Life is short, you need Spark!

从零开始

不需要任何基础，带领您无痛入门 Spark

云计算分布式大数据 Spark 实战高手之路

王家林著

Spark 亚太研究院系列丛书

伴随着大数据相关技术和产业的逐步成熟，继 Hadoop 之后，Spark 技术以其无可比拟的优势，发展迅速，将成为替代 Hadoop 的下一代云计算、大数据核心技术。

本书特点

- 云计算分布式大数据 Spark 实战高手之路三部曲之第一部
- 网络发布版为图文并茂方式，边学习，边演练
- 不需要任何前置知识，从零开始，循序渐进

本书作者

Spark 亚太研究院院长和首席专家，中国目前唯一的移动互联网和云计算大数据集大成者。在 Spark、Hadoop、Android 等方面有丰富的源码、实务和性能优化经验。彻底研究了 Spark 从 0.5.0 到 0.9.1 共 13 个版本的 Spark 源码，并已完成 2014 年 5 月 31 日发布的 Spark1.0 源码研究。

Hadoop 源码级专家，曾负责某知名公司的类 Hadoop 框架开发工作，专注于 Hadoop 一站式解决方案的提供，同时也是云计算分布式大数据处理的最早实践者之一。

Android 架构师、高级工程师、咨询顾问、培训专家。

通晓 Spark、Hadoop、Android、HTML5，迷恋英语播音和健美。

“真相会使你获得自由。”

—— 耶稣《圣经》约翰 8:32KJV

“所有人类的不幸都来源于不能直面事实。”

—— 释迦摩尼

“道法自然”

—— 老子《道德经》第 25 章

《云计算分布式大数据 Spark 实战高手之路》

系列丛书三部曲

《云计算分布式大数据 **Spark** 实战高手之路---从零开始》：

不需要任何基础，带领您无痛入门 Spark 并能够轻松处理 Spark 工程师的日常编程工作，内容包括 Spark 集群的构建、Spark 架构设计、RDD、Shark/SparkSQL、机器学习、图计算、实时流处理、Spark on Yarn、JobServer、Spark 测试、Spark 优化等。

《云计算分布式大数据 **Spark** 实战高手之路---高手崛起》：

大话 Spark 源码，全世界最有情趣的源码解析，过程中伴随诸多实验，解析 Spark 1.0 的任何一句源码！更重要的是，思考源码背后的问题场景和解决问题的设计哲学和实现招式。

