

## PartI: 对比 HDFS 与传统的分布式/网络文件系统

### 一. 定义

1. 分布式文件系统(Distributed File System): 分布式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(可简单的理解为一台计算机)相连。分布式文件系统的设计基于客户机/服务器模式。一个典型的网络可能包括多个供多用户访问的服务器。另外, 对等特性允许一些系统扮演客户机和服务器的双重角色<sup>1</sup>。
2. 网络文件系统(Network File System): 网络文件系统是基于 UDP/IP 协议的应用, 其实现主要是采用远程过程调用 RPC 机制, RPC 提供了一组与机器、操作系统以及低层传送协议无关的存取远程文件的操作。RPC 采用了 XDR 的支持。XDR 是一种与机器无关的数据描述编码的协议, 他以独立与任意机器体系结构的格式对网上传送的数据进行编码和解码, 支持在异构系统之间数据的传送<sup>2</sup>。
3. Hadoop 分布式文件系统 HDFS(Hadoop Distributed File System): HDFS 是指被设计成适合运行在通用硬件上的分布式文件系统。HDFS 采用了主从结构模型, 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<sup>3</sup>。

### 二. HDFS 的优势

1. HDFS 是一个高度容错性的系统, 适合部署在廉价的机器上。整个 HDFS 系统由数百乃至数千个存储着文件数据片段的服务器组成, 每一个组成部分都很可能出现故障。故障的检测和自动快速恢复是 HDFS 的一大优势。
2. HDFS 能提供高吞吐量的数据访问, 非常适合大规模数据集上的应用。HDFS 被设计成适合批量处理且支持大文件的系统。而且它提供很高的聚合数据带宽, 一个集群中支持数百个节点, 一个集群中还应该支持千万级别的文件。运行在 HDFS 之上的程序有大量的数据集。典型的 HDFS 文件大小是 GB 到 TB 的级别。
3. HDFS 放宽了一部分 POSIX 约束, 来实现流式读取文件系统数据的目的。HDFS 的重点是在数据吞吐量, 而不是数据访问的反应时间, POSIX 的很多硬性需求对于 HDFS 应用都是非必须的, 去掉 POSIX 一小部分关键语义可以实现流式读取文件系统数据的目的<sup>4</sup>。

### 三. HDFS 与 linux 文件系统的区别<sup>5</sup>

---

<sup>1</sup> 王雪涛, 刘伟杰. 分布式文件系统[J]. 科技信息: 学术版(11):416-417.

<sup>2</sup> 沈健, 王梦龙主编; 毕雨, 朱斌峰编著. Linux 系统管理: 华东理工大学出版社, 2014.08: 第 103 页

<sup>3</sup> Hadoop 学习之 HDFS 文件读取. CSDN[引用日期 2015-07-05]

<sup>4</sup> 徐志阳, 张文培, 侯青, et al. HDFS 在多用户并发访问系统中的应用研究[J]. 电脑知识与技术, v.12(8):249-252.

<sup>5</sup> 分布式文件系统(HDFS) 与 linux 系统文件系统关系

[https://blog.csdn.net/qq\\_32041579/article/details/77752733](https://blog.csdn.net/qq_32041579/article/details/77752733)

1. HDFS 的 block 对应 linux 文件,linux 文件系统的 block 对应物理磁盘的 block。
2. HDFS 的最小读写单位是 Linux 系统文件, Linux 文件系统的块读写单位是一块磁盘大小。
3. HDFS 的单个文件是由多个小文件组成,而 Linux 文件系统的单个文件是由多个数据块组成。
4. 在 HDFS 中, Namenode 记录 Linux 数据文件所在的 Datanode,而在 Linux 文件系统中, inode 记录文件存放数据区的 block。

