

# HDFS 原理及操作

本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，博主为石山园，博客地址为 <http://www.cnblogs.com/shishanyuan> 。该系列课程是应邀实验楼整理编写的，这里需要赞一下实验楼提供了学习的新方式，可以边看博客边上机实验，课程地址为 <https://www.shiyanlou.com/courses/237>

【注】该系列所使用到安装包、测试数据和代码均可在百度网盘下载，具体地址为 <http://pan.baidu.com/s/10PnDs>，下载该 PDF 文件

## 1 环境说明

部署节点操作系统为 CentOS，防火墙和 SELinux 禁用，创建了一个 shiyanlou 用户并在系统根目录下创建/app 目录，用于存放 Hadoop 等组件运行包。因为该目录用于安装 hadoop 等组件程序，用户对 shiyanlou 必须赋予 rwx 权限（一般做法是 root 用户在根目录下创建/app 目录，并修改该目录拥有者为 shiyanlou(chown -R shiyanlou:shiyanlou /app)）。

### Hadoop 搭建环境：

- 虚拟机操作系统：CentOS6.6 64 位，单核，1G 内存
- JDK：1.7.0\_55 64 位
- Hadoop：1.1.2

## 2 HDFS 原理

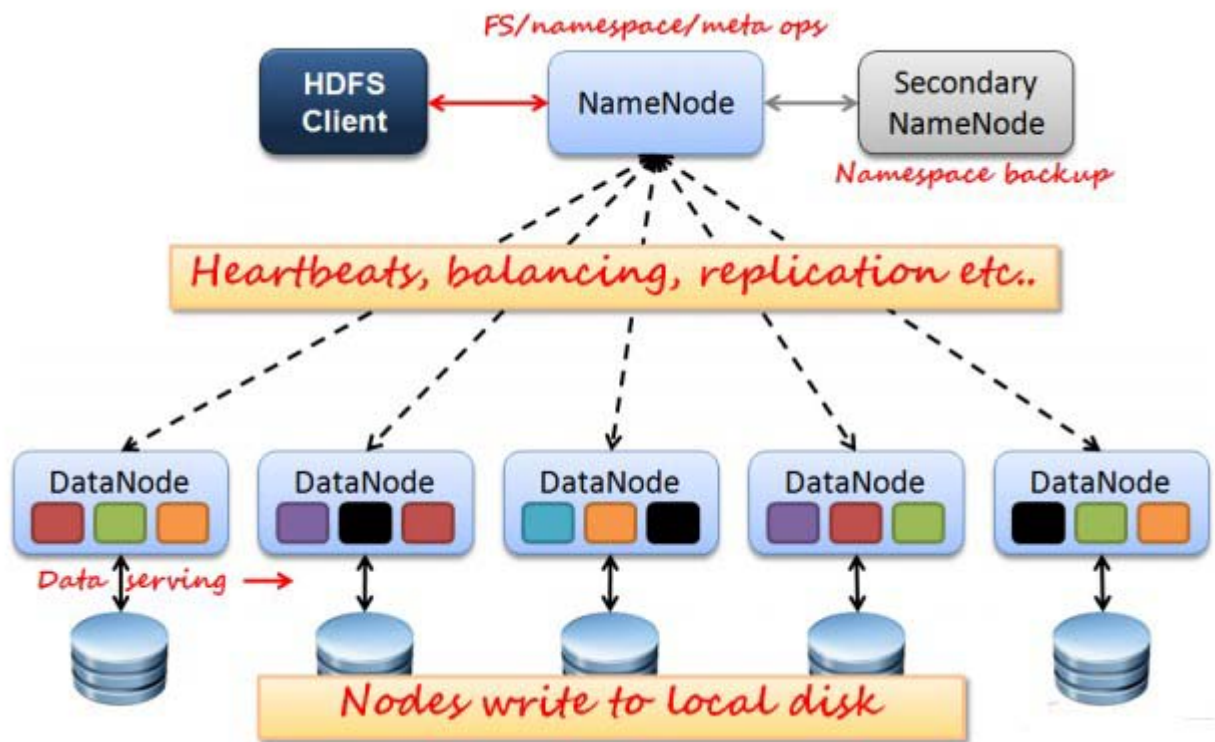
HDFS ( Hadoop Distributed File System ) 是一个分布式文件系统，是谷歌的 GFS 山寨版本。它具有高容错性并提供了高吞吐量的数据访问，非常适合大规模数据集上的应用，它提供了一个高度容错性和高吞吐量的海量数据存储解决方案。

- 高吞吐量访问：HDFS 的每个 Block 分布在不同的 Rack 上，在用户访问时，HDFS 会计算使用最近和访问量最小的服务器给用户。由于 Block 在不同的 Rack 上都有备份，所以不再是单数据访问，所以速度和效率是非常快的。另外 HDFS 可以并行从服务器集群中读写，增加了文件读写的访问带宽。
- 高容错性：系统故障是不可避免的，如何做到故障之后的数据恢复和容错处理是至关重要的。HDFS 通过多方面保证数据的可靠性，多份复制并且分布到物理位置的不同服务器上，数据

