学　　号：16307110435

DATA130015 大规模分布式系统

实 验 报 告

院　　系：　大数据学院

专　　业：　大数据科学与技术

姓　　名：　张霁雯

学　　号：　16307110435

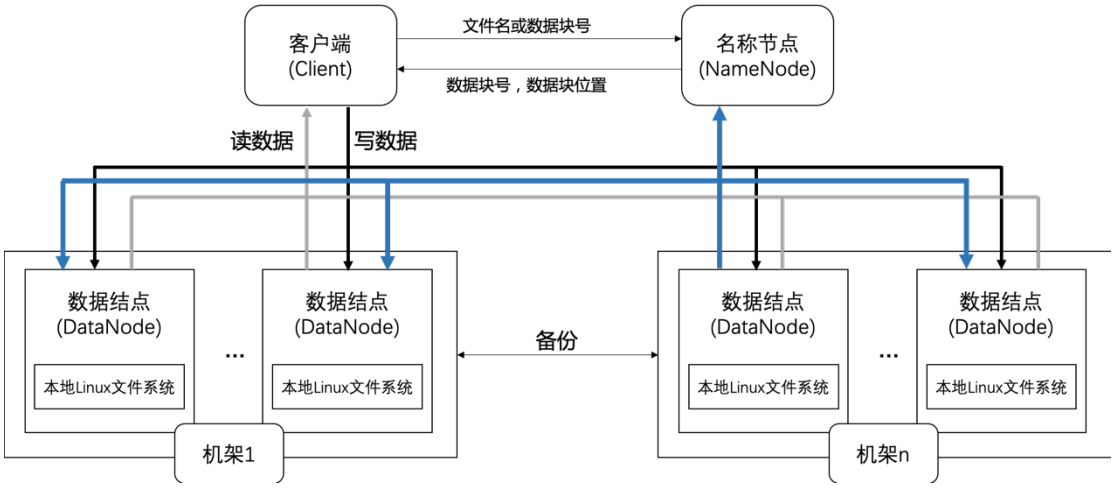
完 成 日 期：　2020 年 3 月 3 日

对 HDFS 与传统的分布式/网络文件系统的分析

Hadoop 是 Apache 软件基金会旗下的一个开源分布式计算平台，为用户提供了系统底层细节和透明的分布式基础架构。由于 Hadoop 是用 Java 语言开发的，因此具有良好的跨平台特性，适宜部署在廉价的计算机集群中。Hadoop 分布式文件系统(HDFS)和 MapReduce 构成了 Hadoop 的核心。

HDFS 是针对谷歌文件系统(Google File System, GFS)的开源实现，是面向普通硬件环境的分布式文件系统。它具有较高的读写速度、很好的容错性和可伸缩性，非常适合大规模数据集上的应用。其冗余数据的存储方式很好的保证了数据的安全性。

HDFS 采用了主从(Master/Slave)结构模型**(图 1)**，一个 HDFS 集群是由一个 NameNode 和若干个 DataNode 组成的。其中 NameNode 作为主服务器，管理文件系统的命名空间和客户端对文件的访问操作；集群中的 DataNode 管理存储的数据。HDFS 具有容错性，每份数据都会有多个副本，默认会有三个。



图片 1：HDFS 体系结构

网络文件系统(NFS)准确的来说并不是一个文件系统，而是由 SUN 公司研制的访问远程文件系统的网络协议。它用于服务器和客户机之间文件访问和共享的通信，从而使客户机能够远程地访问保存在存储设备上的数据。

NFS 的工作原理是使用客户端/服务器架构**(图 2)**，由一个客户端程序和服务器程序组成。服务器程序向其他计算机提供对文件系统的访问，其过程称为输出。NFS 客户端程序对共享文件系统进行访问时，把它们从 NFS 服务器中“输送”出来。它的特点是能够提供透明文件访问以及文件传输；容易扩充新的资源或软件，不需要改变现有的工作环境；性能高，可灵活配置。

