
# 大数据技术基础课程实验报告

## 实验一：基于华为云构建大数据实验环境及 HDFS 实践

付容天

学号 2020211616

班级 2020211310

计算机学院（国家示范性软件学院）

2023 年 3 月 1 日

# 1.1 实验环境搭建

本实验要求购买华为云 ECS 服务器，并构建大数据实验环境。

## 1.1.1 购买 ECS

ECS 指的是弹性云服务器，主要是由 CPU、内存、操作系统、云硬盘组成的基础的计算组件，我们可以像使用自己本地 PC 或物理服务器一样，在云上使用弹性云服务器。我按照实验指导书中给出的方法，购买了华为云 ECS 服务器，配置如下：

| 名称/ID | 监控 | 安全 | 可用区 | 状态 | 规格/镜像 | IP地址 | 计费... | 标签 | 操作 |
|---|---|---|---|---|---|---|---|---|---|
| frt-2020211616-0003<br>ac263b16-3d22-49a0-8ed3-... | | | 可用区2 | 关机 | 2vCPUs \|...<br>CentOS ... | 123....<br>192.... | 按需计费<br>2023/03... | -- | 远程登录 |
| frt-2020211616-0004<br>b907b104-d415-4ebb-a88e-... | | | 可用区2 | 关机 | 2vCPUs \|...<br>CentOS ... | 123....<br>192.... | 按需计费<br>2023/03... | -- | 远程登录 |
| frt-2020211616-0002<br>13c6c104-691e-48ec-8454-... | | | 可用区2 | 关机 | 2vCPUs \|...<br>CentOS ... | 123....<br>192.... | 按需计费<br>2023/03... | -- | 远程登录 |
| frt-2020211616-0001<br>c6f04c73-453b-48f3-8c6d-f... | | | 可用区2 | 关机 | 2vCPUs \|...<br>CentOS ... | 123....<br>192.... | 按需计费<br>2023/03... | -- | 远程登录 |

图 1：ECS 服务器具体配置信息（名称为 frt-2020211616）

## 1.1.2 购买 OBS

OBS 指的是对象存储服务，是一个基于对象的海量存储服务，为客户提供了海量、安全、高可靠、低成本的数据存储能力。按照实验指导书的内容，我创建了如下配置的桶：


图 2：创建 OBS 桶（桶名称 frt-2020211616）

接下来就是创建并行文件系统，并行文件系统是 OBS 提供的一种经过优化的高性能文件系统，提供毫秒级别的访问时延，以及 TB/s 级别带宽和百万级别的

IOPS，能够快速处理高性能计算工作负载。按照实验指导书相关内容，如创建了如下所示的并行文件系统：


图 3：创建并行文件系统（名称 frt-2020211616a）

桶是 OBS 中存储对象的容器，桶中所有对象都处于同一逻辑层级，去除了文件系统中的多层树形目录结构，每个桶都有自己的存储类别、访问权限、所属区域等特性。下面截图展示了我创建的 OBS 桶（学号 2020211616）：


图 4：创建的 OBS 桶（含完整学号）

按照实验指导书，现在来获取 endpoint。Endpoint 从概念上说是用户在订阅主题时指定的接收消息的终端地址，多个 subscription 可以指定同一个 endpoint。含有桶名称和 endpoint 的截图如下所示：


图 5：桶相关信息（含有 endpoint）

接下来就是新增访问密钥，访问密钥包括访问密钥 ID（AS）和秘密访问密钥（SK）两部分，通过 SK 对数据进行签名验证，用于确保请求的机密性、完整性和请求者身份的正确性。新增结果如下图所示：


图 6.1：新增访问密钥

| | A | B | C |
|---|---|---|---|
| 1 | User Name | Access Key Id | Secret Access Key |
| 2 | rongtianfu-bupt | X8PVJIKE02ASGUK8FJIJ | 25mXeC7H2Y7xsuCxD8g6FGUaKhNSRLDW1ksoIBsl |

图 6.2：AK/SK 文件内容

接下来按照实验要求，给出弹性云服务器列表的截图，如下所示：


图 7：弹性云服务器列表（已关机）（学号 2020211616）

## 1.1.3 实验 1.1 的结果分析与总结

在完成了实验 1.1 后，我得到了：

1. 四台以学号（2020211616）命名的服务器
2. 以姓名学号命名的 OBS 桶
3. 保存在本地的 endpoint
4. 保存在本地的 AK/SK 文件

