华东师范大学数据学院上机实践报告

课程名称：分布式模型与编程	年级：2018	上机实践成绩:
指导教师：徐辰		姓名：孙秋实
上机实践名称：HDFS2.x 编程	学号：10185501402	上机实践日期：2021/3/29
上机实践编号：Lab5	组号：Group5	上机实践时间:

Part 1
实验目的

(1) 学习编写简单的基于 Java API 操作 HDFS 的程序

(2) 掌握在 IDEA 中调试 HDFS 相关程序，以及在单机伪分布式、分布式模式下提交运行 HDFS 相关程序的方法

(3) 理解 HDFS 中块的概念

Part 2
实验任务

(1) 完成基于 Java API 编写的“将本地文件上传到 HDFS”的程序

(2) 在 IDEA 中调试以上程序，以及在单机伪分布式、分布式模式下提交运行该程序

Part 3
使用环境

(1) 操作系统：Ubuntu 18.04

(2) JDK 版本：1.8

(3) Hadoop 版本：2.10.1

(4) IDEA 版本：2020.2.3

Part 4
实验过程

Section 1
编写 HDFS 应用程序

pom.xml (exp5) Writer.java

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>exp5</artifactId>
    <version>1.0-SNAPSHOT</version>

    <dependencies>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-common</artifactId>
            <version>2.10.1</version>
        </dependency>

        <dependency...>

        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-client</artifactId>
            <version>2.10.1</version>
        </dependency>

    </dependencies>

</project>
```

图 1: 修改 pom.xml 中的 dependencies

tips: POM 是 Project Object Model 的缩写，即项目对象模型。pom.xml 就是 maven 的配置文件，用以描述项目的各种信息

修改完成后，在菜单界面选择 View→Tool Windows->Maven, 在弹出的界面中点击 Reload All Maven Projects 加载依赖文件

然后就可以开始编写 java 程序了，我们在这里实现将本地文件上传到 HDFS 中，即 HDFS Shell 中的 put 命令。

- 在项目的 “src/main/java” 目录下，选择 New → Package, 输入名称 cn.edu.ecnu.hdfs. example.java.fileupload
- 右键单击建好的包，选择 New → Java Class, 输入名称 Writer

```java
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataOutputStream;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IOUtils;

import java.io.FileInputStream;
import java.io.IOException;
import java.net.URI;
public class Writer {
    public void write(String hdfsFilePath,String localFilePath) throws
            IOException{
        Configuration conf = new Configuration();
        FileSystem fs = FileSystem.get(URI.create(hdfsFilePath),conf);

        FSDataOutputStream hdfsOutputStream = fs.create(new Path(hdfsFilePath));

        FileInputStream localInputStream = new FileInputStream(localFilePath);

        IOUtils.copyBytes(localInputStream,hdfsOutputStream, buffSize: 4096, close: true);
    }
    public static void main(String[] args) throws IOException{
        if(args.length < 2){
            System.err.println("Usage: <hdfsFilePath> <localFilePath>");
            System.exit( code: -1);
        }
        Writer writer = new Writer();
        writer.write(args[0],args[1]);
    }
}
```

图 2: 编写 Writer 类

Section 2
调试 HDFS 应用程序

配置运行环境在 idea 界面点击菜单栏 Run->Edit Configuration, 在弹出的界面中点击 + 号选择 Application, 新建 Application 配置，Name 为 WordCountJava

- 配置 Main Class 为 cn.edu.ecnu.hdfs.example.java.fileupload.Writer
- 配置 Program arguments 为输入路径输出路径

该程序需要传入两个参数:
第一个为 HDFS 文件写入路径，第二个为即将上传到 HDFS 中文件的路径，
如/home/dase-local/input/pd.train/home/dase-local/output/pd.train, 该配置表示将本地”home/dase-local/input”目录中的 “pd.train” 文件上传到 HDFS 的”/user/dase-local/input” 目录中。

Section 3
运行 HDFS 应用程序（伪分布式）

首先做一些准备工作，把之前的 IDEA 项目打包成 jar 包，命名为 Writer.jar，然后上传到我的云主机，如下图所示

