

PACKT
PUBLISHING

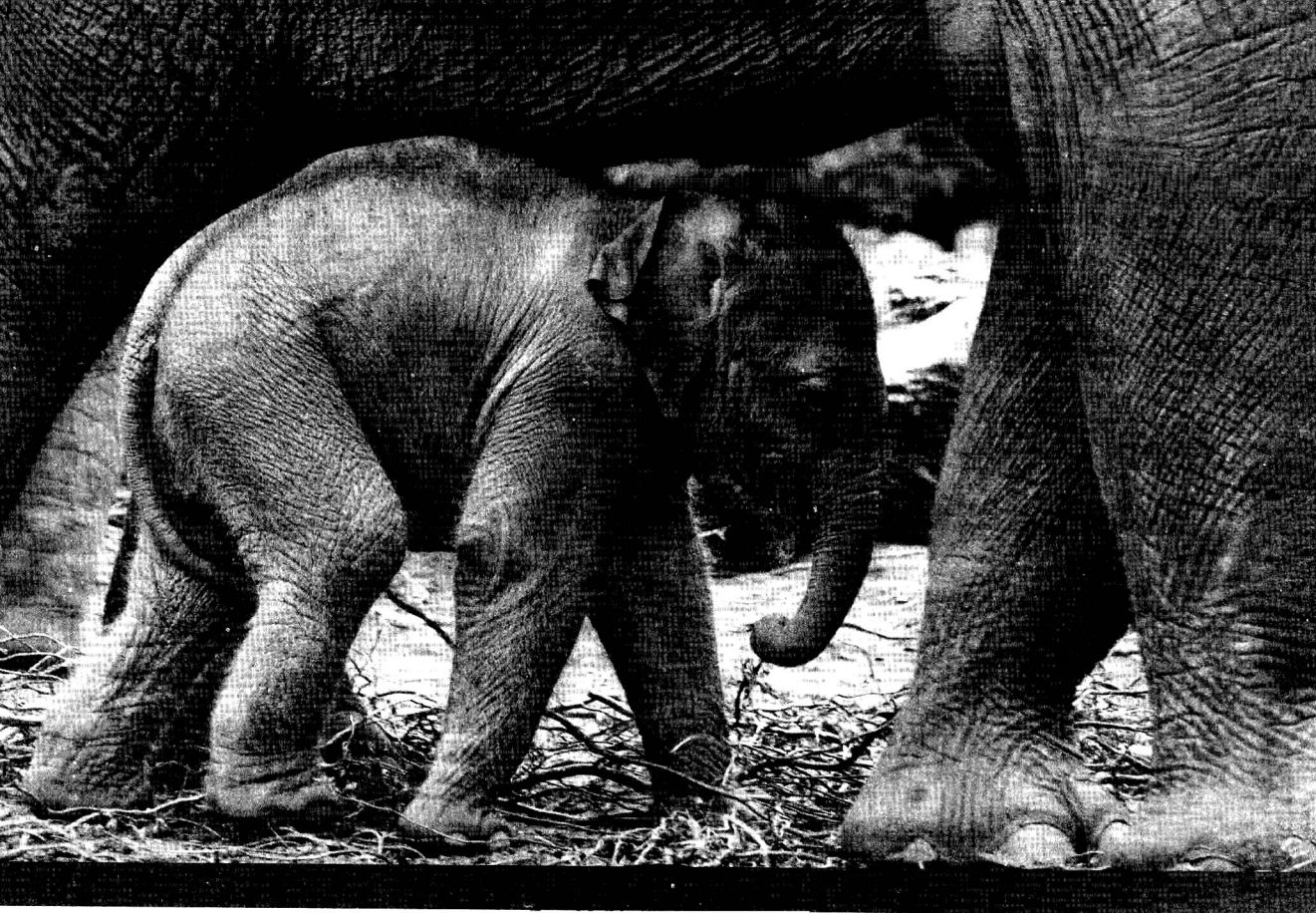
简单、实用的代码示例，快速解决诸多Hadoop相关技术问题

Hadoop 实战手册

Hadoop Real-World Cookbook

[美] Jonathan R. Owens Jon Lentz Brian Femiano 著
傅杰 赵磊 卢学裕 译

 人民邮电出版社
POSTS & TELECOM PRESS



Hadoop 实战手册

[美] Jonathan R. Owens Jon Lentz Brian Femiano 著
傅杰 赵磊 卢学裕 译

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Hadoop实战手册 / (美) 欧文斯 (Owens, J. R.) ,
(美) 伦茨 (Lentz, J.) , (美) 费米亚诺 (Femiano, B.)
著 ; 傅杰, 赵磊, 卢学裕译. — 北京 : 人民邮电出版
社, 2014. 3

