

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 版权所有

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

“所有人类的幸福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。



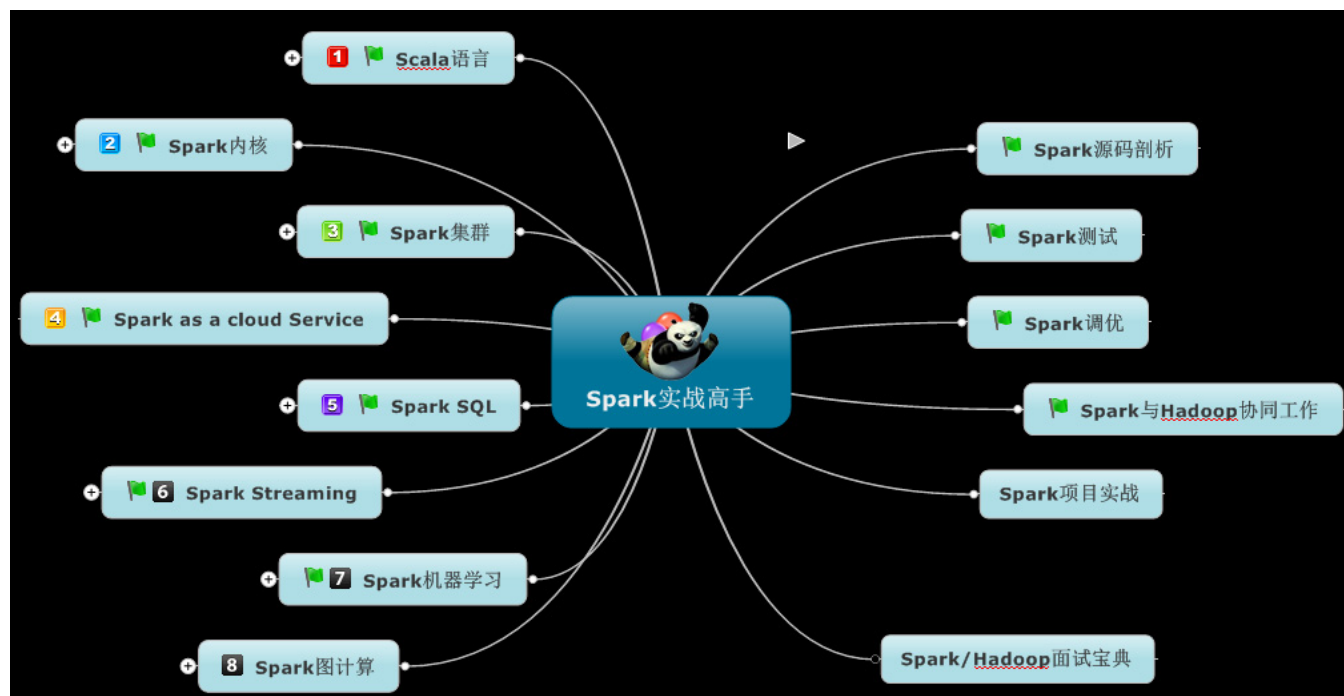
《前言》

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”](#)

第四阶段: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”](#)

第五阶段: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”](#)

第六阶段：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 6 章：Spark SQL 编程动手实战》

Spark SQL 是 Spark 1.0 版本推出以来最引人注目的 Spark 功能，Spark SQL 的前身是 Shark，而 Shark 的前身是 Hive，Shark 比 Hive 在性能高出一到两个数量级，而 Spark SQL 比 Shark 的性能又高出一到两个数量级。

Spark SQL 兼容 SQL、Hive、JSON、Parquet 等操作，同时在最新的版本推出了 JDBC/ODBC 等功能，可以同时使用 Scala、Python、Java 开发基于 Spark SQL API 的数据处理程序。

必须要提出的是 Spark SQL 可以无缝的和 Spark Streaming、GraphX、MLlib 集成，Spark Streaming SQL 以及和外部数据源整合的 Spark SQL 的功能也正在开发中。