《云计算分布式大数据 **Spark** 实战高手之路---高手之巅》：

通过当今主流的 Spark 商业使用方法和最成功的 Hadoop 大型案例让您直达高手之巅，从此一览众山小。

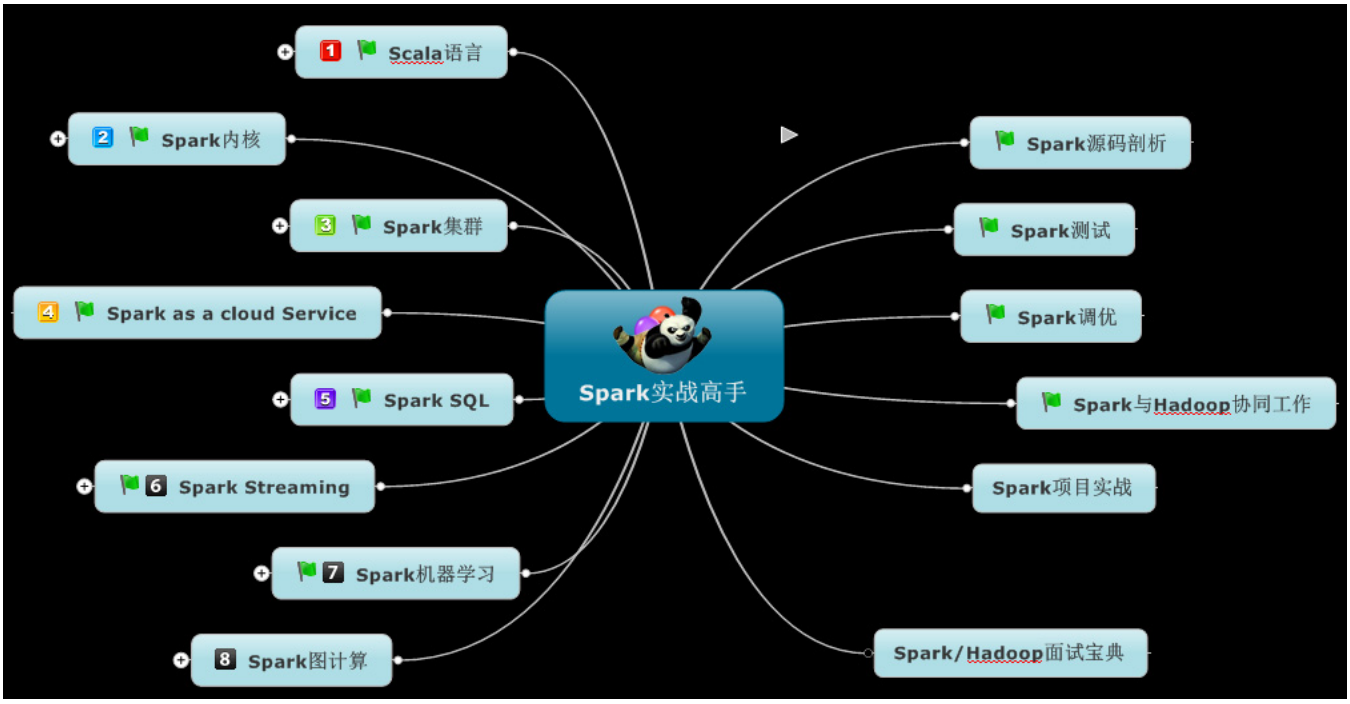
www.sparkinchina.com
TEL: 4006-998-758

《前言》

Spark采用一个统一的技术堆栈解决了云计算大数据的如流处理、图技术、机器学习、NoSQL查询等方面的所有核心问题，具有完善的生态系统，这直接奠定了其一统云计算大数据领域的霸主地位；

要想成为Spark高手，需要经历六大阶段

Spark 实战高手之核心技能点



第一阶段：熟练的掌握Scala语言

1. Spark 框架是采用 Scala 语言编写的，精致而优雅。要想成为 Spark 高手，你就必须阅读 Spark 的源代码，就必须掌握 Scala,；
2. 虽然说现在的 Spark 可以采用多语言 Java、Python 等进行应用程序开发，但是最快速的和支持最好的开发 API 依然并将永远是 Scala 方式的 API，所以你必须掌握 Scala 来编写复杂的和高性能的 Spark 分布式程序；
3. 尤其要熟练掌握 Scala 的 trait、apply、函数式编程、泛型、逆变与协变等；

推荐课程：" 精通Spark的开发语言：Scala最佳实践"

第二阶段：精通Spark平台本身提供给开发者API

1. 掌握Spark中面向RDD的开发模式 掌握各种transformation和action函数的使用；
2. 掌握 Spark 中的宽依赖和窄依赖以及 lineage 机制；
3. 掌握 RDD 的计算流程，例如 Stage 的划分、Spark 应用程序提交给集群的基本过程和 Worker 节点基础的工作原理等

推荐课程： "18 小时内掌握Spark：把云计算大数据速度提高 100 倍以上!"

第三阶段：深入Spark内核

此阶段主要是通过 Spark 框架的源码研读来深入 Spark 内核部分：

1. 通过源码掌握 Spark 的任务提交过程；
2. 通过源码掌握 Spark 集群的任务调度；
3. 尤其要精通 DAGScheduler、TaskScheduler 和 Worker 节点内部的工作的每一步的细节；

推荐课程： “Spark 1.0.0 企业级开发动手：实战世界上第一个Spark 1.0.0 课程，涵盖Spark 1.0.0 所有的企业级开发技术”

第四阶级:掌握基于Spark上的核心框架的使用

Spark 作为云计算大数据时代的集大成者，在实时流处理、图技术、机器学习、NoSQL 查询等方面具有显著的优势，我们使用 Spark 的时候大部分时间都是在使用其上的框架例如 Shark、Spark Streaming 等：

1. Spark Streaming 是非常出色的实时流处理框架 ,要掌握其 DStream、transformation 和 checkpoint 等；
2. Spark 的离线统计分析功能，Spark 1.0.0 版本在 Shark 的基础上推出了 Spark SQL，离线统计分析的功能的效率有显著的提升，需要重点掌握；
3. 对于 Spark 的机器学习和 GraphX 等要掌握其原理和用法；

