# 分布式与并行计算项目设计报告书

### 班级

* 大数据1班

### 小组成员（学号）

* 赵娇（2018080910192）
* 夏雪力（2018080910147）
* 罗莎莎（2018080910086）
* 徐泽伟（2018080910155）

## 前言

  分布式是一种模型结构，区别于核心式，可以从字面理解为“分布在各处”  
分布式的目标是降低单个对象的重要度，从而提升整个系统的性能（稳定性，计算能力等等），不过代价是增加了数据传输量。  
 假如中国专门造了一台超级电脑，用来计算天气预报的数据，所有的电视台都从这台电脑获取数据，然后播放天气预报。那么这种模式就是核心式的，这台电脑处于核心位置，如果这台电脑坏了，不能工作了，那么所有电视台的天气预报都不能进行了。现在这样设计，每个省各自出一台普通电脑，然后这些电脑联合工作，一起计算天气预报，虽然每台电脑的计算能力远没有那台超级电脑强，但是它们协同工作，即使某台或者某几台电脑坏了，计算仍然能继续（坏了的那几台的计算工作将会转移到其他电脑上）。这样就降低了每台电脑的重要程度，使得天气预报系统更加稳定，不过这些电脑之间的协调就比较复杂了。

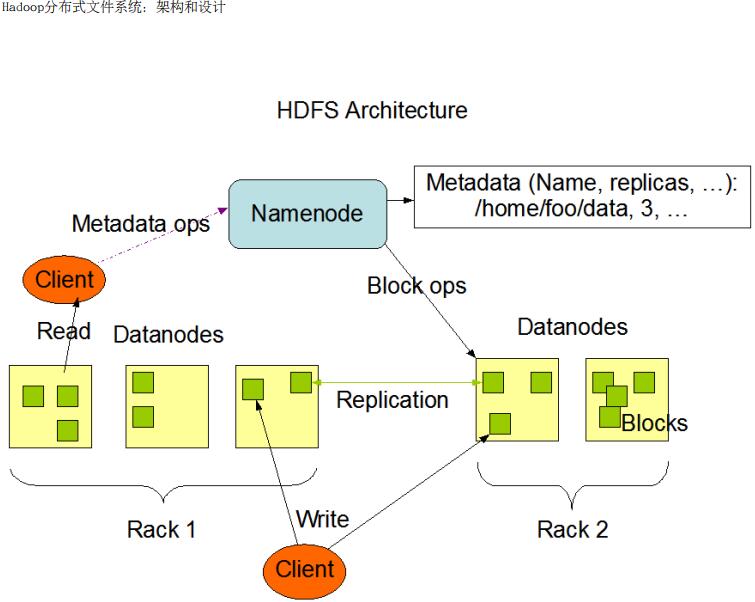
首先我们保证每台计算节点上分别有一个DataNode节点和NodeManager节点。因为都是计算节点，真正干活的，在数量上我们要保证。那么NameNode和ResourceManager是两个非常重要的管理者，我们客户端的请求，第一时间与NameNode和ResourceManager打交道。NameNode负责管理HDFS文件系统的元数据，客户端不管是读文件还是写文件，都要首先找到NameNode获取文件的元数据，再进行文件的操作。ResourceManager也是如此，它负责管理集群中的资源和任务调度，也可以把它视为“大数据操作系统”。客户端能否提交应用并运行，就看你的ResourceManager是否正常。

## 项目框架介绍

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 主机名 | IP地址 | 硬件性能 | namenode | datanode |
| master | 192.168.1.50 | CPU核数:6核  处理器:AMD锐龙5 4500U  运行内存:8GB | 启动 |  |
| slave1 | 192.168.1.57 | CPU核数:4核  处理器:酷睿™ i5-10300H  运行内存:8GB |  | 启动 |
| slave2 | 192.168.1.26 | CPU核数:4核  处理器:酷睿 ™ i5-1135G7  运行内存:16GB |  | 启动 |