本章是基于 Spark SQL 动手编程实践章节，从 Spark SQL 对文本文件的操作入手，到 DSL，到 Parquet，到 JSON、到 Spark SQL 在 IDEA 中的开发和调试等，最后以 Spark SQL 源码分析结束，学习完本章节即可使用 Spark SQL 进行企业级 Spark SQL 开发。

Spark SQL 编程动手实战共分四个部分：

- 第一部分：Spark SQL 操作文本文件和 DSL；
- 第二部分：Spark SQL 操作 Parquet 和 JSON；
- 第三部分：Spark SQL 在 IDEA 中的开发、运行和调试；
- 第四部分：Spark SQL 源码解析；

本讲是 Spark SQL 编程动手实战的第一部分：Spark SQL 操作文本文件和 DSL，具体内容如下所示：

- 1，动手实战操作 Spark SQL 操作文本文件；
- 2，动手实战操作 Spark SQL 操作 DSL；

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Spark SQL操作文本文件8
2. 动手实战Spark SQL操作的DSL.....13



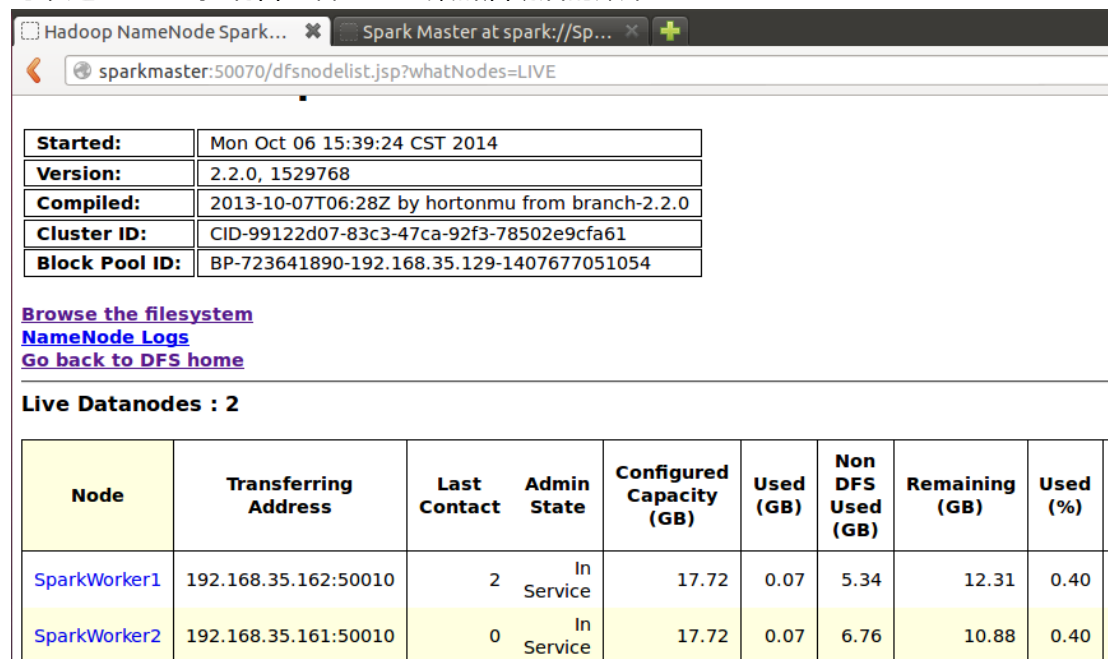
1. 动手实战 Spark SQL 操作文本文件

Step 1 : 启动 Spark 集群并连接上 spark-shell

首先启动 hdfs 集群：

```
root@SparkMaster:/usr/local/hadoop/hadoop-2.2.0/sbin# ./start-dfs.sh
Picked up _JAVA_OPTIONS: -Xms512m -Xmx1024m -XX:PermSize=1024m
Starting namenodes on [SparkMaster]
SparkMaster: starting namenode, logging to /usr/local/hadoop/hadoop-2.2.0/logs/hadoop-root-namenode-SparkMaster.out
SparkWorker2: Warning: Permanently added the ECDSA host key for IP address '192.168.35.161' to the list of known hosts.
SparkWorker1: Warning: Permanently added the ECDSA host key for IP address '192.168.35.162' to the list of known hosts.
SparkWorker2: starting datanode, logging to /usr/local/hadoop/hadoop-2.2.0/logs/hadoop-root-datanode-SparkWorker2.out
SparkWorker1: starting datanode, logging to /usr/local/hadoop/hadoop-2.2.0/logs/hadoop-root-datanode-SparkWorker1.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/hadoop-2.2.0/logs/hadoop-root-secondarynamenode-SparkMaster.out
Picked up _JAVA_OPTIONS: -Xms512m -Xmx1024m -XX:PermSize=1024m
root@SparkMaster:/usr/local/hadoop/hadoop-2.2.0/sbin#
```

可以进入 Web 控制台查看 HDFS 集群启动后的效果：



The screenshot shows the Hadoop NameNode Web console interface. The browser address bar displays 'sparkmaster:50070/dfsnodeList.jsp?whatNodes=LIVE'. The main content area shows a table with cluster metadata and a table of live datanodes.

Started:	Mon Oct 06 15:39:24 CST 2014
