

实验一 Hadoop 集群搭建实验报告

江昱峰 21009200038

2023 年 11 月 13 日

1 实验目的

实践并掌握 Hadoop 集群搭建，具体包括以下四部分内容：

- Linux 环境配置；
- HDFS 伪分布式集群搭建；
- YARN 伪分布式集群搭建；
- 基于 Hadoop 的 PI 值计算。

2 环境配置目的

- Hadoop 是 Java 程序进程，在学习 Hadoop 体系时，必须要有 JDK 环境作为支撑。
- JDK（Java Development Kit）即 Java 开发工具，安装需要配置环境变量（JAVA_HOME，PATH），目的是用到 JDK 程序和编译命令文件时方便在任意目录下可以找到。
- SSH（Secure Shell）是一个建立在应用层上的安全远程管理，可靠地传输协议。为远程连接提供安全，防止远程管理信息泄露。
- 在实际环境当中，服务器部署在机房，为方便开发人员操作，可通过远程连接控制操作。
- 集群中主节点访问需要不断获取从节点服务用户密码，通过配置 SSH 远程免密登录，解决之间频繁获取用户密码问题。

3 实验知识

- JDK 介绍: JDK 是 Java Development Kit 的缩写, 中文称为 Java 开发工具包, 由 SUN 公司提供。它为 Java 程序开发提供了编译和运行环境, 所有的 Java 程序的编写都依赖于它。使用 JDK 可以将 Java 程序编写为字节码文件, 即.class 文件。
- SSH 免密介绍: SSH 为 Secure Shell (安全外壳协议) 的缩写。SSH 是一种网络协议, 用于计算机之间的加密登录。很多 ftp、pop 和 telnet 在本质上都是不安全的, 因为它们在网上用明文传送口令和数据, 别有用心的人非常容易就可以截获这些口令和数据。SSH 就是专为远程登录会话和其他网络服务提供安全性的协议。
- Hadoop 介绍: Hadoop 是一个由 Apache 基金会所开发的分布式系统基础架构。用户可以在不了解分布式底层细节的情况下, 开发分布式程序。充分利用集群的威力进行高速运算和存储。允许使用简单的编程模型在大量计算机集群上对大型数据集进行分布式处理。
- YARN 集群概述: Yarn 是随着 Hadoop 发展而催生的新框架, 全称是 Yet Another Resource Negotiator, 可以翻译为“另一个资源管理器”。Yarn 取代了以前 Hadoop 中 JobTracker 的角色, 因为以前 JT 的任务过重, 负责任务的调度、跟踪、失败重启等过程, 而且只能运行 MapReduce 作业, 不支持其他编程模式, 这也限制了 JT 使用范围, 而 Yarn 应运而生, 解决了这两个问题。用户可以将各种服务框架部署在 YARN 上, 由 YARN 进行统一地管理和资源分配。

