

阿里巴巴、雅虎、Twitter、Groupon等众多知名互联网企业
正在使用的大数据实时计算最新利器

从零开始学 Storm

赵必厦 程丽明 编著



- 国内首本Storm技术教程
- 技术基础、平台部署与应用案例完全解析



清华大学出版社

从零开始学

Storm

赵必厦 程丽明 编著



清华大学出版社
北 京

内 容 简 介

本书由基础知识、安装与部署、研发与维护、进阶知识、企业应用 5 个模块构成，并细分为 20 个章节，其中“基础知识”6 章、“安装与部署”4 章、“研发与维护”4 章、“进阶知识”5 章、“企业应用”1 章，分别介绍了 Storm 的安装与配置、Storm 的基本原理、Topology 组件、Spout 组件、Bolt 组件、ZooKeeper 集群、实战环节等内容，包括理论基础、环境搭建、研发准备、应用案例等。

本书理论联系实际，通过大量实例分析，让读者在较短的时间内掌握 Storm 的使用，搭建并研发出自己的基于 Storm 的大数据处理平台。

本书适合所有大数据处理、实时流数据处理、Storm 的开发者或爱好者，也适合高等院校和培训学校相关专业的师生参考使用。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目 (CIP) 数据

从零开始学 Storm/赵必厦，程丽明编著. —北京：清华大学出版社，2014

ISBN 978-7-302-37245-5

I. ①从… II. ①赵… ②程… III. ①数据处理软件 IV. ①TP274

中国版本图书馆 CIP 数据核字 (2014) 第 154066 号

责任编辑：王金柱

封面设计：王 翔

责任校对：闫秀华

责任印制：何 芊

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者：清华大学印刷厂

装 订 者：三河市新茂装订有限公司

经 销：全国新华书店

开 本：190mm×260mm 印 张：23.75 字 数：608 千字

版 次：2014 年 10 月第 1 版 印 次：2014 年 10 月第 1 次印刷

印 数：1~3000

定 价：59.00 元



前言

Storm是一个免费开源的分布式实时计算系统。Storm能轻松可靠地处理无界的数据流，就像Hadoop批处理一样对数据进行实时处理；但是Storm能持续运作下去，并且Storm的使用十分简单，开发人员可以使用任何编程语言对它进行操作，得到满意的结果。

本书以Storm官方网站最新的Release版本Storm 0.9.0.1进行讲解，从零开始，使读者在较短的时间内系统掌握Storm的理论基础，面向Linux平台搭建与研发自己基于Storm的大数据处理平台。全书分为5个模块，共20章内容，其中“基础知识”6章、“安装与部署”4章、“研发与维护”4章、“进阶知识”5章、“企业应用”1章，分别介绍了Storm的安装与配置、Storm的基本原理、Topology组件、Spout组件、Bolt组件、ZooKeeper集群、实战环节等内容，包括理论基础、环境搭建、研发准备、应用案例等。

作为国内第一本介绍Storm方面的书籍，本书的Storm理论部分主要参考了Storm Wiki；另外，为了更好地理解Storm，也参考了不少Storm爱好者的中文翻译文章；而Storm的应用部分，是本书编者的实战应用经验与理论相结合的结晶。本书在编写过程中，得到了多位Storm开发者/爱好者的帮助以及对本书的校验和建议，在此表示感谢，特别要感谢赵鸿、陈险峰两位同志。