Version:	2.2.0, 1529768
Compiled:	2013-10-07T06:28Z by hortonmu from branch-2.2.0
Cluster ID:	CID-99122d07-83c3-47ca-92f3-78502e9cfa61
Block Pool ID:	BP-723641890-192.168.35.129-1407677051054

Below the metadata table, there are links: [Browse the filesystem](#), [NameNode Logs](#), and [Go back to DFS home](#).


The section **Live Datanodes : 2** contains a table with the following data:

Node	Transferring Address	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)
SparkWorker1	192.168.35.162:50010	2	In Service	17.72	0.07	5.34	12.31	0.40
SparkWorker2	192.168.35.161:50010	0	In Service	17.72	0.07	6.76	10.88	0.40

然后启动 Spark 集群：

```
root@SparkMaster:/usr/local/spark/spark-1.0.2-bin-hadoop2/sbin# ./start-all.sh
starting org.apache.spark.deploy.master.Master, logging to /usr/local/spark/spark-1.0.2-bin-hadoop2/sbin/../logs/spark-root-org.apache.spark.deploy.master.Master-1-SparkMaster.out
SparkWorker1: starting org.apache.spark.deploy.worker.Worker, logging to /usr/local/spark/spark-1.0.2-bin-hadoop2/sbin/../logs/spark-root-org.apache.spark.deploy.worker.Worker-1-SparkWorker1.out
SparkWorker2: starting org.apache.spark.deploy.worker.Worker, logging to /usr/local/spark/spark-1.0.2-bin-hadoop2/sbin/../logs/spark-root-org.apache.spark.deploy.worker.Worker-1-SparkWorker2.out
root@SparkMaster:/usr/local/spark/spark-1.0.2-bin-hadoop2/sbin# jps
Picked up _JAVA_OPTIONS: -Xms512m -Xmx1024m -XX:PermSize=1024m
3759 Jps
3381 SecondaryNameNode
3585 Master
3095 NameNode
root@SparkMaster:/usr/local/spark/spark-1.0.2-bin-hadoop2/sbin#
```

可以进入 Web 控制台查看 Spark 集群启动后的效果：

 **Spark Master at spark://SparkMaster:7077**

URL: spark://SparkMaster:7077

Workers: 2

Cores: 2 Total, 0 Used

Memory: 2.0 GB Total, 0.0 B Used

Applications: 0 Running, 0 Completed

Drivers: 0 Running, 0 Completed

Status: ALIVE

Workers

Id	Address	State
worker-20141006154509-SparkWorker1-41652	SparkWorker1:41652	ALIVE
worker-20141006154509-SparkWorker2-37987	SparkWorker2:37987	ALIVE