实验 1.1 作为这一系列实验的最基础实验，对我们熟悉云端环境、了解实验内容有很大的帮助。在实验 1.1 中没有遇到太大的问题，关键是要注意小心操作，确保每一步不会出错，这样就能顺利完成该实验。

# 1.2 安装 Hadoop 及 HDFS 应用实践

本实验要求在之前购买的华为云服务器上搭建 Hadoop 集群，并使用 IDEA 创建 maven 工程，完成 HDFS 文件读取实践。

## 1.2.1 Hadoop 集群搭建

根据实验指导书的内容，我先后完成了：

1. 通过 WinSCP 上传 Hadoop 安装包
2. 配置服务器间的免密访问
3. 关闭四个服务器上的防火墙，并禁用开机自启
4. 在四个节点上生成密钥，并获得公钥
5. 将四个公钥复制粘贴到四个服务器上的/root/.ssh/authorized_keys 中
6. 编辑四个服务器的 hosts 文件，使其包含从弹性公网地址到节点名称的映射
7. 检查以确保节点间的免密访问
8. 在主节点上拷贝 OpenJDK 安装包，并将安装包分发到三个子节点
9. 在四个节点上安装 OpenJDK，并修改四个节点的 profile 文件（新增配置）
10. 在主节点上安装 Hadoop，并进行必要的配置（环境变量、core-site.xml、hdfs-site.xml、yarn-site.xml、mapred-sit.xml、slaves 等文件）
11. 将主节点上的 Hadoop 包分发到其余子节点的/home/modules 目录（需要预先创建）下面
12. 配置四个节点的环境变量
13. 在四个节点上修改相应的权限
14. 主节点上执行格式化操作，然后启动 Hadoop

执行完毕，在四个节点上给出 JPS 命令，便得到了下面的截图：

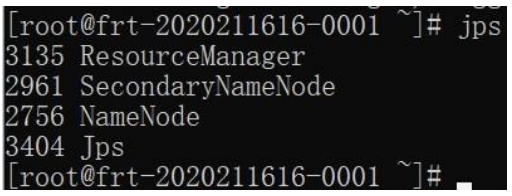


图 8.1: 主节点状态

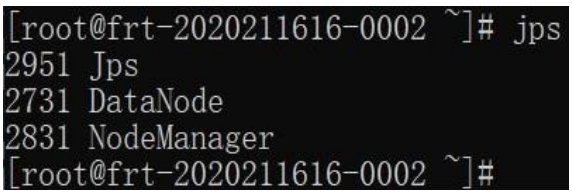


图 8.2: 子节点状态

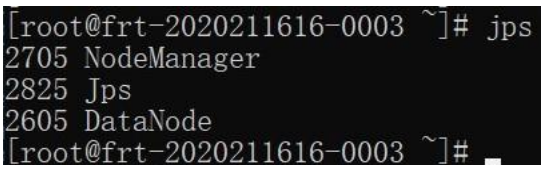


图 8.3: 子节点状态

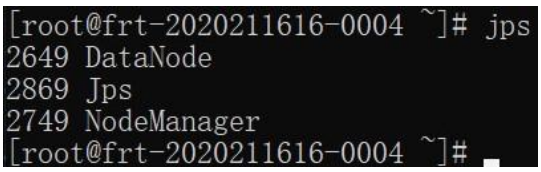


图 8.4: 子节点状态

现在来解释上图的含义。使用 JPS 命令可以列出正在运行的 Java 虚拟机的进程信息，图 8.1 中显示，主节点上有四个正在运行的进程：
（1）ID3135，名称 ResourceManager：该进程用于协调和管理整个 Yarn 集群，当应用程序对集群资源有需求时，该进程生效；
（2）ID2756，名称 NameNode：一般情况下，HDFS 中只包含一个 NameNode，该进程用于跟踪文件、管理文件，并保有文件的相关元数据；
（3）ID2961，名称 SecondaryNameNode：这是为了处理 NameNode 中文件元数据而设置的进程，提供周期检查点和清理任务；
（4）ID3404，名称 JPS：用于查看当前进程。

在子节点中，如图 8.2 所示，则有三个正在运行的进程：
（1）ID2951，名称 JPS：用于查看当前进程；
（2）ID2731，名称 DataNode：负责所在子节点的存储，并且以 heartbeat 机制向 NameNode 定时发送信息，以通知它是活动的；
（3）ID2831，名称 NodeManager：用于处理 ResourceManager 分配的任务。