由于编者水平有限，且本书涉及的知识点较多，书中难免有不妥和错误之处，敬请广大读者批评指正。

书中的源代码主要来自storm-starter项目，下载地址为：<https://github.com/nathanmarz/storm-starter>。

为了更有效地学习，建议读者在开始学习或阅读本书前先了解Linux的基本操作、Shell脚本的基本语法、Java语言的基本语法等内容。

编者

2014年4月

目 录

第 1 章 Storm 简介

1.1	什么是 Storm	2
1.2	Storm 的诞生	3
1.2.1	从 Twitter 说起.....	3
1.2.2	Twitter 需要处理大批实时性要求高的大数据业务	4
1.2.3	Storm 帮助 Twitter 解决实时海量大数据处理问题.....	4
1.3	Storm 的成长	5
1.3.1	Storm 正式开源	5
1.3.2	Storm 的核心技术和基本组成	6
1.3.3	Storm 的项目小组	7
1.3.4	Storm 的技术支持网站	12
1.4	Storm 的优势	15
1.4.1	集成多种技术	15
1.4.2	简单的 API.....	16
1.4.3	可扩展的	16
1.4.4	容错的	16
1.4.5	保证数据处理	17
1.4.6	可以使用任何语言	17
1.4.7	部署和操作简单	17
1.4.8	自由开源	17
1.5	Storm 的应用现状和发展趋势	18
1.5.1	应用现状	18
1.5.2	发展趋势	20
1.6	如何学习 Storm	22
1.7	本书的章节安排及学习建议	23
1.7.1	本书的章节安排	23
1.7.2	关于如何阅读本书的建议	24
1.8	本章小结	24

第 2 章 Storm 的基本知识

2.1 概念 26

2.1.1 元组 (Tuple) 26

2.1.2 流 (Stream) 27

2.1.3 喷口 (Spout) 28

2.1.4 螺栓 (Bolt) 28

2.1.5 拓扑 (Topology) 29

2.1.6 主控节点与工作节点 29

2.1.7 Nimbus 进程与 Supervisor 进程 29

2.1.8 流分组 (Stream grouping) 30

2.1.9 工作进程 (Worker) 30

2.1.10 任务 (Task) 30

2.1.11 执行器 (Executor) 30

2.1.12 可靠性 (Reliability) 30

2.2 Storm 的配置 31

2.2.1 Storm 的配置类型 31

2.2.2 defaults.yaml 文件 32

2.2.3 storm.yaml 文件 35

2.2.4 Config 类 36

2.3 序列化 (Serialization) 37

2.3.1 动态类型 37

2.3.2 自定义序列化 37

2.3.3 Java 序列化 38

2.3.4 特定组件序列化注册 38

2.4 容错机制 39

2.4.1 Worker 进程死亡 39

2.4.2 节点死亡 39

2.4.3 Nimbus 或者 Supervisor 守护进程死亡 39

2.4.4 Nimbus 是否是“单点故障”的 39

2.5 可靠性机制——保证消息处理 40

2.5.1 消息被“完全处理”的含义 40

2.5.2 如果一个消息被完全处理或完全处理失败会发生什么 41

2.5.3 Storm 如何保证可靠性 41

2.5.4 Storm 如何实现可靠性 44

2.5.5 调节可靠性 45

2.6 消息传输机制 46

2.6.1 ZeroMQ 46

2.6.2	Netty	46
2.6.3	自定义消息通信机制	47
2.7	Storm 的开发环境与生产环境	47
2.7.1	开发环境与本地模式	47
2.7.2	生产环境与远程模式	47
2.7.3	开发环境与生产环境的对比	48
2.8	Storm 拓扑的并行度 (parallelism)	49
2.8.1	工作进程、执行器和任务	49
2.8.2	配置拓扑的并行度	50
2.8.3	拓扑示例	51
2.8.4	如何改变运行中拓扑的并行度	52
2.9	Storm 命令行客户端	53
2.10	Javadoc 文档	57
2.11	本章小结	57