```
Last login: Mon Mar 29 19:27:32 on ttys001
 sunqiushi@sunqiushideMacBook-Pro  ~  scp /Users/sunqiushi/Desktop/Writer.jar dase-local@10.24.21.163:~/hadoop-2.10.1/myApp/
The authenticity of host '10.24.21.163 (10.24.21.163)' can't be established.
ECDSA key fingerprint is SHA256:M2mR/pJ+CfVHzcr0XssdykZhiLGRw14JYPeNGr5rPRY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.24.21.163' (ECDSA) to the list of known hosts.
dase-local@10.24.21.163's password:
Writer.jar                                                    100% 2223   200.7KB/s   00:00
 sunqiushi@sunqiushideMacBook-Pro  ~
```

图 3: 把 jar 包传到云主机

tips: 如果不喜欢这么传输的话，可以直接使用 FileZilla 的可视化界面传输，如果是 Windows 机器也可以使用 mobaXterm 来传输。

与之前的实验一样，通过以下方法启动 HDFS 系统

```
dase-local@10-24-21-163:~/softwares/hadoop-2.10.1$ ./sbin/start-dfs.sh
Starting namenodes on [localhost]
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:M2mR/pJ+CfVHzcr0XssdykZhiLGRw14JYPeNGr5rPRY.
Are you sure you want to continue connecting (yes/no)? yes
localhost: Warning: Permanently added 'localhost' (ECDSA) to the list of known h
osts.
localhost: starting namenode, logging to /home/dase-local/softwares/hadoop-2.10.
1/logs/hadoop-dase-local-namenode-10-24-21-163.out
localhost: starting datanode, logging to /home/dase-local/softwares/hadoop-2.10.
1/logs/hadoop-dase-local-datanode-10-24-21-163.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /home/dase-local/softwares/hadoo
p-2.10.1/logs/hadoop-dase-local-secondarynamenode-10-24-21-163.out
dase-local@10-24-21-163:~/softwares/hadoop-2.10.1$ jps
7091 Jps
6965 SecondaryNameNode
6710 DataNode
6505 NameNode
dase-local@10-24-21-163:~/softwares/hadoop-2.10.1$
```

图 4: 启动 HDFS 服务

在单机伪分布式下提交程序

```
dase-local@10-24-21-163:~/softwares/hadoop-2.10.1$ ./bin/hadoop jar ./Writer.jar
 hdfs://localhost:9000/user/dase-local/input/README.txt /home/dase-local/softwar
es/hadoop-2.10.1/README.txt
dase-local@10-24-21-163:~/softwares/hadoop-2.10.1$
```

图 5: 提交程序

这个程序实现了把本地目录的 README.txt 文件上传到 HDFS 的 $/user/dase-local/input$ 目录中的操作，下图使用 Web UI 进行查看

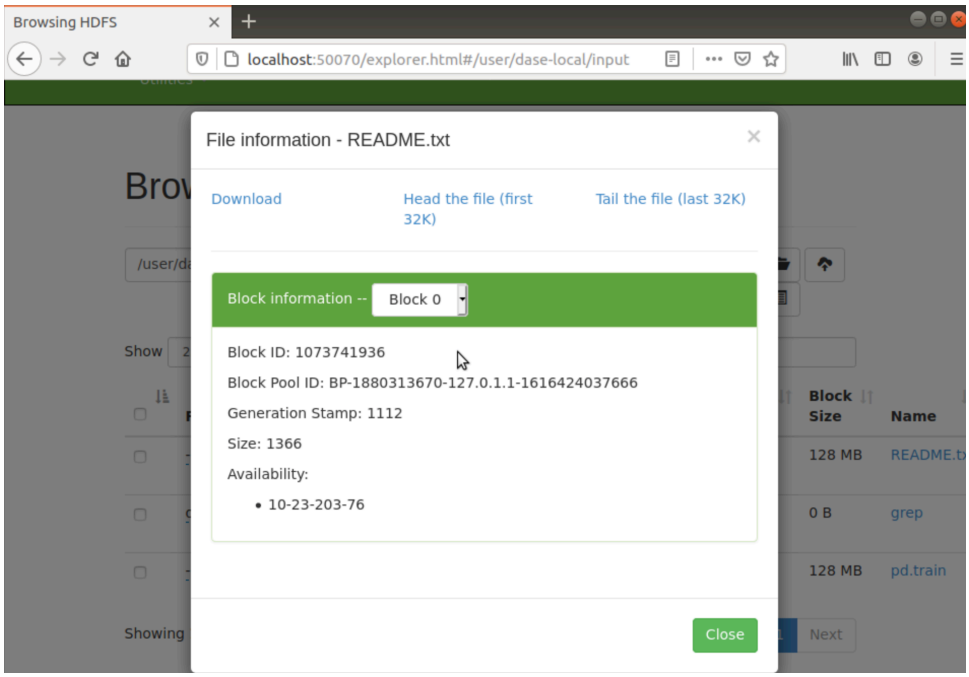

图 6: Web UI 查看提交的文件

关闭 HDFS，准备进行分布式环境下的实验

