

# 命令手册

## 目录

1 概述.....	2
1.1 常规选项.....	2
2 用户命令 .....	3
2.1 archive .....	3
2.2 distcp .....	3
2.3 fs .....	3
2.4 fsck .....	3
2.5 jar .....	4
2.6 job .....	4
2.7 pipes .....	5
2.8 version .....	6
2.9 CLASSNAME .....	6
3 管理命令.....	6
3.1 balancer .....	6
3.2 daemonlog .....	6
3.3 datanode.....	7
3.4 dfsadmin .....	7
3.5 jobtracker .....	8
3.6 namenode .....	9
3.7 secondarynamenode .....	9
3.8 tasktracker .....	9

## 1. 概述

所有的hadoop命令均由bin/hadoop脚本引发。不指定参数运行hadoop脚本会打印所有命令的描述。

用法: `hadoop [--config confdir] [COMMAND] [GENERIC_OPTIONS]  
[COMMAND_OPTIONS]`

Hadoop有一个选项解析框架用于解析一般的选项和运行类。

命令选项	描述
<code>--config confdir</code>	覆盖缺省配置目录。缺省是 <code>\${HADOOP_HOME}/conf</code> 。
GENERIC_OPTIONS	多个命令都支持的通用选项。
COMMAND 命令选项S	各种各样的命令和它们的选项会在下面提到。这些命令被分为 <a href="#">用户命令</a> <a href="#">管理命令</a> 两组。

### 1.1. 常规选项

下面的选项被 [dfsadmin](#), [fs](#), [fsck](#)和 [job](#)支持。应用程序要实现 [Tool](#)来支持 [常规选项](#)。

GENERIC_OPTION	描述
<code>-conf &lt;configuration file&gt;</code>	指定应用程序的配置文件。
<code>-D &lt;property=value&gt;</code>	为指定property指定值value。
<code>-fs &lt;local namenode:port&gt;</code>	指定namenode。
<code>-jt &lt;local jobtracker:port&gt;</code>	指定job tracker。只适用于 <a href="#">job</a> 。
<code>-files &lt;逗号分隔的文件列表&gt;</code>	指定要拷贝到map reduce集群的文件的逗号分隔的列表。只适用于 <a href="#">job</a> 。
<code>-libjars &lt;逗号分隔的jar列表&gt;</code>	指定要包含到classpath中的jar文件的逗号分隔的列表。只适用于 <a href="#">job</a> 。
<code>-archives &lt;逗号分隔的archive列表&gt;</code>	指定要被解压到计算节点上的档案文件的逗号分割的列表。只适用于 <a href="#">job</a> 。

## 2. 用户命令

hadoop集群用户的常用命令。

### 2.1. archive

创建一个hadoop档案文件。参考 [Hadoop Archives](#)。

用法: `hadoop archive -archiveName NAME <src>* <dest>`

命令选项	描述
<code>-archiveName NAME</code>	要创建的档案的名字。
<code>src</code>	文件系统的路径名，和通常含正则表达的一样。
<code>dest</code>	保存档案文件的目标目录。

### 2.2. distcp

递归地拷贝文件或目录。参考[DistCp指南](#)以获取等多信息。

用法: `hadoop distcp <srcurl> <desturl>`

命令选项	描述
<code>srcurl</code>	源Url
<code>desturl</code>	目标Url

### 2.3. fs

用法: `hadoop fs [GENERIC OPTIONS] [COMMAND\_OPTIONS]`

运行一个常规的文件系统客户端。

各种命令选项可以参考[HDFS Shell指南](#)。

### 2.4. fsck

运行HDFS文件系统检查工具。参考[Fsck](#)了解更多。

用法: `hadoop fsck [GENERIC OPTIONS] <path> [-move | -delete |`

`-openforwrite] [-files [-blocks [-locations | -racks]]]`

命令选项	描述
<code>&lt;path&gt;</code>	检查的起始目录。
<code>-move</code>	移动受损文件到/lost+found
<code>-delete</code>	删除受损文件。
<code>-openforwrite</code>	打印出写打开的文件。
<code>-files</code>	打印出正被检查的文件。
<code>-blocks</code>	打印出块信息报告。
<code>-locations</code>	打印出每个块的位置信息。
<code>-racks</code>	打印出data-node的网络拓扑结构。