## PartII: HDFS 伪分布实际部署

### 一. 实验介绍

Hadoop 的部署有三种:单机部署、伪分布式部署和集群部署。伪分布式部署即在一个机器上配置了 hadoop 的所有节点。由 Hadoop 启动的 NameNode、DataNode、JobTracker、TaskTracker 守护进程都在同一台机器上运行,是相互独立的 Java 进程。在伪分布式模式下,Hadoop 使用的是分布式文件系统,各个作业也是由 JobTracker 服务,来管理的独立进程。相较于单机模式(独立模式),伪分布式部署增加了代码调试功能,允许检查内存使用情况,HDFS 输入输出。

### 二. 实验环境

1. Ubuntu18.04
2. jdk 1.8.0\_131
3. hadoop 2.7.7

### 三. 实验过程

1. 搭建 Hadoop 单机环境
  - (1) 更新源并安装 JDK

```
hadoop@ubuntu: ~  
hadoop@ubuntu:~$ sudo apt-get update  
[sudo] password for hadoop:  
Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease  
Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]  
Get:3 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]  
Get:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]  
Get:5 http://security.ubuntu.com/ubuntu xenial-security/main amd64 DEP-11 Metadata [74.8 kB]  
Get:6 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 DEP-11 Metadata [322 kB]  
Get:7 http://security.ubuntu.com/ubuntu xenial-security/main DEP-11 64x64 Icons [83.8 kB]  
Get:8 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 DEP-11 Metadata [124 kB]  
Get:9 http://security.ubuntu.com/ubuntu xenial-security/universe DEP-11 64x64 Icons [194 kB]  
Get:10 http://us.archive.ubuntu.com/ubuntu xenial-updates/main DEP-11 64x64 Icons [238 kB]  
Get:11 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 DEP-11 Metadata [2,464 B]  
Get:12 http://us.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 DEP-11 Metadata [274 kB]  
Get:13 http://us.archive.ubuntu.com/ubuntu xenial-updates/universe DEP-11 64x64
```

图片：更新源

```
hadoop@ubuntu: ~  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.  
hadoop@ubuntu:~$ sudo apt-get install default-jre default-jdk  
[sudo] password for hadoop:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  ca-certificates-java default-jdk-headless default-jre-headless  
  fonts-dejavu-extra java-common libgif7 libice-dev libpthread-stubs0-dev  
  libsm-dev libx11-dev libx11-doc libxau-dev libxcb1-dev libxdmcp-dev  
  libxt-dev openjdk-8-jdk openjdk-8-jdk-headless openjdk-8-jre  
  openjdk-8-jre-headless x11proto-core-dev x11proto-input-dev x11proto-kb-dev  
  xorg-sgml-doctools xtrans-dev  
Suggested packages:  
  default-java-plugin libice-doc libsm-doc libxcb-doc libxt-doc openjdk-8-demo  
  openjdk-8-source visualvm icedtea-8-plugin fonts-ipafont-gothic  
  fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei fonts-indic  
The following NEW packages will be installed:  
  ca-certificates-java default-jdk default-jdk-headless default-jre  
  default-jre-headless fonts-dejavu-extra java-common libgif7 libice-dev  
  libpthread-stubs0-dev libsm-dev libx11-dev libx11-doc libxau-dev libxcb1-dev
```

图片：安装 JDK

## (2) 下载并安装 Hadoop

```
hadoop@ubuntu:~$ wget https://mirrors.tuna.tsinghua.edu.cn/apache/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz
```

图片：从清华镜像下载版本为 2.7.7 的 hadoop

```
hadoop@ubuntu:~$ sudo tar -zxvf hadoop-2.7.7.tar.gz
```