推荐课程： “Spark企业级开发最佳实践”

第五阶级:做商业级别的Spark项目

通过一个完整的具有代表性的 Spark 项目来贯穿 Spark 的方方面面 ,包括项目的架构设计、用到的技术的剖析、开发实现、运维等，完整掌握其中的每一个阶段和细节，这样就可以让您以后可以从容面对绝大多数 Spark 项目。

推荐课程： “Spark架构案例鉴赏：Conviva、Yahoo！、优酷土豆、网易、腾讯、淘宝等公司的实际Spark案例”

第六阶级：提供Spark解决方案

1. 彻底掌握 Spark 框架源码的每一个细节；
2. 根据不同的业务场景的需要提供 Spark 在不同场景的下的解决方案；
3. 根据实际需要，在 Spark 框架基础上进行二次开发，打造自己的 Spark 框架；

推荐课程： “精通Spark：Spark内核剖析、源码解读、性能优化和商业案例实战”

《Spark 书籍第 5 章：Spark API 编程动手实战》

对 Spark 平台开发 API 的掌握是每个大数据 Spark 开发者必须掌握的内容。

Spark 平台基于 **RDD** 给开发者提供了众多友好易用的 **API**，本章将循序渐进的带大家动手实战所有的 **Spark** 核心 **API**，学习完本章即可直接进行 **Spark** 程序的开发！

Spark API 编程动手实战共分四个部分：

- 第一部分：Spark API 动手实战体验
- 第二部分：Spark API 深入实战（一）
- 第三部分：Spark API 深入实战（二）
- 第四部分：Spark API 深入实战（三）

本讲是 **Spark API** 编程动手实战的第二部分：**Spark API** 深入实战（一），具体内容如下所示：

- 1，动手实战 union、groupByKey，join；
- 2，动手实战 reduce、lookup；

不需任何前置知识，从零开始，循序渐进，成为 Spark 高手！

目录

1. 动手实战 union、groupByKey、join

下面看一下 union 的使用：

```
scala> val rdd1 = sc.parallelize(List(('a',1),('b',1)))
rdd1: org.apache.spark.rdd.RDD[(Char, Int)] = ParallelCollectionRDD[20] at parallelize at <console>:12

scala> val rdd2 = sc.parallelize(List(('c',1),('d',1)))
rdd2: org.apache.spark.rdd.RDD[(Char, Int)] = ParallelCollectionRDD[21] at parallelize at <console>:12

scala> rdd1 union rdd2
res12: org.apache.spark.rdd.RDD[(Char, Int)] = UnionRDD[22] at union at <console>:17

scala> val result = rdd1 union rdd2
result: org.apache.spark.rdd.RDD[(Char, Int)] = UnionRDD[23] at union at <console>:16
```