校验功能、后台的连续自检数据一致性功能都为高容错提供了可能。

- 线性扩展 因为 HDFS 的 Block 信息存放到 NameNode 上 ,文件的 Block 分布到 DataNode 上 ,当扩充的时候仅仅添加 DataNode 数量 ,系统可以在不停止服务的情况下做扩充 ,不需要人工干预。

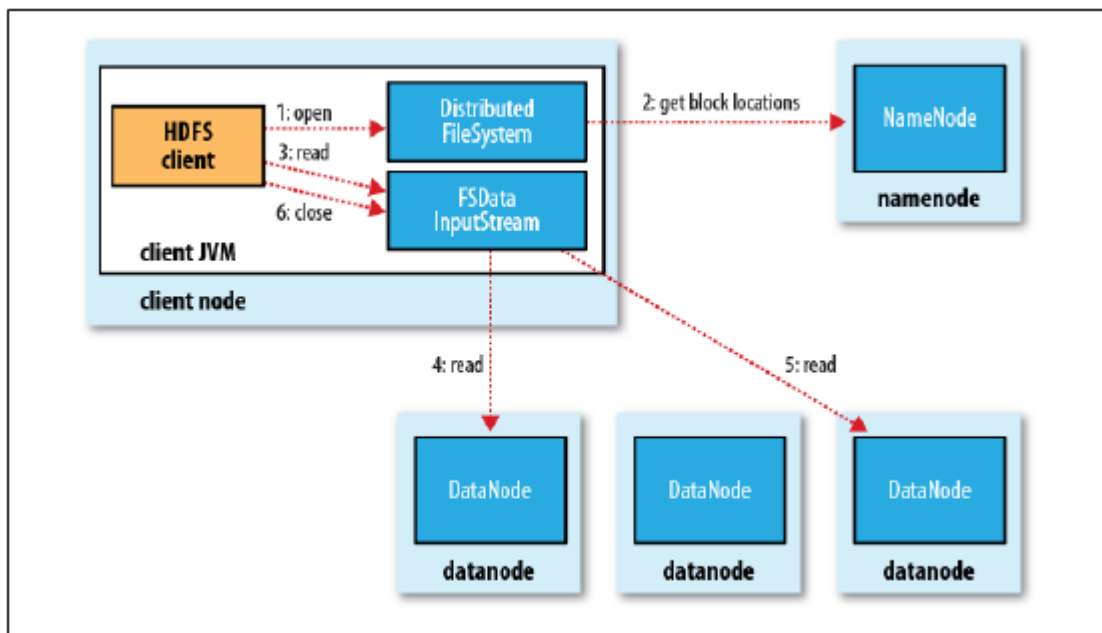
## 2.1 HDFS 架构



如上图所示 HDFS 是 Master 和 Slave 的结构 ,分为 NameNode、Secondary NameNode 和 DataNode 三种角色。

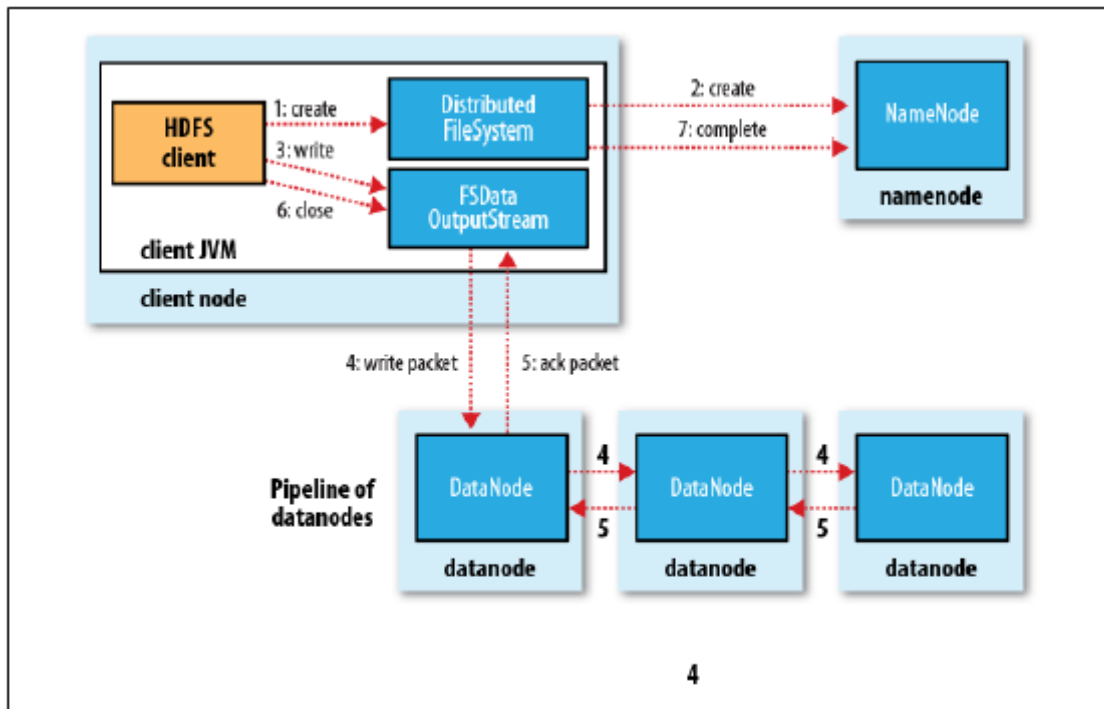
- NameNode :在 Hadoop1.X 中只有一个 Master 节点 ,管理 HDFS 的名称空间和数据块映射信息、配置副本策略和处理客户端请求 ;
- Secondary NameNode :辅助 NameNode ,分担 NameNode 工作 ,定期合并 fsimage 和 fsedits 并推送给 NameNode ,紧急情况下可辅助恢复 NameNode ;
- DataNode :Slave 节点 ,实际存储数据、执行数据块的读写并汇报存储信息给 NameNode ;

