# **一 HDFS概念**

## **1.1 概念**

**HDFS，它是一个文件系统**，全称：**Hadoop Distributed File System**，用于存储文件通过目录树来定位文件；**其次，它是分布式的**，由很多服务器联合起来实现其功能，集群中的服务器有各自的角色。

## **1.2 组成**

1）HDFS集群包括，NameNode和DataNode以及Secondary Namenode。

2）NameNode负责管理整个文件系统的元数据，以及每一个路径（文件）所对应的数据块信息。

3）DataNode 负责管理用户的文件数据块，每一个数据块都可以在多个datanode上存储多个副本。

4）Secondary NameNode用来监控HDFS状态的辅助后台程序，每隔一段时间获取HDFS元数据的快照。

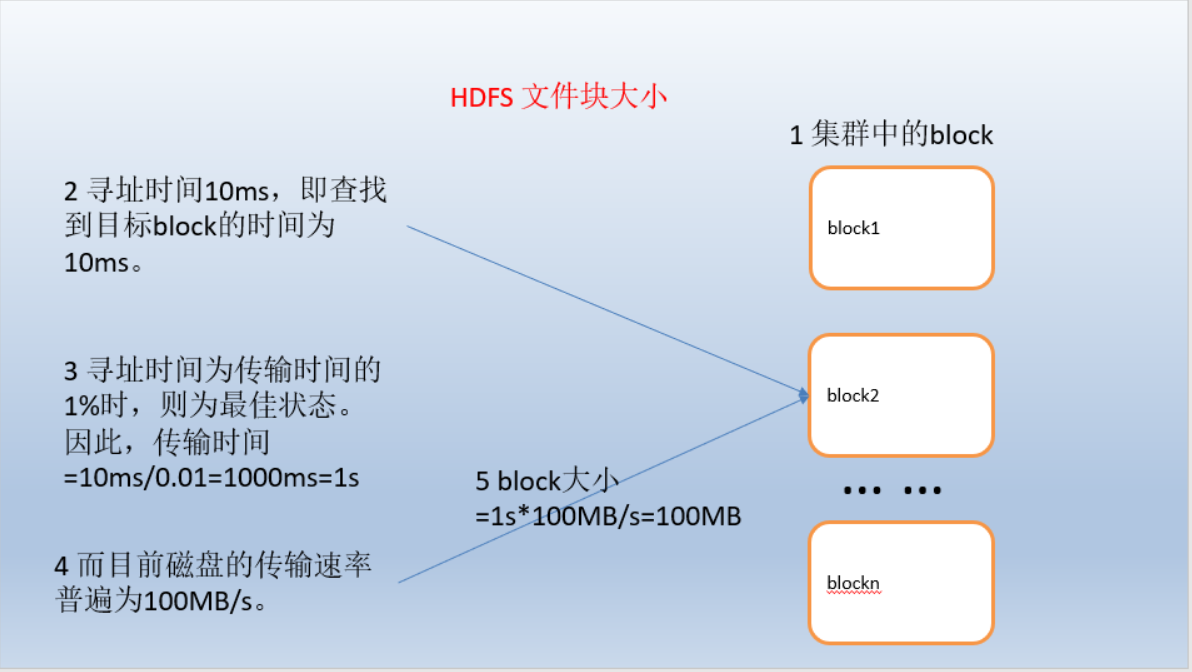
## **1.3 HDFS 文件块大小**

HDFS中的文件在物理上是分块存储（block），块的大小可以通过配置参数( dfs.blocksize)来规定，默认大小在hadoop2.x版本中是128M，老版本中是64M

HDFS的块比磁盘的块大，其目的是为了最小化寻址开销。如果块设置得足够大，从磁盘传输数据的时间会明显大于定位这个块开始位置所需的时间。因而，传输一个由多个块组成的文件的时间取决于磁盘传输速率。