第 3 章 拓扑详解

3.1	什么是拓扑	59
3.2	TopologyBuilder	59
3.3	流分组	61
3.3.1	什么是流分组	61
3.3.2	不同的流分组方式	62
3.4	一个简单的拓扑	66
3.5	在本地模式下运行拓扑	69
3.6	在生产集群上运行拓扑	70
3.6.1	常见的配置	71
3.6.2	杀死拓扑	72
3.6.3	更新运行中的拓扑	72
3.6.4	监控拓扑	72
3.7	拓扑的常见模式	72
3.7.1	流连接 (Stream Join)	72
3.7.2	批处理 (Batching)	73
3.7.3	BasicBolt	73
3.7.4	内存中缓存与字段的组合	73
3.7.5	流的 top N	74
3.7.6	高效保存最近更新缓存对象的 TimeCacheMap (已弃用)	76
3.7.7	分布式 RPC 的 CoordinatedBolt 与 KeyedFairBolt	76
3.8	本地模式与 StormSubmitter 的对比	76

3.9 多语言协议 (Multi-Language Protocol) 78

3.10 使用非 JVM 语言操作 Storm 82

 3.10.1 支持的非 Java 语言 82

 3.10.2 对 Storm 使用非 Java 语言 82

 3.10.3 实现非 Java DSL 的笔记 82

3.11 Hook..... 83

3.12 本章小结 84

第 4 章 组件详解

4.1 基本接口 86

 4.1.1 IComponent 接口 86

 4.1.2 ISpout 接口 86

 4.1.3 IBolt 接口 88

 4.1.4 IRichSpout 与 IRichBolt 接口 89

 4.1.5 IBasicBolt 接口 90

 4.1.6 IStateSpout 与 IRichStateSpout 接口 90

4.2 基本抽象类 91

 4.2.1 BaseComponent 抽象类..... 91

 4.2.2 BaseRichSpout 抽象类 92

 4.2.3 BaseRichBolt 抽象类..... 93

 4.2.4 BaseBasicBolt 抽象类 93

4.3 事务接口 94

 4.3.1 IPartitionedTransactionalSpout 94

 4.3.2 IOpaquePartitionedTransactionalSpout..... 95

 4.3.3 ITransactionalSpout..... 96

 4.3.4 ICommitterTransactionalSpout 98

 4.3.5 IBatchBolt..... 98

4.4 组件之间的相互关系 98

4.5 本章小结 100

第 5 章 Spout 详解

5.1 可靠的与不可靠的消息 102

5.2 Spout 获取数据的方式 104

 5.2.1 直接连接 (Direct Connection) 105

 5.2.2 消息队列 (Enqueued Messages) 106

 5.2.3 DRPC (分布式 RPC) 106

5.3 常用的 Spout..... 107

5.3.1 Kestrel 作为 Spout 的数据源	107
5.3.2 AMQP 作为 Spout 的数据源	107
5.3.3 JMS 作为 Spout 的数据源	107
5.3.4 Redis 作为 Spout 的数据源	108
5.3.5 beanstalkd 作为 Spout 的数据源	108
5.4 学习编写 Spout 类	108
5.5 本章小结	109

第 6 章 Bolt 详解

6.1 Bolt 概述	111
6.2 可靠的与不可靠的 Bolt	112
6.2.1 使用 Anchoring 机制实现可靠的 Bolt	112
6.2.2 使用 IBasicBolt 接口实现自动确认	113
6.3 复合流与复合 Anchoring	114
6.3.1 复合流	114
6.3.2 复合 Anchoring	114
6.4 使用其他语言定义 Bolt	114
6.5 学习编写 Bolt 类	115
6.5.1 可靠的 Bolt	115
6.5.2 不可靠的 Bolt	116
6.6 本章小结	117

第 7 章 ZooKeeper 详解

7.1 ZooKeeper 简介	119
7.2 ZooKeeper 的下载和部署	119
7.2.1 ZooKeeper 的下载	119
7.2.2 ZooKeeper 的部署	120
7.3 ZooKeeper 的配置	121
7.4 ZooKeeper 的运行	123
7.5 ZooKeeper 的本地模式实例	125
7.6 ZooKeeper 的数据模型	126
7.6.1 ZNode	126
7.6.2 ZooKeeper 中的时间	127
7.6.3 ZooKeeper 的 Stat 结构	128
7.7 ZooKeeper 的命令行操作范例	128
7.8 Storm 在 ZooKeeper 中的目录结构	131
7.9 本章小结	132