书名原文: Hadoop real-world solutions cookbook
ISBN 978-7-115-33795-5

I. ①H… II. ①欧… ②伦… ③费… ④傅… ⑤赵…
⑥卢… III. ①数据处理软件—技术手册 IV.
①TP274-62

中国版本图书馆CIP数据核字(2013)第277092号

版权声明

Copyright ©2013 Packt Publishing. First published in the English language under the title *Hadoop Real-World Solutions Book*.

All Rights Reserved.

本书由英国 Packt Publishing 公司授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式或任何手段复制和传播。

版权所有, 侵权必究。

-
- ◆ 著 [美] Jonathan R. Owens Jon Lentz Brian Femiano
译 傅杰 赵磊 卢学裕
责任编辑 杨海玲
责任印制 程彦红 杨林杰
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京天宇星印刷厂印刷
 - ◆ 开本: 800×1000 1/16
印张: 16.25
字数: 318 千字 2014 年 3 月第 1 版
印数: 1-3 500 册 2014 年 3 月北京第 1 次印刷
著作权合同登记号 图字: 01-2013-4468 号

定价: 59.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316
反盗版热线: (010)81055315

前言

本书能帮助开发者更方便地使用 Hadoop，从而熟练地解决问题。读者会更加熟悉 Hadoop 相关的各种工具从而进行最佳的实践。

本书指导读者使用各种工具解决各种问题。这些工具包括：Apache Hive、Pig、MapReduce、Mahout、Giraph、HDFS、Accumulo、Redis 以及 Ganglia。

本书提供了深入的解释以及代码实例。每章的内容包含一组问题集的描述，并对面临的技术挑战提出了解决方案，最后完整地解决了这些问题。每节将单一问题分解成不同的步骤，这样更容易按照步骤执行相关操作。本书覆盖的内容包括：关于 HDFS 的导入、导出数据，使用 Giraph 进行图分析，使用 Hive、Pig 以及 MapReduce 进行批量数据分析，使用 Mahout 进行机器学习方法，调试并修改 MapReduce 作业的错误，使用 Apache Accumulo 对结构数据进行列存储与检索。

本书的示例中涉及的 Hadoop 技术同样也可以应用于读者自己所面对的问题。

本书涵盖哪些内容

第 1 章“Hadoop 分布式文件系统——导入和导出数据”，包含了从一些流行的数据库导入导出数据的方法，包括 MySQL、MongoDB、Greenplum 以及 MSSQL Server。此外，还包括一些辅助工具，例如 Pig、Flume 以及 Sqoop。

第 2 章“HDFS”，介绍从 HDFS 读入或写出数据，介绍了如何使用不同的序列化库，包含 Avro、Thrift 以及 Protocol Buffers。同样包含如何设置数据块大小、备份数以及是否

需要进行 LZO 压缩。

第 3 章“抽取和转换数据”，包含对不同数据源类型进行基本的 Hadoop ETL 操作。不同的工具包括 Hive、Pig 以及 MapReduce JAVA API，用于批量处理数据，输出一份或多份转换数据。

第 4 章“使用 Hive、Pig 以及 MapReduce 处理常见的任务”，关注于如何利用这些工具提供的函数快速解决不同种类的问题。其中包括字符串连接，外部表的映射，简单表的连接，自定义函数以及基于集群进行分发操作。

第 5 章“高级连接操作”，介绍了 MapReduce、Hive 以及 Pig 中复杂而有效的连接技术。章节的内容包括 Pig 中的归并连接、复制连接以及倾斜连接。同样也包含 Hive 的 map 端连接以及全外连接的内容。同时本章也包括对于外部数据存储，如何使用 Redis 进行数据连接。