一个HDFS集群主要由一个NameHode和很多个Datanode组成：Namenode管理文件系统的元数据，而Datanode存储了实际的数据。

HDFS采用master/slave架构。一个HDFS集群是由一个Namenode和一定数目的Datanodes组成。Namenode是一个中心服务器，负责管理文件系统的名字空间(namespace)以及客户端对文件的访问。集群中的Datanode一般是一个节点一个，负责管理它所在节点上的存储。HDFS暴露了文件系统的名字空间，用户能够以文件的形式在上面存储数据。从内部看，一个文件其实被分成一个或多个数据块，这些块存储在一组Datanode上。Namenode执行文件系统的名字空间操作，比如打开﹑关闭﹑重命名文件或目录。它也负责确定数据块到具体Datanode节点的映射。Datanode负责处理文件系统客户端的读写请求。在Namenode的统一调度下进行数据块的创建﹑删除和复制。



## 项目技术介绍

Hadoop项目主要包括以下四个模块

◆ Hadoop Common:

为其他Hadoop模块提供基础设施

◆ Hadoop HDFS:

一个高可靠、高吞吐量的分布式文件系统

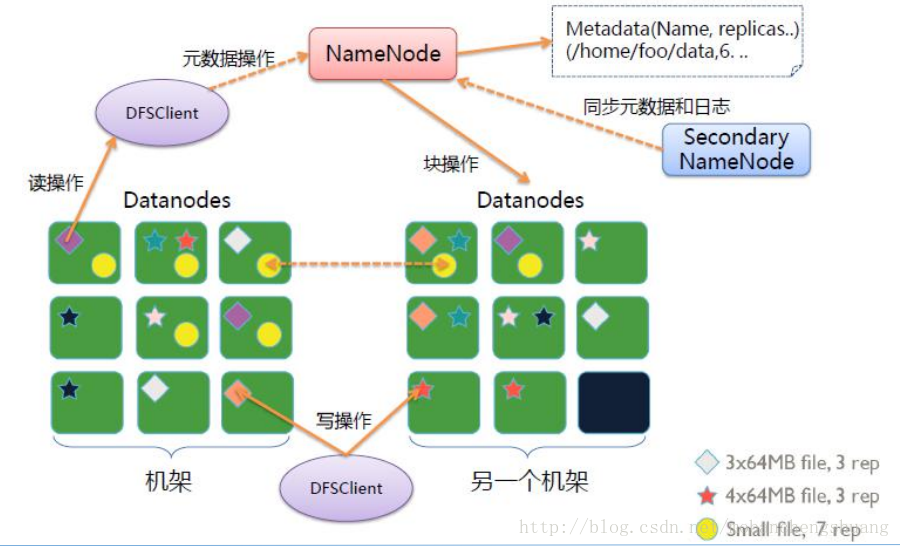
◆ Hadoop MapReduce:

一个分布式的离线并行计算框架

◆ Hadoop YARN:

一个新的MapReduce框架，任务调度与资源管理

**1.HDFS：hadoop Distributed FileSystem（分布式的文件存储系统）**

HDFS服务功能：

◆NameNode（NN）是主节点，存储文件的元数据如文件名，文件目录结构，文件属性（生成时间,副本数,文件权限），以及每个文件的块列表和块所在DataNode等。

◆DataNode（DN）在本地文件系统存储文件块数据，以及块数据的校验和。

◆Secondary NameNode（SNN） 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。

Namenode和Datanode被设计成可以在普通的商用机器上运行。这些机器一般运行着GNU/Linux操作系统(OS)。HDFS采用Java语言开发，因此任何支持Java的机器都可以部署Namenode或Datanode。由于采用了可移植性极强的Java语言，使得HDFS可以部署到多种类型的机器上。一个典型的部署场景是一台机器上只运行一个Namenode实例，而集群中的其它机器分别运行一个Datanode实例。这种架构并不排斥在一台机器上运行多个Datanode，只不过这样的情况比较少见。

