

阿里巴巴、百度、Twitter、雅虎、Groupon等众多知名互联网企业
正在使用的大数据分布式实时计算利器



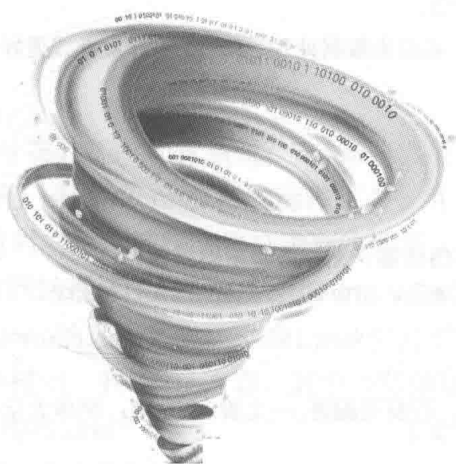
根据storm新版本编写

从零开始学 Storm

(第2版)

清华大学出版社

赵必厦 程丽明 编著



从零开始学 Storm

(第2版)

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 的基本原理、Topology 组件、Spout 组件、Bolt 组件、ZooKeeper 集群、Storm 的安装与配置、实战环节等内容，包括理论基础、环境搭建、研发准备、企业应用等。

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. — 2 版. — 北京：清华大学出版社，2016
ISBN 978-7-302-43807-6

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 (2016) 第 100183 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：李红英

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：三河市君旺印务有限公司

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：24

字 数：614 千字

版 次：2014 年 10 月第 1 版

2016 年 7 月第 2 版

印 次：2016 年 7 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能够轻松可靠地处理无界的数据流，就像Hadoop对数据进行批处理；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Apache Storm官方网站最新的Release版本Storm 0.10.0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的基本原理、Topology组件、Spout组件、Bolt组件、Storm的安装与配置、ZooKeeper集群、实战环节等内容，包括理论基础、环境搭建、研发准备、企业应用等。

本书的Storm理论部分主要参考了Storm官方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，是本书编者实战应用经验与理论相结合的结晶。本书在再版编写过程中，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。本书修正了上一版中的部分错误，根据Apache Storm 0.10.0更新了书中部分的插图、描述、代码等内容。

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，项目网址地址为：

<https://github.com/apache/storm/tree/master/examples/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。

如有任何问题，请发送电子邮件至booksaga@126.com，邮件标题为“从零开始学Storm（第2版）反馈”。

编者

2016年5月

目 录

第 1 章 Storm 简介

1.1	什么是 Storm	1
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	3
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	Apache 接管并孵化 Storm	5
1.3.3	Storm 的核心技术和基本组成	6
1.3.4	Storm 的项目小组	7
1.3.5	Storm 的技术支持网站	10
1.4	Storm 的优势	13
1.4.1	集成多种技术	13
1.4.2	简单的 API	13
1.4.3	可扩展的	14
1.4.4	容错的	14
1.4.5	保证数据处理	14
1.4.6	可以使用任何语言	14
1.4.7	部署和操作简单	15
1.4.8	自由开源	15
1.5	Storm 的应用现状和发展趋势	15
1.5.1	应用现状	16
1.5.2	发展趋势	18
1.6	如何学习 Storm	20
1.7	本书的章节安排及学习建议	21
1.7.1	本书的章节安排	21
1.7.2	关于如何阅读本书的建议	22
1.8	本章小结	23

第 2 章 Storm 的基本知识

2.1	概念	24
2.1.1	元组 (Tuple)	24
2.1.2	流 (Stream)	25
2.1.3	龙卷 (Spout)	26
2.1.4	闪电 (Bolt)	27
2.1.5	拓扑 (Topology)	27
2.1.6	主控节点与工作节点	28
2.1.7	Nimbus 进程与 Supervisor 进程	28
2.1.8	流分组 (Stream grouping)	28
2.1.9	工作进程 (Worker)	28
2.1.10	任务 (Task)	28
2.1.11	执行器 (Executor)	28
2.1.12	可靠性 (Reliability)	29
2.2	Storm 的配置	29
2.2.1	Storm 的配置类型	29
2.2.2	defaults.yaml 文件	30
2.2.3	storm.yaml 文件	33
2.2.4	Config 类	34
2.3	序列化 (Serialization)	35
2.3.1	动态类型	36
2.3.2	自定义序列化	36
2.3.3	Java 序列化	37
2.3.4	特定组件序列化注册	37
2.4	容错机制	37
2.4.1	Worker 进程死亡	37
2.4.2	节点死亡	38
2.4.3	Nimbus 或者 Supervisor 守护进程死亡	38
2.4.4	Nimbus 是否是“单点故障”	38
2.5	可靠性机制——保证消息处理	38
2.5.1	消息被“完全处理”的含义	38
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	39
2.5.3	Storm 如何保证可靠性	40
2.5.4	Storm 如何实现可靠性	43
2.5.5	调节可靠性	44
2.6	消息传输机制	45
2.6.1	ZeroMQ	45
2.6.2	Netty	45