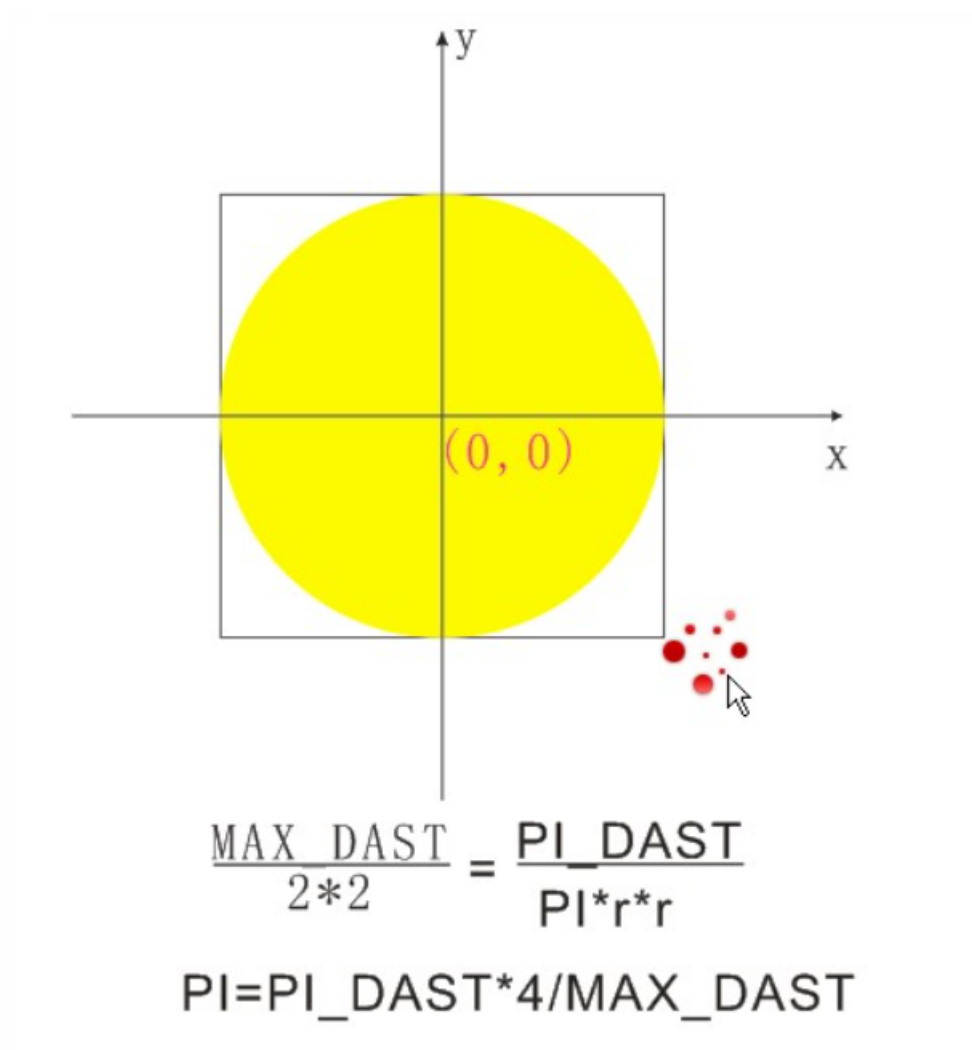
4 实验要求

完成 Hadoop 集群搭建, 具体包括以下四部分:

- Linux 环境配置;
- HDFS 伪分布式集群搭建;
- YARN 伪分布式集群搭建;
- 基于 Hadoop 的 PI 值计算。

5 实验原理

PI 值计算原理：



假如有一个边长为 2 的正方形。以正方形的一个中心点为圆心，以 1 为半径，画一个圆，于是在正方形内就有了一个内切圆。在正方形里随机生成若干个点，则有些点是在圆内，有些点是在圆外。正方形的面积是 4，圆的面积是 Pi 。设点的数量一共是 Max_DAST ，圆内的点数量是 PI_DAST ，在点足够多足够密集的情况下，会近似有 $\text{PI_DAST}/\text{Max_DAST}$ 的比值约

等于圆面积与正方形面积的比值，也就是 $PI_DAST/Max_DAST = \pi/2^2$ ，
即 $\pi = 4*PI_DAST/Max_DAST$ 。

6 实验环境

本次实验环境为青椒课堂平台的 Linux (Ubuntu 20.04) 操作系统。

7 实验步骤

7.1 Linux 环境配置

7.1.1 任务 1：解压缩 JDK 安装包

环境当中已经将安装包提供，可直接使用。JDK 安装包所在路径：`/root/software/jdk-8u221-linux-x64.tar.gz`。

解压缩步骤：

1. 进入 `/root/software/` 目录下。

```
→ ~ cd /root/software
```

2. 解压安装包到当前目录。

```
→ software tar xvzf jdk-8u221-linux-x64.tar.gz
```

3. 查看 `/root/software/` 目录下解压文件。

```
→ software ls /root/software/
hadoop-2.7.7.tar.gz      jdk1.8.0_221      zookeeper-3.4.14.tar.gz
htop-1.0.2.tar.gz       jdk-8u221-linux-x64.tar.gz
htop-2.2.0-3.el7.x86_64.rpm Python-3.6.8.tgz
→ software
```