## 2.2 HDFS 读操作



1. 客户端通过调用 FileSystem 对象的 open() 方法来打开希望读取的文件, 对于 HDFS 来说, 这个对象是分布文件系统的实例;
2. DistributedFileSystem 通过使用 RPC 来调用 NameNode 以确定文件起始块的位置, 同一 Block 按照重复数会返回多个位置, 这些位置按照 Hadoop 集群拓扑结构排序, 距离客户端近的排在前面;
3. 前两步会返回一个 FSDataInputStream 对象, 该对象会被封装成 DFSInputStream 对象, DFSInputStream 可以方便的管理 datanode 和 namenode 数据流, 客户端对这个输入流调用 read() 方法;
4. 存储着文件起始块的 DataNode 地址的 DFSInputStream 随即连接距离最近的 DataNode, 通过对数据流反复调用 read() 方法, 可以将数据从 DataNode 传输到客户端;
5. 到达块的末端时, DFSInputStream 会关闭与该 DataNode 的连接, 然后寻找下一个块的最佳 DataNode, 这些操作对客户端来说是透明的, 客户端的角度看来只是读一个持续不断的流;
6. 一旦客户端完成读取, 就对 FSDataInputStream 调用 close() 方法关闭文件读取。

## 2.3 HDFS 写操作



1. 客户端通过调用 DistributedFileSystem 的 create()方法创建新文件；
2. DistributedFileSystem 通过 RPC 调用 NameNode 去创建一个没有 Blocks 关联的新文件，创建前 NameNode 会做各种校验，比如文件是否存在、客户端有无权限去创建等。如果校验通过，NameNode 会为创建新文件记录一条记录，否则就会抛出 IO 异常；
3. 前两步结束后会返回 FSDataOutputStream 的对象，和读文件的时候相似，FSDataOutputStream 被封装成 DFSOutputStream，DFSOutputStream 可以协调 NameNode 和 Datanode。客户端开始写数据到 DFSOutputStream，DFSOutputStream 会把数据切成一个个小的数据包，并写入内部队列称为“数据队列”(Data Queue)；
4. DataStreamer 会去处理接受 Data Queue，它先询问 NameNode 这个新的 Block 最适合存储的在哪几个 DataNode 里，比如重复数是 3 那么就找到 3 个最适合的 DataNode，把他们排成一个 pipeline.DataStreamer 把 Packet 按队列输出到管道的第一个 Datanode 中，第一个 DataNode 又把 Packet 输出到第二个 DataNode 中，以此类推；
5. DFSOutputStream 还有一个对列叫 Ack Quene，也是有 Packet 组成，等待 DataNode 的收到响应，当 Pipeline 中的所有 DataNode 都表示已经收到的时候，这时 Akc Quene 才会把对应的 Packet 包移除掉；
6. 客户端完成写数据后调用 close()方法关闭写入流；
7. DataStreamer 把剩余的包都刷到 Pipeline 里然后等待 Ack 信息，收到最后一个 Ack 后，通知 NameNode 把文件标示为已完成。