2.6.3 自定义消息通信机制.....	45
2.7 Storm 的开发环境与生产环境.....	46
2.7.1 开发环境与本地模式.....	46
2.7.2 生产环境与远程模式.....	46
2.7.3 开发环境与生产环境的对比.....	47
2.8 Storm 拓扑的并行度 (parallelism)	48
2.8.1 工作进程、执行器和任务.....	48
2.8.2 配置拓扑的并行度.....	49
2.8.3 拓扑示例.....	50
2.8.4 如何改变运行中拓扑的并行度.....	51
2.9 Storm 命令行客户端.....	52
2.10 Javadoc 文档.....	56
2.11 本章小结.....	56

第 3 章 拓扑详解

3.1 什么是拓扑.....	57
3.2 TopologyBuilder.....	57
3.3 流分组.....	59
3.3.1 什么是流分组.....	59
3.3.2 不同的流分组方式.....	60
3.4 一个简单的拓扑.....	64
3.5 在本地模式下运行拓扑.....	67
3.6 在生产集群上运行拓扑.....	68
3.6.1 常见的配置.....	70
3.6.2 杀死拓扑.....	70
3.6.3 更新运行中的拓扑.....	71
3.6.4 监控拓扑.....	71
3.7 拓扑的常见模式.....	71
3.7.1 流连接 (Stream Join)	71
3.7.2 批处理 (Batching)	72
3.7.3 BasicBolt	72
3.7.4 内存中缓存与字段的组合.....	72
3.7.5 流的 top N.....	72
3.7.6 高效保存最近更新缓存对象的 TimeCacheMap (已弃用)	74
3.7.7 分布式 RPC 的 CoordinatedBolt 与 KeyedFairBolt	75
3.8 本地模式与 StormSubmitter 的对比.....	75
3.9 多语言协议 (Multi-Language Protocol)	77
3.10 使用非 JVM 语言操作 Storm.....	81
3.10.1 支持的非 Java 语言.....	81

- 3.10.2 对 Storm 使用非 Java 语言 81
- 3.10.3 实现非 Java DSL 的笔记 82
- 3.11 Hook 82
- 3.12 本章小结 83

第 4 章 组件详解

- 4.1 基本接口 84
 - 4.1.1 IComponent 接口 84
 - 4.1.2 ISpout 接口 85
 - 4.1.3 IBolt 接口 86
 - 4.1.4 IRichSpout 与 IRichBolt 接口 88
 - 4.1.5 IBasicBolt 接口 88
 - 4.1.6 IStateSpout 与 IRichStateSpout 接口 89
- 4.2 基本抽象类 90
 - 4.2.1 BaseComponent 抽象类 90
 - 4.2.2 BaseRichSpout 抽象类 90
 - 4.2.3 BaseRichBolt 抽象类 91
 - 4.2.4 BaseBasicBolt 抽象类 92
- 4.3 事务接口 92
 - 4.3.1 IPartitionedTransactionalSpout 92
 - 4.3.2 IOpaquePartitionedTransactionalSpout 94
 - 4.3.3 ITransactionalSpout 95
 - 4.3.4 ICommitterTransactionalSpout 96
 - 4.3.5 IBatchBolt 97
- 4.4 组件之间的相互关系 97
- 4.5 本章小结 98

第 5 章 Spout 详解

- 5.1 可靠的与不可靠的消息 99
- 5.2 Spout 获取数据的方式 102
 - 5.2.1 直接连接 (Direct Connection) 102
 - 5.2.2 消息队列 (Enqueued Messages) 103
 - 5.2.3 DRPC (分布式 RPC) 104
- 5.3 常用的 Spout 104
 - 5.3.1 Kestrel 作为 Spout 的数据源 104
 - 5.3.2 AMQP 作为 Spout 的数据源 104
 - 5.3.3 JMS 作为 Spout 的数据源 105
 - 5.3.4 Redis 作为 Spout 的数据源 105
 - 5.3.5 beanstalkd 作为 Spout 的数据源 105

