《大规模分布式系统》实验报告

---HDFS 伪分布式部署

姓名：刘佰川 专业：计算机科学与技术（数据科学方向） 学号：16307130214

0. 实验环境

因为 Hadoop 官方真正支持作业的平台只有 Linux，故实验采取在 windows 系统上搭建虚拟机的方式来完成（VMware+Ubuntu18.04）。

1. 实验要求

I. 分析对比 HDFS 与传统的分布式/网络文件系统

II. 进行 HDFS 伪分布实际部署,报告实验过程

2. 实验过程

I. Hadoop 及 HDFS 简介

数据是时代的产物，随着时代的高速发展，数据呈爆发式地增长。存储和读取大量数据便成了问题。近年来，硬盘储存容量快速增加，而访问硬盘的速度却没有太大提升。一个想法就是从多个磁盘并行读取数据，加速数据的访问。Hadoop 正提供了一个稳定的共享存储和分析系统来解决这样的工作。

Hadoop 的核心是分布式文件系统（Hadoop Distributed File System ,HDFS）和 MapReduce。其存储功能由 HDFS 实现，分析功能由 MapReduce 实现。HDFS 有较高的读写速度、很好的安全性、容错性和可伸缩性；MapReduce 允许用户在不了解分布式系统底层细节的情况下开发并行应用程序，保证了分析和处理数据的高效性。Hadoop 使程序员可以在廉价的计算机集群上编写分布式并行程序，完成海量数据的存储与计算。

与普通的磁盘一样，HDFS 也有块的概念，相比于普通磁盘的 512Bytes，HDFS 块的默认大小为 64MB。但不同的是，在 HDFS 中，小于一个块大小的文件不会占据整个块的空间。设置较大的块是为了减小寻址开销，使文件传输的时间取决于磁盘的传输率。在物理结构上，计算机集群的多个节点构成了分布式文件系统，这些节点分为两类，名称节点（NameNode）和数据节点

（DataNode）。名称节点管理文件系统的命名空间，维护整个文件系统树及树内所有的文件和索引目录，因此客户端只有访问名称节点才能找到请求的文件块所在的位置并进行读取。数据节点的工作是存储并提供块的服务，并且名称节点不会永久保存块的位置，数据节点需要定时向名称节点发送它们存储的块的列表。在存储时，由名称节点分配存储位置，客户端把数据写入相应的数据节点；在读取时，客户端从名称节点获得数据节点和文件块的映射关系，访问相应位置的文件块。

由 HDFS 的工作机制可知，名称节点是非常重要的部分。如果运行名称节点的机器被损坏了，我们就无法定位到数据节点的块来重构文件，导致所有文件丢失，Hadoop 提供了两种机制来确保名称节点能够经受住故障。

一是在多个文件系统写入名称节点的持久化状态，如同时在本地磁盘和远程 NFS 挂载上写入。二是使用第二名称节点（Secondary NameNode），其的重要作用是定期通过编辑日志（EditLog）合并命名空间镜像（FsImage），以防编辑日志过大。第二名称节点会定期和名称节点通信，从名称节点获取 FsImage 和 EditLog 并执行合并操作，相当于第二名称节点是名称节点的一个检查点（checkpoint），能够周期性备份名称节点的元数据信息，如果名称节点发生故障，则可以利用第二名称节点进行恢复。但是，第二名称节点的状态是滞后于名称节点的，如果在第二名称节点向名称节点获取 FsImage 和 EditLog 后且在第二名称节点向名称节点传输 FsImage 和 EditLog 前发生故障，必然会损失一些数据。这种情况下，一般把存在 NFS 上的名称节点元数据复制到二级名称节点上并将其作为新的名称节点运行。

II. HDFS VS NFS（网络文件系统）

分布式文件系统是指文件系统是指文件系统管理的物理存储资源不一定直接连接在本地节点上，而是通过计算机网络与节点相连。分布式文件系统的设计基于客户机\服务器模式。HDFS、NFS 都是常见的分布式文件系统。

