



Spark 官方文档翻译

Standalone (V1.1.0)

翻译者 徐敬丽

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 and 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

Standalone (V1.1.0)

(翻译者：徐敬丽)

Spark Standalone Mode , 原文档链接：

<http://spark.apache.org/docs/latest/spark-standalone.html>

目录

第 1 章 Spark单机模式 (Standalone Mode)	6
1.1. 安装Standalone集群.....	6
1.2. 手动开启集群.....	6
1.3. 集群启动脚本.....	7
1.4. Applications的集群挂载.....	10
1.5. 启动编译的Applications.....	10
1.6. 资源调度.....	11
1.7. 监控与日志.....	11
1.8. 与Hadoop集群同机运行.....	12
1.9. 配置网络安全端口.....	12
1.10. 高可用性 (HA)	12
1.10.1. 基于ZooKeeper的Standby Masters.....	12
1.10.2. 基于文件系统的单点恢复.....	14

第 1 章 Spark 单机模式 (Standalone Mode)

<http://spark.apache.org/docs/latest/spark-standalone.html>

Spark 除了可以在Mesos或YARN的集群管理器上运行之外,它自身也提供了一个简单的独立部署 (Standalone deploy) 模式。你可以通过创建master和workers, 或者使用我们提供的 [启动脚本](#), 来手动开启一个standalone集群; 也可以在单个机器上运行这些守护程序进行测试。

1.1. 安装 Standalone 集群

安装Standalone模式集群,只需放置一个编译版本的Spark在集群的每个节点上,然后通过系统的每次释放来获得Spark的预构建版本,或者也可以 [自己创建](#) 一个这样的Spark版本。

1.2. 手动开启集群

启动一个单机模式的 master 服务器可以执行代码：

```
./sbin/start-master.sh
```

Master 服务器一旦被启动,就会打印出自己的URL信息: spark://HOST:PORT, 通过URL可以与workers建立链接,或者把URL作为“master”参数传递给 SparkContext接口。Master的URL信息也可以从它的web用户界面 (UI) 获得, 默认的master web UI地址是: <http://localhost:8080>.

同样,你也可以开启一个或多个workers, 并通过以下代码将它们与 master 建立链接：

```
./bin/spark-class org.apache.spark.deploy.worker.Worker spark://IP:PORT
```

成功运行后, 可以通过 master web UI 查看workers的相关信息 (默认为: <http://localhost:8080>), 这时应该可以看到web UI上列出了新的节点以及它们所分配到的CPU个数和内存量 (减去操作系统 (OS) 的 1G内存)。

最后，给出 master 和 workers 的配置信息选项：

参数	描述
<code>-i IP, --ip IP</code>	IP 地址和 DNS 名称
<code>-p PORT, --port PORT</code>	服务端口（默认值：master 为 7077；worker 为随机值）
<code>--webui-port PORT</code>	Web UI 端口（默认值：master 为 8080，worker 为 8081）
<code>-c CORES, --cores CORES</code>	设置 Spark applications 可用的总 CPU 内核数（默认全部可用）；仅适用于 worker
<code>-m MEM, --memory MEM</code>	设置 Spark applications 可用的总内存，如 1000M 或 2G 等（默认值为 RAM 总内存减去 1GB）；仅适用于 worker
<code>-d DIR, --work-dir DIR</code>	设置文件目录用于缓存空间和工作日志输出（默认为：SPARK_HOME/work）；仅适用于 worker

1.3. 集群启动脚本

如果用启动脚本来启动一个 Spark Standalone 集群，需要事先在 Spark 目录下创建一个 `conf/slaves` 文件，它应该包括所有 worker 机器的主机名，每个主机名占据一行。启动集群前，必须保证 master 机器能够通过 ssh（私钥）无密码链接到每个 workers 机器上。

测

试时，只需把 localhost 放入文件 `conf/slaves` 即可

创建好文件 `conf/slaves` 之后，就可以通过以下 shell 脚本开启或停止 Spark 集群的相关进程，这些 shell 脚本继承了 Hadoop 的 `deploy` 脚本风格，存放在 Spark 目录 `SPARK_HOME/bin` 下：

- `sbin/start-master.sh` – 创建一个 master 实例
- `sbin/start-slaves.sh` – 创建 slave 实例（`conf/slaves` 中指定的每台机器上）
- `sbin/start-all.sh` – 同时创建一个 master 和多个 slave 实例（如上所述）
- `sbin/stop-master.sh` – 停止 master（由 `bin/start-master.sh` 创建）
- `sbin/stop-slaves.sh` – 停止所有 slave 实例（位于 `conf/slaves` 指定的机器上）
- `sbin/stop-all.sh` – 同时停止 master 和 slaves（如上所述）