5.4 学习编写 Spout 类.....	105
5.5 本章小结	106

第 6 章 Bolt 详解

6.1 Bolt 概述	107
6.2 可靠的与不可靠的 Bolt.....	108
6.2.1 使用 Anchoring 机制实现可靠的 Bolt	108
6.2.2 使用 IBasicBolt 接口实现自动确认	109
6.3 复合流与复合 Anchoring.....	110
6.3.1 复合流	110
6.3.2 复合 Anchoring.....	110
6.4 使用其他语言定义 Bolt.....	111
6.5 学习编写 Bolt 类.....	111
6.5.1 可靠的 Bolt.....	111
6.5.2 不可靠的 Bolt.....	112
6.6 本章小结	113

第 7 章 ZooKeeper 详解

7.1 ZooKeeper 简介	114
7.2 ZooKeeper 的下载和部署	114
7.2.1 ZooKeeper 的下载.....	114
7.2.2 ZooKeeper 的部署	115
7.3 ZooKeeper 的配置	117
7.4 ZooKeeper 的运行	119
7.5 ZooKeeper 的本地模式实例	120
7.6 ZooKeeper 的数据模型	121
7.6.1 ZNode	122
7.6.2 ZooKeeper 中的时间	123
7.6.3 ZooKeeper 的 Stat 结构.....	123
7.7 ZooKeeper 的命令行操作范例	124
7.8 Storm 在 ZooKeeper 中的目录结构	127
7.9 本章小结	128

第 8 章 基础软件的安装与使用

8.1 Linux 的基本操作	129
8.1.1 环境变量	129
8.1.2 常用命令	130
8.2 JDK 的下载与配置.....	134
8.2.1 Sun JDK 的下载	134

8.2.2	在 Linux 下安装 JDK	135
8.2.3	在 Windows 下安装 JDK	136
8.3	GitHub 托管项目的下载	141
8.4	Maven 的下载与配置	143
8.4.1	Maven 的下载	143
8.4.2	在 Linux 下部署 Maven	144
8.4.3	在 Windows 下部署 Maven	145
8.5	其他软件——Notepad++	146
8.6	本章小结	147

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	148
9.1.1	搭建 ZooKeeper 集群	148
9.1.2	安装 Storm 的本地依赖	148
9.1.3	下载并解压 Storm 发行版本	151
9.1.4	配置 storm.yaml 文件	153
9.1.5	启动 Storm 的守护进程	154
9.2	本地模式的 Storm 完整的配置命令	157
9.3	本章小结	159

第 10 章 Storm 集群搭建实践

10.1	准备工作	160
10.1.1	概述	160
10.1.2	配置 hosts 文件	161
10.1.3	配置静态 IP	161
10.1.4	集群 SSH 无密码	163
10.1.5	修改主机名	164
10.1.6	关闭防火墙	164
10.1.7	同步时间	164
10.1.8	安装 JDK	165
10.2	ZooKeeper 集群的搭建	166
10.2.1	部署第一个节点	166
10.2.2	部署第 i 个节点	167
10.2.3	启动 ZooKeeper 集群	167
10.2.4	查看 ZooKeeper 状态	168
10.2.5	关闭 ZooKeeper 集群	168
10.2.6	清理 ZooKeeper 集群	168
10.3	Storm 集群的搭建	168
10.3.1	安装 Storm 依赖(每个 Storm 节点)	168

10.3.2	部署第一个节点	169
10.3.3	部署第 i 个节点	171
10.3.4	启动 Storm 守护进程	171
10.4	本章小结	172

第 11 章 准备 Storm 的开发环境

11.1	Storm 的开发环境	173
11.1.1	什么是 Storm 的开发环境	173
11.1.2	如何管理 Storm	173
11.1.3	如何提交拓扑到集群	176
11.2	Eclipse 的下载与配置	176
11.2.1	Eclipse 的下载	176
11.2.2	Eclipse 的配置与运行	177
11.2.3	Eclipse 插件的安装	178
11.3	使用 Maven 管理项目	180
11.3.1	Maven 的下载与配置	180
11.3.2	配置 pom.xml 文件	180
11.3.3	运行 Maven 命令	182
11.4	使用 Nexus 搭建本地 Maven 私服	183
11.4.1	下载 Nexus	183
11.4.2	运行 Nexus	184
11.4.3	登录 Nexus 后台	184
11.4.4	配置 Repositories	185
11.4.5	配置 setting.xml 文件	187
11.4.6	修改 Eclipse 的 Maven 插件的配置	189
11.5	使用 SVN 管理代码版本	189
11.5.1	在 Windows 下搭建 SVN 服务器	189
11.5.2	在 Linux 下搭建 SVN 服务器	191
11.5.3	安装 SVN 客户端	191
11.6	部署单节点的 Storm 集群	192
11.6.1	部署伪分布的 ZooKeeper	192
11.6.2	部署伪分布的 Storm 集群	192
11.7	本章小结	194