前面已经简要介绍了 HDFS，现介绍 NFS 及其与 HDFS 文件系统的差异。

NFS 是 Network File System 的缩写，其最大的功能是可以通过网络，让不同的机器、操作系统可以彼此共享文件。用户可以联结到共享计算机并像访问本地硬盘一样访问共享计算机上的文件。管理员可以建立远程系统上文件的访

问，以至于用户感觉不到他们是在访问远程文件。

其差异主要在于：HDFS 旨在承受失败，NFS 没有任何内置的容错功能，HDFS 在设计之初就充分考虑了实际应用环境的特点，在普通服务器上，硬件出错是一种常态而非异常，因此 HDFS 设计了快速检测硬件故障和自动恢复的机制来保证实现数据的完整性；除容错外，HDFS 确实支持多个文件副本。这消除了（或简化）许多客户端访问单个文件的常见瓶颈，由于文件具有多个副本，因此在不同的物理磁盘上，读取性能的缩放比 NFS 更好；NFS 文件始终都只是存储在文件服务器上，存储的容量不可能突破物理服务器的上限，而 HDFS 采取计算机集群，文件的存储容量显著变大；HDFS 采用了“一次写入、多次读取”的简单文件模型，并且不支持多用户写入及任意修改文件。HDFS 只允许一个文件有一个写入者，不允许多个用户对同一个文件执行写操作，而且只允许对文件执行追加操作，不能执行随机写操作，NFS 由管理员控制客户端的读写权限，但对文件的读写模式没有 HDFS 这样严格的控制。

III. Hadoop 安装与 HDFS 伪分布式部署

配置 ssh

将公钥加入授权，实现无密码登陆

cat ./id_rsa.pub >> ./authorized_keys

安装 JDK

选择最新版本 jdk-8u201-linux-x64.tar.gz

创建文件夹：sudo mkdir -p /usr/local/java

移动下载的文件：sudo mv /home/junjin/jdk-8u201-linux-x64.tar.gz /usr/local/java/

进入目录解压：

cd /usr/local/java

sudo tar xvzf jdk-8u131-Linux-x64.tar.gz

删除安装包：sudo rm jdk-8u181-linux-x64.tar.gz

配置 JDK

打开环境变量的配置文件：sudo vim /etc/profile

```
export JAVA_HOME=/usr/local/java/jdk1.8.0_201
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:${JRE_HOME}/bin:$PATH
```

重新加载环境变量的配置文件：source /etc/profile

检测：java -version

```
junjin@ubuntu:~$ java -version
java version "1.8.0_201"
Java(TM) SE Runtime Environment (build 1.8.0_201-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.201-b09, mixed mode)
```

安装 Hadoop

解压：sudo tar -zxf ~/Downloads/hadoop-2.9.1.tar.gz -C /usr/local

删除安装包：rm ~/Downloads/hadoop-2.9.1.tar.gz

重命名修改权限：

cd /usr/local/

sudo mv ./hadoop-2.9.1/ ./Hadoop

检测：

```
junjin@ubuntu:~$ /usr/local/hadoop/bin/hadoop version
Hadoop 2.9.1
Subversion https://github.com/apache/hadoop.git -r e30710aea4e6e55e69372929106c
f119af06fd0e
Compiled by root on 2018-04-16T09:33Z
Compiled with protoc 2.5.0
From source with checksum 7d6d2b655115c6cc336d662cc2b919bd
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-
```

单机配置：无需做任何东西，上面安装成功即是默认的单机模式

伪分布式配置

对 core-site.xml 和 hdfs-site.xml 进行修改。

```
<configuration>
        <property>
                <name>hadoop.tmp.dir</name>
                <value>/usr/local/hadoop/tmp</value>
                <description>Abase for other temporary directories.</descriptio
n>
        </property>
        <property>
                <name>fs.defaultFS</name>
                <value>hdfs://localhost:9000</value>
        </property>
</configuration>
<local/hadoop/etc/hadoop/core-site.xml" 29L, 1028C          27,37-51      Bot
```

```
<configuration>
        <property>
                <name>dfs.replication</name>
                <value>1</value>
        </property>
        <property>
                <name>dfs.namenode.name.dir</name>
                <value>/usr/local/hadoop/tmp/dfs/name</value>
        </property>
        <property>
                <name>dfs.datanode.data.dir</name>
                <value>/usr/local/hadoop/tmp/dfs/data</value>
        </property>
</configuration>
```