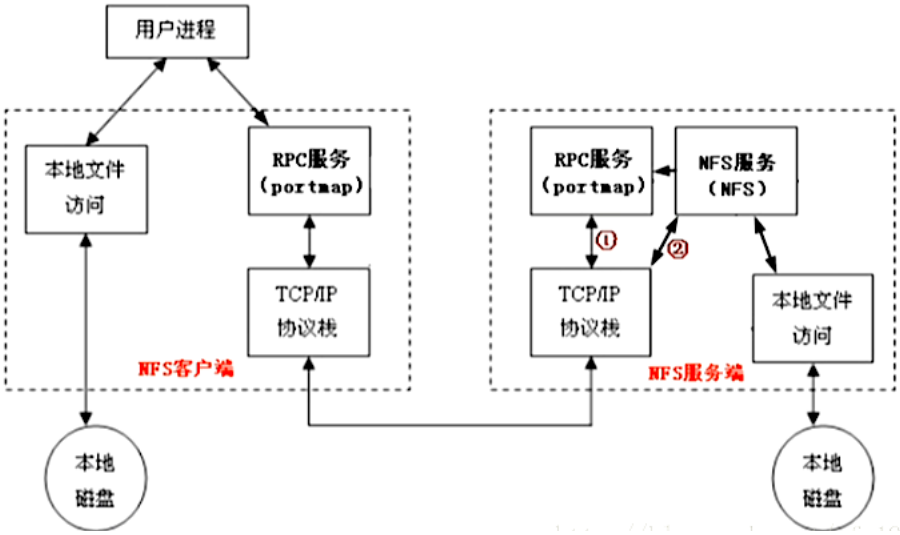
NFS 支持的功能很多，而不同的功能都会使用不同的程序来启动，每启动一个功能就会启用一些端口来传输数据，因此，NFS 的功能所对应的端口无法固定，会造成 NFS 客户端和 NFS 服务端通信障碍。RPC 服务用于解决这个困扰。它会记录每个 NFS 功能所对应的端口号，并且在 NFS 客户端请求时将端口和功能对应的信息传递给请求的 NFS 客户端。

当访问程序通过 NFS 客户端向 NFS 服务器端存取文件时，数据流程大致如下：1)用户访问网站程序，由程序在 NFS 客户端发出存取请求，它的 RPC 服务向服务器端的 RPC 服务发出请求；2)服务器端的 RPC 服务找到对应的已注册的 NFS 端口，通知 NFS 客户端的 RPC 服务；3)此时 NFS 客户端获取到正确的端口，并与 NFS daemon 联机存取数；4)NFS 客户端

把数据存取成功后，返回给前端访问程序，告知用户存取结果，作为网站用户，就完成了一次存取操作。因此，FNS 的各项功能都需要向 RPC 服务注册，所以要先启动 RPC 服务，后启动 NFS 服务。

NFS 为了提高磁盘读写效率，一般以数据块为单位进行传输，而不是以字节为单位。HDFS 也同样采用了块的概念，默认的一个块大小是 64MB。在 HDFS 中的文件会被拆分为多个块，每个块作为独立的单元进行存储。而普通文件系统的块一般只有几千字节(byte)。可以看出，HDFS 在块的大小设计上明显要大于普通文件系统。之所以这么做，是因为 HDFS 要最小化寻址开销：这不仅包括磁盘寻址开销，还包括数据块的定位开销。设计一个比较大的块，可以将上述寻址开销分摊到比较多的数据中，降低单位数据的寻址开销。因此 HDFS 在块的大小设计上明显要大于普通文件系统，以期在处理大规模文件时能获得更好的性能。因此 HDFS 具有以下几个特点：支持大规模文件存储；简化了系统设计，方便了元数据的管理；适合数据备份，每个文件快都可以冗余存储到多个节点上。