第 6 章“大数据分析”，介绍了如何使用 Hadoop 解答关于你的不同数据的各种查询。一些关于 Hive 的例子将展示在不同的分析中如何正确地实现并重复使用用户自定义函数 (UDF)。此外其中有关于 Pig 的两小节，介绍了对于 Audioscrobbler 数据集的各类分析。关于 MapReduce Java API 的一小节介绍了 Combiner 的使用。

第 7 章“高级大数据分析”，展示了使用 Apache Giraph 以及 Mahout 处理不同类型的图分析和机器学习问题。

第 8 章“调试”，帮助你定位并测试 MapReduce 作业。这些例子使用 MRUnit 以及更利于测试的本地模式。此外还强调了使用 counter 以及更新任务状态的重要性，这样做有助于监控 MapReduce 作业。

第 9 章“系统管理”，主要关注如何性能调优 Hadoop 中不同的配置项。下面的内容包含在内：基本的初始化，XML 配置项调整，定位坏数据节点，处理 NameNode 故障，使用 Ganglia 进行性能监控。

第 10 章“使用 Apache Accumulo 进行持久化”，展示了使用 NoSQL 数据存储 Apache Accumulo 带来的很多特性和功能。这些章节利用了这些独特的特性，包括 iterator、combiner、扫描授权以及约束，同时也给出了对于有效建立地理空间行键值以及使用 MapReduce 执行批量分析的例子。

阅读需要的准备

读者需要访问一个伪分布式（单台机器）或完全分布式（多台机器）的集群用于执行

本书中的例子。章节中使用的各种工具需要在集群上安装并正确地配置。此外，本书提供的代码使用不同的语言编写，因此读者能访问的机器最好安装了合适的开发工具。

读者范围

本书使用简要的代码作为例子，展示了可以使用 Hadoop 解决的各类现实中的问题。本书的目的是使不同水平的开发者都能方便地使用 Hadoop 及其工具。Hadoop 的初学者能通过本书加速学习进度并了解现实中 Hadoop 应用的例子。对于有更多经验的 Hadoop 开发者，通过本书，会对许多工具以及技术有新的认识或有一个更清晰的框架，这些东西之前可能你只听说过但并没有真正理解其中的价值。

书中的约定

在你阅读本书时，你会发现书中有各种样式的文本，这些不同样式的文本是用来区分不同类型的信息的。下面是一些不同样式文本的实例，以及相应的说明。

文本中的代码如下所示：“所有的 Hadoop 文件系统 shell 命令行都使用统一的形式 `hadoop fs -COMMAND`。”

代码块如下所示：

```
weblogs = load '/data/weblogs/weblog_entries.txt' as
    (md5:chararray,
     url:chararray,
     date:chararray,
     time:chararray,
     ip:chararray);

md5_grp = group weblogs by md5 parallel 4;

store md5_grp into '/data/weblogs/weblogs_md5_groups.bcp';
```

如果我们希望让代码块中的一特殊部分引起你的注意，相关行或条目会以黑体印刷：

```
weblogs = load '/data/weblogs/weblog_entries.txt' as
    (md5:chararray,
     url:chararray,
     date:chararray,
     time:chararray,
```

```
ip:chararray);
```

```
md5_grp = group weblogs by md5 parallel 4;
```

```
store md5_grp into '/data/weblogs/weblogs_md5_groups.bcp';
```

命令行输入或输出都是以如下样式书写的：

```
hadoop distcp -m 10 hdfs://namenodeA/data/weblogs hdfs://namenodeB/data/weblogs
```

新的术语以及重要文字会以加粗的字体出现。你在屏幕上（如菜单或者对话框中）见到的文字都会是像后面这样出现在正文中：“为了构建 JAR 文件，下载 **Jython Java** 安装程序，并执行该程序，从安装菜单中选择 **Standalone** 选项。”