切换回 hadoop 主目录：/usr/local/Hadoop

NameNode 的格式化：./bin/hdfs namenode -format

开启 NameNode 和 DataNode 守护进程：./sbin/start-dfs.sh

验证：

```
junjin@ubuntu:/usr/local/hadoop$ jps
3040 Jps
2344 SecondaryNameNode
1935 NameNode
2751 DataNode
```

```
junjin@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -ls /user/hadoop/input
Found 8 items
-rw-r--r--   1 junjin supergroup       7861 2019-03-04 17:54 /user/hadoop/input
/capacity-scheduler.xml
-rw-r--r--   1 junjin supergroup       1028 2019-03-04 17:54 /user/hadoop/input
/core-site.xml
-rw-r--r--   1 junjin supergroup      10206 2019-03-04 17:54 /user/hadoop/input
/hadoop-policy.xml
-rw-r--r--   1 junjin supergroup       1069 2019-03-04 17:54 /user/hadoop/input
/hdfs-site.xml
-rw-r--r--   1 junjin supergroup        620 2019-03-04 17:54 /user/hadoop/input
/httpfs-site.xml
-rw-r--r--   1 junjin supergroup       3518 2019-03-04 17:54 /user/hadoop/input
/kms-acls.xml
-rw-r--r--   1 junjin supergroup       5939 2019-03-04 17:54 /user/hadoop/input
/kms-site.xml
-rw-r--r--   1 junjin supergroup        690 2019-03-04 17:54 /user/hadoop/input
/yarn-site.xml
junjin@ubuntu:/usr/local/hadoop$ ./bin/hadoop jar ./share/hadoop/mapreduce/hado
op-mapreduce-examples-2.9.1.jar grep /user/hadoop/input output 'dfs[a-z]+'
```

```
junjin@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -cat output/*
1       dfsadmin
```

结束进程：./sbin/stop-dfs.sh

单机模式这种模式是指在一台单机上运行，没有分布式文件系统，而是直接读写本地操作系统的文件系统。伪分布式运行模式也是在一台单机上运行，但用不同的 JAVA 进程模仿分布式运行中的各类结点。没有所谓的在多台机器上进行真正的分布式计算，故称为"伪分布式"。真正的完全分布式模式是由 3 个及以上的实体机或者虚拟机组件的机群。

3. 实验问题

在配置环境的过程中出现两次报错。

第一次是在开始进程的时候报错：没有 JAVA-HOME

解决：

cd etc/hadoop/

vim hadoop-env.sh

export JAVA_HOME=/usr/local/java/jdk1.8.0_201

第二次是 datanode 未启动，因为在配置 core-site.xml 和 hdfs-site.xml 两个文件时，手动出错，所以多次修改，然后多次使用。每次对 NameNode 格式化的时候都会生成新的 namespaceID，但是在 hadoop.tmp.dir 目录下的 DataNode 还是保留上次的 namespaceID，因为 namespaceID 的不一致，而导致 DataNode 无法启动，所以手动删除。

```
junjin@ubuntu:/usr/local/hadoop$ jps
24305 Jps
23783 NameNode
24185 SecondaryNameNode
junjin@ubuntu:/usr/local/hadoop$
```

解决：

rm -rf data

4. 参考资料

《大数据技术原理与应用》 林子雨 chap2-3

《Hadoop 权威指南》 Tom White chap1-3

https://blog.csdn.net/weixin_42001089/article/details/81865101

https://448230305.iteye.com/blog/2078025

https://www.cnblogs.com/me80/p/7464125.html

https://stackoverrun.com/cn/q/7054377

http://www.jefflei.com/post/1052.html

https://www.cnblogs.com/liango/p/7116620.html