如果寻址时间约为10ms，而传输速率为100MB/s，为了使寻址时间仅占传输时间的1%，我们要将块大小设置约为100MB。默认的块大小128MB。

块的大小：10ms\*100\*100M/s = 100M



# **二 HFDS命令行操作**

## **2.1基本语法**

bin/hadoop fs 具体命令

## **2.2参数大全**

bin/hadoop fs

|  |
| --- |
| [-appendToFile <localsrc> ... <dst>]  [-cat [-ignoreCrc] <src> ...]  [-checksum <src> ...]  [-chgrp [-R] GROUP PATH...]  [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]  [-chown [-R] [OWNER][:[GROUP]] PATH...]  [-copyFromLocal [-f] [-p] <localsrc> ... <dst>]  [-copyToLocal [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]  [-count [-q] <path> ...]  [-cp [-f] [-p] <src> ... <dst>]  [-createSnapshot <snapshotDir> [<snapshotName>]]  [-deleteSnapshot <snapshotDir> <snapshotName>]  [-df [-h] [<path> ...]]  [-du [-s] [-h] <path> ...]  [-expunge]  [-get [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]  [-getfacl [-R] <path>]  [-getmerge [-nl] <src> <localdst>]  [-help [cmd ...]]  [-ls [-d] [-h] [-R] [<path> ...]]  [-mkdir [-p] <path> ...]  [-moveFromLocal <localsrc> ... <dst>]  [-moveToLocal <src> <localdst>]  [-mv <src> ... <dst>]  [-put [-f] [-p] <localsrc> ... <dst>]  [-renameSnapshot <snapshotDir> <oldName> <newName>]  [-rm [-f] [-r|-R] [-skipTrash] <src> ...]  [-rmdir [--ignore-fail-on-non-empty] <dir> ...]  [-setfacl [-R] [{-b|-k} {-m|-x <acl\_spec>} <path>]|[--set <acl\_spec> <path>]]  [-setrep [-R] [-w] <rep> <path> ...]  [-stat [format] <path> ...]  [-tail [-f] <file>]  [-test -[defsz] <path>]  [-text [-ignoreCrc] <src> ...]  [-touchz <path> ...]  [-usage [cmd ...]] |

## **2.3常用命令实操**

（1）-help：输出这个命令参数

bin/hdfs dfs -help rm

（2）-ls: 显示目录信息

hadoop fs -ls /

Hadoop fs -lsr /

（3）-mkdir：在hdfs上创建目录

hadoop fs -mkdir -p /hdfs路径

（4）-moveFromLocal从本地剪切粘贴到hdfs

hadoop fs -moveFromLocal 本地路径 /hdfs路径

（5）--appendToFile ：追加一个文件到已经存在的文件末尾

hadoop fs -appendToFile 本地路径 /hdfs路径

（6）-cat ：显示文件内容

hadoop fs -cat /hdfs路径

（7）-tail -f：监控文件

hadoop fs -tail -f /hdfs路径

（8）-chmod、-chown：linux文件系统中的用法一样，修改文件所属权限

hadoop fs -chmod 777 /hdfs路径

hadoop fs -chown someuser:somegrp /hdfs路径

（9）-cp ：从hdfs的一个路径拷贝到hdfs的另一个路径

hadoop fs -cp /hdfs路径1 / hdfs路径2

（10）-mv：在hdfs目录中移动/重命名 文件

hadoop fs -mv /hdfs路径 / hdfs路径

（11）-get：等同于copyToLocal，就是从hdfs下载文件到本地

hadoop fs -get / hdfs路径 ./本地路径

（12）-getmerge ：合并下载多个文到linux本地，比如hdfs的目录 /aaa/下有多个文件:log.1, *log.2,log.3,...（注：是合成到Linux本地）*

