及运行在 YARN 上的计算框

架, 其中, 核心实现包括基础库、编程接口、ResourceManager 实现、资源调度器实现、NodeManager 实现等, 而计算框架则包括离线计算框架 MapReduce、DAG 计算框架 Tez、实时计算框架 Storm 和内存计算框架 Spark 等。书中不仅详细介绍了 YARN 各个组件和计算框架的内部实现原理, 而且结合源代码进行了深入剖析, 使读者可以快速、全面地学习 Hadoop YARN 设计原理和实现细节。

读者对象

(1) Hadoop 二次开发人员

由于在扩展性、容错性和稳定性等方面的诸多优点, Hadoop 已被越来越多公司采用, 而为了减少开发成本, 大部分公司在 Hadoop 基础上进行二次开发, 以打造属于公司内部的平台。对于这部分 Hadoop 二次开发人员, 深入而又全面地了解 Hadoop 的设计与实现细节是修改 Hadoop 内核的前提, 而本书可帮助这部分读者快速而又全面地了解 Hadoop 实现细节。

(2) Hadoop 应用开发人员

如果要利用 Hadoop 进行高级应用开发, 仅掌握 Hadoop 基本使用方法是远远不够的, 必须对 Hadoop 框架的设计原理、架构和运作机制有一定的了解。对这部分读者而言, 本书将带领他们全面了解 Hadoop 的设计和实现, 加深对 Hadoop 框架的理解, 提高开发水平, 从而编写出更加高效的应用程序。

(3) Hadoop 运维工程师

对于一名合格的 Hadoop 运维工程师而言, 适当地了解 Hadoop 框架的设计原理、架构和运作机制是十分有帮助的, 这不仅可以更快地排除各种可能的 Hadoop 故障, 也能够让 Hadoop 运维人员与研发人员进行更有效地沟通。通过阅读本书, Hadoop 运维人员可以了解到很多从其他书中无法获取的 Hadoop 实现细节。

(4) 开源软件爱好者

Hadoop 是开源软件中的佼佼者, 它在实现的过程中吸收了很多开源领域的优秀思想, 同时也有很多值得学习的创新。尤为值得一提的是, 本书分析 Hadoop 设计和实现的方式也许值得所有开源软件爱好者进行学习和借鉴。通过阅读本书, 这部分读者不仅能领略到开源软件的优秀思想, 还可以掌握分析开源软件源代码的方法和技巧, 从而进一步提高使用开源软件的效率和质量。

如何阅读本书

本书分为四大部分 (不包括附录):

第一部分为基础篇 (第 1 ~ 2 章), 简单地介绍 Hadoop YARN 的环境搭建和基本设计架构, 帮助读者了解一些基础背景知识。

第二部分为 YARN 核心设计篇（第 3 ~ 7 章），着重讲解 YARN 基本库、应用程序设计方法和运行时环境的实现，包括 ResourceManager、NodeManager 和资源调度器等关键组件的内部实现细节。

第三部分为计算框架篇（第 8 ~ 10 章），主要讲解当前比较流行的可运行在 YARN 上的计算框架，包括离线计算框架 MapReduce、DAG 计算框架 Tez、实时计算框架 Storm 和内存计算框架 Spark。

第四部分为高级篇（第 11 ~ 13 章），主要介绍了几个类似于 Hadoop YARN 的开源资源管理系统，包括 Corona、Mesos 等，并总结了资源管理系统的特点及发展趋势。

另外本书最后还添加了几个附录：附录 A 为 YARN 安装指南；附录 B 介绍了常见的 YARN 配置参数；附录 C 介绍了常用的 Hadoop Shell 命令；附录 D 为本书的所有参考资料，包括参考论文、Hadoop jira 和网络资源等。

Hadoop YARN 是 Hadoop 2.0 新引入的系统，对于大部分读者而言，该系统存在很多疑惑与未知之处，而本书正是尝试全方位剖析该系统。为了能够系统化地学习 YARN，推荐读者从第 1 章的基础理论知识开始学习。

勘误和支持

由于笔者的水平有限，加之编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。为此，笔者特意创建一个在线支持与应急方案的站点 <http://hadoop123.com>。你可以将书中的错误发布在 Bug 勘误表页面中，同时如果你遇到任何问题，也可以访问 Q&A 页面，我将尽量在线上为读者提供最满意的解答。如果你有更多的宝贵意见，也欢迎发送邮件至邮箱 dongxicheng@yahoo.com，期待能够得到你们的真挚反馈。

