**1 HDFS的shell（命令行客户端）操作**

注意在Hadoop2.0之前是使用的hadoop命令，那时候的HDFS集群叫做hadoop集群，但是hadoop2.0之后。使用的是hdfs命令，hadoop集群也分为了HDFS集群和Yarn集群！

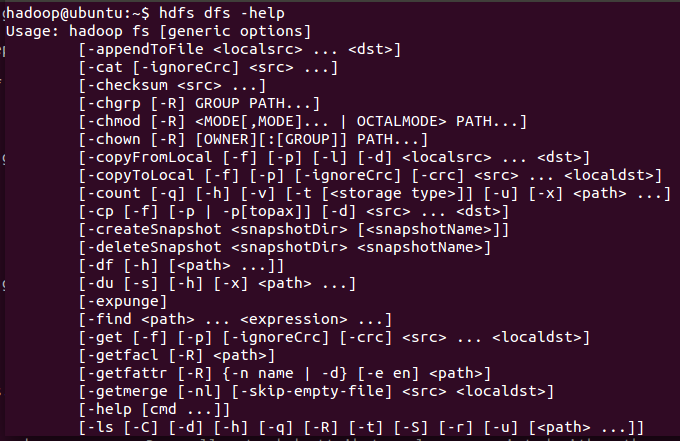
**1.1 HDFS的shell操作**

(1)查看命令帮助：hdfs –help



(2)查看Hadoop集群支持的文件系统命令

hdfs dfs -help



**1.2 HDFS DFS命令详解**

(1)准备工作

　　检查需要处理的数据（假定数据hadoop-test.text已经存放在/data⽬录下）

　　看到测试数据⽬

(2)、文件操作

1）查看⽂件或⽬录

$> hdfs dfs -ls

2）建⽴⽬录(在hdfs的/user/xxx⽬录下操作）

在建立目录前，我们先查看原始的目录状态

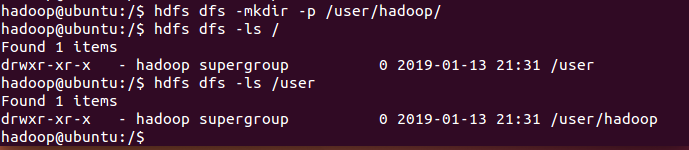
hdfs dfs -ls /



根目录下没有一个文件或文件夹。

执行

hdfs dfs -mkdir -p /user/hadoop/



可以发现文件夹已经在hdfs上创建好了。（在集群中的所有节点都可以查看到）

在/user/hadoop/下创建input文件夹

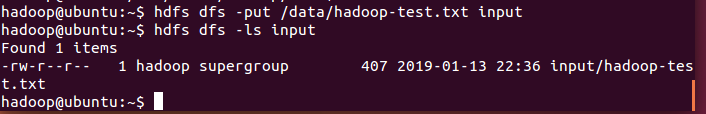
hdfs dfs -mkdir -p /user/hadoop/input/

(hdfs好像没有类似cd命令的操作)

3）⽂件上传(将本地⽂件local\_file上传到hdfs的/user/xxx⽬录下)

hdfs dfs -put local\_file input

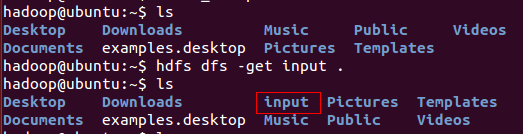
如hdfs dfs -put /data/hadoop-test.txt input

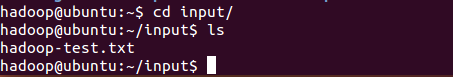


4）下载⽂件或⽬录到本地当前⽬录下

下面是下载input文件夹的例子

hdfs dfs -get input .