方框中出现的为警告或重要的注解。



提示或小技巧出现在这里。

读者反馈

我们总是欢迎来自读者的反馈。请告诉我们你觉得这本书怎么样，你喜欢哪些内容，不喜欢哪些内容。读者的反馈对于帮助我们写出那些对读者真正有用的内容至关重要。

如果是给我们反馈一些普通的信息，你可以给 feedback@packtpub.com 这个邮箱发一封邮件即可，记得在你邮件的标题中提及相应的书名。

如果你是某一方面的专家并且对于写作或者撰稿有兴趣的话，你可以访问 www.packtpub.com/authors，读读我们的作者指南。

客户支持

现在你成为了 Packt 出版社的一名尊敬的用户，为使你的购买物超所值，我们为你准备了一系列的东西。

下载本书中的示例代码

你可以登录账号下载到所有从 <http://www.packtpub.com> 购买的 Packt 的书的示例代码文件。如果你从别的地方购买了此书，可以访问 <http://www.packtpub.com/support> 登记信息，文件将通过邮件直接发给你。

勘误

尽管我们已经非常小心谨慎，以确保内容的准确性，错误还是不可避免。如果你在书中发现了错误（这种错误可能是文字或者代码方面的），若你能向我们报告这些错误，我们将感激不尽。这样做，可以使其他读者免受这些错误带来的困扰，帮助改善该书的下一个版本。如果你发现了什么错误，请访问 <http://www.packtpub.com/support>，选择书名，点击“提交错误”链接，然后输入你发现错误的详细内容，通知我们。一旦你指出的错误得到确认，你提交的内容就会被采纳，并加入一个已经存在的勘误列表中。你也可以访问这个链接 <http://www.packtpub.com/support>，选择书名，查看相应书本已有的勘误表。

关于盗版

所有媒体的版权材料在互联网上被盗版是一个日趋严重的问题。在 Packt，我们对于版权与许可证的保护工作是十分看重的。如果你发现任何非法复制我们作品的现象，无论以任何形式，只要在互联网上，请提供给我们对应的网络地址或网站名称，我们会立即对其行为进行纠正。

请通过 copyright@packtpub.com 联系我们，并附上可疑盗版材料的链接。

我们感谢你对保护作者权益的帮助，你的协助同时也保障了 we 带给你更多有价值内容的能力。

如果你有疑问

如果你对书的某些方面有疑问，你可以通过 questions@packtpub.com 联系我们，我们会尽最大的努力为你解答。

目录

第 1 章 Hadoop 分布式文件系统——导入和导出数据	1
1.1 介绍	1
1.2 使用 Hadoop shell 命令导入和导出数据到 HDFS	2
1.3 使用 distcp 实现集群间数据复制	7
1.4 使用 Sqoop 从 MySQL 数据库导入数据到 HDFS	9
1.5 使用 Sqoop 从 HDFS 导出数据到 MySQL	12
1.6 配置 Sqoop 以支持 SQL Server	15
1.7 从 HDFS 导出数据到 MongoDB	17
1.8 从 MongoDB 导入数据到 HDFS	20
1.9 使用 Pig 从 HDFS 导出数据到 MongoDB	23
1.10 在 Greenplum 外部表中使用 HDFS	24
1.11 利用 Flume 加载数据到 HDFS 中	26
第 2 章 HDFS	28
2.1 介绍	28
2.2 读写 HDFS 数据	29
2.3 使用 LZ0 压缩数据	31
2.4 读写序列化文件数据	34
2.5 使用 Avro 序列化数据	37
2.6 使用 Thrift 序列化数据	41
2.7 使用 Protocol Buffers 序列化数据	44
2.8 设置 HDFS 备份因子	48