致谢

感谢我的导师廖华明副研究员，是她引我进入 Hadoop 世界。

感谢腾讯的蔡斌老师，正是由于他的推荐，才使得两本 Hadoop 书的出版成为可能。

感谢机械工业出版社华章公司的杨福川老师和孙海亮老师在这一年多的时间中始终支持我的写作，他们的鼓励和帮助使我顺利完成了本书。

感谢何鹏、姜冰、郑伟伟、战科宇、周礼、刘晏辰、王群等人给我提供的各种帮助。

最后感谢我的父母，感谢他们的养育之恩，感谢兄长的鼓励和支持，感谢他们时时刻刻给我信心和力量！感谢我的女朋友颢悦对我生活的细心照料与琐事上的宽容。

谨以此书献给我最亲爱的家人，以及众多热爱 Hadoop 的朋友们！

董西成
于北京



目 录

前 言

第一部分 准备篇

第 1 章 环境准备	2
1.1 准备学习环境	2
1.1.1 基础软件下载	2
1.1.2 如何准备 Linux 环境	3
1.2 获取 Hadoop 源代码	5
1.3 搭建 Hadoop 源代码阅读环境	5
1.3.1 创建 Hadoop 工程	5
1.3.2 Hadoop 源代码阅读技巧	8
1.4 Hadoop 源代码组织结构	10
1.5 Hadoop 初体验	12
1.5.1 搭建 Hadoop 环境	12
1.5.2 Hadoop Shell 介绍	15

1.6 编译及调试 Hadoop 源代码	16
1.6.1 编译 Hadoop 源代码	17
1.6.2 调试 Hadoop 源代码	18
1.7 小结	20
第 2 章 YARN 设计理念与基本架构	21
2.1 YARN 产生背景	21
2.1.1 MRv1 的局限性	21
2.1.2 轻量级弹性计算平台	22
2.2 Hadoop 基础知识	23
2.2.1 术语解释	23
2.2.2 Hadoop 版本变迁	25
2.3 YARN 基本设计思想	29
2.3.1 基本框架对比	29
2.3.2 编程模型对比	30
2.4 YARN 基本架构	31
2.4.1 YARN 基本组成结构	32
2.4.2 YARN 通信协议	34
2.5 YARN 工作流程	35
2.6 多角度理解 YARN	36
2.6.1 并行编程	36
2.6.2 资源管理系统	36
2.6.3 云计算	37
2.7 本书涉及内容	38
2.8 小结	38

第二部分 YARN 核心设计篇

第 3 章 YARN 基础库	40
3.1 概述	40
3.2 第三方开源库	41
3.2.1 Protocol Buffers	41

3.2.2	Apache Avro	43
3.3	底层通信库	46
3.3.1	RPC 通信模型	46
3.3.2	Hadoop RPC 的特点概述	48
3.3.3	RPC 总体架构	48
3.3.4	Hadoop RPC 使用方法	49
3.3.5	Hadoop RPC 类详解	51
3.3.6	Hadoop RPC 参数调优	57
3.3.7	YARN RPC 实现	57
3.3.8	YARN RPC 应用实例	61
3.4	服务库与事件库	65
3.4.1	服务库	66
3.4.2	事件库	66
3.4.3	YARN 服务库和事件库的使用方法	68
3.4.4	事件驱动带来的变化	70
3.5	状态机库	72
3.5.1	YARN 状态转换方式	72
3.5.2	状态机类	73
3.5.3	状态机的使用方法	73
3.5.4	状态机可视化	76
3.6	源代码阅读引导	76
3.7	小结	77
3.8	问题讨论	77
第 4 章	YARN 应用程序设计方法	78
4.1	概述	78
4.2	客户端设计	79
4.2.1	客户端编写流程	80
4.2.2	客户端编程库	84
4.3	ApplicationMaster 设计	84
4.3.1	ApplicationMaster 编写流程	84
4.3.2	ApplicationMaster 编程库	92
4.4	YARN 应用程序实例	95

4.4.1 DistributedShell	95
4.4.2 Unmanaged AM	99
4.5 源代码阅读引导	100
4.6 小结	100
4.7 问题讨论	100
第 5 章 ResourceManager 剖析	102
5.1 概述	102
5.1.1 ResourceManager 基本职能	102
5.1.2 ResourceManager 内部架构	103
5.1.3 ResourceManager 事件与事件处理器	106
5.2 用户交互模块	108
5.2.1 ClientRMService	108
5.2.2 AdminService	109
5.3 ApplicationMaster 管理	109
5.4 NodeManager 管理	112
5.5 Application 管理	113
5.6 状态机管理	114
5.6.1 RMAp 状态机	115
5.6.2 RMApAttempt 状态机	119
5.6.3 RMContainer 状态机	123
5.6.4 RMNode 状态机	127
5.7 几个常见行为分析	129
5.7.1 启动 ApplicationMaster	129
5.7.2 申请与分配 Container	132
5.7.3 杀死 Application	134
5.7.4 Container 超时	135
5.7.5 ApplicationMaster 超时	138
5.7.6 NodeManager 超时	138
5.8 安全管理	139
5.8.1 术语介绍	139
5.8.2 Hadoop 认证机制	139
5.8.3 Hadoop 授权机制	142

