

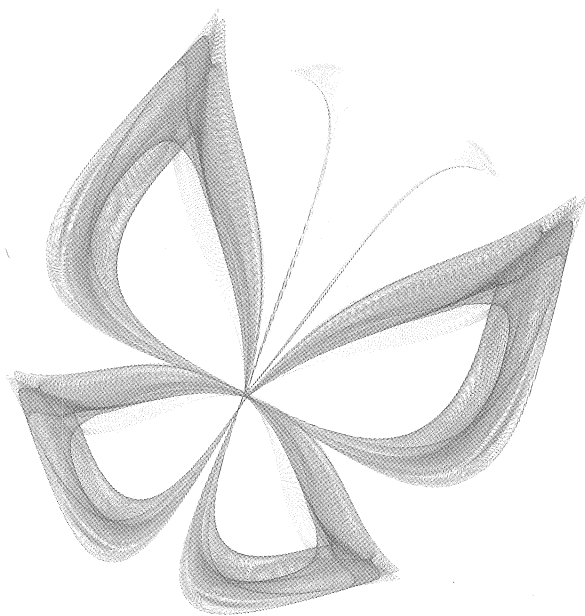


技术丛书

The Art of Spark Kernel Design

# Spark内核设计的艺术 架构设计与实现

耿嘉安◎著



机械工业出版社  
China Machine Press

## 为什么写这本书

给本书写前言时，让我想起了两年前给《深入理解 Spark：核心思想与源码分析》<sup>①</sup>一书写前言的经历。我不禁想起崔护的《题都城南庄》这首诗，诗的内容是：

去年今日此门中，人面桃花相映红。

人面不知何处去，桃花依旧笑春风。

从核心思想和架构来看，Spark 依然是那个 Spark，但是我已经找了一个新的“东家”。我的年龄不知不觉中又长了两岁，Spark 也在大数据领域从“新贵”变成了“老人”。Spark 的版本从 0.x.x 到 2.x.x 基本上也是用了两年时间。

自从《深入理解 Spark：核心思想与源码分析》一书出版后，引起了一些市场反响，更难得的是得到了很多读者的反馈。一些热心的读者通过微信或者邮件向我指出了书中内容的很多不足之处，包括错别字、错误的描述、代码分析有点像流水账、提纲挈领的内容偏少、代码版本过低等。一些错误在修订的版本中得到了解决，有些修正的内容则通过单独写博客来补充。在与读者的沟通过程中，也纠正了我对一些问题的理解偏差。再次深深地感谢广大读者的支持与帮助！

一些读者对《深入理解 Spark：核心思想与源码分析》一书的内容非常肯定，希望能够出第 2 版，高婧雅编辑也一再“怂恿”我，但是我一直没有写第 2 版的打算。我当时希望有人能够以更好的方式写一本介绍和分析 Spark 2.0 版本的源码分析书籍，因为我感觉之前的写作方式的确不是很好。在我心中一直有个矛盾：如果源码太少，源码分析的书籍将退化成单纯讲原理的书籍，对于想深入理解 Spark 实现的读者来说这是不够的；如果源码太多，又让人有堆砌代码或者“混”篇幅的感觉。很多源码分析的书只是简单说说接口或者方法的功能，让人始终有种“雾里看花”的感觉。所以我一直很期待能有更好的方式来写作源码分析类的书。

在一年多的等待中，我始终没有发现类似书籍的出现，于是我打算再做一次尝试。这

---

① 该书由机械工业出版社出版，书号 978-7-111-52234-8。

次摒弃了《深入理解 Spark：核心思想与源码分析》一书中按照代码执行流程分析的方式，改为先从整体上介绍一个系统，然后逐个分析每个组件的功能，最后将这些组件之间的关系用流程图的方式串联起来。本书的写作方式依然犯有代码过多的“毛病”，但我还是期待本书能带来一些新的气象。