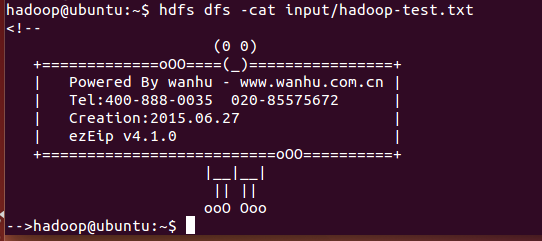




5）浏览hdfs中的⽂件

hdfs dfs -cat input/local\_file

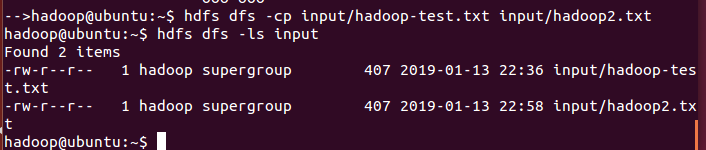
如hdfs dfs -cat input/hadoop-test.txt



6）在hdfs中复制⽂件或⽬录

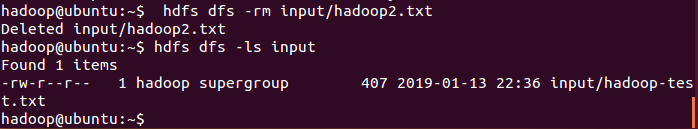
hdfs dfs -cp input/local\_file input/local\_file2

如hdfs dfs -cp input/hadoop-test.txt input/hadoop2.txt



7）删除⽂件或者⽬录

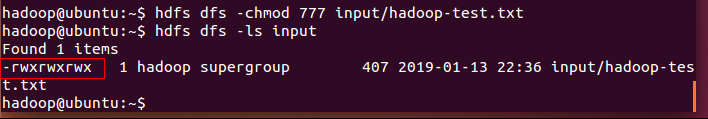
hdfs dfs -rm input/hadoop2.txt



8)修改权限

修改文件权限与linux shell中类似

hdfs dfs –chmod 777 input/hadoop-test.txt



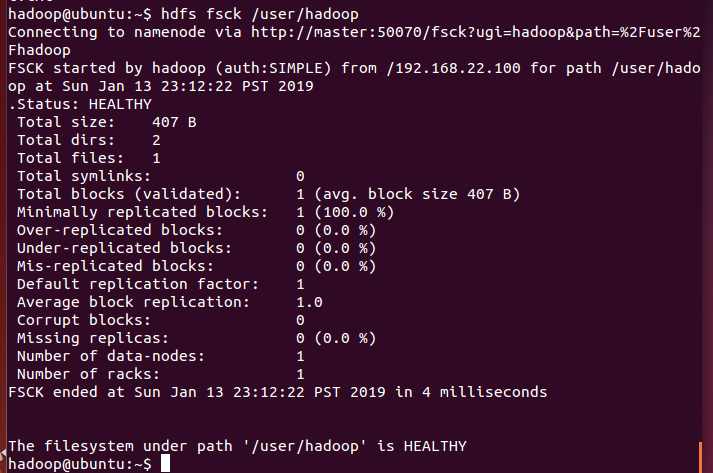
9）查看命令帮助

hdfs dfs -help [cmd]