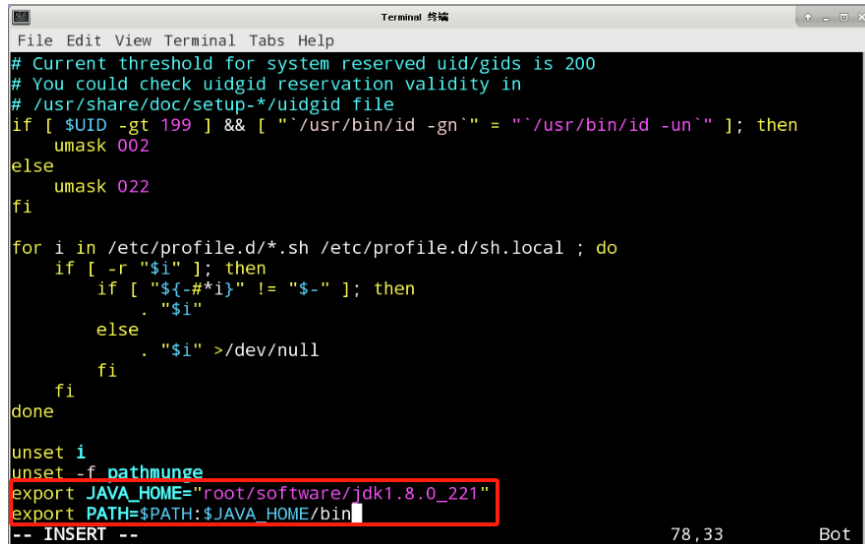
7.1.2 任务 2：配置 JDK 环境变量

配置环境变量步骤：

1. 编辑系统环境变量文件 `/etc/profile`。

```
→ ~ vim /etc/profile
```

2. 在末尾添加 JDK 的安装目录以及 PATH 路径添加 JDK 安装路径。



```
File Edit View Terminal Tabs Help
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi
for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done
unset i
unset -f pathmunge
export JAVA_HOME="/root/software/jdk1.8.0_221"
export PATH=$PATH:$JAVA_HOME/bin
-- INSERT --
78,33 Bot
```

3. 使用 source 命令使配置文件生效。

```
→ software source /etc/profile
```

4. 通过检测命令检测 JDK 环境安装是否成功。

```
→ software whereis java
java: /root/software/jdk1.8.0_221/bin/java
→ software
```

7.1.3 任务 3: SSH 服务获取公钥私钥

环境当中已经下载好 SSH 服务（openssh-server 和 openssh-clients），直接进行操作即可。有两种免密方式，选择两者中任意一种方式即可。此处我选择非对称算法 RSA。

SSH 免密流程：

1. 启动 SSH 服务。

```
→ software /usr/sbin/sshd
```

2. 使用 netstat 命令查看端口号。

```
→ software netstat -atunlp|grep sshd
tcp        0      0 0.0.0.0:22          0.0.0.0:*          LISTEN     624/sshd
tcp6       0      0 :::22              :::*                LISTEN     624/sshd
→ software
```

3. 使用 RSA 算法生成密钥对。输入信息时不需要输入，回车三次即可。

```
→ software ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:tdt0+5RuFHSr22cqemNYSqvju6x1xEqgN2XN5yRKQ2s root@qingjiao
The key's randomart image is:
+---[RSA 2048]---+
|                 |
|      . + .      |
|    . E = o. .   |
|   . * = * .    |
|  . o S + . .   |
| . o o.o.o....  |
|   o.oBo.=      |
|   o..o==..o    |
|  .o*= .o+=o.   |
+---[SHA256]-----+
```

4. 查看/root/.ssh 路径下生成的公钥私钥文件。

```
→ software ls /root/.ssh
id_rsa id_rsa.pub
→ software
```

7.1.4 任务 4: SSH 授权并验证登录

本节任务对 SSH 免密进行授权，使用方式与上一任务使用方式保持一致，否则授权失败。

1. 将获取的公钥文件 id_rsa.pub 拷贝到授权列表文件 authorized_keys

。

```
→ .ssh cp id_rsa.pub authorized_keys
```

2. 修改授权列表文件 `authorized_keys` 的权限，使拥有者可读可写，其他用户无权限。

```
→ .ssh chmod 600 authorized_keys
```