使用 collect 操作查看一下执行结果：

```
scala> result.collect
14/09/29 16:56:33 INFO spark.SparkContext: Starting job: collect at <console>:19
14/09/29 16:56:33 INFO scheduler.DAGScheduler: Got job 14 (collect at <console>:19) with 4 output part
14/09/29 16:56:33 INFO scheduler.DAGScheduler: Final stage: Stage 21(collect at <console>:19)
14/09/29 16:56:33 INFO scheduler.DAGScheduler: Parents of final stage: List()
14/09/29 16:56:34 INFO scheduler.DAGScheduler: Missing parents: List()
14/09/29 16:56:34 INFO scheduler.DAGScheduler: Submitting Stage 21 (UnionRDD[23] at union at <console>
14/09/29 16:56:34 INFO scheduler.DAGScheduler: Submitting 4 missing tasks from Stage 21 (UnionRDD[23]
14/09/29 16:56:34 INFO scheduler.TaskSchedulerImpl: Adding task set 21.0 with 4 tasks
14/09/29 16:56:34 INFO scheduler.TaskSetManager: Starting task 21.0:0 as TID 34 on executor 0: SparkWo
14/09/29 16:56:34 INFO scheduler.TaskSetManager: Serialized task 21.0:0 as 2361 bytes in 1 ms
14/09/29 16:56:34 INFO scheduler.TaskSetManager: Starting task 21.0:1 as TID 35 on executor 1: SparkWo
14/09/29 16:56:34 INFO scheduler.TaskSetManager: Serialized task 21.0:1 as 2361 bytes in 0 ms
14/09/29 16:56:35 INFO scheduler.TaskSetManager: Starting task 21.0:2 as TID 36 on executor 0: SparkWo
14/09/29 16:56:35 INFO scheduler.TaskSetManager: Serialized task 21.0:2 as 2361 bytes in 13 ms
14/09/29 16:56:35 INFO scheduler.TaskSetManager: Starting task 21.0:3 as TID 37 on executor 1: SparkWo
14/09/29 16:56:35 INFO scheduler.TaskSetManager: Serialized task 21.0:3 as 2361 bytes in 0 ms
14/09/29 16:56:35 INFO scheduler.DAGScheduler: Completed ResultTask(21, 1)
14/09/29 16:56:35 INFO scheduler.TaskSetManager: Finished TID 35 in 698 ms on SparkWorker1 (progress:
14/09/29 16:56:35 INFO scheduler.TaskSetManager: Finished TID 34 in 722 ms on SparkWorker2 (progress:
14/09/29 16:56:35 INFO scheduler.DAGScheduler: Completed ResultTask(21, 0)
14/09/29 16:56:35 INFO scheduler.TaskSetManager: Finished TID 37 in 118 ms on SparkWorker1 (progress:
14/09/29 16:56:35 INFO scheduler.DAGScheduler: Completed ResultTask(21, 3)
14/09/29 16:56:35 INFO scheduler.DAGScheduler: Completed ResultTask(21, 2)
14/09/29 16:56:35 INFO scheduler.DAGScheduler: Stage 21 (collect at <console>:19) finished in 0.978 s
14/09/29 16:56:35 INFO spark.SparkContext: Job finished: collect at <console>:19, took 1.914495523 s
res13: Array[(Char, Int)] = Array((a,1), (b,1), (c,1), (d,1))
```

下面看一下 groupByKey 的使用的：

```
scala> val wordcount = rdd.flatMap(_.split(' ')).map((_, 1)).groupByKey
wordcount: org.apache.spark.rdd.RDD[(String, Iterable[Int])] = MappedValuesRDD[28] at groupByKey at <console>:14
```