## 2.4 HDFS 中常用到的命令

- `hadoop fs`

*`hadoop fs -ls /`*

*`hadoop fs -lsr`*

*`hadoop fs -mkdir /user/hadoop`*

*`hadoop fs -put a.txt /user/hadoop/`*

*`hadoop fs -get /user/hadoop/a.txt /`*

*`hadoop fs -cp src dst`*

*`hadoop fs -mv src dst`*

*`hadoop fs -cat /user/hadoop/a.txt`*

*`hadoop fs -rm /user/hadoop/a.txt`*

*`hadoop fs -rmr /user/hadoop/a.txt`*

*`hadoop fs -text /user/hadoop/a.txt`*

*`hadoop fs -copyFromLocal localsrc dst` 与 `hadoop fs -put` 功能类似。*

*`hadoop fs -moveFromLocal localsrc dst` 将本地文件上传到 hdfs , 同时删除本地文件。*

- `hadoop fsadmin`

*`hadoop dfsadmin -report`*

*`hadoop dfsadmin -safemode enter | leave | get | wait`*

*`hadoop dfsadmin -setBalancerBandwidth 1000`*

- `hadoop fsck`

- `start-balancer.sh`

相关 HDFS API 可以到 Apache 官网进行查看：

org.apache.hadoop.fs.FSDataOutputStream	<a href="#">createInternal</a> (org.apache.hadoop.fs.Path f, EnumSet<org.apache.hadoop.fs.CreateFlag> createFlag, org.apache.hadoop.fs.permission.FsPermission absolutePer, int bufferSize, short replication, long blockSize, org.apache.hadoop.util.Progressable progress, int bytesPerChecksum, boolean createParent)
void	<a href="#">createSymlink</a> (org.apache.hadoop.fs.Path target, org.apache.hadoop.fs.Path link, boolean createParent)
boolean	<a href="#">delete</a> (org.apache.hadoop.fs.Path f, boolean recursive)
org.apache.hadoop.fs.BlockLocation[]	<a href="#">getFileBlockLocations</a> (org.apache.hadoop.fs.Path p, long start, long len)
org.apache.hadoop.fs.FileChecksum	<a href="#">getFileChecksum</a> (org.apache.hadoop.fs.Path f)
org.apache.hadoop.fs.FileStatus	<a href="#">getFileLinkStatus</a> (org.apache.hadoop.fs.Path f)
org.apache.hadoop.fs.FileStatus	<a href="#">getFileStatus</a> (org.apache.hadoop.fs.Path f)
org.apache.hadoop.fs.FsStatus	<a href="#">getFsStatus</a> ()
org.apache.hadoop.fs.Path	<a href="#">getLinkTarget</a> (org.apache.hadoop.fs.Path p)
org.apache.hadoop.fs.FsServerDefaults	<a href="#">getServerDefaults</a> ()
int	<a href="#">getUriDefaultPort</a> ()

## 3 测试例子 1

### 3.1 测试例子 1 内容

在 Hadoop 集群中编译并运行《权威指南》中的例 3.2，读取 HDFS 文件内容。

### 3.2 运行代码

```
import java.io.InputStream;

import java.net.URI;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.*;
import org.apache.hadoop.io.IOUtils;

public class FileSystemCat {
    public static void main(String[] args) throws Exception {
        String uri = args[0];
        Configuration conf = new Configuration();
        FileSystem fs = FileSystem.get(URI.create(uri), conf);
        InputStream in = null;
        try {
            in = fs.open(new Path(uri));
            IOUtils.copyBytes(in, System.out, 4096, false);
        }
    }
}
```

```
    } finally {
        IOUtils.closeStream(in);
    }
}
}
```

## 3.3 实现过程

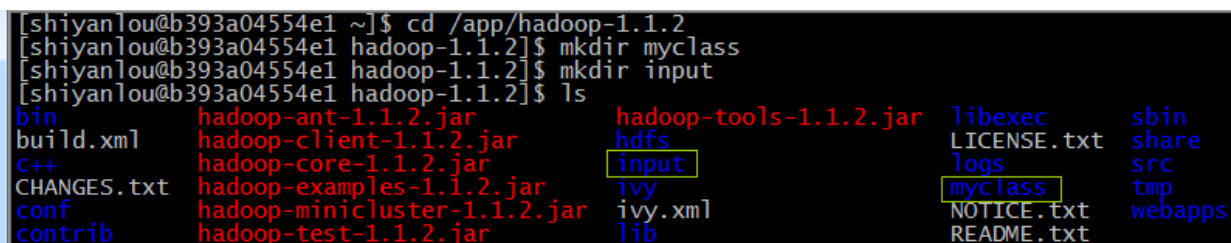
### 3.3.1 创建代码目录

使用如下命令启动 Hadoop

```
cd /app/hadoop-1.1.2/bin
./start-all.sh
```

在/app/hadoop-1.1.2 目录下使用如下命令建立 myclass 和 input 目录：

```
cd /app/hadoop-1.1.2
mkdir myclass
mkdir input
```



```
[shiyanolou@b393a04554e1 ~]$ cd /app/hadoop-1.1.2
[shiyanolou@b393a04554e1 hadoop-1.1.2]$ mkdir myclass
[shiyanolou@b393a04554e1 hadoop-1.1.2]$ mkdir input
[shiyanolou@b393a04554e1 hadoop-1.1.2]$ ls
bin                hadoop-ant-1.1.2.jar          hadoop-tools-1.1.2.jar  libexec            sbin
build.xml          hadoop-client-1.1.2.jar      hdfs                    LICENSE.txt        share
c++                hadoop-core-1.1.2.jar        input                    logs               src
CHANGES.txt      hadoop-examples-1.1.2.jar   ivy                      myclass            tmp
conf               hadoop-minicluster-1.1.2.jar ivy.xml                  NOTICE.txt        webapps
contrib           hadoop-test-1.1.2.jar        lib                       README.txt
```