集群中单一Namenode的结构大大简化了系统的架构。Namenode是所有HDFS元数据的仲裁者和管理者，这样，用户数据永远不会流过Namenode。

1. **MapReduce**

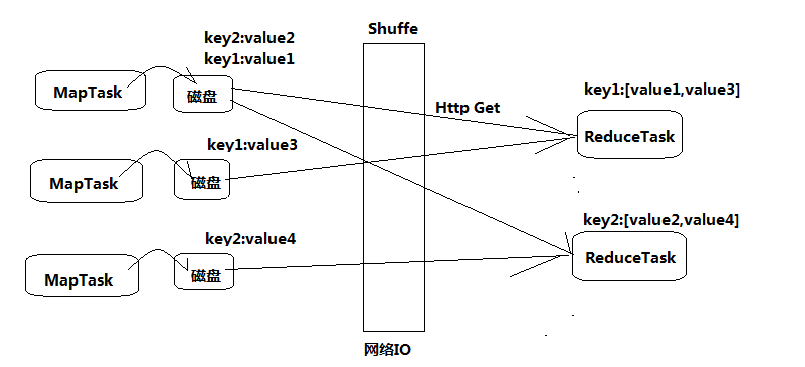
Hadoop Map/Reduce是一个使用简易的软件框架，基于它写出来的应用程序能够运行在由上千个商用机器组成的大型集群上，并以一种可靠容错的方式并行处理上T级别的数据集。

一个Map/Reduce作业(job)通常会把输入的数据集切分为若干独立的数据块，由map任务(task)以完全并行的方式处理它们。框架会对map的输出先进行排序，然后把结果输入给reduce任务。通常作业的输入和输出都会被存储在文件系统中。整个框架负责任务的调度和监控，以及重新执行已经失败的任务。

通常，Map/Reduce框架和分布式文件系统是运行在一组相同的节点上的，也就是说，计算节点和存储节点通常在一起。这种配置允许框架在那些已经存好数据的节点上高效地调度任务，这可以使整个集群的网络带宽被非常高效地利用。

Map/Reduce框架由一个单独的master JobTracker和每个集群节点一个slave TaskTracker共同组成。master负责调度构成一个作业的所有任务，这些任务分布在不同的slave上, master监控它们的执行，重新执行已经失败的任务。而slave仅负责执行由master指派的任务。

应用程序至少应该指明输入/输出的位置(路径)，并通过实现合适的接口或抽象类提供map和reduce函数。再加上其他作业的参数，就构成了作业配置(job configuration)。然后，Hadoop的job client提交作业〈jar包/可执行程序等）和配置信息给JobTracker，后者负责分发这些软件和配置信息给slave﹑调度任务并监控它们的执行，同时提供状态和诊断信息给job-client。

虽然Hadoop框架是用JavaTM实现的，但Map/Reduce应用程序则不一定要用Java来写。

◆将计算过程分为两个阶段：Map和Reduce

Map阶段并行处理输入数据 ；

Reduce阶段对Map结果进行汇总 ；

◆ Shuffle链接Map和Reduce两个阶段（Shuffle通俗的理解就是重新洗牌，打乱原有顺序）

Map Task将数据写到本地磁盘 ；

Reduce Task从每个Map Task上读取一份数据 ；

◆ 仅适合离线批处理

具有很好的容错性和扩展性 ；

适合简单的批处理任务 ；

◆ 缺点明显：

启动开销大，过多使用磁盘导致效率低下等；

1. **YARN（主要是负责硬件资源的合理调用）**

◆ YARN 总体上仍然是Master/Slave 结构，在整个资源管理框架中，ResourceManager 为Master，NodeManager 为Slave。

◆ ResourceManager 负责对各个NodeManager 上的资源进行统一管理和调度；

◆ 当用户提交一个应用程序时，需要提供一个用以跟踪和管理这个程序的ApplicationMaster（主管进程），它负责向ResourceManager 申请资源，并要求NodeManger 启动可以占用一定资源的任务。

◆ 由于不同的ApplicationMaster 被分布到不同的节点上，因此它们之间不会相互影响。

1. Web接口

NameNode和DataNode各自启动了一个内置的Web服务器，显示了集群当前的基本状态和信息。在默认配置下NameNode的首页地址是http://namenode-name:50070/这个页面列出了集群里的所有DataNode和集群的基本状态，这个Web接口也可以用来浏览整个文件系统（使用NameNode首页上的"Browse the file system"链接)。