3. 通过 `hostname` 命令获取本机名。

```
→ .ssh hostname  
qingjiao
```

4. 验证免密登录是否配置成功（查看 `/root` 目录方式验证）。确保提交检测完成之后才可进行退出操作。

```
→ .ssh ssh qingjiao "sudo ls /root"  
The authenticity of host 'qingjiao (10.0.0.30)' can't be established.  
ECDSA key fingerprint is SHA256:gVhMfJLkbMRqLvuxCw0/azLMoKUBS9+OYyQp08pYy+w.  
ECDSA key fingerprint is MD5:49:fd:24:88:63:5d:8c:24:81:2d:ea:2f:19:23:e8:04.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'qingjiao,10.0.0.30' (ECDSA) to the list of known hosts.  
anaconda-ks.cfg  
Desktop  
Documents  
Downloads  
Music  
Pictures  
Public  
software  
Templates  
Videos  
→ .ssh
```

7.2 HDFS 伪分布式集群搭建

7.2.1 任务 1：检测 JDK 环境，解压安装包

1. 使用 `java -version` 命令检测 JDK 环境是否安装成功（安装 Hadoop 集群必须有 JAVA 环境）。

```
→ software java -version  
java version "1.8.0_221"  
Java(TM) SE Runtime Environment (build 1.8.0_221-b11)  
Java HotSpot(TM) 64-Bit Server VM (build 25.221-b11, mixed mode)
```

2. 解压 Hadoop 安装包，安装包已下载，存放在 `/root/software/` 目录下，进入到目录，解压安装包到当前文件。

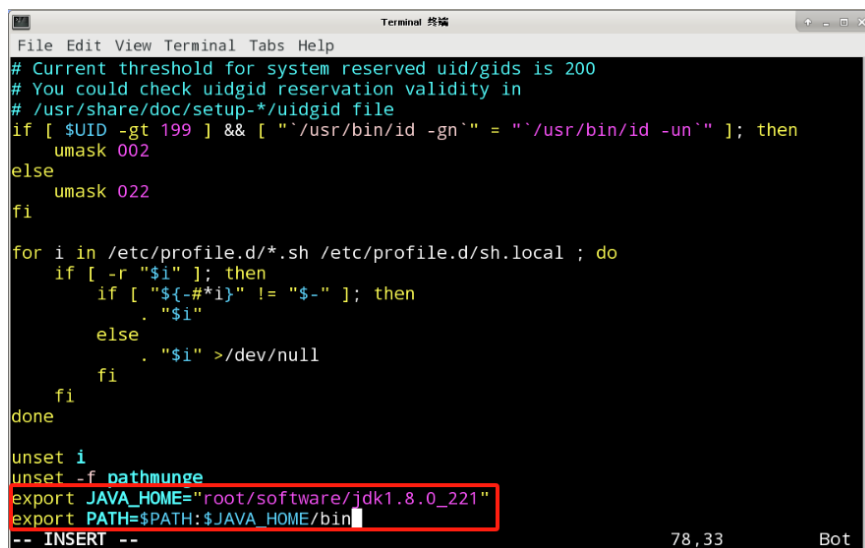
```
→ software tar xvfz hadoop-2.7.7.tar.gz
```

3. 查看当前目录结构，查看解压是否成功。

```
→ software ls
hadoop-2.7.7  hadoop-2.7.7.tar.gz  jdk1.8.0_221
→ software
```

7.2.2 任务 2：配置文件

1. 编辑 hadoop-env.sh，修改 JAVA_HOME 参数为本机 JDK 所在路径（/root/software/jdk1.8.0_221）。



```
File Edit View Terminal Tabs Help
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi
for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done
unset i
unset -f pathmunge
export JAVA_HOME=/root/software/jdk1.8.0_221
export PATH=$PATH:$JAVA_HOME/bin
-- INSERT --
```