第 12 章 开发自己的 Storm 应用

12.1	新建 Maven 项目	195
12.2	修改为适合 Storm 开发的项目	198
12.2.1	对包名进行分类管理	198
12.2.2	修改 pom.xml 文件	199

12.3	编写代码	201
12.3.1	编写 Spout 类	201
12.3.2	编写 Bolt 类	202
12.3.3	编写 Topology 类	203
12.4	本地测试运行	204
12.5	提交到 Storm 集群运行	205
12.5.1	使用 Maven 打包	205
12.5.2	提交 jar 包到集群	205
12.6	本章小结	206

第 13 章 storm-starter 详解

13.1	storm-starter 项目概述	207
13.2	storm-starter 的下载	209
13.3	使用 Maven 进行管理	211
13.3.1	使用 Maven 打包 storm-starter	211
13.3.2	使用 Maven 直接运行 ExclamationTopology	211
13.3.3	使用 Maven 运行单元测试	211
13.4	在 Eclipse 中运行	212
13.4.1	新建 Maven 项目的方式	212
13.4.2	导入已存在的项目的方式	214
13.5	storm-starter 的入门例子	214
13.5.1	ExclamationTopology	214
13.5.2	WordCountTopology	216
13.5.3	ReachTopology	219
13.6	storm-starter 的其他例子	225
13.6.1	BasicDRPCTopology	225
13.6.2	ManualDRPC	226
13.6.3	PrintSampleStream	226
13.6.4	RollingTopWords	227
13.6.5	SkewedRollingTopWords	228
13.6.6	SingleJoinExample	229
13.6.7	TransactionalGlobalCount	230
13.6.8	TransactionalWords	230
13.6.9	WordCountTopologyNode	231
13.7	本章小结	232

第 14 章 研发与集群管理技巧

14.1	使用 daemontools 监控 Storm 进程	233
14.1.1	daemontools 简介	233

14.1.2	安装 daemontools	234
14.1.3	编写监控脚本	234
14.2	使用 Monit 监控 Storm	236
14.2.1	Monit 简介	236
14.2.2	安装 Monit	237
14.2.3	配置 Monit	238
14.2.4	启动 Monit	240
14.2.5	获取 Monit 帮助信息	241
14.3	常用的集群操作命令	242
14.4	使用 Storm 的经验与建议	243
14.5	本章小结	244

第 15 章 DRPC 详解

15.1	概述	245
15.2	DRPCTopologyBuilder	246
15.2.1	LinearDRPCTopologyBuilder	246
15.2.2	LinearDRPCTopologyBuilder 提供的方法	246
15.2.3	LinearDRPCTopologyBuilder 使用范例	248
15.2.4	LinearDRPCTopologyBuilder 的工作原理	249
15.2.5	LinearDRPCTopologyBuilder 目前已弃用	249
15.3	本地模式的 DRPC	249
15.4	远程模式的 DRPC	250
15.5	一个复杂的 DRPC 例子 (计算 reach 值)	250
15.6	非线性 DRPC	253
15.7	本章小结	253

第 16 章 事务拓扑详解

16.1	什么是事务拓扑	254
16.1.1	设计 1	254
16.1.2	设计 2	255
16.1.3	设计 3 (Storm 的设计)	256
16.2	事务拓扑的设计细节	256
16.3	事务拓扑的实现细节	257
16.3.1	事务 Spout 的工作原理	257
16.3.2	“对于给定的事务 id 不能发射相同的 Batch” 的处理	258
16.3.3	更多的细节	260
16.4	事务拓扑 API	260
16.4.1	Bolt	260
16.4.2	事务 Spout	261

16.4.3	配置	262
16.5	TransactionalTopologyBuilder	262
16.5.1	TransactionalTopologyBuilder 提供的方法	262
16.5.2	TransactionalTopologyBuilder 类已弃用	266
16.6	一个简单的例子	266
16.7	本章小结	269

第 17 章 Trident 详解

17.1	Trident 概述	270
17.1.1	简单的例子——单词统计 (TridentWordCount)	270
17.1.2	另一个例子——计算 Reach 值 (TridentReach)	274
17.1.3	字段和元组	275
17.1.4	状态 (State)	276
17.1.5	Trident 拓扑的执行	277
17.2	Trident API	279
17.2.1	概述	279
17.2.2	本地分区操作	279
17.2.3	重新分区操作	283
17.2.4	聚合操作	284
17.2.5	流分组操作	284
17.2.6	合并与连接	285
17.3	Trident 的状态	285
17.3.1	Trident 状态分类	286
17.3.2	事务 Spout (Transactional Spout)	286
17.3.3	不透明事务 Spout (Opaque Transactional Spout)	288
17.3.4	非事务 Spout (Non-transactional Spout)	289
17.3.5	Spout 与 State 之间的联系	289
17.3.6	State API	290
17.3.7	persistentAggregate 方法	294
17.3.8	实现 MapStates	294
17.4	Trident Spout	295
17.4.1	流水线 (Pipelining)	296
17.4.2	Trident Spout 的类型	296
17.5	本章小结	296