## 本书的主要特色

- ❑ 按照源码分析的习惯设计，从脚本分析到初始化，再到核心内容。整个过程遵循由浅入深的基本思路。
- ❑ 每一章先对本章的内容有个总体介绍，然后深入分析各个组件的实现原理，最后将各个组件之间的关系通过执行流程来展现。
- ❑ 本书尽可能地用图来展示原理，以加速读者对内容的掌握。
- ❑ 本书讲解的很多实现及原理都值得借鉴，可以帮助读者提升架构设计、程序设计等方面的能力。
- ❑ 本书尽可能保留较多的源码，以便于初学者能够在脱离办公环境的地方（如地铁、公交等），也能轻松阅读。

## 读者对象

源码阅读是一项苦差事，人力和时间成本都很高，尤其对于刚刚接触 Spark 的人来说更是如此。本书尽可能保留源码，使得分析过程不至于产生跳跃感，目的是降低大多数人的学习门槛。如果你是从事 IT 工作 1～3 年的新人或者希望开始学习 Spark 的核心知识，本书非常适合你。如果你已经对 Spark 有所了解或者已经使用它，还想进一步提高自己，那么本书更适合你。如果你是一个开发新手，对 Java、Linux 等基础知识还不是很了解的话，本书可能不太适合你。如果你已经对 Spark 有深入的研究，本书也许可以作为你的参考资料。

总体来说，本书适合以下人群：

- ❑ 已经了解过 Spark，但还想深入理解 Spark 实现原理的人；
- ❑ 大数据技术爱好者；
- ❑ 对性能优化和部署方案感兴趣的运维工程师与架构师；
- ❑ 开源代码爱好者，喜欢研究源码的同学可以通过本书学到一些阅读源码的方式、方法。

本书不会教你如何开发 Spark 应用程序，而只拿 word count 的经典例子做演示。本书会简单介绍 Hadoop MapReduce、Hadoop YARN、Mesos、Alluxio (Tachyon)、ZooKeeper、HDFS、Akka、Jetty、Netty，但不会过多介绍这些框架的使用，因为市场上已经有丰富的书籍供读者挑选。本书也不会过多介绍 Scala、Java、Shell 的语法，读者可以在市场上选择适合自己的书籍阅读。本书将无比适合那些想要破解“潘多拉魔盒”的人！

## 如何阅读本书

本书一共有 10 章内容，主要包括以下部分。

**准备部分**（第 1~2 章）：简单介绍了 Spark 的环境搭建和基本原理，帮助读者了解一些背景知识。

**基础部分**（第 3~5 章）：介绍 Spark 的基础设施、SparkContext 的初始化、Spark 执行环境等内容。

**核心部分**（第 6~9 章）：这是 Spark 最为核心的部分，包括存储体系、调度系统、计算引擎、部署模式等。

**API 部分**（第 10 章）：这部分主要对 Spark 的新旧 API 进行对比，对新 API 进行介绍。

本书最后的附录中还包括一些内容：附录 A 介绍的是 Spark 中最常用的工具类 Utils；附录 B 是 Akka 的简介；附录 C 为 Jetty 的简介和工具类 JettyUtils 的介绍；附录 D 为 Metrics 库的简介和 Metrics 中部分 API 的介绍；附录 E 演示了 Hadoop 1.0 版本中的 word count 例子；附录 F 介绍了工具类 CommandUtils 的常用方法；附录 G 是关于 Netty 的简介和工具类 NettyUtils 的介绍；附录 H 是对 Spark 中的 RPC 工具类 RpcUtils 的介绍。

为了降低读者阅读理解 Spark 源码的门槛，本书尽可能保留源码实现。本书以 Spark 2.1.0 版本为主，有兴趣的读者也可按照本书的方式，阅读 Spark 的最新源码。

## 勘误

本书内容很多，限于笔者水平有限，书中内容难免有错误之处。如果你对本书有任何问题或者意见，都可以通过邮箱 [beliefer@163.com](mailto:beliefer@163.com) 或者博客 <http://blog.csdn.net/beliefer> 联系我，给我提交你的建议或者想法，我将怀着一颗谦卑之心与大家共同进步。