注意这些脚本只能通过集群的 master 机执行。

你还可以通过创建 `conf/spark-env.sh` 文件，进一步选择性地配置集群的环境变量，从而更好地控制集群。文件 `conf/spark-env.sh` 可以通过 `conf/spark-env.sh.template` 模板来创建，然后拷贝到所有的 worker 机上来使设置生效，以下是可供选择的配置信息：

环境变量	描述
SPARK_MASTER_IP	绑定 master 到一个特定的 IP 地址, 例如公共 IP
SPARK_MASTER_PORT	在另一个端口上启动 master (默认: 7077)
SPARK_MASTER_WEBUI_PORT	Master web UI 端口 (默认值: 8080)
SPARK_MASTER_OPTS	设置只适用于 Master 的配置属性, 形式为 “-Dx = y” (默认值: 无), 配置选项参见下面列表。
SPARK_LOCAL_DIRS	设置 Spark 缓存空间目录, 包括存储在磁盘上的映射输出文件和 RDDs; 这可以是一种快速的本地磁盘系统, 也可以是一个以逗号分隔的位于不同磁盘上的多个目录。
SPARK_WORKER_CORES	设置 Applications 可用的内核总数 (默认: 可用的全部内核).
SPARK_WORKER_MEMORY	设置 Applications 可用的内存总量, 例如 1000M, 2G (默认值: 内存总量减去 1GB)。注意: 单个 Application 的内存配置使用它的 <code>spark.executor.memory</code> 属性
SPARK_WORKER_PORT	在特定端口上启动 worker (默认: 随机值)。
SPARK_WORKER_WEBUI_PORT	Worker web UI 端口 (默认值: 8081)
SPARK_WORKER_INSTANCES	在每台机器上运行的 worker 实例数量 (默认值: 1)。如果你有非常多的机器, 并且希望启动多个 spark-worker 进程, 可以使 worker 的实例数大于 1。如果这样设置, 还要确保设置环境变量 SPARK_WORKER_CORES 来明确限制每个 worker 所允许使用的内核总数, 否则每个 worker 将会尝试使用所有的处理器内核。
SPARK_WORKER_DIR	设置 applications 的运行目录, 包括日志和暂存空间 (默认值: SPARK_HOME /work)。
SPARK_WORKER_OPTS	设置只适用于 Worker 的配置属性, 形式为 “-Dx = y” (默认值: 无), 配置选项参见下面列表。
SPARK_DAEMON_MEMORY	为 Spark- master 和 worker 的守护进程分配内存 (默认值: 512M)。
SPARK_DAEMON_JAVA_OPTS	Spark-master 和 worker 守护进程的 JVM 选项, (默认为: 无)。
SPARK_PUBLIC_DNS	设置 Spark-master 和 workers 的公共 DNS 名称 (默认值: 无)

注意 这些启动脚本文件目前不支持 Windows 系统 如果要在 Windows 上运行 Spark

集群，就要手动启动 master 和 workers.

SPARK_MASTER_OPTS 支持下面的系统属性：

属性名称	默认值	描述
<code>spark.deploy.retainApplications</code>	200	显示完成应用程序（application）的最大数目，较老的 applications 将从 UI 丢弃掉来维持这一限制。
<code>spark.deploy.retainDrivers</code>	200	显示完成驱动程序（driver）的最大数目，较老的 driver 将从 UI 丢弃掉来维持这一限制。
<code>spark.deploy.spreadOut</code>	true	standalone 集群管理器判断是否应该跨节点扩布 applications 或试图整合到尽可能少的节点上。扩布通常更好地使 HDFS 数据本地化，但对计算密集型工作负载整合更有效。
<code>spark.deploy.defaultCores</code>	(infinite)	Spark standalone 集群模式中，在没有设置 <code>spark.cores.max</code> 的情形下，系统默认分配给 applications 的内核数目。这时 applications 总是获取所有可用内核，除非他们自己配置 <code>spark.cores.max</code> 值。在共享集群上，应该设置 <code>spark.cores.max</code> 为较低值，以防止用户默认抓取整个集群。
<code>spark.worker.timeout</code>	60	设置时限（秒），在此后如果 standalone deploy master 仍未接收到 worker 的任何注册信息，就可视此 worker 已经丢失

SPARK_WORKER_OPTS 支持下面的系统属性：

属性名称	默认值	描述
<code>spark.worker.cleanup.enabled</code>	False	启动定期清理 worker/application 目录功能。 注意，这仅影响 standalone 模式，因为 YARN 工作方式不同。