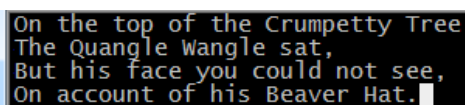
### 3.3.2 建立例子文件上传到 HDFS 中

进入/app/hadoop-1.1.2/input 目录，在该目录中建立 quangle.txt 文件

```
cd /app/hadoop-1.1.2/input
touch quangle.txt
vi quangle.txt
```

内容为：

```
On the top of the Crumpeppy Tree
The Quangle Wangle sat,
But his face you could not see,
On account of his Beaver Hat.
```



```
On the top of the Crumpeppy Tree
The Quangle Wangle sat,
But his face you could not see,
On account of his Beaver Hat.█
```

使用如下命令在 hdfs 中建立目录/class4



```
hadoop fs -mkdir /class4
```

```
hadoop fs -ls /
```

( 如果需要直接使用 hadoop 命令，需要把/app/hadoop-1.1.2 加入到 Path 路径中 )

```
[shiyanolou@b393a04554e1 ~]$ hadoop fs -mkdir /class4
[shiyanolou@b393a04554e1 ~]$ hadoop fs -ls /
Found 2 items
drwxr-xr-x  - shiyanolou supergroup          0 2015-06-01 08:16 /app
drwxr-xr-x  - shiyanolou supergroup          0 2015-06-04 01:40 /class4
[shiyanolou@b393a04554e1 ~]$
```

把例子文件上传到 hdfs 的/class4 文件夹中

```
cd /app/hadoop-1.1.2/input
```

```
hadoop fs -copyFromLocal quangle.txt /class4/quangle.txt
```

```
hadoop fs -ls /class4
```

```
[shiyanolou@b393a04554e1 ~]$ cd /app/hadoop-1.1.2/input
[shiyanolou@b393a04554e1 input]$ hadoop fs -copyFromLocal quangle.txt /class4/quangle.txt
[shiyanolou@b393a04554e1 input]$ hadoop fs -ls /class4
Found 1 items
-rw-r--r--  1 shiyanolou supergroup        119 2015-06-04 01:42 /class4/quangle.txt
[shiyanolou@b393a04554e1 input]$
```

### 3.3.3 配置本地环境

对/app/hadoop-1.1.2/conf 目录中的 hadoop-env.sh 进行配置，如下如所示：

```
cd /app/hadoop-1.1.2/conf
```

```
sudo vi hadoop-env.sh
```

加入对 HADOOP\_CLASSPATH 变量值，值为/app/hadoop-1.1.2/myclass，设置完毕后编译该配置文件，使配置生效

```
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/app/hadoop-1.1.2/myclass
```

```
# The scheduling priority for daemon processes.  See 'man nice'.
# export HADOOP_NICENESS=10

export JAVA_HOME=/app/lib/jdk1.7.0_55
export PATH=$PATH:/app/hadoop-1.1.2/bin
export HADOOP_CLASSPATH=/app/hadoop-1.1.2/myclass
```

### 3.3.4 编写代码

进入/app/hadoop-1.1.2/myclass 目录，在该目录中建立 FileSystemCat.java 代码文件，命令如下：

```
cd /app/hadoop-1.1.2/myclass/
```

```
vi FileSystemCat.java
```

输入代码内容：



```
import java.io.InputStream;
import java.net.URI;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.*;
import org.apache.hadoop.io.IOUtils;

public class FileSystemCat {
    public static void main(String[] args) throws Exception {
        String uri = args[0];
        Configuration conf = new Configuration();
        FileSystem fs = FileSystem.get(URI.create(uri), conf);
        InputStream in = null;
        try {
            in = fs.open(new Path(uri));
            IOUtils.copyBytes(in, System.out, 4096, false);
        } finally {
            IOUtils.closeStream(in);
        }
    }
}
```

### 3.3.5 编译代码

在/app/hadoop-1.1.2/myclass 目录中，使用如下命令编译代码：

```
javac -classpath ../hadoop-core-1.1.2.jar FileSystemCat.java
ls
```

```
[shiyanolou@b393a04554e1 myclass]$ javac -classpath ../hadoop-core-1.1.2.jar FileSystemCat.java
a
[shiyanolou@b393a04554e1 myclass]$ ll
total 8
-rw-rw-r-- 1 shiyanolou shiyanolou 1228 Jun  4 01:47 FileSystemCat.class
-rw-rw-r-- 1 shiyanolou shiyanolou  568 Jun  4 01:47 FileSystemCat.java
[shiyanolou@b393a04554e1 myclass]$
```

### 3.3.6 使用编译代码读取 HDFS 文件

使用如下命令读取 HDFS 中/class4/quangle.txt 内容：

```
hadoop FileSystemCat /class4/quangle.txt
```

```
[shiyanolou@b393a04554e1 conf]$ hadoop FileSystemCat /class4/quangle.txt
On the top of the Crumpetty Tree
The Quangle Wangle sat,
But his face you could not see,
On account of his Beaver Hat.
[shiyanolou@b393a04554e1 conf]$
```