最后启动 spark-shell：

```
root@SparkMaster:/usr/local/spark/spark-1.0.2-bin-hadoop2/bin# ./spark-shell --master spark://SparkMaster:7077 --executor-memory 1g
Spark assembly has been built with Hive, including Datanucleus jars on classpath
Picked up _JAVA_OPTIONS: -Xms512m -Xmx1024m -XX:PermSize=1024m
14/10/06 15:59:52 INFO spark.SecurityManager: Changing view acls to: root
14/10/06 15:59:52 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(root)
14/10/06 15:59:52 INFO spark.HttpServer: Starting HTTP Server
14/10/06 15:59:52 INFO server.Server: jetty-8.y.z-SNAPSHOT
14/10/06 15:59:52 INFO server.AbstractConnector: Started SocketConnector@0.0.0.0:57353
Welcome to

  ____  _
 / ___|| | | |
| |___| |_| |
 \___ \|  __/
     || |___
     ||___|

 version 1.0.2

Using Scala version 2.10.4 (Java HotSpot(TM) Client VM, Java 1.7.0_67)
Type in expressions to have them evaluated.
Type :help for more information.
```

Step 2 : 动手编写 Spark SQL 写代码

首先创建 SparkContext 上下文：

```
scala> val sqlContext = new org.apache.spark.sql.SQLContext(sc)
sqlContext: org.apache.spark.sql.SQLContext = org.apache.spark.sql.SQLContext@b03b35
```

接着引入隐式转换，用于把 RDD 转为 SchemaRDD：

```
scala> import sqlContext._
import sqlContext._

scala>
```

接下来定义一个 case class 来用于描述和存储 SQL 表中的每一行数据：

```
scala> case class Person(name: String, age: Int)
defined class Person

scala> 
```

接下来要加载数据，这里的测试数据是 people.txt 文件：

Contents of directory [/data/sqlData](#)

Goto :

[Go to parent directory](#)

Name	Type	Size	Replication	Block Size	M
people.txt	file	32 B	2	128 MB	2

其内容为：

File: [/data/sqlData/people.txt](#)

Goto :

[Go back to dir listing](#)

[Advanced view/download options](#)

```
Michael, 29
Andy, 30
Justin, 19
```

加载数据：

```
scala> val people = sc.textFile("hdfs://SparkMaster:9000/data/sqlData/people.txt")
    .map(_._split(",")).map(p => Person(p(0), p(1).trim.toInt))
14/10/06 16:54:10 WARN util.SizeEstimator: Failed to check whether UseCompressed
Oops is set; assuming yes
14/10/06 16:54:10 INFO storage.MemoryStore: ensureFreeSpace(132596) called with
curMem=0, maxMem=622775500
14/10/06 16:54:10 INFO storage.MemoryStore: Block broadcast_0 stored as values t
o memory (estimated size 129.5 KB, free 593.8 MB)
people: org.apache.spark.rdd.RDD[Person] = MappedRDD[3] at map at <console>:22
scala>
```

注册成名称为 people 的 table：

```
scala> people.registerAsTable("people")
scala>
```

此时 people 依旧是一个 MappedRDD：

```
scala> people.toDebugString
14/10/06 16:57:58 INFO mapred.FileInputFormat: Total input paths to process : 1
res1: String =
MappedRDD[3] at map at <console>:22 (2 partitions)
  MappedRDD[2] at map at <console>:22 (2 partitions)
    MappedRDD[1] at textFile at <console>:22 (2 partitions)
      HadoopRDD[0] at textFile at <console>:22 (2 partitions)
scala>
```

接下进行 SQL 查询操作：

```
scala> val teenagers = sqlContext.sql("SELECT name FROM people WHERE age >= 13 A
ND age <= 19")
teenagers: org.apache.spark.sql.SchemaRDD =
SchemaRDD[6] at RDD at SchemaRDD.scala:103
== Query Plan ==
== Physical Plan ==
Project [name#0]
  Filter ((age#1 >= 13) && (age#1 <= 19))
    ExistingRDD [name#0,age#1], MapPartitionsRDD[4] at mapPartitions at basicOpera
tors.scala:208
scala>
```