## 致谢

感谢我们生活在信息时代，让我们有机会接触互联网与大数据；感谢父母多年来在学习、工作及生活上的帮助与支持；感谢妻子在生活中的照顾和谦让。

感谢高婧雅编辑给予本书出版的大力支持与帮助。

感谢我在大数据路上的领路人——和仲；感谢热衷于技术的王欢对本书内容提出的宝贵建议；感谢对本书内容进行审阅的余尧尧和马晓波；感谢对本书内容有过帮助的读者朋友们。

耿嘉安

# 目 录 *Contents*

本书赞誉	
前言	
第 1 章 环境准备	1
1.1 运行环境准备	2
1.1.1 安装 JDK	2
1.1.2 安装 Scala	2
1.1.3 安装 Spark	3
1.2 Spark 初体验	4
1.2.1 运行 spark-shell	4
1.2.2 执行 word count	5
1.2.3 剖析 spark-shell	9
1.3 阅读环境准备	14
1.3.1 安装 SBT	15
1.3.2 安装 Git	15
1.3.3 安装 Eclipse Scala IDE 插件	15
1.4 Spark 源码编译与调试	17
1.5 小结	23
第 2 章 设计理念与基本架构	24
2.1 初识 Spark	25
2.1.1 Hadoop MRv1 的局限	25
2.1.2 Spark 的特点	26
2.1.3 Spark 使用场景	28
2.2 Spark 基础知识	29
2.3 Spark 基本设计思想	31
2.3.1 Spark 模块设计	32
2.3.2 Spark 模型设计	34
2.4 Spark 基本架构	36
2.5 小结	38
第 3 章 Spark 基础设施	39
3.1 Spark 配置	40
3.1.1 系统属性中的配置	40
3.1.2 使用 SparkConf 配置的 API	41
3.1.3 克隆 SparkConf 配置	42
3.2 Spark 内置 RPC 框架	42
3.2.1 RPC 配置 TransportConf	45
3.2.2 RPC 客户端工厂 Transport-ClientFactory	47
3.2.3 RPC 服务端 TransportServer	53
3.2.4 管道初始化	56
3.2.5 TransportChannelHandler 详解	57
3.2.6 服务端 RpcHandler 详解	63
3.2.7 服务端引导程序 Transport-ServerBootstrap	68

3.2.8 客户端 TransportClient 详解 .....	71	4.15 SparkContext 的伴生对象 .....	130
3.3 事件总线 .....	78	4.16 小结 .....	131
3.3.1 ListenerBus 的继承体系 .....	79		
3.3.2 SparkListenerBus 详解 .....	80	<b>第 5 章 Spark 执行环境</b> .....	132
3.3.3 LiveListenerBus 详解 .....	83	5.1 SparkEnv 概述 .....	133
3.4 度量系统 .....	87	5.2 安全管理器 SecurityManager .....	133
3.4.1 Source 继承体系 .....	87	5.3 RPC 环境 .....	135
3.4.2 Sink 继承体系 .....	89	5.3.1 RPC 端点 RpcEndpoint .....	136
3.5 小结 .....	92	5.3.2 RPC 端点引用 RpcEndpointRef .....	139
		5.3.3 创建传输上下文 TransportConf .....	142
<b>第 4 章 SparkContext 的初始化</b> .....	93	5.3.4 消息调度器 Dispatcher .....	142
4.1 SparkContext 概述 .....	94	5.3.5 创建传输上下文 Transport-	
4.2 创建 Spark 环境 .....	97	Context .....	154
4.3 SparkUI 的实现 .....	100	5.3.6 创建传输客户端工厂 Transport-	
4.3.1 SparkUI 概述 .....	100	ClientFactory .....	159
4.3.2 WebUI 框架体系 .....	102	5.3.7 创建 TransportServer .....	160
4.3.3 创建 SparkUI .....	107	5.3.8 客户端请求发送 .....	162
4.4 创建心跳接收器 .....	111	5.3.9 NettyRpcEnv 中的常用方法 .....	173
4.5 创建和启动调度系统 .....	112	5.4 序列化管理器 SerializerManager .....	175
4.6 初始化块管理器 BlockManager .....	114	5.5 广播管理器 BroadcastManager .....	178
4.7 启动度量系统 .....	114	5.6 map 任务输出跟踪器 .....	185
4.8 创建事件日志监听器 .....	115	5.6.1 MapOutputTracker 的实现 .....	187
4.9 创建和启动 ExecutorAllocation-		5.6.2 MapOutputTrackerMaster 的	
Manager .....	116	实现原理 .....	191
4.10 ContextCleaner 的创建与启动 .....	120	5.7 构建存储体系 .....	199
4.10.1 创建 ContextCleaner .....	120	5.8 创建度量系统 .....	201
4.10.2 启动 ContextCleaner .....	120	5.8.1 MetricsConfig 详解 .....	203
4.11 额外的 SparkListener 与启动事件		5.8.2 MetricsSystem 中的常用方法 .....	207
总线 .....	122	5.8.3 启动 MetricsSystem .....	209
4.12 Spark 环境更新 .....	123	5.9 输出提交协调器 .....	211
4.13 SparkContext 初始化的收尾 .....	127	5.9.1 OutputCommitCoordinator-	
4.14 SparkContext 提供的常用方法 .....	128	Endpoint 的实现 .....	211