图片：解压 hadoop-2.7.6.tar.gz

```
hadoop@ubuntu:~$ sudo mv hadoop-2.7.7 /usr/local/hadoop
hadoop@ubuntu:~$ ll /usr/local/hadoop/
total 144
drwxr-xr-x  9 ysy  staff   4096 Jul 18  2018 ./
drwxr-xr-x 11 root  root    4096 Feb 26 01:46 ../
drwxr-xr-x  2 ysy  staff   4096 Jul 18  2018 bin/
drwxr-xr-x  3 ysy  staff   4096 Jul 18  2018 etc/
drwxr-xr-x  2 ysy  staff   4096 Jul 18  2018 include/
drwxr-xr-x  3 ysy  staff   4096 Jul 18  2018 lib/
drwxr-xr-x  2 ysy  staff   4096 Jul 18  2018 libexec/
-rw-r--r--  1 ysy  staff  86424 Jul 18  2018 LICENSE.txt
-rw-r--r--  1 ysy  staff  14978 Jul 18  2018 NOTICE.txt
-rw-r--r--  1 ysy  staff   1366 Jul 18  2018 README.txt
drwxr-xr-x  2 ysy  staff   4096 Jul 18  2018 sbin/
drwxr-xr-x  4 ysy  staff   4096 Jul 18  2018 share/
hadoop@ubuntu:~$
```

图片：移动 hadoop 到目录 usr/local/hadoop

(3) Hadoop 环境设置

```
hadoop@ubuntu:~$ sudo gedit ~/.bashrc
```

图片：打开环境配置文件

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64 # JDK安装目录
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
export JAVA_LIBRARY_PATH=$HADOOP_HOME/lib/native:$JAVA_LIBRARY_PATH
```

图片：环境配置

```
hadoop@ubuntu:~$ source ~/.bashrc
```

图片：保存环境配置文件

```

hadoop@ubuntu:~$ source ~/.bashrc
hadoop@ubuntu:~$ hadoop
Usage: hadoop [--config confdir] [COMMAND | CLASSNAME]
  CLASSNAME                run the class named CLASSNAME
or
where COMMAND is one of:
  fs                        run a generic filesystem user client
  version                  print the version
  jar <jar>                run a jar file
                           note: please use "yarn jar" to launch
                           YARN applications, not this command.
  checknative [-a|-h]     check native hadoop and compression libraries availability
  distcp <srcurl> <desturl> copy file or directories recursively
  archive -archiveName NAME -p <parent path> <src>* <dest> create a hadoop archive
  classpath                prints the class path needed to get the
                           interact with credential providers
                           Hadoop jar and the required libraries
  daemonlog                get/set the log level for each daemon
  trace                   view and modify Hadoop tracing settings

Most commands print help when invoked w/o parameters.
hadoop@ubuntu:~$

```

图片：hadoop 环境配置成功

## 2. HDFS 伪分布实际部署

### (1) 打开 hadoop-env.sh 并配置 java 环境变量

```

hadoop@ubuntu:~$ sudo gedit /usr/local/hadoop/etc/hadoop/hadoop-env.sh

```

图片：打开 hadoop-env.sh

```

# The java implementation to use.
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

```

图片：配置 java 环境变量

### (2) 编辑 core-site.xml 文件。core-site.xml 文件中包含端口号的信息，分配给文件系统存储，用于存储所述数据存储器的限制大小。打开 core-site.xml 并在，标记之间添加属性。

```

hadoop@ubuntu:~$ sudo gedit /usr/local/hadoop/etc/hadoop/core-site.xml

```

图片：打开 core-site.xml

```

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>

```

图片：添加属性

### (3) 编辑 yarn-site.xml 文件。yarn-site.xml 文件用于配置成 yarn 在 Hadoop 中。打开 yarn-site.xml 文件，并在文件中的标签之间添加属性。

```
hadoop@ubuntu:~$ sudo gedit /usr/local/hadoop/etc/hadoop/yarn-site.xml
```

图片：打开 yarn-site.xml

```
<configuration>
<!-- Site specific YARN configuration properties -->
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>
```

图片：添加属性

- (4) 编辑 mapred-site.xml 文件。此文件用于指定正在使用 MapReduce 框架。缺省情况下，包含 Hadoop 的模板 yarn-site.xml。首先，它需要从 mapred-site.xml 复制。获得 mapred-site.xml 模板文件。随后再打开文件添加相关属性。

```
hadoop@ubuntu:~$ sudo cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml
hadoop@ubuntu:~$ sudo gedit /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml
```

图片：获得模板文件并打开

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

图片：添加属性

- (5) 编辑 hdfs-site.xml 文件。hdfs-site.xml 文件中包含 NameNode 路径信息，是存储 Hadoop 基础工具的地方。打开这个文件，并在这个文件中的标签之间添加属性。