**1.3 HDFS管理命令**

(1)系统⽬录检查

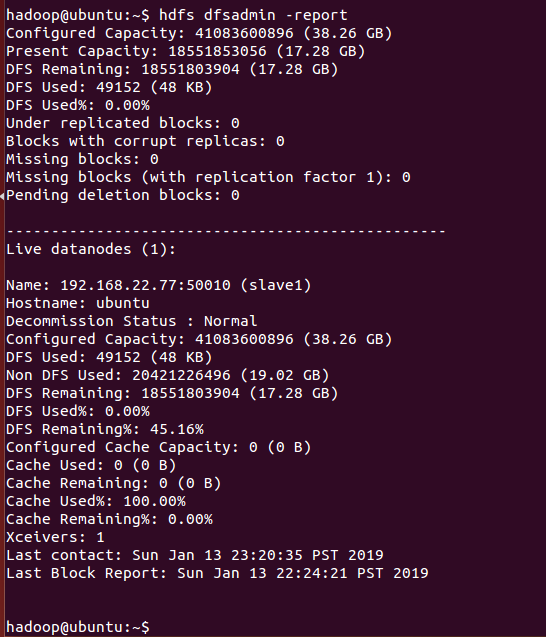
hdfs fsck /user/hadoop #系统⽬录详细检测



hdfs fsck /user/hadoop -files -blocks -locations –racks

(2)检测DataNode报告

hdfs dfsadmin –report



(3) 权限管理

hdfs fs -chmod 666 /user/xxx

注：xxx充当占位符的作用，请按照实际情况修改，下文不再重复说明。

(4 ) hdfs空间⽬录配额设置

hdfs dfsadmin -setSpaceQuota [N] /user/xxx

(5)hdfs空间⽬录配额清除

hdfs dfsadmin -clrSpaceQuota /user/xxx

(6)查看⽬录配额设置

hdfs fs -count -q /user/xxx

(7)删除DataNode

hdfs dfsadmin -refreshNodes

总结：

　　注意HDFS中的文件系统也是和linux文件系统一样的。既然是文件系统也有根目录和家目录，在HDFS中“/”代表的就是根目录，而“/user”等于linux中的“/usr”，下一级目录代表的就是用户了。

但是HDFS没有这个用户，所以就有了虚拟用户，也就是你当前HDFS的Linux系统当前用户就会默认为它的虚拟用户。当我们有一个zyh用户时，所以你就可以创建一个/user/zyh/下面的目录，就代表着家目录。

**2搭建Hadoop客户端与Java访问HDFS集群**

**2.1 Hadoop客户端配置**

其实在前面配置的每一个集群节点都可以做一个Hadoop客户端。但是我们一般都不会拿用来做集群的服务器来做客户端，需要单独的配置一个客户端。

像服务端的Hadoop一样，参考安装说明书安装好java和Hadoop，但作为客户端的Hadoop配置要简单的多。

(1) 客户端配置core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://192.168.22.100:9000</value>

</property>

</configuration>



(2) 客户端配置mapred-site.xml

<configuration>

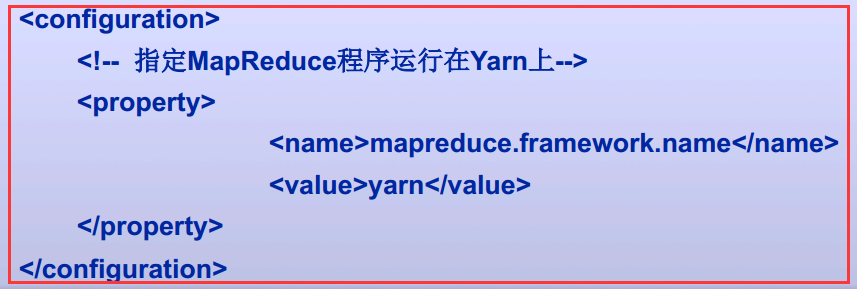
<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>