2. 配置核心组件 core-site.xml，在配置 <configuration></configuration> 中添加如下内容：

```
→ hadoop vim core-site.xml
```



```
File Edit View Terminal Tabs Help
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

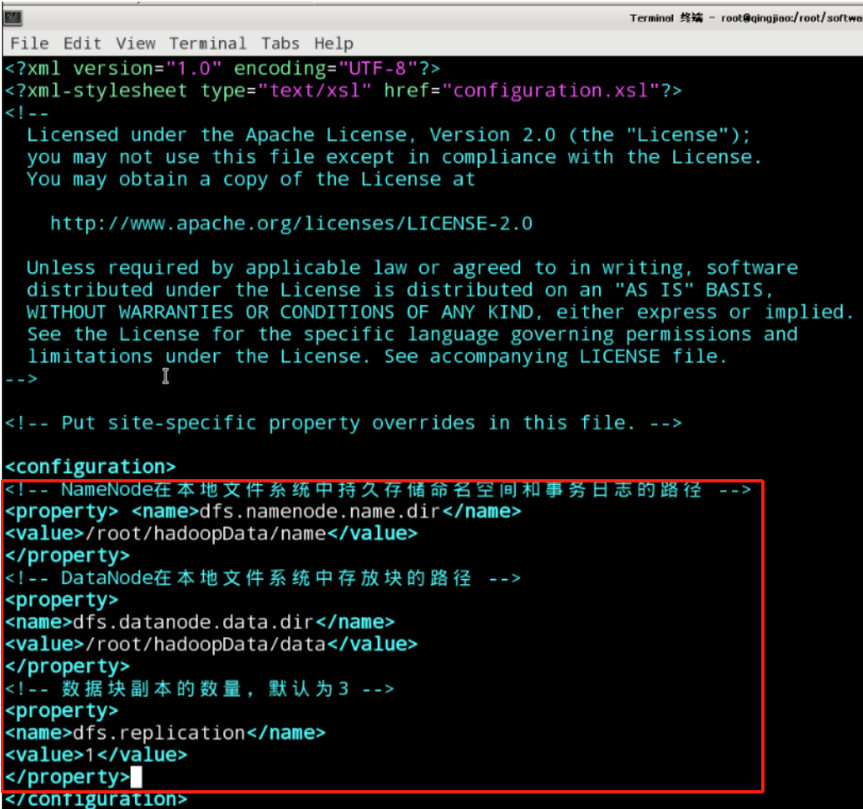
  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

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

<configuration>
<!-- HDFS集群中NameNode的URI (包括协议、主机名称、端口号), 默认为file:/// -->
<property>
<name>fs.defaultFS</name>
<!-- 用于指定NameNode的地址 -->
<value>hdfs://localhost:9000</value></property>
<!-- Hadoop运行时产生文件的临时存储目录 -->
<property>
<name>hadoop.tmp.dir</name><value>/root/hadoopData/temp</value>
</property>
</configuration>
```

3. 配置文件系统 hdfs-site.xml, 添加如下内容:

```
→ hadoop vim hdfs-site.xml
```



```
File Edit View Terminal Tabs Help
Terminal 终端 - root@qingjiao:/root/software

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

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

<configuration>
<!-- NameNode在本地文件系统中持久存储命名空间和事务日志的路径 -->
<property> <name>dfs.namenode.name.dir</name>
<value>/root/hadoopData/name</value>
</property>
<!-- DataNode在本地文件系统中存放块的路径 -->
<property>
<name>dfs.datanode.data.dir</name>
<value>/root/hadoopData/data</value>
</property>
<!-- 数据块副本的数量，默认为3 -->
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>
```

4. 配置 slaves 文件, 修改从节点主机名为本主机名。(见上图)

7.2.3 任务 3: 配置 Hadoop 环境变量

1. 打开/etc/profile 文件, 添加 HADOOP_HOME 路径和 PATH 路径。

```
File Edit View Terminal Tabs Help
pathmunge /usr/local/sbin after
pathmunge /usr/sbin after
fi

HOSTNAME=/usr/bin/hostname 2>/dev/null
HISTSIZE=1000
if [ "$HISTCONTROL" = "ignorespace" ] ; then
    export HISTCONTROL=ignoreboth
else
    export HISTCONTROL=ignoredups
fi

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset _f_pathmunge
export HADOOP_HOME=/root/software/hadoop-2.7.7
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

2. 使用 source 配置文件立即生效。