```
hadoop@ubuntu:~$ sudo gedit /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

图片：打开 hdfs-site.xml

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/usr/local/hadoop/hadoop_data/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/usr/local/hadoop/hadoop_data/hdfs/datanode</value>
  </property>
</configuration>
```

图片：添加属性

(6) 创建并格式化 HDFS 目录

```
hadoop@ubuntu:~$ sudo mkdir -p /usr/local/hadoop/hadoop_data/hdfs/namenode
hadoop@ubuntu:~$ sudo mkdir -p /usr/local/hadoop/hadoop_data/hdfs/datanode
hadoop@ubuntu:~$ sudo chown hduser:hduser -R /usr/local/hadoop
chown: invalid user: 'hduser:hduser'
hadoop@ubuntu:~$ sudo chown hadoop:hadoop -R /usr/local/hadoop
```

图片：创建并格式化 HDFS 目录

(7) 启动 Hadoop 文件系统

```
hadoop@ubuntu:~$ start-dfs.sh
Starting namenodes on [localhost]
hadoop@localhost's password:
localhost: namenode running as process 2775. Stop it first.
hadoop@localhost's password:
localhost: datanode running as process 2931. Stop it first.
Starting secondary namenodes [0.0.0.0]
hadoop@0.0.0.0's password:
0.0.0.0: secondarynamenode running as process 3833. Stop it first.
```

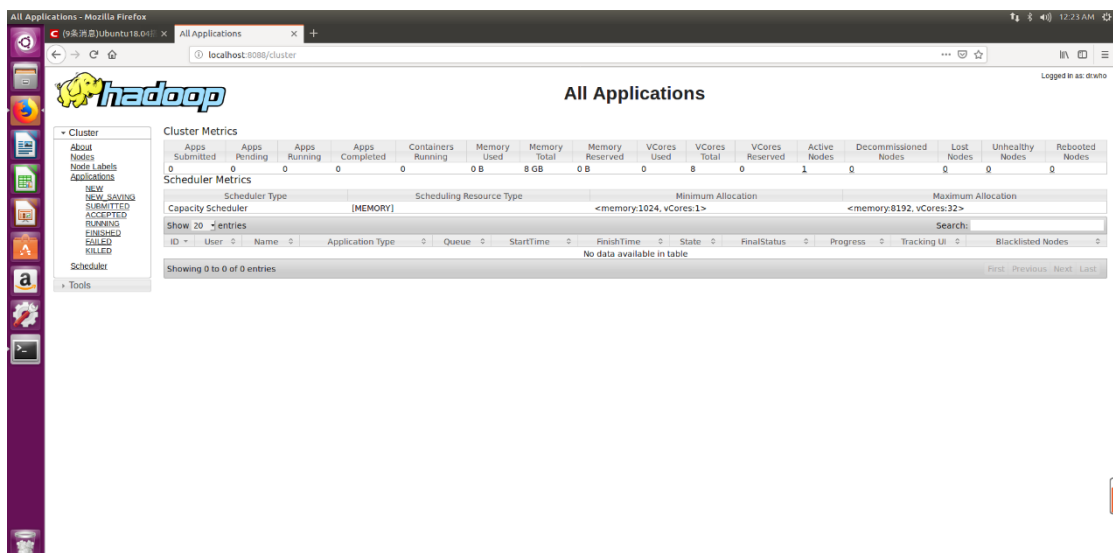
图片：启动 Hadoop 文件系统

(8) 启动 yarn 守护进程