		定期清理功能一旦启动，无论 application 是否正在工作，application 目录都会被清理掉。
spark.worker.cleanup.interval	1800 (30 minutes)	控制时间间隔（以秒为单位），用来使 worker 清理旧的 application 在本地机器上的工作目录
spark.worker.cleanup.appDataTtl	7 * 24 * 3600 (7 days)	设置 worker 能够保留 application 工作目录的时长（以秒为单位），即生命周期，它取决于可用磁盘空间的数量。Application 日志和 jar 文件会被同时下载到每个 application 的工作目录。随着时间的推移，工作目录将会迅速填满磁盘空间，特别是在频繁运行 jobs 的情况下。

1.4. Applications 的集群挂载

配置好集群之后就可以在集群上挂载一个应用(application)，只需把master的URL：spark://IP:PORT作为参数传递给 [SparkContext](#)接口的构造方法。

执行以下命令就可以在集群上运行一个交互的 Spark- shell 脚本：

```
./bin/spark-shell --master spark://IP:PORT
```

你也可以配置属性--cores <numCores>去控制 spark-shell 在集群上所使用的 CPU 内核数。

1.5. 启动编译的 Applications

脚本 [spark-submit](#) 提供了standalone集群提交编译的spark-application的最直接的方式。对于standalone 集群，spark目前仅支持客户端作业进程内的驱动程序（[client deploy mode](#)）。

如果spark-application通过脚本 [spark-submit](#) 启动,那么application包就会自动分配到所有的worker节点上。如果application还同时依赖于其他的包文件,就应该用 --jars 标志来指定它们,不同包文件之间用逗号分隔(如--jars jar1,jar2)。有关application的配置和执行环境,参考 [Spark配置](#)。

1.6. 资源调度

Standalone的集群模式目前只支持简单的FIFO作业调度方式。但是,你可以通过控制Spark作业能够获取的最大资源数来实现大量的并发作业。在默认配置的情形下,一个作业会占用集群中的全部资源,但是这在每次只有一个作业进程的情况下才有意义。在初始化SparkContext接口之前,你可以在 [SparkConf](#)对象中设置spark.cores.max来覆盖系统设置的可用内核数属性,例如:

```
val conf = new SparkConf()
    .setMaster(...)
    .setAppName(...)
    .set("spark.cores.max", "10")

val sc = new SparkContext(conf)
```

另外,也可以在集群的 master 进程中配置脚本文件 spark.deploy.defaultCores 来改变作业运行所需最大内核数的默认值属性,但是这需要将下面的配置选项添加到 conf/spark-env.sh 文件中:

```
export SPARK_MASTER_OPTS="-Dspark.deploy.defaultCores=<value>"
```

这在共享集群中是很有用的,因为用户不可能单独配置自己作业的最大可用内核数属性。

1.7. 监控与日志

Spark 单机模式通过 web UI 来监控集群, master 和每一个 worker 都有自己的 web UI 用于显示集群和作业的统计资料。默认情形下,用户可以访问 master 的主端口:8080 进入到 master 的 web UI, 还可以通过配置文件或命令行选项来改变端口值。

此外, 每个作业输出的详细日志也可以写入到它工作目录中(默认为: SPARK_HOME/work)。在工作目录中,每个作业都对应两个文件 stdout 和 stderr, 这两个文件包含了该作业所有控制台输出的信息。

1.8. 与 Hadoop 集群同机运行

Spark 可以作为一个单独的服务与 Hadoop 集群同机运行。要想从 Spark 的作业进程中访问到 Hadoop 集群的数据，只需使用 `hdfs://URL`（通常为 `hdfs://<namenode>:9000/path`，用户可以从 Hadoop 节点的 web UI 上找到正确的 URL）。或者你也可以建立一个独立的 Spark 集群，并且在整个网络上赋予它 HDFS 权限；这可能要比本地磁盘存储更慢，但如果在相同的局域网中运行，就不必担心（例如你可以把几台 Spark 机和 Hadoop 机放在同一个机架上）。

1.9. 配置网络安全端口

Spark大量使用网络资源的特性，使得它的运行环境对防火墙设置具有严格的要求。需要配置的端口的完整列表参见：[安全页面](#)。

1.10. 高可用性（HA）

通常情况下，单机模式的集群调度具有一定的容错能力（目前 Spark 通过移除失败作业到其它的workers上来实现自身的抗故障特性）。然而，调度器用master来做调度决策，这无形中将会产生单点故障的问题，如果 master 崩溃，就没有可创建的新的作业了。如何解决单点故障的问题，Spark 提出了两种方案，具体如下：

1.10.1. 基于 ZooKeeper 的 Standby Masters

1.10.1.1. 概述

将 Standalone 集群连接到同一个 ZooKeeper 实例并启动多个 Master，利用 ZooKeeper 提供的选举和状态保存功能，可以使一个 Master 被选举，而其他 Master 处于 Standby 状态。如果现任 Master 死去，另一个 Master 会通过选举产生，并恢复到旧的 Master 状态，然后恢复调度。整个恢复过程可能要 1-2 分钟，这个过程只会影响新的 Applications 的调度，对于在故障切换过程中正在运行的 Application 没有任何影响。