## 2.5. jar

运行jar文件。用户可以把他们的Map Reduce代码捆绑到jar文件中，使用这个命令执行。

用法: `hadoop jar <jar> [mainClass] args...`

streaming作业是通过这个命令执行的。参考[Streaming examples](#)中的例子。

Word count例子也是通过jar命令运行的。参考[Wordcount example](#)。

## 2.6. job

用于和Map Reduce作业交互和命令。

用法: `hadoop job [GENERIC OPTIONS] [-submit <job-file>] | [-status <job-id>] | [-counter <job-id> <group-name> <counter-name>] | [-kill <job-id>] | [-events <job-id> <from-event-#> <#-of-events>] | [-history [a11] <jobOutputDir>] | [-list [a11]] | [-kill-task <task-id>] | [-fail-task <task-id>]`

命令选项	描述
<code>-submit &lt;job-file&gt;</code>	提交作业

<code>-status &lt;job-id&gt;</code>	打印map和reduce完成百分比和所有计数器。
<code>-counter &lt;job-id&gt; &lt;group-name&gt; &lt;counter-name&gt;</code>	打印计数器的值。
<code>-kill &lt;job-id&gt;</code>	杀死指定作业。
<code>-events &lt;job-id&gt; &lt;from-event-#&gt; &lt;#-of-events&gt;</code>	打印给定范围内jobtracker接收到的事件细节。
<code>-history [all] &lt;jobOutputDir&gt;</code>	<code>-history &lt;jobOutputDir&gt;</code> 打印作业的细节、失败及被杀死原因的细节。更多的关于一个作业的细节比如成功的任务，做过的任务尝试等信息可以通过指定[all]选项查看。
<code>-list [all]</code>	<code>-list all</code> 显示所有作业。 <code>-list</code> 只显示将要完成的作业。
<code>-kill-task &lt;task-id&gt;</code>	杀死任务。被杀死的任务不会不利于失败尝试。
<code>-fail-task &lt;task-id&gt;</code>	使任务失败。被失败的任务会对失败尝试不利。

## 2.7. pipes

运行pipes作业。

用法: `hadoop pipes [-conf <path>] [-jobconf <key=value>, <key=value>, ...]  
[-input <path>] [-output <path>] [-jar <jar file>] [-inputformat <class>]  
[-map <class>] [-partitioner <class>] [-reduce <class>] [-writer <class>]  
[-program <executable>] [-reduces <num>]`

命令选项	描述
<code>-conf &lt;path&gt;</code>	作业的配置
<code>-jobconf &lt;key=value&gt;, &lt;key=value&gt;, ...</code>	增加/覆盖作业的配置项
<code>-input &lt;path&gt;</code>	输入目录
<code>-output &lt;path&gt;</code>	输出目录
<code>-jar &lt;jar file&gt;</code>	Jar文件名
<code>-inputformat &lt;class&gt;</code>	InputFormat类
<code>-map &lt;class&gt;</code>	Java Map类

-partitioner <class>	Java Partitioner
-reduce <class>	Java Reduce类
-writer <class>	Java RecordWriter
-program <executable>	可执行程序URI
-reduces <num>	reduce个数

## 2.8. version

打印版本信息。

用法: `hadoop version`

## 2.9. CLASSNAME

hadoop脚本可用于调调用任何类。

用法: `hadoop CLASSNAME`

运行名字为CLASSNAME的类。

## 3. 管理命令

hadoop集群管理员常用的命令。

### 3.1. balancer

运行集群平衡工具。管理员可以简单的按Ctrl-C来停止平衡过程。参考[Rebalancer](#)了解更多。

用法: `hadoop balancer [-threshold <threshold>]`

命令选项	描述
-threshold <threshold>	磁盘容量的百分比。这会覆盖缺省的阈值。

### 3.2. daemonlog

获取或设置每个守护进程的日志级别。

用法: `hadoop daemonlog -getlevel <host:port> <name>`

用法: `hadoop daemonlog -setlevel <host:port> <name> <level>`

命令选项	描述
<code>-getlevel &lt;host:port&gt; &lt;name&gt;</code>	打印运行在<host:port>的守护进程的日志级别。这个命令内部会连接 <code>http://&lt;host:port&gt;/logLevel?log=&lt;name&gt;</code>
<code>-setlevel &lt;host:port&gt; &lt;name&gt; &lt;level&gt;</code>	设置运行在<host:port>的守护进程的日志级别。这个命令内部会连接 <code>http://&lt;host:port&gt;/logLevel?log=&lt;name&gt;</code>

### 3.3. datanode

运行一个HDFS的datanode。

用法: `hadoop datanode [-rollback]`

命令选项	描述
<code>-rollback</code>	将datanode回滚到前一个版本。这需要在停止datanode, 分发老的hadoop版本之后使用。