## 4 测试例子 2

### 4.1 测试例子 2 内容

在本地文件系统生成一个大约 100 字节的文本文件，写一段程序读入这个文件并将其第 101-120 字节的内容写入 HDFS 成为一个新文件。

### 4.2 运行代码

**注意：**在编译前请先删除中文注释！

```
import java.io.File;
```

```

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.OutputStream;
import java.net.URI;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataInputStream;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IOUtils;
import org.apache.hadoop.util.Progressable;

public class LocalFile2Hdfs {
    public static void main(String[] args) throws Exception {

        // 获取读取源文件和目标文件位置参数
        String local = args[0];
        String uri = args[1];

        FileInputStream in = null;
        OutputStream out = null;
        Configuration conf = new Configuration();
        try {
            // 获取读入文件数据
            in = new FileInputStream(new File(local));

            // 获取目标文件信息
            FileSystem fs = FileSystem.get(URI.create(uri), conf);
            out = fs.create(new Path(uri), new Progressable() {
                @Override
                public void progress() {
                    System.out.println("*");
                }
            });

            // 跳过前100个字符
            in.skip(100);
            byte[] buffer = new byte[20];

            // 从101的位置读取20个字符到buffer中
            int bytesRead = in.read(buffer);
            if (bytesRead >= 0) {
                out.write(buffer, 0, bytesRead);
            }
        } finally {
            IOUtils.closeStream(in);
            IOUtils.closeStream(out);
        }
    }
}

```

```
    }  
  }  
}
```

## 4.3 实现过程

### 4.3.1 编写代码

进入/app/hadoop-1.1.2/myclass 目录,在该目录中建立 LocalFile2Hdfs.java 代码文件,命令如下:

```
cd /app/hadoop-1.1.2/myclass/  
vi LocalFile2Hdfs.java
```

输入代码内容:

```
import java.io.File;  
import org.apache.hadoop.fs.FSDataInputStream;  
import org.apache.hadoop.fs.FileSystem;  
import org.apache.hadoop.fs.Path;  
import org.apache.hadoop.io.IOUtils;  
import org.apache.hadoop.util.Progressable;  
  
public class LocalFile2Hdfs {  
    public static void main(String[] args) throws Exception {  
  
        String local = args[0];  
        String uri = args[1];  
  
        FileInputStream in = null;  
        OutputStream out = null;  
        Configuration conf = new Configuration();  
        try {  
            in = new FileInputStream(new File(local));  
  
            FileSystem fs = FileSystem.get(URI.create(uri), conf);  
            out = fs.create(new Path(uri), new Progressable() {  
                @Override  
                public void progress() {  
                    System.out.println("#");  
                }  
            });  
  
            in.skip(100);  
            byte[] buffer = new byte[20];  
  
            int bytesRead = in.read(buffer);  
            if (bytesRead >= 0) {  
                out.write(buffer, 0, bytesRead);  
            }  
        } finally {  
            IOUtils.closeStream(in);  
            IOUtils.closeStream(out);  
        }  
    }  
}
```

### 4.3.2 编译代码

在/app/hadoop-1.1.2/myclass 目录中,使用如下命令编译代码:

```
javac -classpath ../hadoop-core-1.1.2.jar LocalFile2Hdfs.java
```

```
[shiyanolou@b393a04554e1 myclass]$ javac -classpath ../hadoop-core-1.1.2.jar LocalFile2Hdfs.java
va
[shiyanolou@b393a04554e1 myclass]$ ll
total 20
-rw-rw-r-- 1 shiyanolou shiyanolou 1228 Jun  4 01:47 FileSystemCat.class
-rw-rw-r-- 1 shiyanolou shiyanolou  568 Jun  4 01:47 FileSystemCat.java
-rw-rw-r-- 1 shiyanolou shiyanolou  566 Jun  4 01:56 LocalFile2Hdfs$1.class
-rw-rw-r-- 1 shiyanolou shiyanolou 1465 Jun  4 01:56 LocalFile2Hdfs.class
-rw-rw-r-- 1 shiyanolou shiyanolou 1126 Jun  4 01:55 LocalFile2Hdfs.java
[shiyanolou@b393a04554e1 myclass]$
```

### 4.3.3 建立测试文件

进入/app/hadoop-1.1.2/input 目录，在该目录中建立 local2hdfs.txt 文件

```
cd /app/hadoop-1.1.2/input/
```

```
vi local2hdfs.txt
```

内容为：

*Washington (CNN) -- Twitter is suing the U.S. government in an effort to loosen restrictions on what the social media giant can say publicly about the national security-related requests it receives for user data.*

*The company filed a lawsuit against the Justice Department on Monday in a federal court in northern California, arguing that its First Amendment rights are being violated by restrictions that forbid the disclosure of how many national security letters and Foreign Intelligence Surveillance Act court orders it receives -- even if that number is zero.*