5.9.2	OutputCommitCoordinator 的实现 .....	212	6.8.1	BlockManagerMaster 的职责 .....	285
5.9.3	OutputCommitCoordinator 的工作原理 .....	216	6.8.2	BlockManagerMasterEndpoint 详解 .....	286
5.10	创建 SparkEnv .....	217	6.8.3	BlockManagerSlaveEndpoint 详解 .....	289
5.11	小结 .....	217	6.9	Block 传输服务 .....	290
<b>第 6 章</b>	<b>存储体系 .....</b>	<b>219</b>	6.9.1	初始化 NettyBlockTransfer-Service .....	291
6.1	存储体系概述 .....	220	6.9.2	NettyBlockRpcServer 详解 .....	292
6.1.1	存储体系架构 .....	220	6.9.3	Shuffle 客户端 .....	296
6.1.2	基本概念 .....	222	6.10	DiskBlockObjectWriter 详解 .....	305
6.2	Block 信息管理器 .....	227	6.11	小结 .....	308
6.2.1	Block 锁的基本概念 .....	227	<b>第 7 章</b>	<b>调度系统 .....</b>	<b>309</b>
6.2.2	Block 锁的实现 .....	229	7.1	调度系统概述 .....	310
6.3	磁盘 Block 管理器 .....	234	7.2	RDD 详解 .....	312
6.3.1	本地目录结构 .....	234	7.2.1	为什么需要 RDD .....	312
6.3.2	DiskBlockManager 提供的方法 .....	236	7.2.2	RDD 实现的初次分析 .....	313
6.4	磁盘存储 DiskStore .....	239	7.2.3	RDD 依赖 .....	316
6.5	内存管理器 .....	242	7.2.4	分区计算器 Partitioner .....	318
6.5.1	内存池模型 .....	243	7.2.5	RDDInfo .....	320
6.5.2	StorageMemoryPool 详解 .....	244	7.3	Stage 详解 .....	321
6.5.3	MemoryManager 模型 .....	247	7.3.1	ResultStage 的实现 .....	322
6.5.4	UnifiedMemoryManager 详解 .....	250	7.3.2	ShuffleMapStage 的实现 .....	323
6.6	内存存储 MemoryStore .....	252	7.3.3	StageInfo .....	324
6.6.1	MemoryStore 的内存模型 .....	253	7.4	面向 DAG 的调度器 DAGScheduler .....	326
6.6.2	MemoryStore 提供的方法 .....	255	7.4.1	JobListener 与 JobWaiter .....	326
6.7	块管理器 BlockManager .....	265	7.4.2	ActiveJob 详解 .....	328
6.7.1	BlockManager 的初始化 .....	265	7.4.3	DAGSchedulerEventProcessLoop 的简要介绍 .....	328
6.7.2	BlockManager 提供的方法 .....	266	7.4.4	DAGScheduler 的组成 .....	329
6.8	BlockManagerMaster 对 Block-Manager 的管理 .....	285	7.4.5	DAGScheduler 提供的常用方法 .....	330