第 18 章 Storm 的内部实现

18.1	文件系统分析	297
18.2	数据目录结构	298
18.2.1	Nimbus 节点的目录结构	299

18.2.2	Supervisor 节点的目录结构	299
18.3	代码库的结构	300
18.3.1	storm.thrift	301
18.3.2	Java 接口	316
18.3.3	实现	316
18.4	拓扑的生命周期	318
18.4.1	启动拓扑	319
18.4.2	监控拓扑	321
18.4.3	杀死拓扑	321
18.5	Acking 框架的实现	322
18.5.1	异或计算的基本原理	322
18.5.2	Acking 框架的实现原理	322
18.5.3	Acker 的 execute 方法	323
18.5.4	待定元组 (pending tuple) 和 RotatingMap	323
18.6	Metric	324
18.7	本章小结	329

第 19 章 Storm 相关的其他项目

19.1	JStorm 项目	330
19.1.1	项目简介	330
19.1.2	下载与部署	331
19.1.3	源代码编译	332
19.2	storm-deploy 项目	332
19.3	Storm 与 Kafka	333
19.3.1	Kafka 简介	333
19.3.2	Kafka 的安装	333
19.3.3	启动服务	334
19.3.4	测试运行	335
19.3.5	Storm 与 Kafka 的项目	337
19.4	storm-kestrel 项目	338
19.4.1	storm-kestrel 项目简介	338
19.4.2	使用 storm-kestrel 项目	338
19.4.3	Kestrel 服务器和队列	339
19.4.4	添加元素到 kestrel	339
19.4.5	从 Kestrel 中移除元素	340
19.4.6	持续添加元素到 Kestrel	341
19.4.7	使用 KestrelSpout	342
19.4.8	执行	342
19.5	本章小结	343

第 20 章 企业应用案例

20.1	Storm 席卷众多互联网企业	344
20.1.1	Storm 的典型应用场景	344
20.1.2	Storm 的三大基本应用	345
20.2	Storm 在 Twitter 中的应用	345
20.2.1	Twitter 公司简介	345
20.2.2	Storm 帮助 Twitter 提升产品性能	346
20.2.3	MapR 在 Twitter 中的应用简介	346
20.3	Storm 在阿里巴巴集团的应用	348
20.3.1	阿里巴巴集团简介	348
20.3.2	Storm 在阿里巴巴的应用	348
20.3.3	Storm 在淘宝公司的应用	350
20.3.4	Storm 在支付宝公司的应用	350
20.4	其他应用 Storm 的知名企业和项目	351
20.5	本章小结	367
参考资料		368

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, 读者将会对 Storm 有一个基本的了解。

1.1 什么是 Storm

Storm 是一个分布式计算框架, 主要使用 Clojure 与 Java 语言编写, 最初是由 Nathan Marz 带领 Backtype 公司的团队创建, 在 Backtype 公司被 Twitter 公司收购后进行开源。最初的版本是在 2011 年 9 月 17 日发行, 版本号为 0.5.0。

2013 年 9 月, Apache 基金会开始接管并孵化 Storm 项目。Apache Storm 是在 Eclipse Public License 下进行开发的, 它提供给大多数企业使用。在 Apache 孵化器项目中, 使用 Git 作为版本控制, 使用 Atlassian JIRA 作为问题跟踪。经过了 1 年多时间, 2014 年 9 月, Storm 项目成为 Apache 的顶级项目。目前, Storm 的最新版本为 Apache Storm 0.10.0。

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 对数据进行批处理; 但是 Storm 能持续地运作下去, 并且要掌握 Storm 十分简单, 开发人员可以使用任何编程语言对它进行操作, 而得到满意的结果。

Storm 能用到很多场景中, 包括实时分析、在线机器学习、连续计算、分布式 RPC、ETL 等。Storm 的处理速度非常快, 每个节点每秒钟可以处理 100 万条消息。同时, Storm 是可伸缩的、容错的, 并且能保证数据根据用户的设定被妥善处理, 便于进行设置和操作。Storm 的抽象示意图如图 1.1 所示。