



Spark 官方文档翻译

Bagel 编程指南 (v1.1.0)

翻译者 蔡立宇

Spark 官方文档翻译团成员

前言

世界上第一个Spark 1.1.0 中文文档问世了！

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

Spark是当今大数据领域最活跃最热门的高效大数据通用计算平台，基于RDD，Spark成功的构建起了一体化、多元化的大数据处理体系，在“One Stack to rule them all”思想的引领下，Spark成功的使用Spark SQL、Spark Streaming、MLLib、GraphX近乎完美的解决了大数据中Batch Processing、Streaming Processing、Ad-hoc Query等三大核心问题，更为美妙的是在Spark中Spark SQL、Spark Streaming、MLLib、GraphX四大子框架和库之间可以无缝的共享数据和操作，这是当今任何大数据平台都无可匹敌的优势。

在实际的生产环境中，世界上已经出现很多一千个以上节点的Spark集群，以eBay为例，eBay的Spark集群节点已经超过2000个，Yahoo 等公司也在大规模的使用Spark，国内的淘宝、腾讯、百度、网易、京东、华为、大众点评、优酷土豆等也在生产环境下深度使用Spark。2014 Spark Summit上的信息，Spark已经获得世界20家顶级公司的支持，这些公司中包括Intel、IBM等，同时更重要的是包括了最大的四个Hadoop发行商，都提供了对Spark非常强有力的支持。

与Spark火爆程度形成鲜明对比的是Spark人才的严重稀缺，这一情况在中国尤其严重，这种人才的稀缺，一方面是由于Spark技术在2013、2014年才在国内的一些大型企业里面被逐步应用，另一方面是由于匮乏Spark相关的中文资料和系统化的培训。为此，Spark亚太研究院和51CTO联合推出了“Spark亚太研究院决胜大数据时代100期公益大讲堂”，来推动Spark技术在国内的普及及落地。

具体视频信息请参考 http://edu.51cto.com/course/course_id-1659.html

与此同时，为了向Spark学习者提供更为丰富的学习资料，Spark亚太研究院发起并号召，结合网络社区的力量构建了Spark中文文档专家翻译团队，历经1个月左右的艰苦努力和反复修改，Spark中文文档V1.1终于完成。尤其值得一提的是，在此次中文文档的翻译期间，Spark官方团队发布了Spark 1.1.0版本，为了让学习者了解到最新的内容，Spark中文文档专家翻译团队主动提出基于最新的Spark 1.1.0版本，更新了所有已完成的翻译内容，在此，我谨代表Spark亚太研究院及广大Spark学习爱好者向专家翻译团队所有成员热情而专业的工作致以深刻的敬意！

当然，作为世界上第一份相对系统的Spark中文文档，不足之处在所难免，大家有任何建议或者意见都可以发邮件到marketing@sparkinchina.com；同时如果您想加入

Spark中文文档翻译团队，也请发邮件到marketing@sparkinchina.com进行申请；Spark中文文档的翻译是一个持续更新的、不断版本迭代的过程，我们会尽全力给大家提供更高质量的Spark中文文档翻译。

最后，也是最重要的，请允许我荣幸的介绍一下我们的Spark中文文档第一个版本翻译的专家团队成员，他们分别是（排名不分先后）：

- ▶ 傅智勇，《快速开始(v1.1.0)》（和唐海东翻译的是同一主题，大家可以对比参考）
- ▶ 吴洪泽，《Spark机器学习库（v1.1.0）》（其中聚类和降维部分是蔡立宇翻译）
- ▶ 武扬，《在Yarn上运行Spark（v1.1.0）》《Spark 调优(v1.1.0)》
- ▶ 徐骄，《Spark配置(v1.1.0)》《Spark SQL编程指南(v1.1.0)》（Spark SQL和韩保礼翻译的是同一主题，大家可以对比参考）
- ▶ 蔡立宇，《Bagel 编程指南(v1.1.0)》
- ▶ harli，《Spark 编程指南（v1.1.0）》
- ▶ 吴卓华，《图计算编程指南(1.1.0)》
- ▶ 樊登贵，《EC2(v1.1.0)》《Mesos(v1.1.0)》
- ▶ 韩保礼，《Spark SQL编程指南(v1.1.0)》（和徐骄翻译的是同一主题，大家可以对比参考）
- ▶ 颜军，《文档首页(v1.1.0)》
- ▶ Jack Niu，《Spark实时流处理编程指南(v1.1.0)》
- ▶ 俞杭军，《sbt-assembly》《使用Maven编译Spark(v1.1.0)》
- ▶ 唐海东，《快速开始(v1.1.0)》（和傅智勇翻译的是同一主题，大家可以对比参考）
- ▶ 刘亚卿，《硬件配置(v1.1.0)》《Hadoop 第三方发行版(v1.1.0)》《给Spark提交代码(v1.1.0)》
- ▶ 耿元振《集群模式概览(v1.1.0)》《监控与相关工具(v1.1.0)》《提交应用程序(v1.1.0)》
- ▶ 王庆刚，《Spark作业调度(v1.1.0)》《Spark安全(v1.1.0)》
- ▶ 徐敬丽，《Spark Standalone 模式（v1.1.0）》

另外关于Spark API的翻译正在进行中，敬请关注。

Life is short, You need Spark!