7.4.6 DAGScheduler 与 Job 的提交	334	分配	402
7.4.7 构建 Stage	337	7.10.6 TaskSchedulerImpl 的调度	
7.4.8 提交 ResultStage	341	流程	405
7.4.9 提交还未计算的 Task	343	7.10.7 TaskSchedulerImpl 对执行	
7.4.10 DAGScheduler 的调度流程	347	结果的处理	406
7.4.11 Task 执行结果的处理	348	7.10.8 TaskSchedulerImpl 的常用方法	409
7.5 调度池 Pool	351	7.11 小结	412
7.5.1 调度算法	352	第 8 章 计算引擎	413
7.5.2 Pool 的实现	354	8.1 计算引擎概述	414
7.5.3 调度池构建器	357	8.2 内存管理器与执行内存	417
7.6 任务集合管理器 TaskSetManager	363	8.2.1 ExecutionMemoryPool 详解	417
7.6.1 Task 集合	363	8.2.2 MemoryManager 模型与执行	
7.6.2 TaskSetManager 的成员属性	364	内存	420
7.6.3 调度池与推断执行	366	8.2.3 UnifiedMemoryManager 与执行	
7.6.4 Task 本地性	370	内存	421
7.6.5 TaskSetManager 的常用方法	373	8.3 内存管理器与 Tungsten	423
7.7 运行器后端接口 LauncherBackend	383	8.3.1 MemoryBlock 详解	423
7.7.1 BackendConnection 的实现	384	8.3.2 MemoryManager 模型与	
7.7.2 LauncherBackend 的实现	386	Tungsten	425
7.8 调度后端接口 SchedulerBackend	389	8.3.3 Tungsten 的内存分配器	425
7.8.1 SchedulerBackend 的定义	389	8.4 任务内存管理器	431
7.8.2 LocalSchedulerBackend 的实现		8.4.1 TaskMemoryManager 详解	431
分析	390	8.4.2 内存消费者	439
7.9 任务结果获取器 TaskResultGetter	394	8.4.3 执行内存整体架构	441
7.9.1 处理成功的 Task	394	8.5 Task 详解	443
7.9.2 处理失败的 Task	396	8.5.1 任务上下文 TaskContext	443
7.10 任务调度器 TaskScheduler	397	8.5.2 Task 的定义	446
7.10.1 TaskSchedulerImpl 的属性	397	8.5.3 ShuffleMapTask 的实现	449
7.10.2 TaskSchedulerImpl 的初始化	399	8.5.4 ResultTask 的实现	450
7.10.3 TaskSchedulerImpl 的启动	399	8.6 IndexShuffleBlockResolver 详解	451
7.10.4 TaskSchedulerImpl 与 Task 的		8.7 采样与估算	455
提交	400	8.7.1 SizeTracker 的实现分析	455
7.10.5 TaskSchedulerImpl 与资源			



8.7.2	SizeTracker 的工作原理 .....	457	9.2.2	运行 Task .....	530
8.8	特质 WritablePartitionedPair-Collection .....	458	9.3	local 部署模式 .....	535
8.9	AppendOnlyMap 的实现分析 .....	460	9.4	持久化引擎 PersistenceEngine .....	537
8.9.1	AppendOnlyMap 的容量增长 .....	461	9.4.1	基于文件系统的持久化引擎 .....	539
8.9.2	AppendOnlyMap 的数据更新 .....	462	9.4.2	基于 ZooKeeper 的持久化引擎 .....	541
8.9.3	AppendOnlyMap 的缓存聚合算法 .....	464	9.5	领导选举代理 .....	542
8.9.4	AppendOnlyMap 的内置排序 .....	466	9.6	Master 详解 .....	546
8.9.5	AppendOnlyMap 的扩展 .....	467	9.6.1	启动 Master .....	549
8.10	PartitionedPairBuffer 的实现分析 .....	469	9.6.2	检查 Worker 超时 .....	553
8.10.1	PartitionedPairBuffer 的容量增长 .....	469	9.6.3	被选举为领导时的处理 .....	554
8.10.2	PartitionedPairBuffer 的插入 .....	470	9.6.4	一级资源调度 .....	558
8.10.3	PartitionedPairBuffer 的迭代器 .....	471	9.6.5	注册 Worker .....	568
8.11	外部排序器 .....	472	9.6.6	更新 Worker 的最新状态 .....	570
8.11.1	ExternalSorter 详解 .....	473	9.6.7	处理 Worker 的心跳 .....	570
8.11.2	ShuffleExternalSorter 详解 .....	487	9.6.8	注册 Application .....	571
8.12	Shuffle 管理器 .....	490	9.6.9	处理 Executor 的申请 .....	573
8.12.1	ShuffleWriter 详解 .....	491	9.6.10	处理 Executor 的状态变化 .....	573
8.12.2	ShuffleBlockFetcherIterator 详解 .....	502	9.6.11	Master 的常用方法 .....	574
8.12.3	BlockStoreShuffleReader 详解 .....	510	9.7	Worker 详解 .....	578
8.12.4	SortShuffleManager 详解 .....	513	9.7.1	启动 Worker .....	581
8.13	map 端与 reduce 端的 Shuffle 组合 .....	516	9.7.2	向 Master 注册 Worker .....	584
8.14	小结 .....	519	9.7.3	向 Master 发送心跳 .....	589
第 9 章	部署模式 .....	520	9.7.4	Worker 与领导选举 .....	591
9.1	心跳接收器 HeartbeatReceiver .....	521	9.7.5	运行 Driver .....	593
9.2	Executor 的实现分析 .....	527	9.7.6	运行 Executor .....	594
9.2.1	Executor 的心跳报告 .....	528	9.7.7	处理 Executor 的状态变化 .....	599
			9.8	StandaloneAppClient 实现 .....	600
			9.8.1	ClientEndpoint 的实现分析 .....	601
			9.8.2	StandaloneAppClient 的实现分析 .....	606
			9.9	StandaloneSchedulerBackend 的实现分析 .....	607