```
hadoop@ubuntu:~$ start-yarn.sh
starting yarn daemons
resourcemanager running as process 4090. Stop it first.
hadoop@localhost's password:
localhost: nodemanager running as process 4402. Stop it first.
```

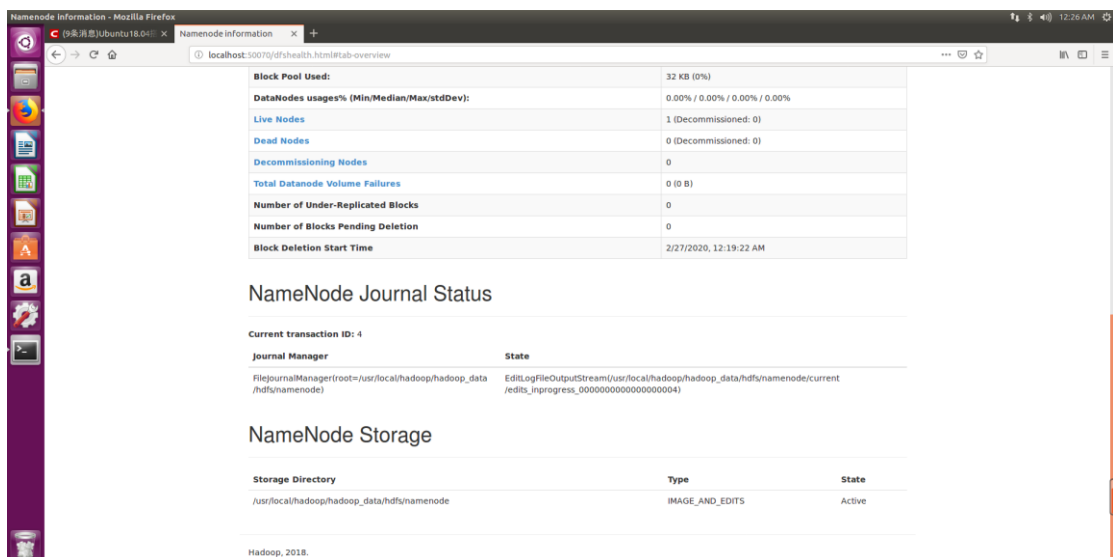
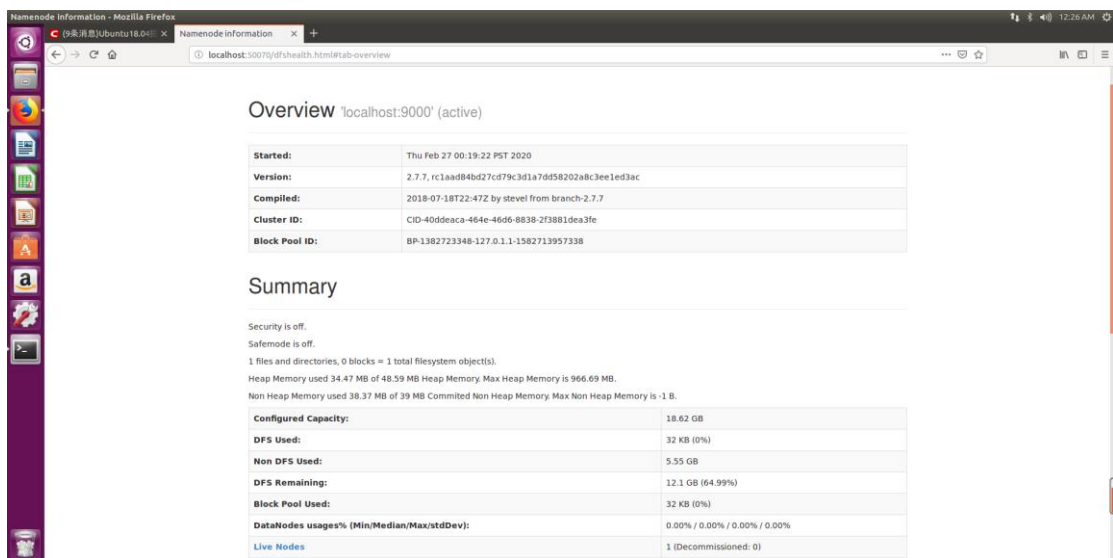
图片：启动 yarn 守护进程

## 四. 实验结果



图片：localhost:8088





图片: localhost:50070

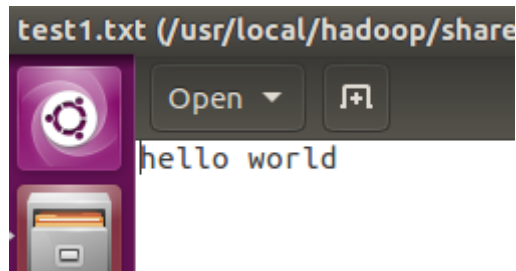
```
hadoop@ubuntu:/$ jps
2931 DataNode
4019 ResourceManager
4147 NodeManager
49493 Jps
2775 NameNode
3833 SecondaryNameNode
```

图片: 部署成功

## 五. wordcount 实例测试

(1) 自定义需要计数的文件 test1.txt





图片：test1.txt

- (2) 在 dfs 上创建一个目录/use/Hadoop