*Twitter vice president Ben Lee wrote in a blog post that it's suing in an effort to publish the full version of a "transparency report" prepared this year that includes those details. The San Francisco-based firm was unsatisfied with the Justice Department's move in January to allow technological firms to disclose the number of national security-related requests they receive in broad ranges.*

```
Washington (CNN) -- Twitter is suing the U.S. government in an effort to loosen restrictions
on what the social media giant can say publicly about the national security-related requests
it receives for user data.
The company filed a lawsuit against the Justice Department on Monday in a federal court in no
rthern California, arguing that its First Amendment rights are being violated by restrictions
that forbid the disclosure of how many national security letters and Foreign Intelligence Su
rveillance Act court orders it receives -- even if that number is zero.
Twitter vice president Ben Lee wrote in a blog post that it's suing in an effort to publish t
he full version of a "transparency report" prepared this year that includes those details.
The San Francisco-based firm was unsatisfied with the Justice Department's move in January to
allow technological firms to disclose the number of national security-related requests they
receive in broad ranges.
```

### 4.3.4 使用编译代码上传文件内容到 HDFS

使用如下命令读取 local2hdfs 第 101-120 字节的内容写入 HDFS 成为一个新文件：

```
cd /app/hadoop-1.1.2/input
```

```
hadoop LocalFile2Hdfs local2hdfs.txt /class4/local2hdfs_part.txt
```

```
[shiyanolou@b393a04554e1 ~]$ cd /app/hadoop-1.1.2/input
[shiyanolou@b393a04554e1 input]$ hadoop LocalFile2Hdfs local2hdfs.txt /class4/local2hdfs_part.txt
*
[shiyanolou@b393a04554e1 input]$ hadoop fs -ls /class4
Found 2 items
-rw-r--r-- 1 shiyanolou supergroup 20 2015-06-04 02:03 /class4/local2hdfs_part.txt
-rw-r--r-- 1 shiyanolou supergroup 119 2015-06-04 01:42 /class4/quangle.txt
[shiyanolou@b393a04554e1 input]$
```

### 4.3.5 验证是否成功

使用如下命令读取 local2hdfs\_part.txt 内容：

```
hadoop fs -cat /class4/local2hdfs_part.txt
```

```
[shiyanolou@b393a04554e1 ~]$ hadoop fs -ls /class4
Found 2 items
-rw-r--r-- 1 shiyanolou supergroup 20 2015-06-04 02:03 /class4/local2hdfs_part.txt
-rw-r--r-- 1 shiyanolou supergroup 119 2015-06-04 01:42 /class4/quangle.txt
[shiyanolou@b393a04554e1 ~]$ hadoop fs -cat /class4/local2hdfs_part.txt
the social media gi[shiyanolou@b393a04554e1 ~]$
```

## 5 测试例子 3

### 5.1 测试例子 3 内容

测试例子 2 的反向操作，在 HDFS 中生成一个大约 100 字节的文本文件，写一段程序读入这个文件，并将其第 101-120 字节的内容写入本地文件系统成为一个新文件。

### 5.2 程序代码

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.OutputStream;
import java.net.URI;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataInputStream;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IOUtils;

public class Hdfs2LocalFile {
    public static void main(String[] args) throws Exception {

        String uri = args[0];
        String local = args[1];

        FSDataInputStream in = null;
```

```
OutputStream out = null;
Configuration conf = new Configuration();
try {
    FileSystem fs = FileSystem.get(URI.create(uri), conf);
    in = fs.open(new Path(uri));
    out = new FileOutputStream(local);

    byte[] buffer = new byte[20];
    in.skip(100);
    int bytesRead = in.read(buffer);
    if (bytesRead >= 0) {
        out.write(buffer, 0, bytesRead);
    }
} finally {
    IOUtils.closeStream(in);
    IOUtils.closeStream(out);
}
}
```

## 5.3 实现过程

### 5.3.1 编写代码

进入/app/hadoop-1.1.2/myclass 目录,在该目录中建立 Hdfs2LocalFile.java 代码文件,命令如下:

```
cd /app/hadoop-1.1.2/myclass/
vi Hdfs2LocalFile.java
```

输入代码内容:

```

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.OutputStream;
import java.net.URI;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataInputStream;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IOUtils;

public class Hdfs2LocalFile {
    public static void main(String[] args) throws Exception {

        String uri = args[0];
        String local = args[1];

        FSDataInputStream in = null;
        OutputStream out = null;
        Configuration conf = new Configuration();
        try {
            FileSystem fs = FileSystem.get(URI.create(uri), conf);
            in = fs.open(new Path(uri));
            out = new FileOutputStream(local);

            byte[] buffer = new byte[20];
            in.skip(100);
            int bytesRead = in.read(buffer);
            if (bytesRead >= 0) {
                out.write(buffer, 0, bytesRead);
            }
        } finally {
            IOUtils.closeStream(in);
            IOUtils.closeStream(out);
        }
    }
}

```

### 5.3.2 编译代码

在/app/hadoop-1.1.2/myclass 目录中，使用如下命令编译代码：