```
dase-local@10-24-21-163:~/softwares/hadoop-2.10.1$ ./sbin/stop-dfs.sh
Stopping namenodes on [localhost]
localhost: stopping namenode
localhost: stopping datanode
Stopping secondary namenodes [0.0.0.0]
0.0.0.0: stopping secondarynamenode
dase-local@10-24-21-163:~/softwares/hadoop-2.10.1$
```

图 7: 关闭 HDFS 服务

Section 4
运行 HDFS 应用程序（分布式系统）

以下是在分布式环境（4 台机器）进行实验，以 dase-dis 用户身份执行，与上述操作一样，还是先把 jar 包复制到 dase-dis 用户下的 $/hadoop-2.10.1/myApp$ 目录之中

随后，主节点使用 start-dfs.sh 启动 HDFS 服务，然后让客户端提交应用程序

```
tangqiong@tangqiongdeMacBook-Pro Writer % scp Writer.jar dase-dis@10.24.21.106:/home/dase-dis/hadoop-2.10.1/myAp
p
dase-dis@10.24.21.106's password:
Writer.jar                                    100% 2222   255.8KB/s   00:00
tangqiong@tangqiongdeMacBook-Pro Writer %
```

图 8: Client 提交应用程序

```
dase-dis@ecnu02:~/.ssh$ jps
4532 DataNode
5158 NodeManager
5306 Jps
```

图 9: 从节点状态

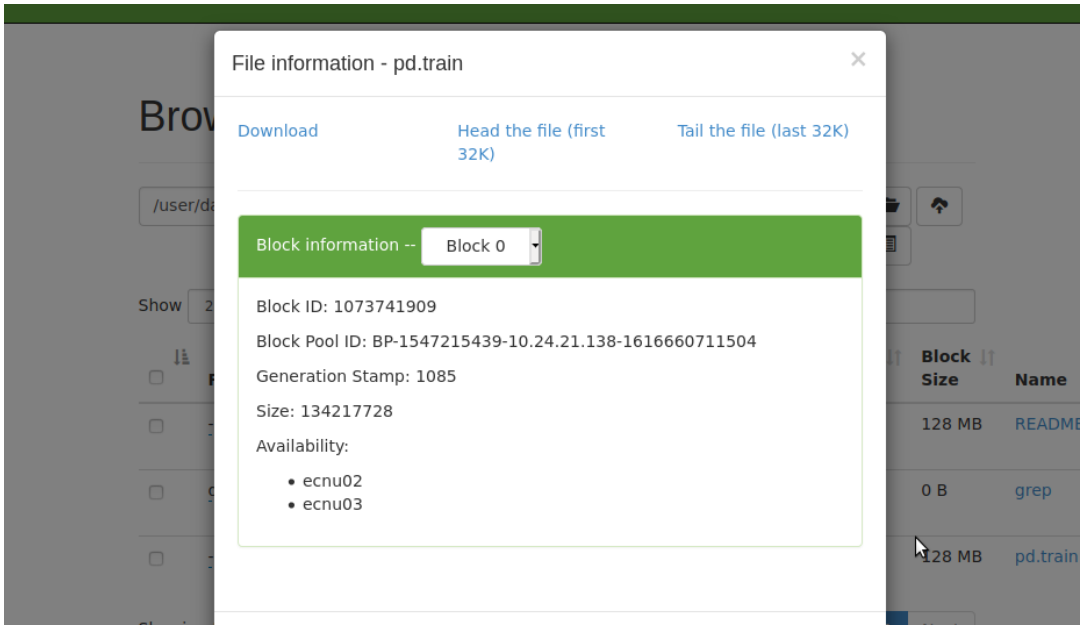

图 10: 分布式模式下应用程序结果

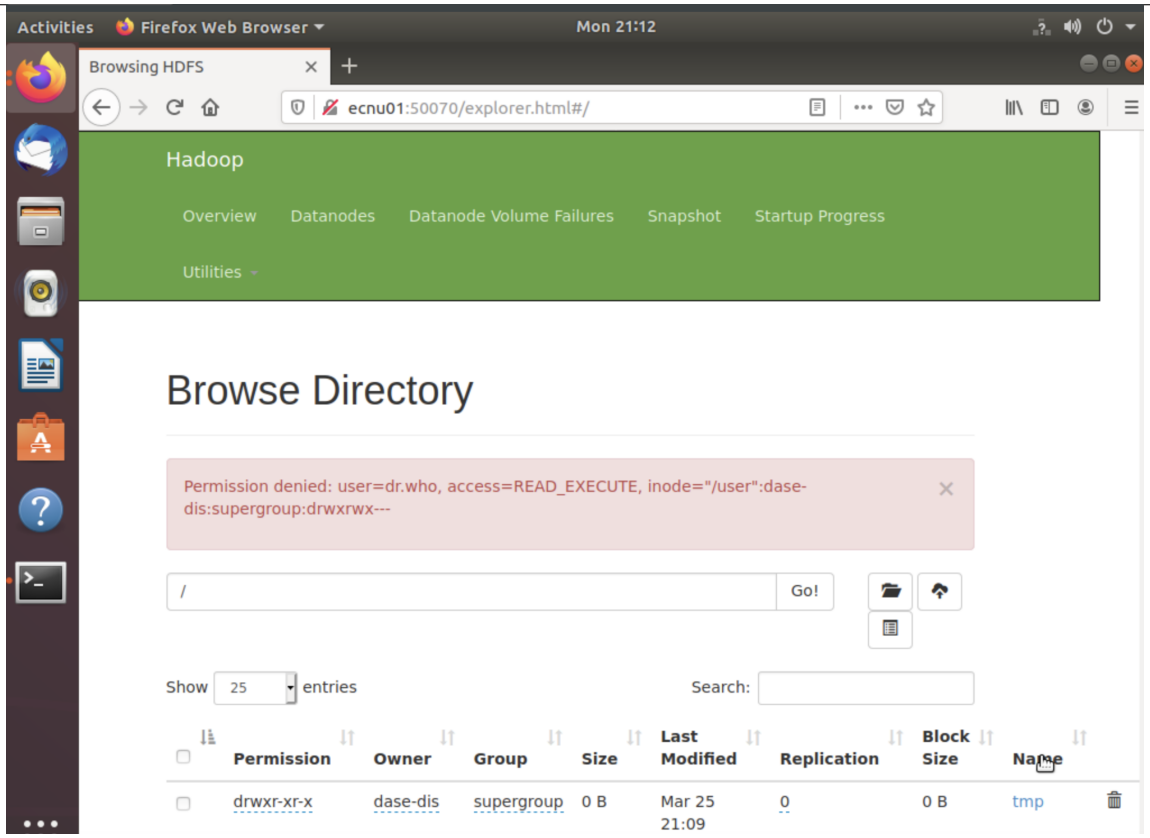

图 11: 访问应用程序结果详情