第 8 章 基础软件的安装与使用

8.1	Linux 的基本操作	134
8.1.1	环境变量	134
8.1.2	常用命令	135
8.2	JDK 的下载与配置	138
8.2.1	Sun JDK 的下载	138
8.2.2	在 Linux 下安装 JDK	140
8.2.3	在 Windows 下安装 JDK	141
8.3	GitHub 托管项目的下载	146
8.4	Maven 的下载与配置	147
8.4.1	Maven 的下载	148
8.4.2	在 Linux 下部署 Maven	149
8.4.3	在 Windows 下部署 Maven	149
8.5	其他软件——Notepad++	151
8.6	本章小结	151

第 9 章 Storm 的安装与配置

9.1	Storm 集群的安装步骤与准备工作	153
9.1.1	搭建 ZooKeeper 集群	153
9.1.2	安装 Storm 的本地依赖	153
9.1.3	下载并解压 Storm 发行版本	156
9.1.4	配置 storm.yaml 文件	157
9.1.5	启动 Storm 的守护进程	159
9.2	本地模式的 Storm 完整的配置命令	161
9.3	本章小结	162

第 10 章 Storm 集群搭建实践

10.1	准备工作	164
10.1.1	概述	164
10.1.2	配置 hosts 文件	164
10.1.3	配置静态 IP	165
10.1.4	集群 SSH 无密码	166
10.1.5	修改主机名	167
10.1.6	关闭防火墙	168
10.1.7	同步时间	168

10.1.8	安装 JDK	168
10.2	ZooKeeper 集群的搭建	169
10.2.1	部署第一个节点	169
10.2.2	部署第 i 个节点	170
10.2.3	启动 ZooKeeper 集群	171
10.2.4	查看 ZooKeeper 状态	171
10.2.5	关闭 ZooKeeper 集群	171
10.2.6	清理 ZooKeeper 集群	171
10.3	Storm 集群的搭建	172
10.3.1	安装 Storm 依赖 (每个 Storm 节点)	172
10.3.2	部署第一个节点	173
10.3.3	部署第 i 个节点	174
10.3.4	启动 Storm 守护进程	174
10.4	本章小结	175

第 11 章 准备 Storm 的开发环境

11.1	Storm 的开发环境	177
11.1.1	什么是 Storm 的开发环境	177
11.1.2	如何管理 Storm	177
11.1.3	如何提交拓扑到集群	181
11.2	Eclipse 的下载与配置	182
11.2.1	Eclipse 的下载	182
11.2.2	Eclipse 的配置与运行	183
11.2.3	Eclipse 插件的安装	184
11.3	使用 Maven 管理项目	186
11.3.1	Maven 的下载与配置	186
11.3.2	配置 pom.xml 文件	186
11.3.3	运行 Maven 命令	188
11.4	使用 Nexus 搭建本地 Maven 私服	188
11.4.1	下载 Nexus	189
11.4.2	运行 Nexus	189
11.4.3	登录 Nexus 后台	190
11.4.4	配置 Repositories	191
11.4.5	配置 setting.xml 文件	192
11.4.6	修改 Eclipse 的 Maven 插件的配置	194
11.5	使用 SVN 管理代码版本	195
11.5.1	在 Windows 下搭建 SVN 服务器	195
11.5.2	在 Linux 下搭建 SVN 服务器	196

11.5.3	安装 SVN 客户端	197
11.6	部署单节点的 Storm 集群	197
11.6.1	部署伪分布的 ZooKeeper	197
11.6.2	部署伪分布的 Storm 集群	198
11.7	本章小结	199