此时 teenagers 已经通过隐式转换成为了 SchemaRDD，我们看一下其 lineage：

```
scala> teenagers.toDebugString
14/10/07 22:35:08 INFO mapred.FileInputFormat: Total input paths to process : 1
res1: String =
(2) SchemaRDD[6] at RDD at SchemaRDD.scala:103
== Query Plan ==
== Physical Plan ==
Project [name#0]
  Filter ((age#1 >= 13) && (age#1 <= 19))
    ExistingRDD [name#0,age#1], MapPartitionsRDD[4] at mapPartitions at basicOperators.scala:208
      | MapPartitionsRDD[8] at mapPartitions at basicOperators.scala:42
      | MapPartitionsRDD[7] at mapPartitions at basicOperators.scala:57
      | MapPartitionsRDD[4] at mapPartitions at basicOperators.scala:208
      | MappedRDD[3] at map at <console>:19
      | MappedRDD[2] at map at <console>:19
      | hdfs://SparkMaster:9000/data/sqlData/people.txt MappedRDD[1] at textFile at <console>:19
      | hdfs://SparkMaster:9000/data/sqlData/people.txt HadoopRDD[0] at textFile at <console>:19
scala>
```

通过 collect 操作出发 Job 的提交和执行：

```
scala> teenagers.map(t => "Name: " + t(0)).collect().foreach(println)
14/10/07 22:36:50 INFO spark.SparkContext: Starting job: collect at <console>:20
14/10/07 22:36:50 INFO scheduler.DAGScheduler: Got job 0 (collect at <console>:20) with 2 output partitions (allowLocal=false)
14/10/07 22:36:50 INFO scheduler.DAGScheduler: Final stage: Stage 0(collect at <console>:20)
14/10/07 22:36:50 INFO scheduler.DAGScheduler: Parents of final stage: List()
14/10/07 22:36:50 INFO scheduler.DAGScheduler: Missing parents: List()
14/10/07 22:36:50 INFO scheduler.DAGScheduler: Submitting Stage 0 (MappedRDD[9] at map at <console>:20), which has no missing parents
14/10/07 22:36:50 INFO storage.MemoryStore: ensureFreeSpace(5600) called with currentMem=156597, maxMem=277877882
14/10/07 22:36:50 INFO storage.MemoryStore: Block broadcast_1 stored as values in memory (estimated size 5.5 KB, free 264.9 MB)
14/10/07 22:36:50 INFO storage.MemoryStore: ensureFreeSpace(2986) called with currentMem=162197, maxMem=277877882
14/10/07 22:36:50 INFO storage.MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated size 2.9 KB, free 264.8 MB)
14/10/07 22:36:50 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on SparkMaster:49779 (size: 2.9 KB, free: 265.0 MB)
14/10/07 22:36:50 INFO storage.BlockManagerMaster: Updated info of block broadcast_1_piece0
14/10/07 22:36:51 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from Stage 0 (MappedRDD[9] at map at <console>:20)
```

```
kWorker1/192.168.35.163:46389]
14/10/07 22:36:52 INFO network.SendingConnection: Initiating connection to [SparkWorker2/192.168.35.161:37367]
14/10/07 22:36:52 INFO network.SendingConnection: Connected to [SparkWorker1/192.168.35.163:46389], 1 messages pending
14/10/07 22:36:52 INFO network.SendingConnection: Connected to [SparkWorker2/192.168.35.161:37367], 1 messages pending
14/10/07 22:36:52 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on SparkWorker1:46389 (size: 2.9 KB, free: 265.0 MB)
14/10/07 22:36:52 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on SparkWorker2:37367 (size: 2.9 KB, free: 265.0 MB)
14/10/07 22:36:52 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on SparkWorker1:46389 (size: 12.5 KB, free: 265.0 MB)
14/10/07 22:36:53 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on SparkWorker2:37367 (size: 12.5 KB, free: 265.0 MB)
14/10/07 22:36:59 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 8123 ms on SparkWorker1 (1/2)
14/10/07 22:37:00 INFO scheduler.DAGScheduler: Stage 0 (collect at <console>:20) finished in 9.557 s
14/10/07 22:37:00 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 9561 ms on SparkWorker2 (2/2)
14/10/07 22:37:00 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
14/10/07 22:37:00 INFO spark.SparkContext: Job finished: collect at <console>:20, took 10.065465777 s
Name: Justin
scala>
```