## 项目实施过程

一、准备过程

硬件：三台装有ubuntu系统的电脑、一个交换机、三条网线、U盘

软件：hadoop2.8.5

Jdk-15.0.1

1. 搭建过程

1、创建hadoop用户

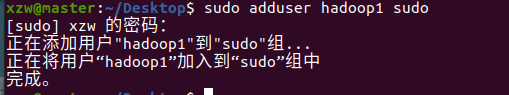
sudo adduser hadoop1

1

2

2、hadoop用户增加管理员权限

sudo adduser hadoop sudo



1. 切换到hadoop用户下

4、查看当前挂载

sudo fdisk -l

7

5、在/mnt目录下面新建一个udisk目录

sudo mkdir /mnt/udisk

9

6、载入U盘，挂载U盘

sudo mount /dev/sda4 /mnt/udisk

10

7、从U盘拷贝文档到指定目录

sudo cp /mnt/udisk/hadoop-2.8.5.tar.gz /home/hadoop1/hadoop-2.8.5.tar.gz

11

sudo cp /mnt/udisk/jdk-15.0.1\_linux-x64\_bin.tar.gz /home/hadoop1/jdk-15.0.1\_linux-x64\_bin.tar.gz12

8、解压打包文件到指定目录

sudo tar -zxvf /home/hadoop1/hadoop-2.8.5.tar.gz -C /usr/opt/hadoop

13

sudo tar -zxvf /home/hadoop1/jdk-15.0.1\_linux-x64\_bin.tar.gz -C /usr/local/java14

9、安装java环境

sudo update-alternatives --install /usr/bin/java java /usr/local/java/jdk-15.0.1/bin/java 300

15

sudo update-alternatives --install /usr/bin/javac javac /usr/local/java/jdk-15.0.1/bin/javac 300

16

10、配置HDFS的文件

cd /usr/opt/hadoop/hadoop-2.8.5/etc/hadoop

17

sudo cat /mnt/udisk/core-site.xml >> /usr/opt/hadoop/hadoop-2.8.5/etc/hadoop/core-site.xml

sudo cat /mnt/udisk/hdfs-site.xml >> /usr/opt/hadoop/hadoop-2.8.5/etc/hadoop/hdfs-site.xml

sudo cat /mnt/udisk/mapred-site.xml >> /usr/opt/hadoop/hadoop-2.8.5/etc/hadoop/mapred-site.xml.template

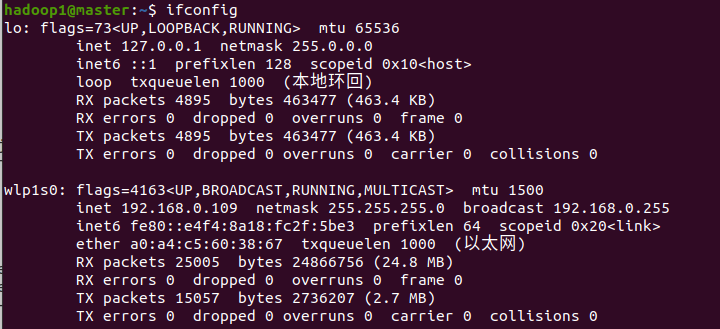
sudo cat /mnt/udisk/yarn-site.xml >> /usr/opt/hadoop/hadoop-2.8.5/etc/hadoop/yarn-site.xml

sudo cat /mnt/udisk/profile >> /etc/profile

sudo cat /mnt/udisk/bashrc >> ./.bashrc

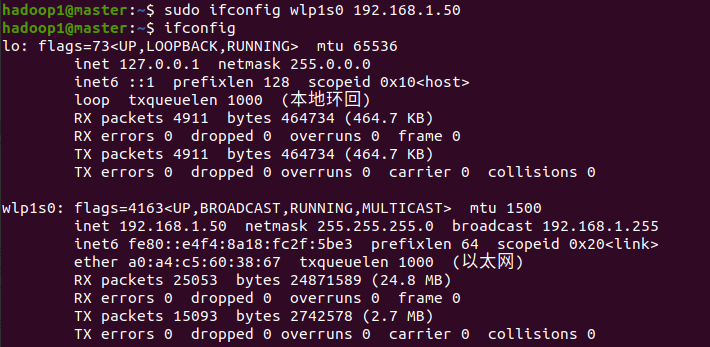
sudo cat /mnt/udisk/hadoop-env.sh >> /usr/opt/hadoop/hadoop-2.8.5/etc/hadoop/hadoop-env.sh