```
hadoop@ubuntu:/usr/local/hadoop/share/hadoop/mapreduce$ hdfs dfs -mkdir /use
hadoop@ubuntu:/usr/local/hadoop/share/hadoop/mapreduce$ hdfs dfs -mkdir /use/hadoop
```

图片：创建目录

- (3) 上传本地文件 test1.txt 到上述目录中

```
hadoop@ubuntu:/usr/local/hadoop/share/hadoop/mapreduce$ hdfs dfs -put test1.txt /use/hadoop
```

图片：上传本地文件

- (4) 查看上传的文件及文件内容

```
hadoop@ubuntu:/usr/local/hadoop/share/hadoop/mapreduce$ hdfs dfs -ls /use
Found 1 items
drwxr-xr-x - hadoop supergroup          0 2020-03-02 16:50 /use/hadoop
hadoop@ubuntu:/usr/local/hadoop/share/hadoop/mapreduce$ hdfs dfs -ls /use/hadoop
Found 1 items
-rw-r--r--  3 hadoop supergroup        12 2020-03-02 16:50 /use/hadoop/test1.txt
hadoop@ubuntu:/usr/local/hadoop/share/hadoop/mapreduce$ hdfs dfs -cat /use/hadoop/*
hello world
```

图片：查看文件内容

- (5) 执行 wordcount 实例

```
hadoop@ubuntu:/usr/local/hadoop/share/hadoop/mapreduce$ hadoop jar hadoop-mapreduce-examples-2.7.7.jar wordcount /use/hadoop/test1.txt /use/hadoop/wct
```

图片：执行 wordcount 实例

```
20/03/02 16:55:37 INFO mapreduce.Job: map 0% reduce 0%
20/03/02 16:55:49 INFO mapreduce.Job: map 100% reduce 0%
20/03/02 16:55:59 INFO mapreduce.Job: map 100% reduce 100%
```

图片：执行过程

- (6) 查看输出结果

```
hadoop@ubuntu:/usr/local/hadoop/share/hadoop/mapreduce$ hdfs dfs -cat /use/hadoop/wct/part-r-00000
hello 1
world 1
```

图片：输出结果

## 六. 实验总结

本次实验分为两个部分，第一部分分析对比了 HDFS 与传统的分布式/网络文件系统的差异，第二部分详细介绍了 HDFS 伪分布实际部署过程并通过 wordcount 实

例验证部署成功。

## 七. 参考文献

1. 王雪涛, 刘伟杰. 分布式文件系统[J]. 科技信息: 学术版(11):416-417.
2. 沈健, 王梦龙主编; 毕雨, 朱斌峰编著. Linux 系统管理: 华东理工大学出版社, 2014.08: 第 103 页
3. Hadoop 学习之 HDFS 文件读取. CSDN[引用日期 2015-07-05]
4. 徐志阳, 张文培, 侯青, et al. HDFS 在多用户并发访问系统中的应用研究[J]. 电脑知识与技术, v. 12(8):249-252.
5. 分布式文件系统(HDFS)与 linux 系统文件系统关系  
[https://blog.csdn.net/qq\\_32041579/article/details/77752733](https://blog.csdn.net/qq_32041579/article/details/77752733)