当然，HDFS 特殊的设计也有一些局限性，主要包括：不适合低延迟数据访问，因为 HDFS 是面对大规模数据批量处理设计的，采用流式数据读取，延迟较高；无法高效存储大量小文件，过多的小文件会导致 NameNode 寻找元数据的效率降低，此外在使用 MapReduce 的时候会产生过多的 Map 任务，增加线程管理开销；不支持多用户写入以及任意修改文件。



图片 2：NFS 体系结构

HDFS 和 NFS 最大的区别就在于容错性。HDFS 设计初衷就是容错，而 NFS 没有内置任何容错功能。除了容错性以外，HDFS 的多副本机制减轻了常见的多客户端访问单个文件的瓶颈问题。因为文件有多个副本在不同的物理磁盘上，使得一个文件的数据能够在不同的数据结点上实现并发访问，大大提高了数据的访问速度，所以 HDFS 的读取性能会优于 NFS。但是由于在 HDFS 中，文件的修改还涉及对文件副本的修改，因此在对数据的写入、修改操作上不如 NFS 快捷方便。

HDFS 伪分布实际部署过程

环境：阿里云 ECS Ubuntu16.04 + JDK7　　部署版本：Hadoop 2.9.2

【创建 hadoop 用户】

```
1. sudo useradd -m hadoop -s /bin/bash
2. sudo passwd hadoop  # 此处需要输入密码
3. sudo adduser hadoop sudo
```

【准备工作：更新 apt，安装 ssh server】

```
1. sudo apt-get update
2. # 防止后续安装 JDK 失败
3. sudo apt-get install python-software-properties
4. sudo apt-get install software-properties-common
5. sudo add-apt-repository ppa:openjdk-r/ppa
6. sudo apt-get update
```

```
1. sudo apt-get install openssh-server
2. # 配置 ssh 免密登陆
3. ssh localhost; exit
4. cd ~/.ssh/
5. ssh-keygen -t rsa
6. cat ./id_rsa.pub >> ./authorized_keys
```

【安装 Java 环境】

```
1. # 下载 JDK7
2. sudo apt-get install openjdk-7-jre openjdk-7-jdk
3. # 查看 java 安装路径，添加到系统路径中
4. dpkg -L openjdk-7-jdk | grep '/bin/javac'
5. vim ~/.bashrc
6. export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
7. source ~/.bashrc
8. # 测试安装是否成功
9. echo $JAVA_HOME
10. java -version
```

代码运行结果：

```
hadoop@iZ2zebcud10fnwn2l31klfZ:~$ dpkg -L openjdk-7-jdk | grep '/bin/javac'
/usr/lib/jvm/java-7-openjdk-amd64/bin/javac
hadoop@iZ2zebcud10fnwn2l31klfZ:~$ vim ~/.bashrc
hadoop@iZ2zebcud10fnwn2l31klfZ:~$ source ~/.bashrc
hadoop@iZ2zebcud10fnwn2l31klfZ:~$ echo $JAVA_HOME
/usr/lib/jvm/java-7-openjdk-amd64
hadoop@iZ2zebcud10fnwn2l31klfZ:~$ java -version
java version "1.7.0_95"
OpenJDK Runtime Environment (IcedTea 2.6.4) (7u95-2.6.4-3)
OpenJDK 64-Bit Server VM (build 24.95-b01, mixed mode)
```

【安装 Hadoop 2.9.2】

```
# 下载 hadoop2.9.2 安装包
wget http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-2.9.2/hadoop-2.9.2.tar.gz
sudo tar -zxf hadoop-2.9.2.tar.gz -C /usr/local  # 解压
cd /usr/local/
sudo mv ./hadoop-2.9.2/ ./hadoop
sudo chown -R hadoop ./hadoop  # 给用户 hadoop 提权限
# 测试安装是否成功
cd /usr/local/hadoop
./bin/hadoop version
```