11、查看ip

ifconfig

12、修改静态ip

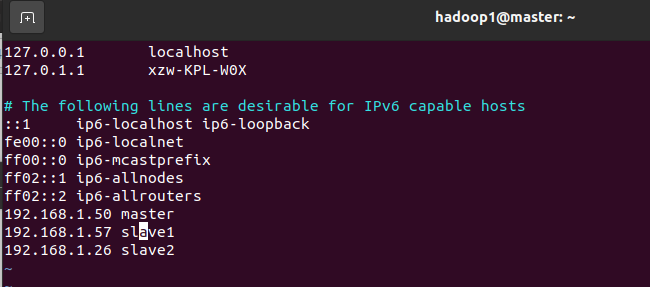
sudo ifconfig wlp1s0 192.168.1.50



13、修改自己所用节点的ip映射

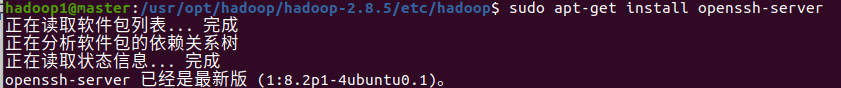
sudo vim /etc/hosts

29



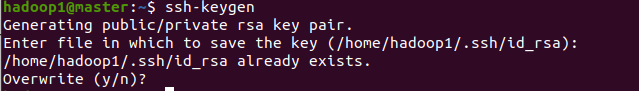
14、安装opeanssh-server

sudo apt-get install openssh-server



15、生成公钥

ssh-keygen

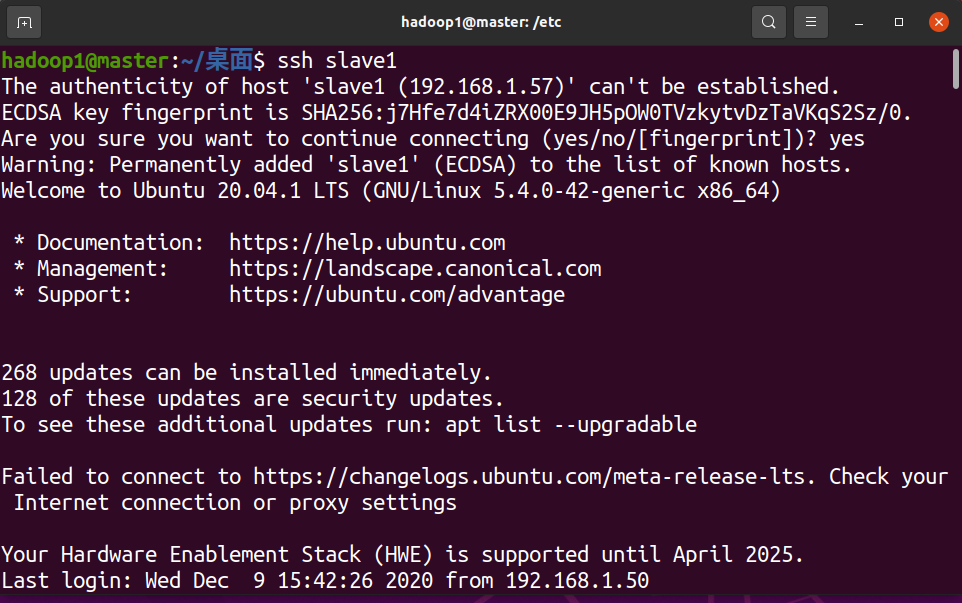


16、公钥发送实现免密登陆

ssh-copy-id slave1@192.168.1.26