hadoop fs -getmerge /aaa/log.\* ./log.sum

合成到不同的目录：hadoop fs -getmerge /hdfs1路径 /hdfs2路径 /

（13）-put：等同于copyFromLocal

hadoop fs -put /本地路径 /hdfs路径

（14）-rm：删除文件或文件夹

hadoop fs -rm -r /hdfs路径

（15）-df ：统计文件系统的可用空间信息

hadoop fs -df -h / hdfs路径

（16）-du统计文件夹的大小信息

[itstar@bigdata111 hadoop-2.8.4]$ hadoop fs -du -s -h / hdfs路径

188.5 M /user/itstar/wcinput

[itstar@bigdata111 hadoop-2.8.4]$ hadoop fs -du -h / hdfs路径

188.5 M / hdfs路径

97 / hdfs路径

（17）-count：统计一个指定目录下的文件节点数量

hadoop fs -count /aaa/

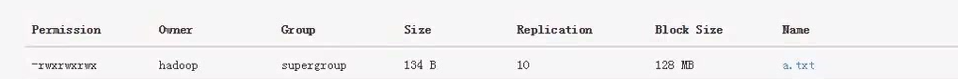
[itstar@bigdata111 hadoop-2.8.4]$ hadoop fs -count / hdfs路径

1 2 197657784 / hdfs路径

嵌套文件层级； 包含文件的总数

（18）-setrep：设置hdfs中文件的副本数量：3是副本数，可改

hadoop fs -setrep 3 / hdfs路径



这里设置的副本数只是记录在namenode的元数据中，是否真的会有这么多副本，还得看datanode的数量。因为目前只有3台设备，最多也就3个副本，只有节点数的增加到10台时，副本数才能达到10。

# **三、IDEA环境准备**

{$MAVEN\_HOME/conf/settings}

|  |
| --- |
| <!--本地仓库所在位置-->  <localRepository>F:\m2\repository</localRepository>  <!--使用阿里云镜像去下载Jar包，速度更快-->  <mirrors>  <mirror>  <id>alimaven</id>  <name>aliyun maven</name>  <url>http://maven.aliyun.com/nexus/content/groups/public/</url>  <mirrorOf>central</mirrorOf>  </mirror>  </mirrors> |

## **3.1 Maven配置**

## **3.2 Maven准备**

|  |
| --- |
| <dependencies>  <dependency>  <groupId>org.apache.hadoop</groupId>  <artifactId>hadoop-common</artifactId>  <version>2.8.4</version>  </dependency>  <dependency>  <groupId>org.apache.hadoop</groupId>  <artifactId>hadoop-hdfs</artifactId>  <version>2.8.4</version>  </dependency>  <dependency>  <groupId>org.apache.hadoop</groupId>  <artifactId>hadoop-client</artifactId>  <version>2.8.4</version>  </dependency>  <dependency>  <groupId>org.projectlombok</groupId>  <artifactId>lombok</artifactId>  <version>1.16.10</version>  </dependency>  <dependency>  <groupId>log4j</groupId>  <artifactId>log4j</artifactId>  <version>1.2.17</version>  </dependency>  <dependency>  <groupId>org.slf4j</groupId>  <artifactId>slf4j-api</artifactId>  <version>1.7.7</version>  </dependency>  <!-- https://mvnrepository.com/artifact/junit/junit -->  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>4.12</version>  <scope>test</scope>  </dependency>  </dependencies> |

## **3.3 IDEA准备**

1）配置HADOOP\_HOME环境变量

2）采用hadoop编译后的bin 、lib两个文件夹（如果不生效，重新启动IDEA）

3）创建第一个java工程

|  |
| --- |
| **public** **class** HdfsClientDemo1 {  **public** **static** **void** main(String[] args) **throws** Exception {  // 1 获取文件系统  Configuration configuration = **new** Configuration();  // 配置在集群上运行  configuration.set("fs.defaultFS", "hdfs://bigdata111:9000");  FileSystem fileSystem = FileSystem.*get*(configuration);    // 直接配置访问集群的路径和访问集群的用户名称  // FileSystem fileSystem = FileSystem.get(new URI("hdfs://bigdata111:9000"),configuration, "itstar");    // 2 把本地文件上传到文件系统中  fileSystem.copyFromLocalFile(**new** Path("f:/hello.txt"), **new** Path("/hello1.copy.txt"));    // 3 关闭资源  fileSystem.close();  System.*out*.println("over");  }  } |

4）执行程序

注：eclipse运行时可能需要配置用户名称

客户端去操作hdfs时，是有一个用户身份的。默认情况下，hdfs客户端api会从jvm中获取一个参数来作为自己的用户身份：-DHADOOP\_USER\_NAME=itstar，itstar为用户名称。