5.9 容错机制	144
5.9.1 Hadoop HA 基本框架	145
5.9.2 YARN HA 实现	148
5.10 源代码阅读引导	149
5.11 小结	151
5.12 问题讨论	152
第 6 章 资源调度器	153
6.1 资源调度器背景	153
6.2 HOD 调度器	154
6.2.1 Torque 资源管理器	154
6.2.2 HOD 作业调度	155
6.3 YARN 资源调度器的基本架构	157
6.3.1 基本架构	157
6.3.2 资源表示模型	160
6.3.3 资源调度模型	161
6.3.4 资源抢占模型	164
6.4 YARN 层级队列管理机制	169
6.4.1 层级队列管理机制	169
6.4.2 队列命名规则	171
6.5 Capacity Scheduler	172
6.5.1 Capacity Scheduler 的功能	172
6.5.2 Capacity Scheduler 实现	176
6.6 Fair Scheduler	179
6.6.1 Fair Scheduler 功能介绍	180
6.6.2 Fair Scheduler 实现	182
6.6.3 Fair Scheduler 与 Capacity Scheduler 对比	183
6.7 其他资源调度器介绍	184
6.8 源代码阅读引导	185
6.9 小结	186
6.10 问题讨论	187

第 7 章 NodeManager 剖析	188
7.1 概述	188
7.1.1 NodeManager 基本职能	188
7.1.2 NodeManager 内部架构	190
7.1.3 NodeManager 事件与事件处理器	193
7.2 节点健康状况检测	194
7.2.1 自定义 Shell 脚本	194
7.2.2 检测磁盘损坏数目	196
7.3 分布式缓存机制	196
7.3.1 资源可见性与分类	198
7.3.2 分布式缓存实现	200
7.4 目录结构管理	203
7.4.1 数据目录管理	203
7.4.2 日志目录管理	203
7.5 状态机管理	206
7.5.1 Application 状态机	207
7.5.2 Container 状态机	210
7.5.3 LocalizedResource 状态机	213
7.6 Container 生命周期剖析	214
7.6.1 Container 资源本地化	214
7.6.2 Container 运行	218
7.6.3 Container 资源清理	222
7.7 资源隔离	224
7.7.1 Cgroups 介绍	224
7.7.2 内存资源隔离	228
7.7.3 CPU 资源隔离	230
7.8 源代码阅读引导	234
7.9 小结	235
7.10 问题讨论	236

第三部分 计算框架篇

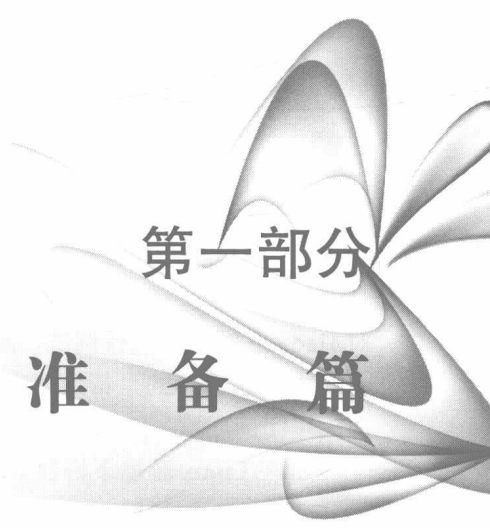
第 8 章 离线计算框架 MapReduce	238
8.1 概述	238
8.1.1 基本构成	238
8.1.2 事件与事件处理器	240
8.2 MapReduce 客户端	241
8.2.1 ApplicationClientProtocol 协议	242
8.2.2 MRClientProtocol 协议	243
8.3 MRAppMaster 工作流程	243
8.4 MR 作业生命周期及相关状态机	246
8.4.1 MR 作业生命周期	246
8.4.2 Job 状态机	249
8.4.3 Task 状态机	253
8.4.4 TaskAttempt 状态机	255
8.5 资源申请与再分配	259
8.5.1 资源申请	259
8.5.2 资源再分配	262
8.6 Container 启动与释放	263
8.7 推测执行机制	264
8.7.1 算法介绍	265
8.7.2 推测执行相关类	266
8.8 作业恢复	267
8.9 数据处理引擎	269
8.10 历史作业管理器	271
8.11 MRv1 与 MRv2 对比	273
8.11.1 MRv1 On YARN	273
8.11.2 MRv1 与 MRv2 架构比较	274
8.11.3 MRv1 与 MRv2 编程接口兼容性	274
8.12 源代码阅读引导	275
8.13 小结	277
8.14 问题讨论	277