```
→ software source /etc/profile
→ software
```

3. 使用命令检测 Hadoop 环境变量是否设置成功。

```
→ software hadoop version
Hadoop 2.7.7
Subversion Unknown -r c1aad84bd27cd79c3d1a7dd58202a8c3ee1ed3ac
Compiled by stevel on 2018-07-18T22:47Z
Compiled with protoc 2.5.0
From source with checksum 792e15d20b12c74bd6f19a1fb886490
This command was run using /root/software/hadoop-2.7.7/share/hadoop/common/hadoop-common-2.7.7.jar
→ software
```

7.2.4 任务 4: HDFS 集群测试

1. 格式化文件系统。(格式化文件系统仅用于第一次启动 HDFS)

```

→ software cd hadoop-2.7.7/sbin
→ sbin hdfs namenode -format
23/11/11 11:06:59 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = qingjiao/172.22.2.4
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 2.7.7
STARTUP_MSG: classpath = /root/software/hadoop-2.7.7/etc/hadoop:/root/software/hadoop-2.7.7/share/hadoop/common/lib/log4j-1.2.17.jar:/root/software/hadoop-2.7.7/share/hadoop/common/lib/xmlenc-0.52.jar:/root/software/hadoop-2.7.7/share/hadoop/common/lib/junit-4.11.jar:/root/software/hadoop-2.7.7/share/hadoop/common/lib/httpclient-4.2.5.jar:/root/software/hadoop-2.7.7/share/hadoop/common/lib/jets3t-0.9.0.jar:/root/software/hadoop-2.7.

```

2. 使用脚本命令一键启动。

```

→ sbin start-dfs.sh
Starting namenodes on [localhost]
localhost: Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
localhost: starting namenode, logging to /root/software/hadoop-2.7.7/logs/hadoop-root-namenode-qingjiao.out
localhost: starting datanode, logging to /root/software/hadoop-2.7.7/logs/hadoop-root-datanode-qingjiao.out
Starting secondary namenodes [0.0.0.0]
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 /root/software/hadoop-2.7.7/logs/hadoop-root-secondarynamenode-qingjiao.out
→ sbin

```

3. 查看进程启动情况。

```

→ sbin jps
1353 DataNode
1514 SecondaryNameNode
1227 NameNode
1644 Jps
→ sbin

```

4. 使用脚本命令一键退出。

```

→ sbin exit

```

7.3 YARN 伪分布式集群搭建

7.3.1 任务 1：配置环境变量

配置环境变量 yarn-env.sh:

1. 使用命令查看是否存在 jdk 环境，通过 echo 命令获取本机 JDK 所在路径。

```

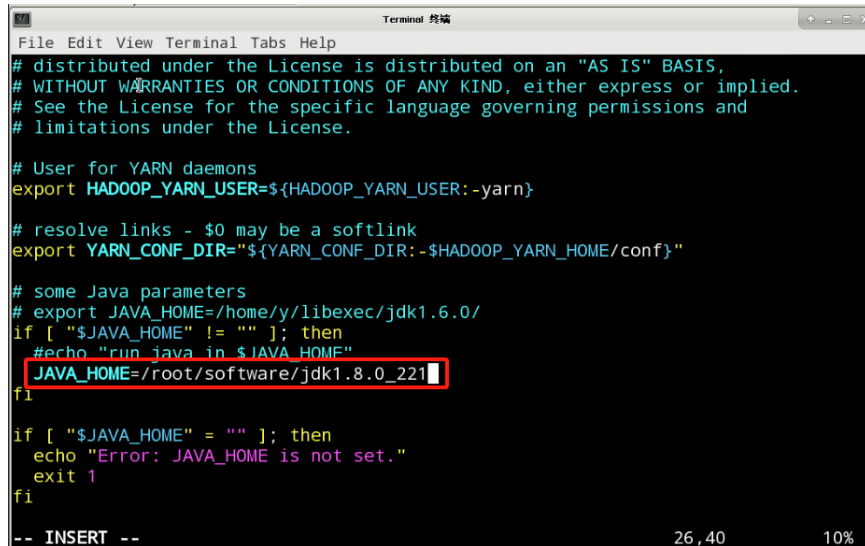
→ ~ echo $JAVA_HOME
/root/software/jdk1.8.0_221
→ ~

```

2. 使用命令打开 yarn-env.sh 文件。