可以发现计算结果为 Justin。

2. 动手实战 Spark SQL 操作的 DSL

DSL 是 Domain Specific Language 的缩写，使用 DSL 我们可以直接基于读取的 RDD 数据进行 SQL 操作，无需注册成 Table。

我们推出并重新启动 spark-shell：

```
scala> exit
warning: there were 1 deprecation warning(s); re-run with -deprecation for details
root@SparkMaster:/usr/local/spark/spark-1.1.0-bin-hadoop2.4/bin# ./spark-shell -
-master spark://SparkMaster:7077
```


我们同样使用 “people.txt” 的数据：

```
scala> val sqlContext = new org.apache.spark.sql.SQLContext(sc)
sqlContext: org.apache.spark.sql.SQLContext = org.apache.spark.sql.SQLContext@1c8f13

scala> import sqlContext._
import sqlContext._

scala> case class Person(name: String, age: Int)
defined class Person

scala> val people = sc.textFile("hdfs://SparkMaster:9000/data/sqlData/people.txt")
    .map(_._split(",")).map(p => Person(p(0), p(1).trim.toInt))
14/10/08 08:58:39 WARN util.SizeEstimator: Failed to check whether UseCompressed
Oops is set; assuming yes
14/10/08 08:58:40 INFO storage.MemoryStore: ensureFreeSpace(143748) called with
curMem=0, maxMem=277877882
14/10/08 08:58:40 INFO storage.MemoryStore: Block broadcast_0 stored as values i
n memory (estimated size 140.4 KB, free 264.9 MB)
14/10/08 08:58:41 INFO storage.MemoryStore: ensureFreeSpace(12849) called with c
urMem=143748, maxMem=277877882
14/10/08 08:58:41 INFO storage.MemoryStore: Block broadcast_0_piece0 stored as b
ytes in memory (estimated size 12.5 KB, free 264.9 MB)
14/10/08 08:58:41 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in mem
ory on SparkMaster:34701 (size: 12.5 KB, free: 265.0 MB)
14/10/08 08:58:41 INFO storage.BlockManagerMaster: Updated info of block broadca
st_0_piece0
```

此时可以直接进行 SQL 查询操作:

```
scala> // The following is the same as 'SELECT name FROM people WHERE age >= 10
AND age <= 19'

scala> val teenagers = people.where('age >= 10').where('age <= 19').select('name')
teenagers: org.apache.spark.sql.SchemaRDD =
SchemaRDD[8] at RDD at SchemaRDD.scala:103
== Query Plan ==
== Physical Plan ==
Project [name#0]
  Filter ((age#1 >= 10) && (age#1 <= 19))
    ExistingRDD [name#0,age#1], MapPartitionsRDD[4] at mapPartitions at basicOpera
tors.scala:208

scala> █
```