第 9 章 DAG 计算框架 Tez	278
9.1 背景	278
9.2 Tez 数据处理引擎	281
9.2.1 Tez 编程模型	281
9.2.2 Tez 数据处理引擎	282
9.3 DAG Master 实现	284
9.3.1 DAG 编程模型	284
9.3.2 MR 到 DAG 转换	286
9.3.3 DAGAppMaster	288
9.4 优化机制	291
9.4.1 当前 YARN 框架存在的问题	291
9.4.2 Tez 引入的优化技术	292
9.5 Tez 应用场景	292
9.6 与其他系统比较	294
9.7 小结	295
第 10 章 实时 / 内存计算框架 Storm/Spark	296
10.1 Hadoop MapReduce 的短板	296
10.2 实时计算框架 Storm	296
10.2.1 Storm 编程模型	297
10.2.2 Storm 基本架构	302
10.2.3 Storm On YARN	304
10.3 内存计算框架 Spark	307
10.3.1 Spark 编程模型	308
10.3.2 Spark 基本架构	312
10.3.3 Spark On YARN	316
10.3.4 Spark/Storm On YARN 比较	317
10.4 小结	317

第四部分 高级篇

第 11 章 Facebook Corona 剖析	320
11.1 概述	320
11.1.1 Corona 的基本架构	320
11.1.2 Corona 的 RPC 协议与序列化框架	322
11.2 Corona 设计特点	323
11.2.1 推式网络通信模型	323
11.2.2 基于 Hadoop 0.20 版本	324
11.2.3 使用 Thrift	324
11.2.4 深度集成 Fair Scheduler	324
11.3 工作流程介绍	324
11.3.1 作业提交	325
11.3.2 资源申请与任务启动	326
11.4 主要模块介绍	327
11.4.1 ClusterManager	327
11.4.2 CoronaJobTracker	330
11.4.3 CoronaTaskTracker	333
11.5 小结	335
第 12 章 Apache Mesos 剖析	336
12.1 概述	336
12.2 底层网络通信库	337
12.2.1 libprocess 基本架构	338
12.2.2 一个简单示例	338
12.3 Mesos 服务	340
12.3.1 SchedulerProcess	341
12.3.2 Mesos Master	342
12.3.3 Mesos Slave	343
12.3.4 ExecutorProcess	343
12.4 Mesos 工作流程	344
12.4.1 框架注册过程	344

12.4.2 Framework Executor 注册过程	345
12.4.3 资源分配到任务运行过程	345
12.4.4 任务启动过程	347
12.4.5 任务状态更新过程	347
12.5 Mesos 资源分配策略	348
12.5.1 Mesos 资源分配框架	349
12.5.2 Mesos 资源分配算法	349
12.6 Mesos 容错机制	350
12.6.1 Mesos Master 容错	350
12.6.2 Mesos Slave 容错	351
12.7 Mesos 应用实例	352
12.7.1 Hadoop On Mesos	352
12.7.2 Storm On Mesos	353
12.8 Mesos 与 YARN 对比	354
12.9 小结	355
第 13 章 YARN 总结与发展趋势	356
13.1 资源管理系统设计动机	356
13.2 资源管理系统架构演化	357
13.2.1 集中式架构	357
13.2.2 双层调度架构	358
13.2.3 共享状态架构	358
13.3 YARN 发展趋势	359
13.3.1 YARN 自身的完善	359
13.3.2 以 YARN 为核心的生态系统	361
13.3.3 YARN 周边工具的完善	363
13.4 小结	363
附录 A YARN 安装指南	364
附录 B YARN 配置参数介绍	367
附录 C Hadoop Shell 命令介绍	371
附录 D 参考资料	374



第一部分 准备篇

由于 MRv1 (MapReduce version 1) 在扩展性、可靠性、资源利用率和多框架等方面存在明显不足, 故 Apache 开始尝试对 MapReduce 进行升级改造, 进而诞生了下一代 MapReduce 计算框架 MRv2 (MapReduce version 2)。由于 MRv2 将资源管理模块构建成了一个独立的通用系统 YARN, 这使得 MRv2 的核心从单一计算框架 MapReduce 转移为通用资源管理系统 YARN。本书第一部分将介绍学习 MRv2 前的准备工作, 并给出 MRv2 和 YARN 的基本概念和架构。