HDFS 是用 Java 为 Hadoop 框架编写的分布式、可扩展且可移植的文件系统，从图 8.1 到图 8.4 四张图，我们可以直观地看出我们搭建的这个 HDFS 的特点：采用了主从架构，由一个 NameNode 和三个 DataNode 组成。其中，NameNode 负责管理文件系统的名字空间以及客户端对文件的访问；而 DataNode 则一般是一个节点有一个，负责管理所在节点上的存储。

## 1.2.2 创建并运行 maven 工程

在这一部分的实验中，我首先完成了：
1. IDEA、maven 等必要软件的安装
2. Maven 工程的创建，以及 pom.xml 配置文件的修改
3. 语言环境、编译环境等的修改和确定
4. 在华为云控制台上放开必要的端口（添加入方向规则）
5. 本地和服务器的 hosts 文件的修改
6. 编写 ExeHDFS 类，修改为含有名字和学号的程序

完成上面的内容后，得到了：
1. 一个 Hadoop 集群，其中一个主节点、三个子节点
2. 一个 maven 工程项目

然后运行 maven 工程，得到下面的内容：

```
View file:
log4j:WARN No appenders could be found for logger (org.apache.hadoop.metrics2.li
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more in
name: hdfs://123.249.126.97:8020/tmp, folder: true, size: 0
Upload file:
Upload successfully!
Write file:
hdfs://123.249.126.97/frt_2020211616.txt
Download file:
Download successfully!
View file:
name: hdfs://123.249.126.97:8020/frt_2020211616.txt, folder: false, size: 65
name: hdfs://123.249.126.97:8020/tmp, folder: true, size: 0
name: hdfs://123.249.126.97:8020/upload_2020211616.txt, folder: false, size: 65

Process finished with exit code 0
```

图 9：Java 运行结果（本人学号 2020211616）


图 10：下载的 txt 文件内容（本人学号 2020211616）

下面对这个运行结果进行分析。主程序先执行 testView()函数查看当前 HDFS 中的文件，如果为空则会输出“HDFS is empty.”，如果非空则会输出相应的内容，例如图 9 中的输出结果表示其中有一个 tmp 文件夹（size=0，说明为空文件夹）。然后执行 testUpload()函数，尝试上传本地文件，上传成功会输出相应的提示信息。再执行 testCreate()函数，在 HDFS 上创建一个文本文件（名为 frt_2020211616.txt），并写入指定信息。之后执行 testDownload()函数，下载指定文件（下载结果如图 10 所示）。最后再一次执行 testView()函数，可以看到此时 HDFS 上有三个文件，分别是：原本的 tmp 空文件夹、上传的 upload_2020211616.txt 文件、创建的 frt_2020211616.txt 文件。

## 1.2.3 实验 1.2 的结果分析与总结

这一部分的实验让我们初步体验了在 Hadoop 集群上完成 HDFS 文件的学些操作，并针对出现的问题，去查看相应的日志文件，来寻找解决方案。在这一部分的实验中，我先后遇到了下面几个印象深刻的问题：

1. Upload 失败：查看日志可以发现是权限问题，hadoop-2.7.7 文件夹的权限不知为何被关闭，故再次打开，上传成功；
2. Upload 失败：这次 upload 失败的原因与上次不同，查看日志发现是

NameNode 与 DataNode 所属的 clusterID 不同，这是在主节点多次执行格式化所导致的。通过摸索，我发现这个问题有两种处理方法：（1）找到子节点 dfs 文件夹中的 VERSION 文件，将其中的 clusterID 手动修改为与主节点一致；（2）直接删除四个节点的 dfs 文件夹，重新在主节点上执行格式化；

3. Upload 失败：这又是一次原因不同的 upload 失败，我将 50010 和 50020 两个端口放开，解决了这个问题；
4. Write 失败：查看日志，发现无法顺利写入到子节点，排查问题，发现是因为子节点与主节点之间的免密通信不充分，对此进行修正，解决了问题；
5. 每次重启服务器时，/etc/hosts 文件中会自动添加主机名解析，这可以通过注释掉/etc/cloud/cloud.cfg 文件中的 manage_etc_hosts:localhost 来解决这个问题。