使用 toDebugString 查看一下 lineage 关系：

```
scala> teenagers.toDebugString
14/10/08 09:01:31 INFO mapred.FileInputFormat: Total input paths to process : 1
res0: String =
(2) SchemaRDD[8] at RDD at SchemaRDD.scala:103
== Query Plan ==
== Physical Plan ==
Project [name#0]
  Filter ((age#1 >= 10) && (age#1 <= 19))
    ExistingRDD [name#0,age#1], MapPartitionsRDD[4] at mapPartitions at basicOperators.scala:208
      | MapPartitionsRDD[10] at mapPartitions at basicOperators.scala:42
      | MapPartitionsRDD[9] at mapPartitions at basicOperators.scala:57
      | MapPartitionsRDD[4] at mapPartitions at basicOperators.scala:208
      | MappedRDD[3] at map at <console>:19
      | MappedRDD[2] at map at <console>:19
      | hdfs://SparkMaster:9000/data/sqlData/people.txt MappedRDD[1] at textFile at <console>:19
      | hdfs://SparkMaster:9000/data/sqlData/people.txt HadoopRDD[0] at textFile at <console>:19
scala>
```

可以发现使用 DSL 的时候 teenagers 在内部已经被隐式转换成为了 SchemaRDD 的实例。

把结果打印出来：

```
scala> teenagers.map(t => "Name: " + t(0)).collect().foreach(println)
14/10/08 17:00:54 INFO mapred.FileInputFormat: Total input paths to process : 1
14/10/08 17:00:55 INFO spark.SparkContext: Starting job: collect at <console>:24
14/10/08 17:00:55 INFO scheduler.DAGScheduler: Got job 0 (collect at <console>:24) with 2 output partitions (allowLocal=false)
14/10/08 17:00:55 INFO scheduler.DAGScheduler: Final stage: Stage 0 (collect at <console>:24)
14/10/08 17:00:55 INFO scheduler.DAGScheduler: Parents of final stage: List()
14/10/08 17:00:55 INFO scheduler.DAGScheduler: Missing parents: List()
14/10/08 17:00:55 INFO scheduler.DAGScheduler: Submitting Stage 0 (MappedRDD[9] at map at <console>:24), which has no missing parents
14/10/08 17:00:56 INFO storage.MemoryStore: ensureFreeSpace(5632) called with curMem=156597, maxMem=277877882
14/10/08 17:00:56 INFO storage.MemoryStore: Block broadcast_1 stored as values in memory (estimated size 5.5 KB, free 264.9 MB)
14/10/08 17:00:56 INFO storage.MemoryStore: ensureFreeSpace(3032) called with curMem=162229, maxMem=277877882
14/10/08 17:00:56 INFO storage.MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated size 3.0 KB, free 264.8 MB)
14/10/08 17:00:56 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on SparkMaster:38240 (size: 3.0 KB, free: 265.0 MB)
14/10/08 17:00:56 INFO storage.BlockManagerMaster: Updated info of block broadcast_1_piece0
14/10/08 17:00:56 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from Stage 0 (MappedRDD[9] at map at <console>:24)
14/10/08 17:00:56 INFO scheduler.TaskSchedulerImpl: Adding task set 0.0 with 2 tasks
14/10/08 17:00:56 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 0.0 (TID 0, SparkWorker1, NODE_LOCAL, 1203 bytes)
14/10/08 17:00:56 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 0.0 (TID 1, SparkWorker2, NODE_LOCAL, 1203 bytes)
14/10/08 17:00:59 INFO network.ConnectionManager: Accepted connection from [SparkWorker2/192.168.35.161:48222]
14/10/08 17:00:59 INFO network.SendingConnection: Initiating connection to [SparkWorker2/192.168.35.161:35277]
14/10/08 17:00:59 INFO network.ConnectionManager: Connected to [SparkWorker2/192.168.35.161:35277], 1 messages pending
14/10/08 17:00:59 INFO network.ConnectionManager: Accepted connection from [SparkWorker1/192.168.35.163:37303]
14/10/08 17:00:59 INFO network.SendingConnection: Initiating connection to [SparkWorker1/192.168.35.163:47158]
14/10/08 17:00:59 INFO network.ConnectionManager: Connected to [SparkWorker1/192.168.35.163:47158], 1 messages pending
14/10/08 17:01:00 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on SparkWorker2:35277 (size: 3.0 KB, free: 265.0 MB)
14/10/08 17:01:00 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on SparkWorker1:47158 (size: 3.0 KB, free: 265.0 MB)
14/10/08 17:01:05 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on SparkWorker2:35277 (size: 12.5 KB, free: 265.0 MB)
14/10/08 17:01:06 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on SparkWorker1:47158 (size: 12.5 KB, free: 265.0 MB)
14/10/08 17:02:12 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 76340 ms on SparkWorker2 (1/2)
14/10/08 17:02:13 INFO scheduler.DAGScheduler: Stage 0 (collect at <console>:24) finished in 76.937 s
14/10/08 17:02:13 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 76919 ms on SparkWorker1 (2/2)
14/10/08 17:02:13 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
14/10/08 17:02:13 INFO spark.SparkContext: Job finished: collect at <console>:24, took 77.627147606 s
Name: Justin
scala>
```

