

本地对 HDFS 文件系统操作

对 hdfs 文件系统的操作有很多，下面只是做了几个比较常用的简单的功能：包括文件上传下载，创建/删除文件（文件夹），显示文件内容，列出目录下的文件。实现这些功能需要引入 Hadoop 下的库。包括 common，hdfs，mapreduce，yarn 下的所有包，本次运行环境是基于 Hadoop2.6.0，运行的时候既可以在本地模式下运行，也可以在分布式环境下运行。需要指出的是，这些操作是指在本地与 hdfs 文件系统之间。下面列出几个简单的相关操作的方法。完整的代码以及参考手册我将一并上传到 wiki 上。

上传：

```
public static void upLoad(Configuration conf,String uri, String local, String
remote) throws IOException{
    FileSystem fs = FileSystem.get(URI.create(uri), conf); //获取链接
    fs.copyFromLocalFile(new Path(local), new Path(remote)); //执行 fs 提
    供的拷贝方法
    System.out.println("copy from : " + local + " to " + remote);
    fs.close();
}
```

下载：

```
public static void download(Configuration conf , String uri ,String remote,
String local) throws IOException {
    Path r_path = new Path(remote);
    FileSystem fs = FileSystem.get(URI.create(uri), conf);
    fs.copyToLocalFile(r_path, new Path(local));
    System.out.println("download: from" + remote + " to " + local);
    fs.close();
}
```

创建文件：

```
public static void createFile(Configuration conf, String uri, String remote)
throws IOException{
    FileSystem fs = FileSystem.get(URI.create(uri),conf);
    Path r_path = new Path(remote);
    if(!fs.isFile(r_path)){
        fs.create(r_path);
        System.out.println("在 hdfs 上创建文件"+remote);
    }else{
        System.out.println("文件已经存在");
    }
}
```

```
}
```

删除文件：

```
public static void delete(Configuration conf, String uri, String remote)
throws IOException{
    FileSystem fs = FileSystem.get(URI.create(uri),conf);
    Path r_path = new Path(remote);
    fs.deleteOnExit(r_path);
    System.out.println("Deleted:" + remote);
    fs.close();
}
```

参考资料：

1. Hadoop-mapreduce 开发手册