



Lab 2: 计算层软件架构实验

Hanchuan Xu
xhc@hit.edu.cn

April 21, 2021

实验目的

- 学习构建集群的方法
- 学会使用典型的负载均衡框架 **Nginx**
- 学会使用**Hadoop** 进行**map-reduce**编程
- 能够灵活应用计算层中间件到实际系统

实验要求

- 组队：2人结对成组
- 内容：2+1 (2个必选，1个可选)
 - 2.1 Nginx集群负载均衡实验（必选）
 - 2.2 Hadoop分布式计算实验（必选）
 - 2.3 ActiveMQ异步消息推送实验（可选）
 - 结合《软件过程与工具》课程中进销存系统(或其他实际软件系统)进行计算层架构重构，支持海量用户的在线高并发请求场景
 - 在实验报告中给出关键过程的细节
- 实验最终提交物、结果考核
 - 实验报告和程序提交到乐学网：<http://cms.hit.edu.cn>
 - 提交物命名：其中一名同学的学号-姓名.zip/rar
 - 第10周实验课时，老师和TA现场检查程序
 - **提交时间：第10周周末，5月16日23:55分之前**

2.1 Nginx集群负载均衡实验

■ 1. 建立虚拟机环境

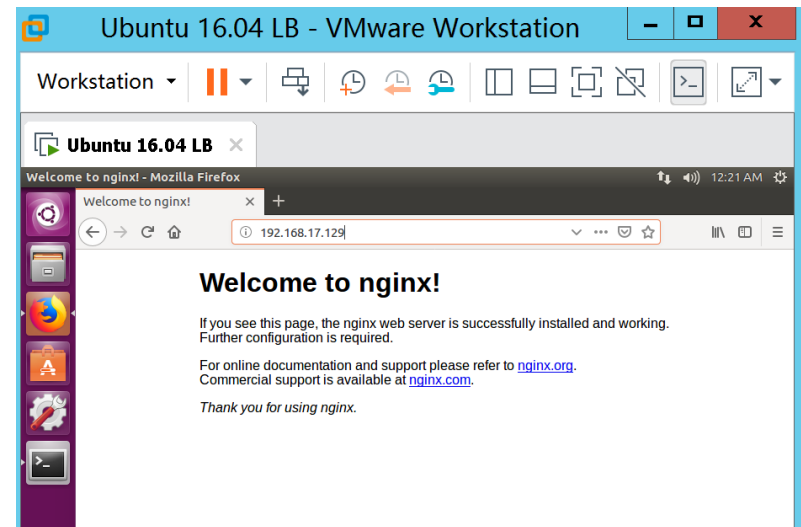
- 根据所用操作系统(windows或macOS), 下载VMware Workstation/VirtualBox/Hyper-v等某个虚拟机软件, 并安装
- 下载ubuntu安装文件<https://cn.ubuntu.com/download>, 桌面版和服务版均可, 安装进虚拟机
- 设置网络参数, 保证可以访问internet

2.1 Nginx集群负载均衡实验

■ 2. 搭建基本的Nginx框架

- 在ubuntu系统里安装Nginx
- 练习Nginx的基本操作（启动、重启、停止等）
- 配置单机Nginx环境，主配置文件为/etc/nginx/nginx.conf，主配置文件可以通过include语句引入额外配置文件

```
user www-data;  
worker_processes auto;  
pid /run/nginx.pid;  
events {  
worker_connections 768;  
}  
...  
http {  
include /etc/nginx/conf.d/*.conf;  
include /etc/nginx/sites-enabled/*;  
}
```