使用 collect 查看一下结果：

```
scala> wordcount.collect
14/09/29 17:00:33 INFO spark.SparkContext: Starting job: collect at <console>:17
14/09/29 17:00:33 INFO scheduler.DAGScheduler: Registering RDD 25 (map at <console>:14)
14/09/29 17:00:33 INFO scheduler.DAGScheduler: Got job 15 (collect at <console>:17) with 2 output partitions (allowLocal=fal
14/09/29 17:00:33 INFO scheduler.DAGScheduler: Final stage: Stage 22(collect at <console>:17)
14/09/29 17:00:33 INFO scheduler.DAGScheduler: Parents of final stage: List(Stage 23)
14/09/29 17:00:33 INFO scheduler.DAGScheduler: Missing parents: List(Stage 23)
14/09/29 17:00:33 INFO scheduler.DAGScheduler: Submitting Stage 23 (MappedRDD[25] at map at <console>:14), which has no miss
14/09/29 17:00:34 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from Stage 23 (MappedRDD[25] at map at <console>:1
14/09/29 17:00:34 INFO scheduler.TaskSchedulerImpl: Adding task set 23.0 with 2 tasks
14/09/29 17:00:34 INFO scheduler.TaskSetManager: Starting task 23.0:0 as TID 38 on executor 0: SparkWorker2 (PROCESS_LOCAL)
14/09/29 17:00:34 INFO scheduler.TaskSetManager: Serialized task 23.0:0 as 1883 bytes in 0 ms
14/09/29 17:00:34 INFO scheduler.TaskSetManager: Starting task 23.0:1 as TID 39 on executor 1: SparkWorker1 (PROCESS_LOCAL)
14/09/29 17:00:34 INFO scheduler.TaskSetManager: Serialized task 23.0:1 as 1883 bytes in 0 ms
14/09/29 17:00:34 INFO scheduler.DAGScheduler: Completed ShuffleMapTask(23, 0)
14/09/29 17:00:34 INFO scheduler.TaskSetManager: Finished TID 38 in 443 ms on SparkWorker2 (progress: 1/2)
14/09/29 17:00:34 INFO scheduler.TaskSetManager: Finished TID 39 in 487 ms on SparkWorker1 (progress: 2/2)
14/09/29 17:00:34 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 23.0, whose tasks have all completed, from pool
14/09/29 17:00:34 INFO scheduler.DAGScheduler: Completed ShuffleMapTask(23, 1)
14/09/29 17:00:34 INFO scheduler.DAGScheduler: Stage 23 (map at <console>:14) finished in 0.490 s
14/09/29 17:00:34 INFO scheduler.DAGScheduler: looking for newly runnable stages
14/09/29 17:00:34 INFO scheduler.DAGScheduler: running: Set()
14/09/29 17:00:34 INFO scheduler.DAGScheduler: waiting: Set(Stage 22)
14/09/29 17:00:34 INFO scheduler.DAGScheduler: failed: Set()
14/09/29 17:00:34 INFO scheduler.DAGScheduler: Missing parents for Stage 22: List()
14/09/29 17:00:34 INFO scheduler.DAGScheduler: Submitting Stage 22 (MappedValuesRDD[28] at groupByKey at <console>:14), whic
14/09/29 17:00:34 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from Stage 22 (MappedValuesRDD[28] at groupByKey a
14/09/29 17:00:34 INFO scheduler.TaskSchedulerImpl: Adding task set 22.0 with 2 tasks
14/09/29 17:00:34 INFO scheduler.TaskSetManager: Starting task 22.0:0 as TID 40 on executor 1: SparkWorker1 (PROCESS_LOCAL)
```

```
14/09/29 17:00:35 INFO spark.SparkContext: Job finished: collect at <console>:17, took 1.949010192 s
14/09/29 17:00:35 INFO scheduler.TaskSetManager: Finished TID 41 in 1186 ms on SparkWorker2 (progress: 2/2)
14/09/29 17:00:35 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 22.0, whose tasks have all completed, from pool
res14: Array[(String, Iterable[Int])] = Array((means,ArrayBuffer(1)), (under,ArrayBuffer(1, 1)), (this,ArrayBuffer(1, 1, 1, 1)), (Because,Arra
Buffer(1)), (agree,ArrayBuffer(1)), (Python,ArrayBuffer(1, 1)), (cluster.,ArrayBuffer(1)), (its,ArrayBuffer(1)), (YARN,,ArrayBuffer(1, 1, 1)),
(have,ArrayBuffer(1, 1)), (MRv1,,ArrayBuffer(1)), (pre-built,ArrayBuffer(1)), (locally.,ArrayBuffer(1)), (locally,ArrayBuffer(1, 1)), (changed
ArrayBuffer(1)), (MapReduce,ArrayBuffer(1, 1)), (only,ArrayBuffer(1)), (Configuration,ArrayBuffer(1)), (requests,ArrayBuffer(1)), (basic,Array
uffer(1)), (sc.parallelize(1,ArrayBuffer(1)), (documentation,ArrayBuffer(1)), (first,ArrayBuffer(1)), (This,ArrayBuffer(1, 1)), (several,Array
uffer(1)), (without,ArrayBuffer(1)), ("yarn-client",ArrayBuffer(1)), ([params]`.,A...
scala>
```

join 操作就是一个笛卡尔积操作的过程，具体示例如下所示：

```
scala> val rdd1 = sc.parallelize(List(('a',1),('a',2),('b',3),('b',4)))
rdd1: org.apache.spark.rdd.RDD[(Char, Int)] = ParallelCollectionRDD[34] at parallelize at <console>:12

scala> val rdd2 = sc.parallelize(List(('a',5),('a',6),('b',7),('b',8)))
rdd2: org.apache.spark.rdd.RDD[(Char, Int)] = ParallelCollectionRDD[35] at parallelize at <console>:12
```

对 rdd1 和 rdd2 执行 join 操作：

```
scala> val result = rdd1 join rdd2
result: org.apache.spark.rdd.RDD[(Char, (Int, Int))] = FlatMappedValuesRDD[38] at join at <console>:16
```