了解更多有关ZooKeeper的入门知识请参见[这里](#)。

1.10.1.2. 配置

为了启用这个恢复模式，需要用下面的配置选项在 spark-env 中对环境变量

SPARK_DAEMON_JAVA_OPTS 进行设置:

系统属性	描述
spark.deploy.recoveryMode	设置 ZOOKEEPER 启用 standby Master 恢复模式（缺省值：NONE）
spark.deploy.zookeeper.url	ZooKeeper 集群的 URL (例如：192.168.1.100:2181,192.168.1.101:2181).
spark.deploy.zookeeper.dir	设置 ZooKeeper 集群中用于存储恢复状态的文件目录（缺省值：/spark）

可能的陷阱：将 Standalone 集群连接到同一个 ZooKeeper 实例并启动多个 Masters, 如果不能正确配置 Master 的 ZooKeeper 信息, Masters 就不会发现彼此而把它们看作是 leaders. 这将会形成一个不健康的集群状态（所有 Masters 之间彼此相互独立地调度）。

1.10.1.3. 细节

完成 ZooKeeper 集群的设置后，启用其高可用性是非常简单的。首先启动这个 ZooKeeper 集群，然后在不同节点上启动多个 Masters，注意这些节点需要具有相同的 ZooKeeper 配置（ZooKeeper URL 和目录），而且 Masters 可以随时被添加和移除。

当调用新的 Applications 或添加新的 Workers 时，它们需要知道当前 Master 的 IP 地址，

这种 HA 方案处理这种情况很简单，只需要使 SparkContext 接口指向一个 Master 列表就可以了，如 spark://host1:port1,host2:port2。SparkContext 就会尝试着依次对 Master 列表进行注册并轮询，如果 host1 出现故障，配置信息对于寻找 host2 仍然会有效。

注册接入 Master 和正常运行没有什么重要不同，当集群启动时，application 或 worker 需要能够找到并注册接入到当前的 lead Master，一旦注册成功，它们就会被并入系统（例如储存在 ZooKeeper 中）。发生故障切换时，新被选中的 Master 就会联系所有先前已经注册成功的 applications 和 Workers，并告知它们领导关系的变化。因此从某种意义上讲，这些 applications 和 workers 并不需要事先知道新 Master 的存在性。

正是由于这一属性，新的 Masters 可以在任何时候被创建。唯一需要注意的是，在这个新的 Masters 成为 leader 的情形下，新的 applications 和 workers 能够找到并注册接入到这个 Master。一旦注册成功，这个 application 或 worker 就会被启用。

1.10.2. 基于文件系统的单点恢复

1.10.2.1. 概述

尽管 ZooKeeper 是用于生产模式高可用性的最佳途径，但是当 Master 发生故障时，如果你想重新启动 Master，文件系统（FILESYSTEM）模式才能起作用。Spark 提供了目录来保存 Application 和 worker 的注册状态信息，一旦 Master 发生故障，就可以通过重新启动 Master 进程，恢复 Application 和 worker 的旧有状态。

1.10.2.2. 配置

基于文件系统的单点恢复，主要是在 spark-env 里对 SPARK_DAEMON_JAVA_OPTS 设置：

系统属性	描述
<code>spark.deploy.recoveryMode</code>	设置文件系统启用单点恢复模式（缺省值：NONE）
<code>spark.deploy.recoveryDirectory</code>	Spark-Master 可进入的文件目录，用于存储作业的恢复状态

1.10.2.3. 细节

- 这个解决方案可以与一个监视器进程（如 [monit](#)）协同运行，或者通过重启来开启手动恢复模式
- 尽管文件系统具有一定的恢复作用，但这种方案对于特定的开发和测试目的未必总是最优的。特别地，通过 stop-master.sh 停止一个 master 并不能清除它的恢复状态，这样每当启用一个新的 master 时，就会进入恢复模式。这将会额外增加达到 1 秒的启动时间来等待所有已经注册的 application 和 worker 超时。
- 尽管并未获得官方支持，但是你也可以增加一个 NFS 目录作为恢复目录。如果原始的 master 节点完全死亡，你就可以在不同的节点上重新启动一个 master，它将会正确地

恢复所有先前注册的 Workers/Applications 信息 (相当于 ZooKeeper 恢复)。 尽管如此，将来新的 application 必须能够找到新的 Master 进行注册。

■ 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、商业案例探询！

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！
- ▶ 如果你是对 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