2.9 设置 HDFS 块大小	49
第 3 章 抽取和转换数据	51
3.1 介绍	51
3.2 使用 MapReduce 将 Apache 日志转换为 TSV 格式	52
3.3 使用 Apache Pig 过滤网络服务器日志中的爬虫访问量	54
3.4 使用 Apache Pig 根据时间戳对网络服务器日志数据排序	57
3.5 使用 Apache Pig 对网络服务器日志进行会话分析	59
3.6 通过 Python 扩展 Apache Pig 的功能	61
3.7 使用 MapReduce 及二次排序计算页面访问量	62
3.8 使用 Hive 和 Python 清洗、转换地理事件数据	67
3.9 使用 Python 和 Hadoop Streaming 执行时间序列分析	71
3.10 在 MapReduce 中利用 MultipleOutputs 输出多个文件	75
3.11 创建用户自定义的 Hadoop Writable 及 InputFormat 读取地理事件数据	78
第 4 章 使用 Hive、Pig 和 MapReduce 处理常见的任务	85
4.1 介绍	85
4.2 使用 Hive 将 HDFS 中的网络日志数据映射为外部表	86
4.3 使用 Hive 动态地为网络日志查询结果创建 Hive 表	87
4.4 利用 Hive 字符串 UDF 拼接网络日志数据的各个字段	89
4.5 使用 Hive 截取网络日志的 IP 字段并确定其对应的国家	92
4.6 使用 MapReduce 对新闻档案数据生成 n-gram	94
4.7 通过 MapReduce 使用分布式缓存查找新闻档案数据中包含关键词的行	98
4.8 使用 Pig 加载一个表并执行包含 GROUP BY 的 SELECT 操作	102
第 5 章 高级连接操作	104
5.1 介绍	104
5.2 使用 MapReduce 对数据进行连接	104
5.3 使用 Apache Pig 对数据进行复制连接	108
5.4 使用 Apache Pig 对有序数据进行归并连接	110
5.5 使用 Apache Pig 对倾斜数据进行倾斜连接	111
5.6 在 Apache Hive 中通过 map 端连接对地理事件进行分析	113
5.7 在 Apache Hive 通过优化的全外连接分析地理事件数据	115

5.8 使用外部键值存储 (Redis) 连接数据	118
第 6 章 大数据分析	123
6.1 介绍	123
6.2 使用 MapReduce 和 Combiner 统计网络日志数据集中的独立 IP 数	124
6.3 运用 Hive 日期 UDF 对地理事件数据集中的时间日期进行转换与排序	129
6.4 使用 Hive 创建基于地理事件数据的每月死亡报告	131
6.5 实现 Hive 用户自定义 UDF 用于确认地理事件数据的来源可靠性	133
6.6 使用 Hive 的 map/reduce 操作以及 Python 标记最长的无暴力发生的时间区间	136
6.7 使用 Pig 计算 Audioscrobbler 数据集中艺术家之间的余弦相似度	141
6.8 使用 Pig 以及 datafu 剔除 Audioscrobbler 数据集中的离群值	145
第 7 章 高级大数据分析	147
7.1 介绍	147
7.2 使用 Apache Giraph 计算 PageRank	147
7.3 使用 Apache Giraph 计算单源最短路径	150
7.4 使用 Apache Giraph 执行分布式宽度优先搜索	158
7.5 使用 Apache Mahout 计算协同过滤	165
7.6 使用 Apache Mahout 进行聚类	168
7.7 使用 Apache Mahout 进行情感分类	171
第 8 章 调试	174
8.1 介绍	174
8.2 在 MapReduce 中使用 Counters 监测异常记录	174
8.3 使用 MRUnit 开发和测试 MapReduce	177
8.4 本地模式下开发和测试 MapReduce	179
8.5 运行 MapReduce 作业跳过异常记录	182
8.6 在流计算作业中使用 Counters	184
8.7 更改任务状态显示调试信息	185
8.8 使用 illustrate 调试 Pig 作业	187
第 9 章 系统管理	189
9.1 介绍	189
9.2 在伪分布模式下启动 Hadoop	189

9.3	在分布式模式下启动 Hadoop	192
9.4	添加一个新节点	195
9.5	节点安全退役	197
9.6	NameNode 故障恢复	198
9.7	使用 Ganglia 监控集群	199
9.8	MapReduce 作业参数调优	201
第 10 章	使用 Apache Accumulo 进行持久化	204
10.1	介绍	204
10.2	在 Accumulo 中设计行键存储地理事件	205
10.3	使用 MapReduce 批量导入地理事件数据到 Accumulo	213
10.4	设置自定义字段约束 Accumulo 中的地理事件数据	220
10.5	使用正则过滤器限制查询结果	225
10.6	使用 SumCombiner 计算同一个键的不同版本的死亡数总和	228
10.7	使用 Accumulo 实行单元级安全的扫描	232
10.8	使用 MapReduce 聚集 Accumulo 中的消息源	237