tips: 不论从哪个节点访问都可能会被 permission denied，使用 chmod 可以解决这个问题

Part 5
思考题

Section 1
HDFS 上每个文件块多大？结合 Web UI 说明

如下图所示，每个文件块大小为 128MB（事实上 128MB 是默认的设置）

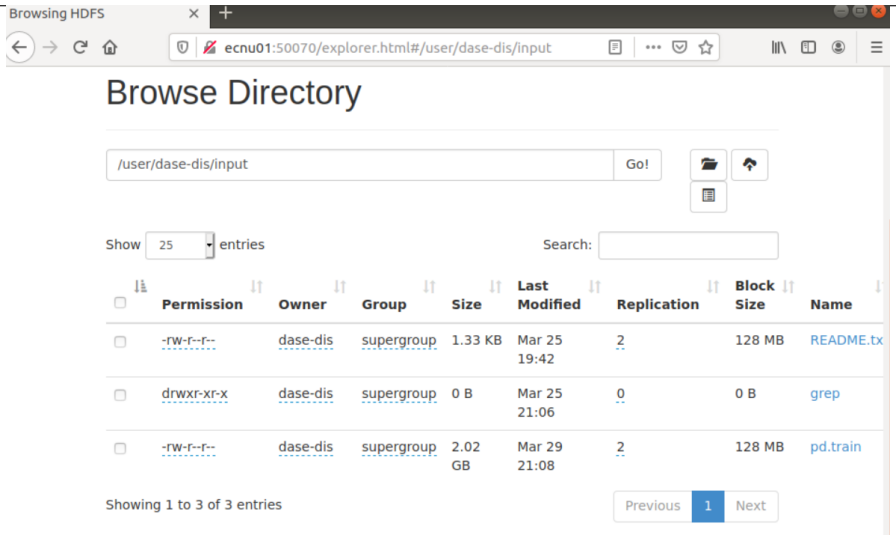

图 12: Block Size

Section 2

HDFS 上每个文件的副本数如何指定?

Answer: 在 HDFS 的配置文件 *hdfs-site.xml* 中可以指定，可以通过修改 dfs.replication 的值来指定每个文件的副本数。

Part 6

实验总结

踩坑:

通过 Web UI 访问会出现多个 Permission Denied 的问题，都是可以通过 chmod 777 来解决的，所以主节点每次访问的时候都可以试着先提高自己的权限，但是在工业应用这么做应该是很危险的。

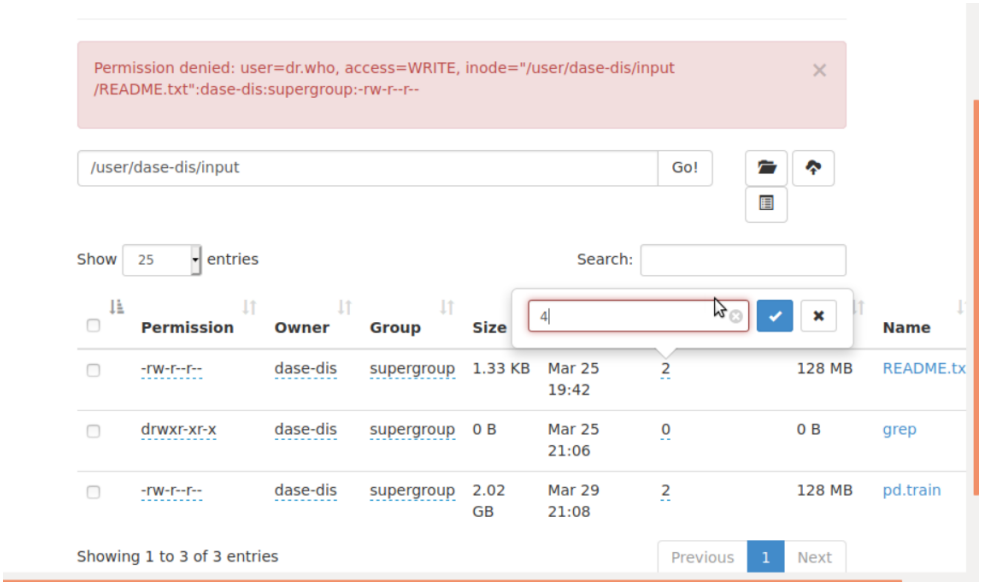

图 13: Permission Denied Example