第 12 章 开发自己的 Storm 应用

12.1	新建 Maven 项目	201
12.2	修改为适合 Storm 开发的项目	203
12.2.1	对包名进行分类管理	203
12.2.2	修改 pom.xml 文件	204
12.3	编写代码	206
12.3.1	编写 Spout 类	206
12.3.2	编写 Bolt 类	207
12.3.3	编写 Topology 类	208
12.4	本地测试运行	209
12.5	提交到 Storm 集群运行	210
12.5.1	使用 Maven 打包	210
12.5.2	提交 jar 包到集群	210
12.6	本章小结	211

第 13 章 storm-starter 详解

13.1	storm-starter 项目概述	213
13.2	storm-starter 的下载	215
13.3	使用 Maven 进行管理	215
13.3.1	使用 Maven 打包 storm-starter	215
13.3.2	使用 Maven 直接运行 WordCountTopology	216
13.3.3	使用 Maven 运行单元测试	216
13.4	在 Eclipse 中运行	216
13.4.1	新建 Maven 项目的方式	216
13.4.2	导入已存在的项目的方式	218
13.5	storm-starter 的入门例子	219
13.5.1	ExclamationTopology	219
13.5.2	WordCountTopology	222
13.5.3	ReachTopology	224
13.6	storm-starter 的其他例子	230
13.6.1	BasicDRPCTopology	230

13.6.2	ManualDRPC	231
13.6.3	PrintSampleStream.....	231
13.6.4	RollingTopWords	232
13.6.5	SingleJoinExample.....	233
13.6.6	TransactionalGlobalCount	234
13.6.7	TransactionalWords.....	234
13.7	本章小结	235

第 14 章 研发与集群管理技巧

14.1	使用 daemontools 监控 Storm 进程	237
14.1.1	daemontools 简介	237
14.1.2	安装 daemontools	237
14.1.3	编写监控脚本	238
14.2	使用 Monit 监控 Storm.....	240
14.2.1	Monit 简介	240
14.2.2	安装 Monit.....	240
14.2.3	配置 Monit.....	242
14.2.4	启动 Monit.....	244
14.2.5	获取 Monit 帮助信息	245
14.3	常用的集群操作命令	246
14.4	使用 Storm 的经验与建议	247
14.5	本章小结	248

第 15 章 DRPC 详解

15.1	概述	250
15.2	DRPCTopologyBuilder	251
15.2.1	LinearDRPCTopologyBuilder.....	251
15.2.2	LinearDRPCTopologyBuilder 提供的方法	251
15.2.3	LinearDRPCTopologyBuilder 使用范例	252
15.2.4	LinearDRPCTopologyBuilder 的工作原理	253
15.2.5	LinearDRPCTopologyBuilder 目前已弃用	254
15.3	本地模式的 DRPC.....	254
15.4	远程模式的 DRPC.....	254
15.5	一个复杂的 DRPC 例子（计算 reach 值）	255
15.6	非线性 DRPC.....	257
15.7	本章小结	257

第 16 章 事务拓扑详解

16.1	什么是事务拓扑	259
16.1.1	设计 1	259
16.1.2	设计 2	260
16.1.3	设计 3 (Storm 的设计)	260
16.2	事务拓扑的设计细节	261
16.3	事务拓扑的实现细节	261
16.3.1	事务 Spout 的工作原理	262
16.3.2	“对于给定的事务 id 不能发射相同的 Batch” 的处理	262
16.3.3	更多的细节	264
16.4	事务拓扑 API	264
16.4.1	Bolt	264
16.4.2	事务 Spout	266
16.4.3	配置	266
16.5	TransactionalTopologyBuilder	267
16.5.1	TransactionalTopologyBuilder 提供的方法	267
16.5.2	TransactionalTopologyBuilder 类已弃用	270
16.6	一个简单的例子	270
16.7	本章小结	273