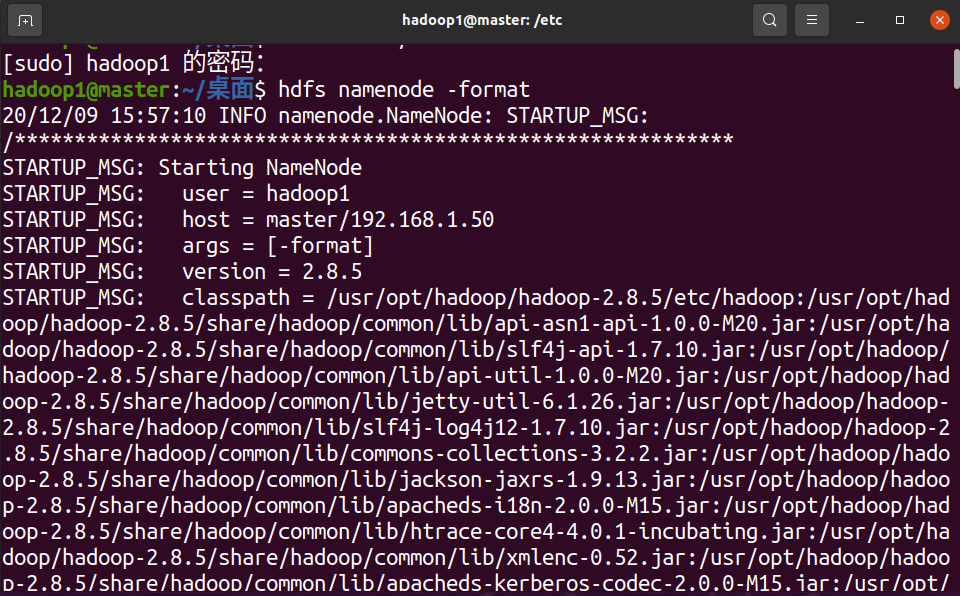
32

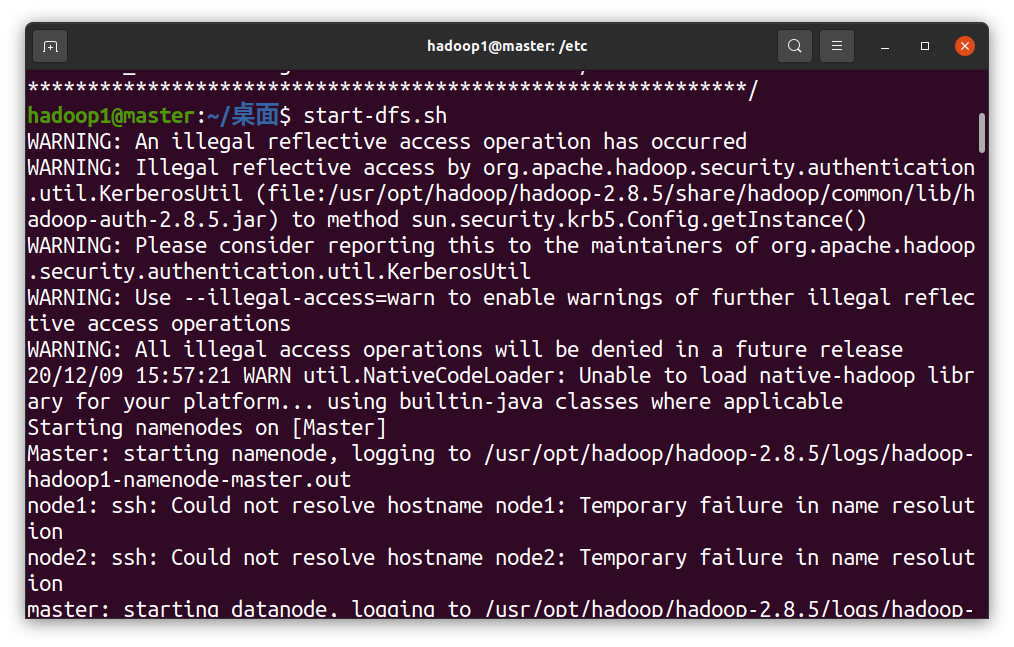
1. 用 ssh slave1 验证是否可以免密登录

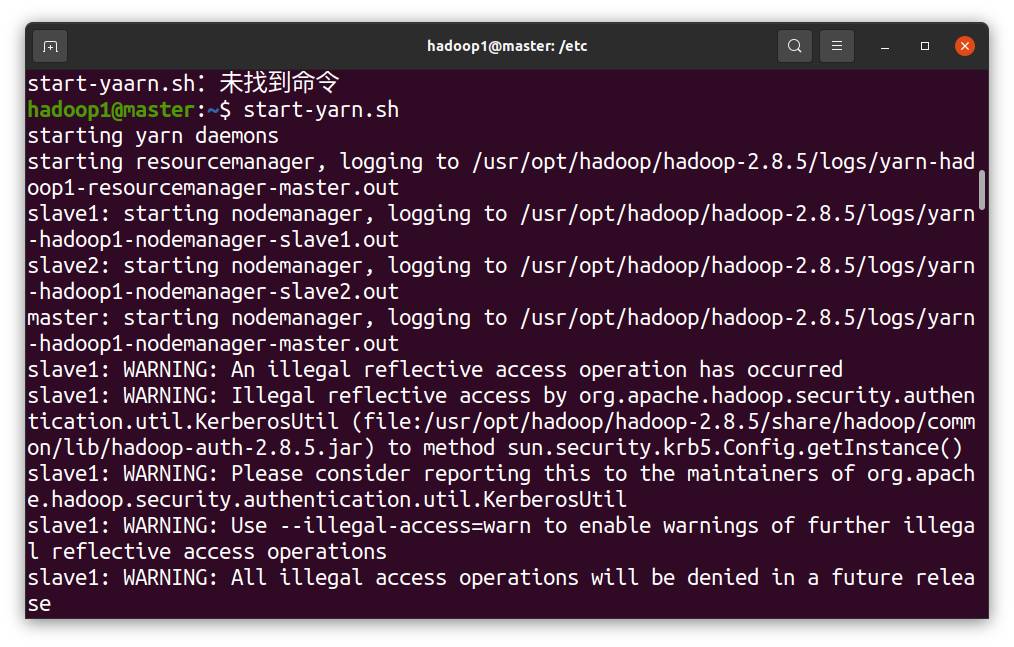


1. 首次启动 Hadoop 需要将 master 节点格式化

hdfs namenode -format

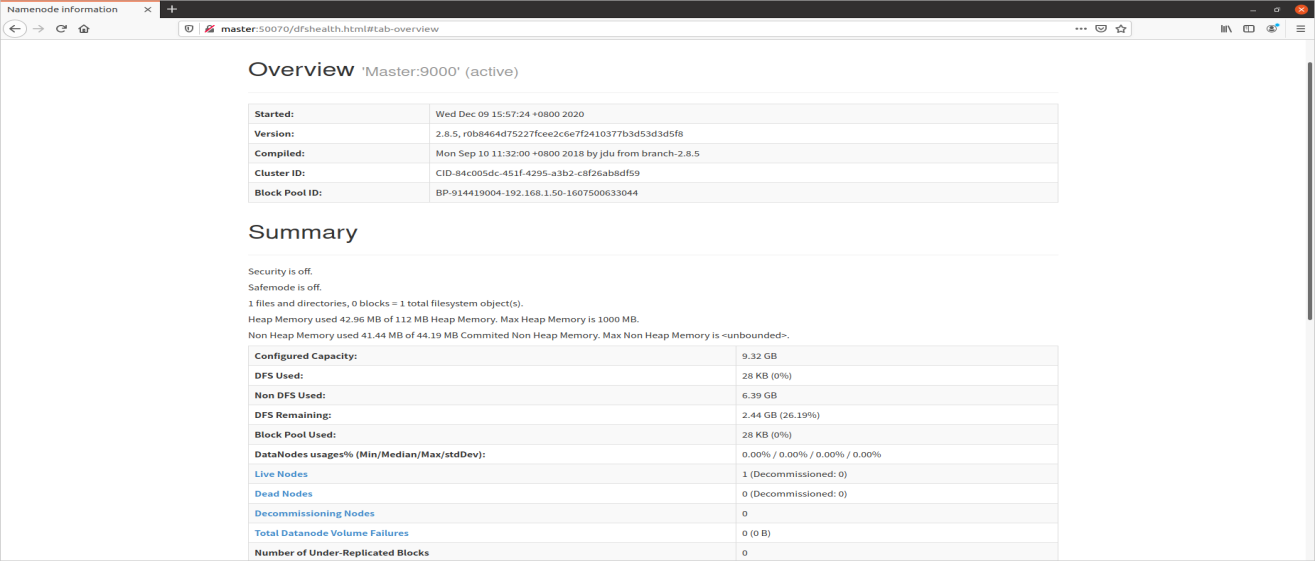


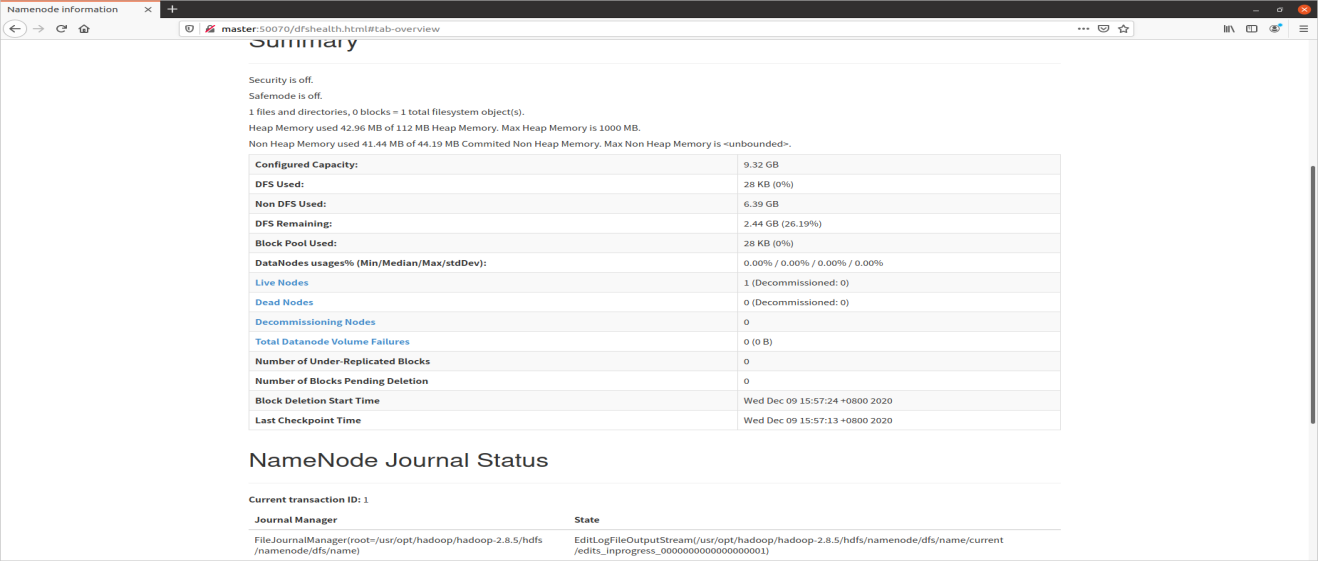
19、启动start-dfs.sh、start-yarn.sh



20、通过jps查看各个节点的启动进程