(3) 客户端配置yarn-site.xml

<configuration>

<!-- Site specific YARN configuration properties -->

<property>

<name>yarn.resourcemanager.hostname</name>

<value>192.168.22.100</value>

</property>

</configuration>



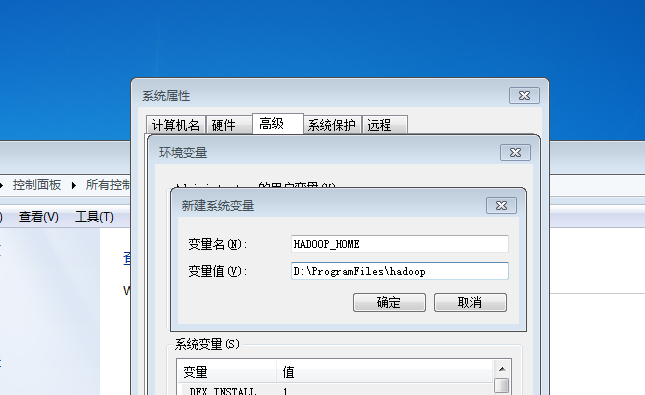
以上就搭建了一个Hadoop的客户端。

开发往往在windows环境下进行，如果在window上安装客户端，还需以下额外的操作。

先安装jdk，然后将下载的hadoop安装包解压的制度路径。

1）配置环境

.计算机->属性->高级系统设置->高级选项卡->环境变量->单击新建HADOOP\_HOME



注意：路径到bin的上级目录即可。

2)Path环境变量下配置”%HADOOP\_HOME%\bin;”变量

3) 修改hadoop安装目录下的etc/hadoop下**hadoop-env.cmd**文件，将JAVA\_HOME用 @rem注释掉，编辑为JAVA\_HOME的路径，然后保存。



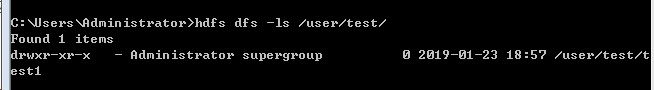
测试一下，/user/test是事先在hdfs上创建的测试目录，被赋予777的权限，该目录下有一个test1文件夹。我们在windows主机上运行cmd执行查看文件列表的命令

hdfs dfs –l /user/test



可以发现虽然能查看到文件列表，但抛了异常。原版的Hadoop不支持Windows系统，我们需要修改一些配置方便在Windows上运行

4）在网上下载名为hadooponwindows-master.zip的文件(能支持在windows运行hadoop的工具)，解压后将其中的bin目录下的所有文件粘贴覆盖到hadoop安装目录的bin下。然后再测试一遍上面的操作。



没有抛异常。这样，在windows上的客户端算是安装好了。

**2.2 Java访问HDFS集群**

**2.2.1 HDFS的Java访问接口**

1）org.apache.hadoop.fs.FileSystem

是一个通用的文件系统API，提供了不同文件系统的统一访问方式。

2）org.apache.hadoop.fs.Path

是Hadoop文件系统中统一的文件或目录描述，类似于java.io.File对本地文件系统的文件或目录描述。

3）org.apache.hadoop.conf.Configuration

读取、解析配置文件(如core-site.xml/hdfs-default.xml/hdfs-site.xml等)，或添加配置的工具类

4）org.apache.hadoop.fs.FSDataOutputStream

对Hadoop中数据输出流的统一封装

5）org.apache.hadoop.fs.FSDataInputStream

对Hadoop中数据输入流的统一封装

**2.2.2 Java访问HDFS主要编程步骤**

1）构建Configuration对象，读取并解析相关配置文件

Configuration conf=new Configuration();

2）设置相关属性

conf.set("fs.defaultFS","hdfs://IP:9000");

3）获取特定文件系统实例fs（以HDFS文件系统实例）

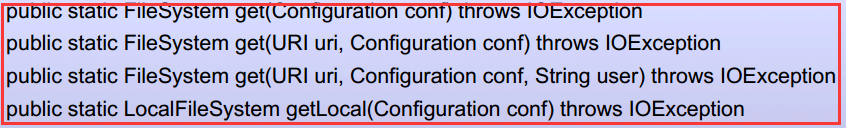
FileSystem fs=FileSystem.get(new URI("hdfs://IP:9000"),conf);

4）通过文件系统实例fs进行文件操作(以删除文件未例)

fs.delete(new Path("/user/liuhl/someWords.txt"));