第 17 章 Trident 详解

17.1	Trident 概述	275
17.1.1	简单的例子——单词统计 (TridentWordCount)	275
17.1.2	另一个例子——计算 Reach 值 (TridentReach)	278
17.1.3	字段和元组	280
17.1.4	状态 (State)	281
17.1.5	Trident 拓扑的执行	282
17.2	Trident API	283
17.2.1	概述	283
17.2.2	本地分区操作	283
17.2.3	重新分区操作	288
17.2.4	聚合操作	288
17.2.5	流分组操作	288
17.2.6	合并与连接	289
17.3	Trident 的状态	290
17.3.1	Trident 状态分类	290

17.3.2	事务 Spout (Transactional Spout)	291
17.3.3	不透明事务 Spout (Opaque Transactional Spout)	292
17.3.4	非事务 Spout (Non-transactional Spout)	294
17.3.5	Spout 与 State 之间的联系	294
17.3.6	State API	294
17.3.7	persistentAggregate 方法	298
17.3.8	实现 MapStates	299
17.4	Trident Spout	299
17.4.1	流水线 (Pipelining)	300
17.4.2	Trident Spout 的类型	300
17.5	本章小结	300

第 18 章 Storm 的内部实现

18.1	文件系统分析	302
18.2	数据目录结构	303
18.2.1	Nimbus 节点的目录结构	303
18.2.2	Supervisor 节点的目录结构	304
18.3	代码库的结构	305
18.3.1	storm.thrift	305
18.3.2	Java 接口	312
18.3.3	实现	312
18.4	拓扑的生命周期	314
18.4.1	启动拓扑	315
18.4.2	监控拓扑	317
18.4.3	杀死拓扑	317
18.5	Acking 框架的实现	318
18.5.1	异或计算的基本原理	318
18.5.2	Acking 框架的实现原理	318
18.5.3	Acker 的 execute 方法	319
18.5.4	待定元组 (pending tuple) 和 RotatingMap	319
18.6	Metric	319
18.7	本章小结	325

第 19 章 Storm 相关的其他项目

19.1	JStorm 项目	327
19.1.1	项目简介	327
19.1.2	下载与部署	328

19.1.3	源代码编译	329
19.2	storm-deploy 项目	329
19.3	Storm 与 Kafka	329
19.3.1	Kafka 简介	329
19.3.2	Kafka 的安装	330
19.3.3	启动服务	331
19.3.4	测试运行	331
19.3.5	Storm 与 Kafka 的项目	333
19.4	storm-kestrel 项目	334
19.4.1	storm-kestrel 项目简介	334
19.4.2	使用 storm-kestrel 项目	334
19.4.3	Kestrel 服务器和队列	335
19.4.4	添加元素到 kestrel	336
19.4.5	从 Kestrel 中移除元素	336
19.4.6	持续添加元素到 Kestrel	337
19.4.7	使用 KestrelSpout	338
19.4.8	执行	339
19.5	本章小结	339

第 20 章 企业应用案例

20.1	Storm 席卷众多互联网企业	341
20.1.1	Storm 的典型应用场景	341
20.1.2	Storm 的三大基本应用	341
20.2	Storm 在 Twitter 中的应用	342
20.2.1	Twitter 公司简介	342
20.2.2	Storm 帮助 Twitter 提升产品性能	343
20.2.3	MapR 在 Twitter 中的应用简介	343
20.3	Storm 在阿里巴巴集团的应用	344
20.3.1	阿里巴巴集团简介	345
20.3.2	Storm 在阿里巴巴的应用	345
20.3.3	Storm 在淘宝公司的应用	346
20.3.4	Storm 在支付宝公司的应用	347
20.4	其他应用 Storm 的知名企业和项目	347
20.5	本章小结	363
	参考资料	364

第 1 章

Storm 简介

本章主要介绍 Storm 的起源，Storm 的基本概念、特征和优势，从哪里可以找到 Storm 的学习和源代码参考的网站，以及 Storm 的应用现状和发展趋势。通过本章的学习，读者将对 Storm 有一个基本的了解。