第 1 章

Hadoop 分布式文件系统—— 导入和导出数据

本章我们将介绍：

- 使用 Hadoop shell 命令导入和导出数据到 HDFS
- 使用 `distcp` 实现集群间数据复制
- 使用 Sqoop 从 MySQL 数据库导入数据到 HDFS
- 使用 Sqoop 从 HDFS 导出数据到 MySQL
- 配置 Sqoop 以支持 SQL Server
- 从 HDFS 导出数据到 MongoDB
- 从 MongoDB 导入数据到 HDFS
- 使用 Pig 从 HDFS 导出数据到 MongoDB
- 在 Greenplum 外部表中使用 HDFS
- 利用 Flume 加载数据到 HDFS 中

1.1 介绍

在一个经典的数据架构中，Hadoop 是处理复杂数据流的核心。数据往往是从许多分散的系统中收集而来，并导入 **Hadoop 分布式文件系统（HDFS）** 中，然后通过 MapReduce 或者其他基于 MapReduce 封装的语言（如 Hive、Pig 和 Cascading 等）进行处理，最后将这些已经过滤、转换和聚合过的结果导出到一个或多个外部系统中。

举个比较具体的例子，一个大型网站可能会做一些关于网站点击率的基础数据分析。从多个服务器中采集页面访问日志，并将其推送到 HDFS 中。启动一个 MapReduce 作业，并将这些数据作为 MapReduce 的输入，接下来数据将被解析、汇总以及与 IP 地址进行关联计算，最终得出 URL、页面访问量和每个 cookie 的地理位置数据。生成的相关结果可以导入关系型数据库中。即席查询 (Ad-hoc query)^①此时就可以构建在这些数据上了。分析师可以快速地生成各种报表数据，例如，当前的独立用户数、用户访问最多的页面、按地区对用户进行拆分及其他的数据汇总。

本章的重点将关注 HDFS 数据的导入与导出，主要内容包含与本地文件系统、关系数据库、NoSQL 数据库、分布式数据库以及其他 Hadoop 集群之间数据的互相导入和导出。

1.2 使用 Hadoop shell 命令导入和导出数据到 HDFS

HDFS 提供了许多 shell 命令来实现访问文件系统的功能，这些命令都是构建在 HDFS FileSystem API 之上的。Hadoop 自带的 shell 脚本是通过命令行来执行所有操作的。这个脚本的名称叫做 `hadoop`，通常安装在 `$HADOOP_BIN` 目录下，其中 `$HADOOP_BIN` 是 Hadoopbin 文件完整的安装目录，同时有必要将 `$HADOOP_BIN` 配置到 `$PATH` 环境变量中，这样所有的命令都可以通过 `hadoop fs -command` 这样的形式来执行。

如果需要获取文件系统的所有命令，可以运行 `hadoop` 命令传递不带参数的选项 `fs`。

`hadoop fs`

```
[cloudera@localhost Desktop]$ hadoop fs
Usage: hadoop fs [generic options]
[-cat [-ignoreCrc] <src> ...]
[-chgrp [-R] GROUP PATH ...]
[-chmod [-R] <MODE>[,<MODE>]... | <OCTALMODE> PATH...]
[-chown [-R] [OWNER][:[GROUP]] PATH...]
[-copyFromLocal <localsrc> ... <dst>]
[-copyToLocal [-ignoreCrc] [-crc] <src> ... <localdst>]
[-count [-q] <path> ...]
[-cp <src> ... <dst>]
[-df [-h] [<path> ...]]
[-du [-s] [-h] <path> ...]
[-expunge]
[-get [-ignoreCrc] [-crc] <src> ... <localdst>]
[-getmerge [-nl] <src> <localdst>]
[-help [cmd ...]]
[-ls [-d] [-h] [-R] [<path> ...]]
[-mkdir [-p] <path> ...]
[-moveFromLocal <localsrc> ... <dst>]
[-moveToLocal <src> <localdst>]
[-mv <src> ... <dst>]
[-put <localsrc> ... <dst>]
[-rm [-f] [-r|-R] [--skiptrash] <src> ...]
```

^① 即席查询是用户根据自己的需求，灵活地选择查询条件，系统能够根据用户的选择生成相应的统计报表。

```

[-rmdir [--ignore-fail-on-non-empty] <dir> ...]
[-setrep [-R] [-w] <rep> <path/file> ...]
[-stat [format] <path> ...]
[-tail [-f] <file>]
[-test [-azd] <path>]
[-text [-ignoreCrc] <src> ...]
[-touchz <path> ...]
[-usage [cmd ...]]