```
→ ~ cd /root/software/hadoop-2.7.7/etc/hadoop
→ hadoop vim yarn-env.sh
→ hadoop
```

3. 修改 JAVA_HOME 参数为本机 JDK 所在路径。



```
File Edit View Terminal Tabs Help
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# User for YARN daemons
export HADOOP_YARN_USER=${HADOOP_YARN_USER:-yarn}

# resolve links - $0 may be a softlink
export YARN_CONF_DIR="${YARN_CONF_DIR:-$HADOOP_YARN_HOME/conf}"

# some Java parameters
# export JAVA_HOME=/home/y/libexec/jdk1.6.0/
if [ "$JAVA_HOME" != "" ]; then
    #echo "run java in $JAVA_HOME"
    JAVA_HOME=/root/software/jdk1.8.0_221
fi

if [ "$JAVA_HOME" = "" ]; then
    echo "Error: JAVA_HOME is not set."
    exit 1
fi

-- INSERT --
```

7.3.2 任务 2：配置计算框架

配置计算框架 mapred-site.xml

1. 进入 \$HADOOP_HOME/etc/hadoop/目录下将 mapred-site.xml.template 文件复制并改名为 mapred-site.xml。

```
→ hadoop cp mapred-site.xml.template mapred-site.xml
→ hadoop
```

2. 打开 mapred-site.xml 文件，进入编辑模式。

```
→ hadoop vim mapred-site.xml
```

3. 添加内容。

```
File Edit View Terminal Tabs Help
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

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

<configuration>
<!-- 指定使用 YARN 运行 MapReduce 程序，默认为 local -->
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value> </property>
</configuration>
```

7.3.3 任务 3：配置 YARN 系统

配置 YARN 系统 yarn-site.xml。

1. 打开 YARN 核心配置文件 yarn-site.xml。

```
→ hadoop vim yarn-site.xml
```

2. 在文件 <configuration></configuration> 中间添加配置内容：

```
File Edit View Terminal Tabs Help
<?xml version="1.0"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<configuration>

<!-- Site specific YARN configuration properties -->
<!-- NodeManager上运行的附属服务，也可以理解为 reduce 获取数据的方式 -->
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
</configuration>
```

7.3.4 任务 4：启动检测 YARN 集群

YARN 伪分布式集群检测：

1. 使用脚本命令一键启动 YARN 集群。

```
→ hadoop cd ../../sbin
→ sbin start-yarn.sh
starting yarn daemons
chown: missing operand after '/root/software/hadoop-2.7.7/logs'
Try 'chown --help' for more information.
starting resourcemanager, logging to /root/software/hadoop-2.7.7/logs/yarn--resourceman
anager-qingjiao.out
localhost: Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
localhost: starting nodemanager, logging to /root/software/hadoop-2.7.7/logs/yarn-root-nodemanager-qingjiao.out
→ sbin
```

2. 查看进程是否启动 ResourceManager 和 NodeManager。

```
→ sbin jps
609 ResourceManager
708 NodeManager
1006 Jps
→ sbin
```

3. 使用脚本命令一键关闭 YARN 集群。

```
→ sbin exit
```

7.4 Hadoop 初体验——PI 值计算

任务 1：PI 值计算实现步骤：

1. 启动 Hadoop 集群，系统环境已经启动。可通过 jps 命令进行查看。

```
→ ~ cd /root/software/hadoop-2.7.7/sbin
→ sbin jps
1171 NodeManager
1063 ResourceManager
903 SecondaryNameNode
552 NameNode
1471 Jps
703 DataNode
→ sbin
```

2. 进入到官方提供 jar 包目录 `$HADOOP_HOME/share/hadoop/mapreduce/`。

```
→ sbin cd $HADOOP_HOME/share/hadoop/mapreduce/  
→ mapreduce
```

3. 通过查看命令查看目录下 PI 值计算所用 jar 包。
4. 使用命令运行该 jar 包并指定 map 数为 10，总点数为 100。