使用 collect 查看执行结果：

```
14/09/29 17:16:20 INFO spark.MapOutputTrackerMasterActor: Asked to send map output locations for shuffle 7 to spark@SparkWorker1
14/09/29 17:16:20 INFO spark.MapOutputTrackerMaster: Size of output statuses for shuffle 7 is 150 bytes
14/09/29 17:16:20 INFO spark.MapOutputTrackerMasterActor: Asked to send map output locations for shuffle 7 to spark@SparkWorker2
14/09/29 17:16:20 INFO scheduler.DAGScheduler: Completed ResultTask(27, 1)
14/09/29 17:16:20 INFO scheduler.TaskSetManager: Finished TID 53 in 132 ms on SparkWorker1 (progress: 1/2)
14/09/29 17:16:20 INFO scheduler.TaskSetManager: Finished TID 52 in 149 ms on SparkWorker2 (progress: 2/2)
14/09/29 17:16:20 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 27.0, whose tasks have all completed, from pool
14/09/29 17:16:20 INFO scheduler.DAGScheduler: Completed ResultTask(27, 0)
14/09/29 17:16:20 INFO scheduler.DAGScheduler: Stage 27 (collect at <console>:19) finished in 0.147 s
14/09/29 17:16:20 INFO spark.SparkContext: Job finished: collect at <console>:19, took 0.446298932 s
res16: Array[(Char, (Int, Int))] = Array((b,(3,7)), (b,(3,8)), (b,(4,7)), (b,(4,8)), (a,(1,5)), (a,(1,6)), (a,(2,5)), (a,(2,6)))
```

可以看出 join 操作完全是一个笛卡尔积的操作；

2. 动手实战 reduce、lookup

reduce 本身在 RDD 操作中属于一个 action 类型的操作，会导致 Job 的提交和执行：

```
scala> val rdd = sc.parallelize(List(1,2,3,4,5))
rdd: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[39] at parallelize at <console>:12

scala> rdd.reduce(_+_)
14/09/29 17:25:03 INFO spark.SparkContext: Starting job: reduce at <console>:15
14/09/29 17:25:03 INFO scheduler.DAGScheduler: Got job 18 (reduce at <console>:15) with 2 output part
14/09/29 17:25:03 INFO scheduler.DAGScheduler: Final stage: Stage 30(reduce at <console>:15)
14/09/29 17:25:03 INFO scheduler.DAGScheduler: Parents of final stage: List()
14/09/29 17:25:03 INFO scheduler.DAGScheduler: Missing parents: List()
14/09/29 17:25:03 INFO scheduler.DAGScheduler: Submitting Stage 30 (ParallelCollectionRDD[39] at para
sing parents
14/09/29 17:25:03 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from Stage 30 (ParallelColl
12)
14/09/29 17:25:03 INFO scheduler.TaskSchedulerImpl: Adding task set 30.0 with 2 tasks
14/09/29 17:25:03 INFO scheduler.TaskSetManager: Starting task 30.0:0 as TID 54 on executor 0: SparkW
14/09/29 17:25:03 INFO scheduler.TaskSetManager: Serialized task 30.0:0 as 1055 bytes in 0 ms
14/09/29 17:25:03 INFO scheduler.TaskSetManager: Starting task 30.0:1 as TID 55 on executor 1: SparkW
14/09/29 17:25:03 INFO scheduler.TaskSetManager: Serialized task 30.0:1 as 1059 bytes in 0 ms
14/09/29 17:25:03 INFO scheduler.TaskSetManager: Finished TID 55 in 232 ms on SparkWorker1 (progress:
14/09/29 17:25:03 INFO scheduler.DAGScheduler: Completed ResultTask(30, 1)
14/09/29 17:25:03 INFO scheduler.DAGScheduler: Completed ResultTask(30, 0)
14/09/29 17:25:03 INFO scheduler.DAGScheduler: Stage 30 (reduce at <console>:15) finished in 0.235 s
14/09/29 17:25:03 INFO scheduler.TaskSetManager: Finished TID 54 in 245 ms on SparkWorker2 (progress:
14/09/29 17:25:03 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 30.0, whose tasks have all comple
14/09/29 17:25:03 INFO spark.SparkContext: Job finished: reduce at <console>:15, took 0.315574366 s
res17: Int = 15

scala>
```