21、通过 Web 页面看到查看 DataNode 和 NameNode 的状态：<http://master:50070/>





## 项目结果分析和总结

本学期的课程为我们打开了一扇全新的大门，离开了虚拟机，给自己的电脑装上了Ubuntu系统，通过一学期的学习，更加深入了解并习惯了linux系统的使用，令我们印象最深刻的特别是通过ssh命令连接上别人的电脑并进行控制。

搭建中遇到的问题：

1.在配置免密之前一直没有成功，明明配了免密还要一直提示输入密码，这个问题是因为我在之前配置文件主机名映射的问题

  (1).首先cd ~ 到家目录 ls -la查看是否有.ssh配置文件

  (2).如果有rm -rf .ssh 删除配置文件重新生成

  (3).ssh-keygen -t rsa 生成密钥 然后一直回车

  (4).ssh-copy-id 192.168.1.50

例如：ssh-copy-id 192.168.1.50 ssh-copy-id 192.168.1.57(注意的是需要连接2台就得配置2台，配置包括本机)

2.启动之后一直不显示进程或进程缺少

(1).在配置文件没有书写问题时，cd / 到根目录 删除之前生成的文件 是hadoop1 切记其他虚拟机文件也得删除

(2).删除之后重新 格式化 hdfs namenode -format

(3).成功以后启动集群 start-all.ssh

本次项目是以成功安装Ubuntu系统为基础，搭建网络环境为保障，从最先安装和配置jdk开始，安装hadoop单机模式，到伪分布式，再到完全分布式，最后到实现HDFS可用。整个过程难度是逐层递增的。Hadoop作为大数据计算框架，所以核心的关键点就是分布式集群的搭建。

越学习越发现，有时候努力很重要，但方法更重要。知识之间是相关联的，学习的过程还需要我们不断的反思和实践总结，此次实验使我们受益匪浅。