```
^C- mapreduce hadoop jar hadoop-mapreduce-examples-2.7.7.jar pi 10 10  
Number of Maps = 10  
Samples per Map = 10  
Wrote input for Map #0  
Wrote input for Map #1  
Wrote input for Map #2  
Wrote input for Map #3  
Wrote input for Map #4  
Wrote input for Map #5  
Wrote input for Map #6  
Wrote input for Map #7  
Wrote input for Map #8  
Wrote input for Map #9  
Starting Job  
23/11/11 13:48:47 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032  
23/11/11 13:48:47 INFO input.FileInputFormat: Total input paths to process : 10  
23/11/11 13:48:47 INFO mapreduce.JobSubmitter: number of splits:10  
23/11/11 13:48:48 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1699681107545_0003  
23/11/11 13:48:48 INFO impl.YarnClientImpl: Submitted application application_1699681107545_0003  
23/11/11 13:48:48 INFO mapreduce.Job: The url to track the job: http://qingjiao:8088/proxy/application_1699681107545_0003/  
23/11/11 13:48:48 INFO mapreduce.Job: Running job: job_1699681107545_0003  
23/11/11 13:48:55 INFO mapreduce.Job: Job job_1699681107545_0003 running in uber mode : false  
23/11/11 13:48:55 INFO mapreduce.Job: map 0% reduce 0%  
23/11/11 13:49:05 INFO mapreduce.Job: map 10% reduce 0%  
23/11/11 13:49:06 INFO mapreduce.Job: map 60% reduce 0%  
23/11/11 13:49:13 INFO mapreduce.Job: map 70% reduce 0%  
23/11/11 13:49:14 INFO mapreduce.Job: map 100% reduce 0%  
23/11/11 13:49:15 INFO mapreduce.Job: map 100% reduce 100%  
23/11/11 13:49:16 INFO mapreduce.Job: Job job_1699681107545_0003 completed successfully  
23/11/11 13:49:16 INFO mapreduce.Job: Counters: 49  
File System Counters  
FILE: Number of bytes read=226  
FILE: Number of bytes written=1353528  
FILE: Number of read operations=0  
FILE: Number of large read operations=0  
FILE: Number of write operations=0  
HDFS: Number of bytes read=2630  
HDFS: Number of bytes written=215
```

5. 查看计算结果并进行记录。

```
Estimated value of Pi is 3.20000000000000000000
```

计算结果为 3.2000。

6. 使用命令再次运行该 jar 包并指定 map 数为 100，总点数为 10000。


```
→ mapreduce hadoop jar hadoop-mapreduce-examples-2.7.7.jar pi 100 100
Number of Maps = 100
Samples per Map = 100
Wrote input for Map #0
Wrote input for Map #1
Wrote input for Map #2
Wrote input for Map #3
Wrote input for Map #4
Wrote input for Map #5
Wrote input for Map #6
Wrote input for Map #7
Wrote input for Map #8
Wrote input for Map #9
Wrote input for Map #10
Wrote input for Map #11
Wrote input for Map #12
Wrote input for Map #13
Wrote input for Map #14
Wrote input for Map #15
Wrote input for Map #16
Wrote input for Map #17
Wrote input for Map #18
Wrote input for Map #19
Wrote input for Map #20
Wrote input for Map #21
Wrote input for Map #22
Wrote input for Map #23
Wrote input for Map #24
Wrote input for Map #25
Wrote input for Map #26
Wrote input for Map #27
Wrote input for Map #28
Wrote input for Map #29
Wrote input for Map #30
```

7. 再次查看计算结果并和上次计算结果对比。

```
Estimated value of Pi is 3.14080000000000000000
```

计算结果为 3.1408。

8 实验结果截图

9 结果分析

就总体趋势而言，当 mapreduce 的切片越细，即 map、总点数越大时，计算结果越精确。

10 困难解决

1.1.4（任务 4：SSH 授权并验证登录）部分一直无法通过评测点。

11 心得体会

做完本次实验，除了掌握了实验目的部分中所有内容的收获之外，我还有以下几点心得体会：

- 实践并掌握了 SSH 服务获取公钥私钥、SSH 授权并验证登录、配置 Hadoop 环境变量、配置计算框架、配置 YARN 系统等内容。
- 对比分析了 HDFS 伪分布式集群搭建、YARN 伪分布式集群搭建的异同点。
- 基于 Hadoop 进行了编程实操。