Spark亚太研究院院长 王家林
2014 年 10 月

Spark 亚太研究院决胜大数据时代 100 期公益大讲堂

简介

作为下一代云计算的核心技术，Spark性能超Hadoop百倍，算法实现仅有其 1/10 或 1/100,是可以革命Hadoop的目前唯一替代者，能够做Hadoop做的一切事情，同时速度比Hadoop快了 100 倍以上。目前Spark已经构建了自己的整个大数据处理生态系统，国外一些大型互联网公司已经部署了Spark。甚至连Hadoop的早期主要贡献者Yahoo现在也在多个项目中部署使用Spark；国内的淘宝、优酷土豆、网易、Baidu、腾讯、皮皮网等已经使用Spark技术用于自己的商业生产系统中，国内外的应用开始越来越广泛。Spark正在逐渐走向成熟，并在这个领域扮演更加重要的角色，刚刚结束的2014 Spark Summit上的信息，Spark已经获得世界 20 家顶级公司的支持，这些公司中包括Intel、IBM等，同时更重要的是包括了最大的四个Hadoop发行商都提供了对非常强有力的支持Spark的支持。

鉴于Spark的巨大价值和潜力，同时由于国内极度缺乏Spark人才，Spark亚太研究院在完成了对Spark源码的彻底研究的同时，不断在实际环境中使用Spark的各种特性的基础之上，推出了Spark亚太研究院决胜大数据时代 100 期公益大讲堂，希望能够帮助大家了解Spark的技术。同时，对Spark人才培养有近一步需求的企业和个人，我们将以公开课和企业内训的方式，来帮助大家进行Spark技能的提升。同样，我们也为企业提供一体化的顾问式服务及Spark一站式项目解决方案和实施方案。

Spark亚太研究院决胜大数据时代 100 期公益大讲堂是国内第一个Spark课程免费线上讲座，每周一期，从 7 月份起，每周四晚 20:00-21:30，与大家不见不散！老师将就Spark内核剖析、源码解读、性能优化及商业实战案例等精彩内容与大家分享，干货不容错过！

时间：从 7 月份起，每周一期，每周四晚 20:00-21:30

形式：腾讯课堂在线直播

学习条件：对云计算大数据感兴趣的技术人员

课程学习地址：http://edu.51cto.com/course/course_id-1659.html

Bagel 编程指南(v1.1.0)

(翻译者：蔡立宇)

Bagel Programming Guide , 原文档链

接：<http://spark.apache.org/docs/latest/bagel-programming-guide.html>

目录

第一章Bagel 编程指南(v1.1.0)	6
引用Bagel	6
编程模型	6
合并器	8
聚合器	8
操作	9
1.1 Actions	9
1.2Types	9
接下来... ..	10

第一章 Bagel 编程指南(v1.1.0)

Bagel模块即将被 [GraphX](#)所替代，对于新的用户我们建议使用GraphX。

Bagel 是Google的 [Pregel](#) 图处理框架在Spark上的实现，目前它支持基本的图计算、合并和聚合操作。

在 Pregel 编程模型里，每个任务被分解为顺序执行的多个迭代。每一次迭代称为一个超步。在每一个超步里，图中的各节点通过执行用户定义的函数来更新自身状态，并且向其他节点发送它们下一次迭代所需要的信息。

这里，我们将以 PageRank 算法在 Bagel 框架下的实现为例，来演示 Pregel 的编程模块和它的功能。

引用 Bagel

若需在你的程序中使用 Bagel，需要加入如下内容的 SBT 或相应的 Maven 依赖来引用它：

```
groupId = org.apache.spark  
artifactId = spark-bagel_2.10  
version = 1.1.0-SNAPSHOT
```

编程模型

在Bagel编程模型下，一个图被表示为一个以(K,V)键值对为基本元素的 [分布式数据集](#)。在每一个键值对里，K是节点的ID，而V值则是节点和该节点所附带的状态信息。在每一个超步里，Bagel 对每一个节点都调用用户所定义的函数。该函数以该节点当前的状态以及在上一个超步中从其他节点接收到的信息为输入，输出则为该节点的状态的更新以及发送给其他节点的消息列。

以 PageRank 在 Bagel 框架下的实现为例，节点可代表网页，边则表示网页之间的链接，而消息则表示当前页面与该消息接收页面所共享的 PageRank 值。

具体实现时，我们需先扩展默认的 Vertex 类来多存一个 Double 值。这个值用于存储节点当前的 PageRank 值。Message 和 Edge 类也做类似的扩展。注意，扩展后的类需用 @serializable 标注，以使得 Spark 可在各机器之间传输这类对象。我们同样需要导入 Bagel 的类和隐式转换。

```
import org.apache.spark.bagel._
import org.apache.spark.bagel.Bagel._

@serializable class PREdge(val targetId: String) extends Edge

@serializable class PRVertex(
  val id: String, val rank: Double, val outEdges: Seq[Edge],
  val active: Boolean) extends Vertex

@serializable class PRMessage(
  val targetId: String, val rankShare: Double) extends Message
```