可以发现结果依旧是 Justin，这和前面的计算是一样的。

■ Spark 亚太研究院

Spark 亚太研究院是中国最专业的一站式大数据 Spark 解决方案供应商和高品质大数据企业级完整培训与服务供应商，以帮助企业规划、架构、部署、开发、培训和使用 Spark 为核心，同时提供 Spark 源码研究和应用技术训练。针对具体 Spark 项目，提供完整而彻底的解决方案。包括 Spark 一站式项目解决方案、Spark 一站式项目实施方案及 Spark 一体化顾问服务。

官网：www.sparkinchina.com

■ 近期活动



- ▶ 2014 年亚太地区规格最高的 Spark 技术盛会！
- ▶ 面向大数据、云计算开发者、技术爱好者的饕餮盛宴！
- ▶ 云集国内外 Spark 技术领军人物及灵魂人物！
- ▶ 技术交流、应用分享、源码研究、商业案例探讨！

时间：2014 年 12 月 6-7 日

地点：北京珠三角万豪酒店

Spark 亚太峰会网址：<http://www.sparkinchina.com/meeting/2014yt/default.asp>



- ▶ 如果你是对 Spark 有浓厚兴趣的初学者，在这里你会有绝佳的入门和实践机会！
- ▶ 如果你是 Spark 的应用高手，在这里以“武”会友，和技术大牛们尽情切磋！
- ▶ 如果你是对 Spark 有深入独特见解的专家，在这里可以尽情展现你的才华！

比赛时间：

2014 年 9 月 30 日—12 月 3 日

Spark 开发者大赛网址：<http://www.sparkinchina.com/meeting/2014yt/dhhd.asp>

■ 视频课程：

《大数据 Spark 实战高手之路》 国内第一个 Spark 视频系列课程

从零起步，分阶段无任何障碍逐步掌握大数据统一计算平台 Spark，从 Spark 框架编写和开发语言 Scala 开始，到 Spark 企业级开发，再到 Spark 框架源码解析、Spark 与 Hadoop 的融合、商业案例和企业面试，一次性彻底掌握 Spark，成为云计算大数据时代的幸运儿和弄潮儿，笑傲大数据职场和人生！

- ▶ 第一阶段：熟练的掌握 Scala 语言
课程学习地址：<http://edu.51cto.com/pack/view/id-124.html>
- ▶ 第二阶段：精通 Spark 平台本身提供给开发者 API
课程学习地址：<http://edu.51cto.com/pack/view/id-146.html>
- ▶ 第三阶段：精通 Spark 内核
课程学习地址：<http://edu.51cto.com/pack/view/id-148.html>
- ▶ 第四阶段：掌握基于 Spark 上的核心框架的使用
课程学习地址：<http://edu.51cto.com/pack/view/id-149.html>
- ▶ 第五阶段：商业级别大数据中心黄金组合：Hadoop+ Spark
课程学习地址：<http://edu.51cto.com/pack/view/id-150.html>
- ▶ 第六阶段：Spark 源码完整解析和系统定制
课程学习地址：<http://edu.51cto.com/pack/view/id-151.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：2014 年 10 月 26-28 日北京、2014 年 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