```

这些按照功能进行命名的命令的名称与 Unix shell 命令非常相似。使用 help 选项可以获得某个具体命令的详细说明。

```
hadoop fs -help ls
```

```

[cloudera@localhost Desktop]$ hadoop fs -help ls
-ls [-d] [-h] [-R] [<path> ...]:      List the contents that match the specified file pattern. If
path is not specified, the contents of /usr/<currentUser>
will be listed. Directory entries are of the form
  dirName (full path) <dir>
and file entries are of the form
  fileName(full path) <r n> size
where n is the number of replicas specified for the file
and size is the size of the file, in bytes.
  -d Directories are listed as plain files.
  -h Formats the sizes of files in a human-readable fashion
rather than a number of bytes.
  -R Recursively list the contents of directories.
[cloudera@localhost Desktop]$

```

这些 shell 命令和其简要的参数描述可在官方在线文档 http://hadoop.apache.org/docs/r1.0.4/file_system_shell.html 中进行查阅。

在这一节中，我们将使用 Hadoop shell 命令将数据导入 HDFS 中，以及将数据从 HDFS 中导出。这些命令更多地用于加载数据，下载处理过的数据，管理文件系统，以及预览相关数据。掌握这些命令是高效使用 HDFS 的前提。

准备工作

你需要在 <http://www.packtpub.com/support> 这个网站上下载数据集 `weblog_entries.txt`。

操作步骤

完成以下步骤，实现将 `weblog_entries.txt` 文件从本地文件系统复制到 HDFS 上的一个指定文件夹下。

1. 在 HDFS 中创建一个新文件夹，用于保存 `weblog_entries.txt` 文件：

```
hadoop fs -mkdir /data/weblogs
```

2. 将 `weblog_entries.txt` 文件从本地文件系统复制到 HDFS 刚创建的新文件夹下：

```
hadoop fs -copyFromLocal weblog_entries.txt /data/weblogs
```

3. 列出 HDFS 上 `weblog_entries.txt` 文件的信息：

```
hadoop fs -ls /data/weblogs/weblog_entries.txt
```

```
[cloudera@localhost Desktop]$ hadoop fs -ls /data/weblogs
Found 1 items
-rw-r--r-- 1 cloudera supergroup 254129 2012-12-31 11:06 /data/weblogs/weblog_entries.txt
[cloudera@localhost Desktop]$
```



在 Hadoop 处理的一些结果数据可能会直接被外部系统使用，可能需要其他系统做更进一步的处理，或者 MapReduce 处理框架根本就不符合该场景，任何类似的情况下都需要从 HDFS 上导出数据。下载数据最简单的办法就是使用 Hadoop shell。

4. 将 HDFS 上的 weblog_entries.txt 文件复制到本地系统的当前文件夹下：

```
hadoop fs -copyToLocal /data/weblogs/weblog_entries.txt ./weblog_entries.txt
```

```
[cloudera@localhost data]$ hadoop fs -copyToLocal /data/weblogs/weblog_entries.txt ./weblog_entries.txt
[cloudera@localhost data]$ ls -ltr
total 252
-rwxr-xr-x 1 cloudera cloudera 254129 Dec 31 11:15 weblog_entries.txt
[cloudera@localhost data]$
```

复制 HDFS 的文件到本地文件系统时，需要保证本地文件系统的空间可用以及网络连接的速度。HDFS 中的文件大小在几个 TB 到几十个 TB 是很常见的。在 1Gbit 网络环境下，从 HDFS 中导出 10 TB 数据到本地文件系统，最好的情况下也要消耗 23 个小时，当然这还要保证本地文件系统的空间是可用的。



下载本书的示例代码

你可以从 <http://www.packtpub.com> 下载你买过的任何书的示例代码，如果你买过本书还可以访问 <http://www.packtpub.com/support>，并进行注册来让文件直接发送到你的邮箱