```
javac -classpath ../hadoop-core-1.1.2.jar Hdfs2LocalFile.java
```

```

[shiyanolou@b393a04554e1 myclass]$ javac -classpath ../hadoop-core-1.1.2.jar Hdfs2LocalFile.java
[shiyanolou@b393a04554e1 myclass]$ ll
total 28
-rw-rw-r-- 1 shiyanolou shiyanolou 1228 Jun  4 01:47 FileSystemCat.class
-rw-rw-r-- 1 shiyanolou shiyanolou  568 Jun  4 01:47 FileSystemCat.java
-rw-rw-r-- 1 shiyanolou shiyanolou 1356 Jun  4 02:16 Hdfs2LocalFile.class
-rw-rw-r-- 1 shiyanolou shiyanolou  961 Jun  4 02:16 Hdfs2LocalFile.java
-rw-rw-r-- 1 shiyanolou shiyanolou  566 Jun  4 01:56 LocalFile2Hdfs$.class
-rw-rw-r-- 1 shiyanolou shiyanolou 1465 Jun  4 01:56 LocalFile2Hdfs.class
-rw-rw-r-- 1 shiyanolou shiyanolou 1126 Jun  4 01:55 LocalFile2Hdfs.java
[shiyanolou@b393a04554e1 myclass]$

```

### 5.3.3 建立测试文件

进入/app/hadoop-1.1.2/input 目录，在该目录中建立 hdfs2local.txt 文件

```
cd /app/hadoop-1.1.2/input/
vi hdfs2local.txt
```

内容为：

*The San Francisco-based firm was unsatisfied with the Justice Department's move in January to allow technological firms to disclose the number of national security-related requests they receive in broad ranges.*



*"It's our belief that we are entitled under the First Amendment to respond to our users' concerns and to the statements of U.S. government officials by providing information about the scope of U.S. government surveillance -- including what types of legal process have not been received," Lee wrote. "We should be free to do this in a meaningful way, rather than in broad, inexact ranges."*

```
The San Francisco-based firm was unsatisfied with the Justice Department's move in January to allow technological firms to disclose the number of national security-related requests they receive in broad ranges. "It's our belief that we are entitled under the First Amendment to respond to our users' concerns and to the statements of U.S. government officials by providing information about the scope of U.S. government surveillance -- including what types of legal process have not been received," Lee wrote. "We should be free to do this in a meaningful way, rather than in broad, inexact ranges."
```

在/app/hadoop-1.1.2/input 目录下把该文件上传到 hdfs 的/class4/文件夹中

```
hadoop fs -copyFromLocal hdfs2local.txt /class4/hdfs2local.txt
```

```
hadoop fs -ls /class4/
```

```
[shiyanolou@b393a04554e1 input]$ ls
hdfs2local.txt  local2hdfs.txt  quangle.txt
[shiyanolou@b393a04554e1 input]$ hadoop fs -copyFromLocal hdfs2local.txt /class4/hdfs2local.txt
[shiyanolou@b393a04554e1 input]$ hadoop fs -ls /class4/
Found 3 items
-rw-r--r--  1 shiyanolou supergroup    601 2015-06-04 02:19 /class4/hdfs2local.txt
-rw-r--r--  1 shiyanolou supergroup     20 2015-06-04 02:03 /class4/local2hdfs_part.txt
-rw-r--r--  1 shiyanolou supergroup    119 2015-06-04 01:42 /class4/quangle.txt
```

### 5.3.4 使用编译代码把文件内容从 HDFS 输出到文件系统中

使用如下命令读取 hdfs2local.txt 第 101-120 字节的内容写入本地文件系统成为一个新文件：

```
hadoop Hdfs2LocalFile /class4/hdfs2local.txt hdfs2local_part.txt
```

```
[shiyanolou@b393a04554e1 input]$ hadoop fs -ls /class4/
Found 3 items
-rw-r--r--  1 shiyanolou supergroup    601 2015-06-04 02:19 /class4/hdfs2local.txt
-rw-r--r--  1 shiyanolou supergroup     20 2015-06-04 02:03 /class4/local2hdfs_part.txt
-rw-r--r--  1 shiyanolou supergroup    119 2015-06-04 01:42 /class4/quangle.txt
[shiyanolou@b393a04554e1 input]$
[shiyanolou@b393a04554e1 input]$ hadoop Hdfs2LocalFile /class4/hdfs2local.txt hdfs2local_part.txt
[shiyanolou@b393a04554e1 input]$
```

### 5.3.5 验证是否成功

使用如下命令读取 hdfs2local\_part.txt 内容：

```
cat hdfs2local_part.txt
```

```
[shiyanolou@b393a04554e1 input]$ ll
total 16
-rw-rw-r--  1 shiyanolou shiyanolou  20 Jun  4 02:21 hdfs2local_part.txt
-rw-rw-r--  1 shiyanolou shiyanolou 601 Jun  4 02:18 hdfs2local.txt
-rw-rw-r--  1 shiyanolou shiyanolou 960 Jun  4 02:01 local2hdfs.txt
-rw-rw-r--  1 shiyanolou shiyanolou 119 Jun  4 01:37 quangle.txt
[shiyanolou@b393a04554e1 input]$ cat hdfs2local_part.txt
technological firms [shiyanolou@b393a04554e1 input]$
```