代码运行结果：

```
hadoop@iZ2zebcud10fnwn2l31klfZ:/usr/local$ cd /usr/local/hadoop
hadoop@iZ2zebcud10fnwn2l31klfZ:/usr/local/hadoop$ ./bin/hadoop version
Hadoop 2.9.2
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r 826afbeae31ca687bc2f8471dc841b66ed2c6704
Compiled by ajisaka on 2018-11-13T12:42Z
Compiled with protoc 2.5.0
From source with checksum 3a9939967262218aa556c684d107985
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-2.9.2.jar
```

【Hadoop 伪分布式配置】

1. 修改配置文件

```
cd /usr/local/hadoop/etc/hadoop/
vim core-site.xml
vim hdfs-site.xml
```

(1)将 core-site.xml 文件修改为

```
<configuration>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>file:/usr/local/hadoop/tmp</value>
        <description>Abase for other temporary directories.</description>
    </property>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

(2)将 hdfs-site.xml 文件修改为

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
```

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

2. NameNode 格式化

```
1. cd /usr/local/hadoop
2. ./bin/hdfs namenode -format
```

代码运行结果：

```
20/03/03 17:30:38 INFO namenode.FSImage: Allocated new BlockPoolId: BP-961987996-172.17.42.189-1583227838561
20/03/03 17:30:38 INFO common.Storage: Storage directory /usr/local/hadoop/tmp/dfs/name has been successfully
 formatted.
20/03/03 17:30:38 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/local/hadoop/tmp/dfs/name/curre
nt/fsimage.ckpt_0000000000000000000 using no compression
20/03/03 17:30:38 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/hadoop/tmp/dfs/name/current/fsim
age.ckpt_0000000000000000000 of size 324 bytes saved in 0 seconds .
20/03/03 17:30:38 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
20/03/03 17:30:38 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at iZ2zebcud10fnwn2l31klfZ/172.17.42.189
************************************************************/
```

3. 开启 NameNode 和 DataNode 守护进程

```
1. cd /usr/local/hadoop
2. ./sbin/start-dfs.sh
3.
4. # 出现 Error: JAVA_HOME is not set and could not be found.
5. vim /usr/local/hadoop/etc/hadoop/hadoop-env.sh
6. # 将 JAVA_HOME=${JAVA_HOME}改为
7. JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
8. ./sbin/start-dfs.sh
```

代码运行结果：

```
hadoop@iZ2zebcud10fnwn2l31klfZ:/usr/local/hadoop$ ./sbin/start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-namenode-iZ2zebcud10fnwn2l3
1klfZ.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hadoop-datanode-iZ2zebcud10fnwn2l3
1klfZ.out
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is SHA256:qFHiB/TyfmFkHC1Tljzt8X9nl2qW2OdP/4IiqF8hUKU.
Are you sure you want to continue connecting (yes/no)? yes
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-secondarynamenode-iZ
2zebcud10fnwn2l31klfZ.out
hadoop@iZ2zebcud10fnwn2l31klfZ:/usr/local/hadoop$ jps
11240 Jps
10803 NameNode
11129 SecondaryNameNode
10949 DataNode
```

【运行 Hadoop 伪分布式实例】

```
# 创建用户目录
./bin/hdfs dfs -mkdir -p /user/hadoop
# 将 etc/hadoop 文件夹下的配置文件复制到 input 文件夹中
./bin/hdfs dfs -mkdir input
./bin/hdfs dfs -put ./etc/hadoop/*.xml input
# 查看文件列表
./bin/hdfs dfs -ls input

# 执行字数统计测试样例
./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar grep input output 'dfs[a-z.]+'
# 查看 hdfs 中的输出结果
./bin/hdfs dfs -cat output/*
```

代码运行结果：

```
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=219
        File Output Format Counters
                Bytes Written=77
hadoop@iZ2zebcud10fnwn2l31klfZ:/usr/local/hadoop$ ./bin/hdfs dfs -cat output/*
1       dfsadmin
1       dfs.replication
1       dfs.namenode.name.dir
1       dfs.datanode.data.dir
```