### 3.4. dfsadmin

运行一个HDFS的dfsadmin客户端。

用法: `hadoop dfsadmin [GENERIC OPTIONS] [-report] [-safemode enter | leave | get | wait] [-refreshNodes] [-finalizeUpgrade] [-upgradeProgress status | details | force] [-metasave filename] [-setQuota <quota> <dirname>...<dirname>] [-clrQuota <dirname>...<dirname>] [-help [cmd]]`

命令选项	描述
<code>-report</code>	报告文件系统的基本信息和统计信息。
<code>-safemode enter   leave   get   wait</code>	安全模式维护命令。安全模式是Namenode的一个状态, 这种状态下, Namenode <ol style="list-style-type: none"> <li>1. 不接受对名字空间的更改(只读)</li> <li>2. 不复制或删除块</li> </ol> Namenode会在启动时自动进入安全模式, 当配置的块最小百分比数满足最小的副本数条件时, 会自动离开安全模式。安全模式可以手动进入, 但

	是这样的话也必须手动关闭安全模式。
<code>-refreshNodes</code>	重新读取hosts和exclude文件，更新允许连到Namenode的或那些需要退出或入编的Datanode的集合。
<code>-finalizeUpgrade</code>	终结HDFS的升级操作。Datanode删除前一个版本的工作目录，之后Namenode也这样做。这个操作完结整个升级过程。
<code>-upgradeProgress status   details   force</code>	请求当前系统的升级状态，状态的细节，或者强制升级操作进行。
<code>-metasave filename</code>	保存Namenode的主要数据结构到hadoop.log.dir属性指定的目录下的<filename>文件。对于下面的每一项，<filename>中都会一行内容与之对应 <ol style="list-style-type: none"> <li>1. Namenode收到的Datanode的心跳信号</li> <li>2. 等待被复制的块</li> <li>3. 正在被复制的块</li> <li>4. 等待被删除的块</li> </ol>
<code>-setQuota &lt;quota&gt; &lt;dirname&gt;...&lt;dirname&gt;</code>	为每个目录 <dirname>设定配额<quota>。目录配额是一个长整型整数，强限制了目录树下的名字个数。 命令会在这个目录上工作良好，以下情况会报错： <ol style="list-style-type: none"> <li>1. N不是一个正整数，或者</li> <li>2. 用户不是管理员，或者</li> <li>3. 这个目录不存在或是文件，或者</li> <li>4. 目录会马上超出新设定的配额。</li> </ol>
<code>-clrQuota &lt;dirname&gt;...&lt;dirname&gt;</code>	为每一个目录<dirname>清除配额设定。 命令会在这个目录上工作良好，以下情况会报错： <ol style="list-style-type: none"> <li>1. 这个目录不存在或是文件，或者</li> <li>2. 用户不是管理员。</li> </ol> 如果目录原来没有配额不会报错。
<code>-help [cmd]</code>	显示给定命令的帮助信息，如果没有给定命令，则显示所有命令的帮助信息。

### 3.5. jobtracker

运行MapReduce job Tracker节点。

用法: `hadoop jobtracker`

### 3.6. namenode

运行namenode。有关升级，回滚，升级终结的更多信息请参考[升级和回滚](#)。

用法: `hadoop namenode [-format] | [-upgrade] | [-rollback] | [-finalize] | [-importCheckpoint]`

命令选项	描述
<code>-format</code>	格式化namenode。它启动namenode，格式化namenode，之后关闭namenode。
<code>-upgrade</code>	分发新版本的hadoop后，namenode应以upgrade选项启动。
<code>-rollback</code>	将namenode回滚到前一版本。这个选项要在停止集群，分发老的hadoop版本后使用。
<code>-finalize</code>	finalize会删除文件系统的前一状态。最近的升级会被持久化，rollback选项将再不可用，升级终结操作之后，它会停掉namenode。
<code>-importCheckpoint</code>	从检查点目录装载镜像并保存到当前检查点目录，检查点目录由fs.checkpoint.dir指定。

### 3.7. secondarynamenode

运行HDFS的secondary namenode。参考[Secondary Namenode](#)了解更多。

用法: `hadoop secondarynamenode [-checkpoint [force]] | [-geteditsize]`

命令选项	描述
<code>-checkpoint [force]</code>	如果EditLog的大小 $\geq$ fs.checkpoint.size，启动Secondary namenode的检查点过程。如果使用了-force，将不考虑EditLog的大小。
<code>-geteditsize</code>	打印EditLog大小。

### 3.8. tasktracker

运行MapReduce的task Tracker节点。

用法: `hadoop tasktracker`