下面看一下 lookup 的使用：

```
scala> val rdd2 = sc.parallelize(List(('a',5),('a',6),('b',7),('b',8)))
rdd2: org.apache.spark.rdd.RDD[(Char, Int)] = ParallelCollectionRDD[40] at parallelize

scala> rdd2.lookup('a')
14/09/29 17:27:55 INFO spark.SparkContext: Starting job: lookup at <console>:15
14/09/29 17:27:55 INFO scheduler.DAGScheduler: Got job 19 (lookup at <console>:15) with
14/09/29 17:27:55 INFO scheduler.DAGScheduler: Final stage: Stage 31(lookup at <console
14/09/29 17:27:55 INFO scheduler.DAGScheduler: Parents of final stage: List()
14/09/29 17:27:55 INFO scheduler.DAGScheduler: Missing parents: List()
14/09/29 17:27:55 INFO scheduler.DAGScheduler: Submitting Stage 31 (MappedRDD[42] at lo
14/09/29 17:27:55 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from Stage 31
14/09/29 17:27:55 INFO scheduler.TaskSchedulerImpl: Adding task set 31.0 with 2 tasks
14/09/29 17:27:55 INFO scheduler.TaskSetManager: Starting task 31.0:0 as TID 56 on exec
14/09/29 17:27:55 INFO scheduler.TaskSetManager: Serialized task 31.0:0 as 1676 bytes i
14/09/29 17:27:55 INFO scheduler.TaskSetManager: Starting task 31.0:1 as TID 57 on exec
14/09/29 17:27:55 INFO scheduler.TaskSetManager: Serialized task 31.0:1 as 1676 bytes i
14/09/29 17:27:56 INFO scheduler.TaskSetManager: Finished TID 56 in 72 ms on SparkWorke
14/09/29 17:27:56 INFO scheduler.DAGScheduler: Completed ResultTask(31, 0)
14/09/29 17:27:56 INFO scheduler.TaskSetManager: Finished TID 57 in 216 ms on SparkWork
14/09/29 17:27:56 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 31.0, whose tasks h
14/09/29 17:27:56 INFO scheduler.DAGScheduler: Completed ResultTask(31, 1)
14/09/29 17:27:56 INFO scheduler.DAGScheduler: Stage 31 (lookup at <console>:15) finish
14/09/29 17:27:56 INFO spark.SparkContext: Job finished: lookup at <console>:15, took 0
res18: Seq[Int] = WrappedArray(5, 6)

scala>
```

Spark 亚太研究院

Spark 亚太研究院，提供 Spark、Hadoop、Android、Html5、云计算和移动互联网一站式解决方案。以帮助企业规划、部署、开发、培训和使用为核心，并规划和实施人才培训完整路径，提供源码研究和应用技术训练。

近期活动及相关课程

1、决战云计算大数据时代 Spark 亚太研究院 100 期公益大讲堂

每周四晚上 20：00—21:00
课程介绍：http://edu.51cto.com/course/course_id-1659.html#showDesc
报名参与：http://ke.qq.com/cgi-bin/courseDetail?course_id=6167

2、大数据 Spark 实战高手之路—熟练掌握 Scala 语言视频课程

国内第一个 Scala 视频学习课程！
成为 Spark 高手必备技能，必修课程！
现在购买，即可享受套餐优惠！

课程地址： http://edu.51cto.com/pack/view/id-124.html

近期公开课：

《决胜大数据时代：Hadoop、Yarn、Spark 企业级最佳实践》

集大数据领域最核心三大技术：Hadoop 方向 50%：掌握生产环境下、源码级别下的 Hadoop 经验，解决性能、集群难点问题；Yarn 方向 20%：掌握最佳的分布式集群资源管理框架，能够轻松使用 Yarn 管理 Hadoop、Spark 等；Spark 方向 30%：未来统一的大数据框架平台，剖析 Spark 架构、内核等核心技术，对未来转向 SPARK 技术，做好技术储备。课程内容落地性强，即解决当下问题，又有助于驾驭未来。
开课时间： 10 月 26—28 日北京、11 月 1—3 日深圳

咨询电话：4006-998-758

QQ 交流群： 1 群：317540673 （已满）
2 群：297931500

微信公众号：spark-china