工作原理

Hadoop shell 非常轻量地封装在 HDFS FileSystem API 之上。在执行 hadoop 命令时，如果传进去的参数是 fs，实际上执行的是 org.apache.hadoop.fs.FsShell 这个类。在 0.20.2 版本中 FsShell 实例化了一个 org.apache.hadoop.fs.FileSystem 对象，并且将命令行参数与类方法映射起来。比如，执行 hadoop fs -mkdir /data/weblogs 相当于调用 FileSystem.mkdirs(new Path("/data/weblogs"))。同样，运行 hadoop fs -copyFromLocal weblog_entries.txt /data/weblogs 相当于在调用 FileSystem.copyFromLocal(new Path("weblog_entries.txt"), new Path("/data/weblogs"))。HDFS 数据复制到本地系统的实现方式也是一样，等同于调用 FileSystem.copyToLocal(new Path("/data/weblogs/weblog_entries.txt"), new Path("./weblog_entries.txt"))。更多关于文件系统

的接口信息描述可以见官方文档 <http://hadoop.apache.org/docs/r1.0.4/api/org/apache/hadoop/fs/FileSystem.html>。

`mkdir` 可以通过 `hadoop fs -mkdir PATH1 PATH2` 的形式来执行。例如，`hadoop fs -mkdir /data/weblogs/12012012 /data/weblogs/12022012` 将会在 HDFS 系统上分别创建两个文件夹 `/data/weblogs/12012012` 和 `/data/weblogs/12022012`。如果文件夹创建成功则返回 0，否则返回-1。

```
hadoop fs -mkdir /data/weblogs/12012012 /data/weblogs/12022012
hadoop fs -ls /data/weblogs
```

```
[cloudera@localhost data]$ hadoop fs -mkdir /data/weblogs/12012012 /data/weblogs/12022012
[cloudera@localhost data]$ hadoop fs -ls /data/weblogs
Found 3 items
drwxr-xr-x - cloudera supergroup          0 2012-12-31 11:18 /data/weblogs/12012012
drwxr-xr-x - cloudera supergroup          0 2012-12-31 11:18 /data/weblogs/12022012
-rw-r--r-- 1 cloudera supergroup    254129 2012-12-31 11:06 /data/weblogs/weblog_entries.txt
[cloudera@localhost data]$
```

`copyFromLocal` 可以通过 `hadoop fs -copyFromLocal LOCAL_FILE_PATH URI` 的形式来执行，如果 URI 的地址（指的是 `HDFS://filesystemName:9000` 这个串）没有明确给出，则默认会读取 `core-site.xml` 中的 `fs.default.name` 这个属性。上传成功返回 0，否则返回-1。

`copyToLocal` 命令可以通过 `hadoop fs -copyToLocal [-ignorecrc] [-crc] URILOCAL_FILE_PATH` 的形式来执行。如果 URI 的地址没有明确的给出，则默认会读取 `core-site.xml` 中的 `fs.default.name` 这个属性。`copyToLocal` 会执行 CRC (Cyclic Redundancy Check) 校验保证已复制的数据的正确性，一个失败的副本复制可以通过参数 `-ignorecrc` 来强制执行，还可以通过 `-crc` 参数在复制文件的同时也复制 `crc` 校验文件。

更多参考

`put` 命令与 `copyFromLocal` 类似，`put` 更通用一些，可以复制多个文件到 HDFS 中，也能接受标准输入流。

任何使用 `copyToLocal` 的地方都可以用 `get` 替换，两者的实现一模一样。

在使用 `MapReduce` 处理大数据时，其输出结果可能是一个或者多个文件。最终输出结果的文件个数是由 `mapred.reduce.tasks` 的值决定的。我们可以在 `Jobconf` 类中通过 `setNumReduceTasks()` 方法来设置这个属性，改变提交作业的 `reduce` 个数，每个 `reduce` 将对应输出一个文件。该参数是客户端参数，非集群参数，针对不同的作业应该设置不同的值。其默认值为 1，意味着所有 `Map`（映射函数，以下都用 `Map` 表示）的输出结果都将复