2.2.3 使用FileSystem API读取数据文件

有两个静态工厂方法来获取FileSystem实例文件系统。



常用的就第二个和第四个

**2.3 实战Java访问HDFS集群**

下载示例项目

git clone <https://github.com/wallezhou/j2hdfs.git>

查看hdfs文件案例

com.walle.j2hdfs.main.CatDemo

本地文件上传到hdfs案例

com.walle.j2hdfs.main.UploadDemo

hdfs文件下载到本地

com.walle.j2hdfs.main.DownloadDemo

删除hdfs文件

com.walle.j2hdfs.main.DeleteDemo

本地文件/文件夹压缩上传到hdfs案例

com.walle.j2hdfs.main.CompressedUploadDemo

hdfs文件/文件夹解压到本地案例

com.walle.j2hdfs.main.DecompressionDownloadDemo

错误总结：

org.apache.hadoop.security.AccessControlException: Permission denied: user=Administrator, access=WRITE, inode="/user/hadoop":hadoop:supergroup:drwxr-xr-x

这是由于在window上Administator用户没有hdfs上/user/hadoop的写权限。

在 hdfs-site.xml 总添加参数：

<property>

<name>dfs.permissions</name>

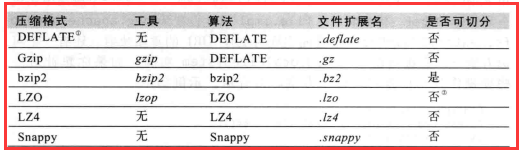
<value>false</value>

</property>

或者修改目录的权限。

文件压缩

减少储存文件所需要的磁盘空间，并加速数据在网络和磁盘上的传输。这两个在大数据处理大龄数据时相当重要！



Hadoop对前面三种有默认集成，有就是说Hadoop支持DEFLATE、Gzip、bzip2三种压缩格式。而后面三种Hadoop没有支持，要用的话要自己去官网下载相应的源码去编译加入到Hadoop才能用。

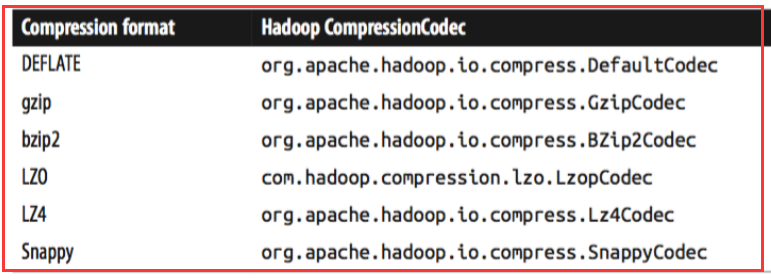
注意：

1）这里我要说的是“是否分割”，当我们一个文件去压缩即使有非常好的压缩算法，但是它的大小还是超过了一个数据块的大小，这时就涉及到分割了。所以说在以后的压缩我们大多数情况下会使用bzip2。

2）Gzip和bzip2比较时，bzip2的压缩率（压缩之后的大小除以源文件的大小）要小，所以说bzip2的压缩效果好。而这里就会压缩和解压缩的时候浪费更多的时间。就是我们常说的“用时间换取空间”。

编解码器（Codec）

codec实现了一种压缩-加压缩算法（意思就是codec使用相关的算法对数据进行编解码）。在Hadoop中，一个对CompressionCodec接口的实现代表一个codec。



对于不同的压缩算法有不同的编解码器。我们要对一个文件进行压缩需要编码器，对一个压缩文件进行解压需要解码器。那我们怎么样去获取编解码器呢？

有两种方式：

一是：根据扩展名让程序自己去选择相应的编解码器。比如说：我在本地有一个文件是 user.txt我们通过-Dinput=user.txt去上传这个文件到集群，在集群中我们把它指定到-Doutput=/user.txt.gz.。这是我们程序的相关的类会根据你的扩展名（这里是.gz）获取相应的压缩编解码器。在Hadoop中有一个CompressionCodecFactory会根据扩展名获取相应的编解码器对象 。

二是：我们自己去指定编解码器。为什么要去指定呢？比如说，我在本地有一个文件是user.txt.gz，其实这个压缩文件是使用的是bzip2的压缩算法压缩的。（因为我自己去更改了它的扩展名），所以这时候就要自己去指定编解码器。