然后，我们从一个文本文件中导入一个样例图，形成其相应的分布式数据集，然后将数据集打包为 PRVertex 对象。为避免在后续多次访问时所需要的转换操作，我们将其缓存起来。

```
val input = sc.textFile("data/ml1b/pagerank_data.txt")

val numVerts = input.count()

val verts = input.map(line => {
  val fields = line.split('\t')
  val (id, linksStr) = (fields(0), fields(1))
  val links = linksStr.split(',').map(new PREdge(_))
  (id, new PRVertex(id, 1.0 / numVerts, links, true))
}).cache
```

接下来，我们定义供调用的入口函数，该函数的输入参数有 verts、一个作为消息的空的分布式数据集，以及一个自定义的迭代 10 次的 PageRank 函数。

```
val emptyMsgs = sc.parallelize(List[(String, PRMessage)]())

def compute(self: PRVertex, msgs: Option[Seq[PRMessage]], superstep: Int)
: (PRVertex, Iterable[PRMessage]) = {
  val msgSum = msgs.getOrElse(List()).map(_._rankShare).sum
  val newRank =
    if (msgSum != 0)
```



```

    0.15 / numVerts + 0.85 * msgSum
  else
    self.rank
  val halt = superstep >= 10
  val msgsOut =
    if (!halt)
      self.outEdges.map(edge =>
        new PRMessage(edge.targetId, newRank / self.outEdges.size))
    else
      List()
  (new PRVertex(self.id, newRank, self.outEdges, !halt), msgsOut)
}

```

val result = Bagel.run(sc, verts, emptyMsgs)()(compute) 【这行估计本来是上面代码的最后一行，但原始网页中把它分到正文里了。
】

最后，我们输出结果：

```
println(result.map(v => "%s\t%s\n".format(v.id, v.rank)).collect.mkString)
```

合并器

一般来说，向其他节点发送消息会有较高的网络通讯代价。对于某些特定的算法而言，或可通过 Combiner(合并器)来减少所需的通讯。举例来说，如果计算函数，也就是用户自定义的函数，接收的是整形数值且只是要对接收到的值求和，那么在 Bagel 框架下或可先通过对发往同一个节点的多条消息进行求和来合并这些消息，从而减少通讯。

在 Bagel 所提供的合并器功能里，用户可自定义一组合并函数来将消息转换为他们合并后的形式。

例如: *PageRank with combiners*

聚合器

聚合器在每一个超步后对所有的节点执行一个 Reduce(精简)操作，并将结果提供给下一个超步中的各节点。

对于聚合功能，用户也可自定义一个聚合函数，该函数将作用于每一个节点。

操作

如下是 Bagel API 中支持的 Actions 和 Types。详见 Bagel.scala。

1.1 Actions

```
/** 完整形式 */

Bagel.run(sc, vertices, messages, combiner, aggregator, partitioner, numSplits)(compute)
// 计算的输入为 (vertex: V, combinedMessages: Option[C], aggregated: Option[A], superstep: Int)
// , 对应的返回值是 (newVertex: V, outMessages: Array[M])

/** 简写形式 */

Bagel.run(sc, vertices, messages, combiner, partitioner, numSplits)(compute)
// 计算的输入为 (vertex: V, combinedMessages: Option[C], superstep: Int)
// , 相应的返回值是 (newVertex: V, outMessages: Array[M])

Bagel.run(sc, vertices, messages, combiner, numSplits)(compute)
// 计算的输入为 (vertex: V, combinedMessages: Option[C], superstep: Int)
// , 相应的返回值是 (newVertex: V, outMessages: Array[M])

Bagel.run(sc, vertices, messages, numSplits)(compute)
// 计算的输入为 (vertex: V, messages: Option[Array[M]], superstep: Int)
// , 相应的返回值是 (newVertex: V, outMessages: Array[M])
```

1.2 Types

```
trait Combiner[M, C] {
  def createCombiner(msg: M): C
  def mergeMsg(combiner: C, msg: M): C
  def mergeCombiners(a: C, b: C): C
}

trait Aggregator[V, A] {
  def createAggregator(vert: V): A
  def mergeAggregators(a: A, b: A): A
}
```

```
trait Vertex {  
  def active: Boolean  
}  
  
trait Message[K] {  
  def targetId: K  
}
```

接下来...

在 `examples/src/main/scala/org/apache/spark/examples/bagel` 中包含了两份示例程序。运行他们只需向 Spark 中已包含的 `bin/run-example` 脚本传入它们的类名即可。比如：

```
./bin/run-example org.apache.spark.examples.bagel.WikipediaPageRank
```

在不传入任何参数时，这些程序将会提示使用指南。

■ 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