9.9.1 StandaloneSchedulerBackend 的属性 .....	607	9.13 其他部署方案 .....	639
9.9.2 DriverEndpoint 的实现分析 .....	609	9.13.1 YARN .....	639
9.9.3 StandaloneSchedulerBackend 的启动 .....	614	9.13.2 Mesos .....	644
9.9.4 StandaloneSchedulerBackend 的停止 .....	617	9.14 小结 .....	646
9.9.5 StandaloneSchedulerBackend 与资源分配 .....	618	<b>第 10 章 Spark API</b> .....	647
<b>9.10 CoarseGrainedExecutorBackend 详解</b> .....	619	10.1 基本概念 .....	648
9.10.1 CoarseGrainedExecutorBackend 进程 .....	620	10.2 数据源 DataSource .....	650
9.10.2 CoarseGrainedExecutorBackend 的功能分析 .....	622	10.2.1 DataSourceRegister 详解 .....	650
<b>9.11 local-cluster 部署模式</b> .....	625	10.2.2 DataSource 详解 .....	651
9.11.1 启动本地集群 .....	625	10.3 检查点的实现 .....	655
9.11.2 local-cluster 部署模式的启动过程 .....	627	10.3.1 CheckpointRDD 的实现 .....	655
9.11.3 local-cluster 部署模式下 Executor 的分配过程 .....	628	10.3.2 RDDCheckpointData 的实现 .....	660
9.11.4 local-cluster 部署模式下的任务提交执行过程 .....	629	10.3.3 ReliableRDDCheckpointData 的实现 .....	662
<b>9.12 Standalone 部署模式</b> .....	631	10.4 RDD 的再次分析 .....	663
9.12.1 Standalone 部署模式的启动过程 .....	632	10.4.1 转换 API .....	663
9.12.2 Standalone 部署模式下 Executor 的分配过程 .....	634	10.4.2 动作 API .....	665
9.12.3 Standalone 部署模式的资源回收 .....	635	10.4.3 检查点 API 的实现分析 .....	667
9.12.4 Standalone 部署模式的容错机制 .....	636	10.4.4 迭代计算 .....	669
		10.5 数据集合 Dataset .....	671
		10.6 DataFrameReader 详解 .....	673
		10.7 SparkSession 详解 .....	676
		10.7.1 SparkSession 的构建器 Builder .....	676
		10.7.2 SparkSession 的 API .....	679
		10.8 word count 例子 .....	679
		10.8.1 Job 准备阶段 .....	680
		10.8.2 Job 的提交与调度 .....	685
		10.9 小结 .....	689
		<b>附录</b> .....	690