2.1 Nginx集群负载均衡实验

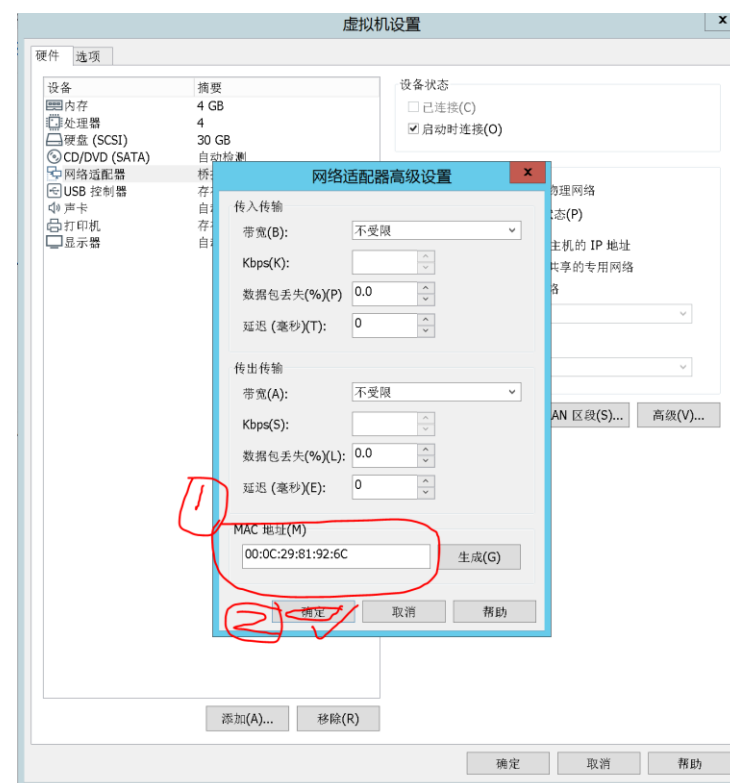
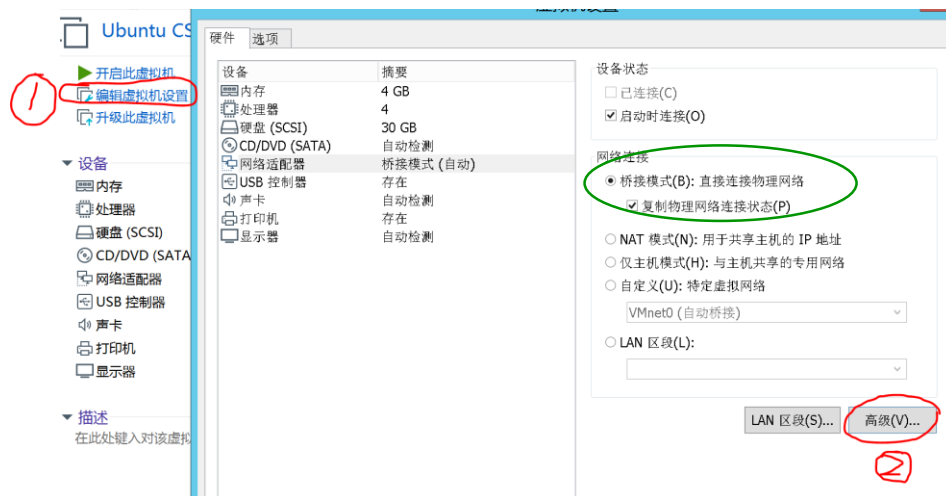
■ 3. 自行学习配置指令

- upstream块的构成与含义
- server块的构成与含义
- location块的构成与含义

2.1 Nginx集群负载均衡实验

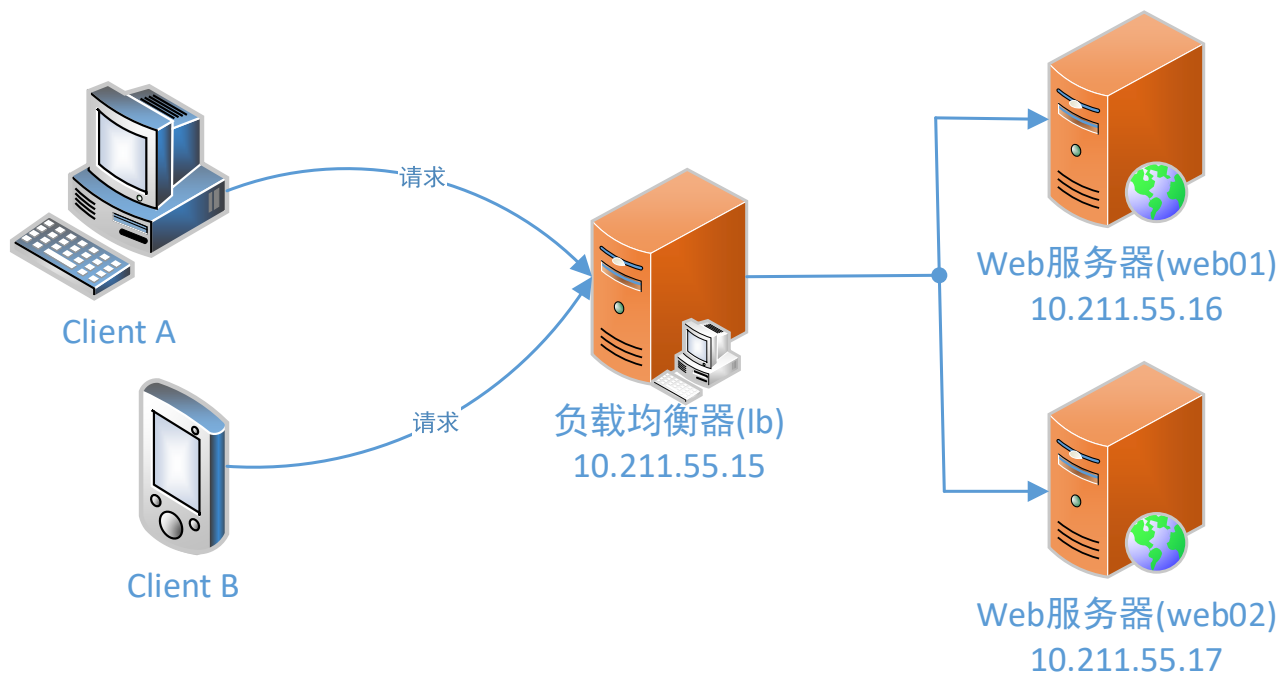
■ 4.建立虚拟多机环境(至少三个)

- 利用虚拟机软件的克隆/迁移功能复制多份虚拟机
- 保证在虚拟机关机状态下修改其网卡mac地址
- 虚拟机开机后修改ip地址, 让虚拟机之间可以互相访问(至少ping通, 或者进一步可以互相ssh连接, 参考2.2的步骤3)



2.1 Nginx集群负载均衡实验

- 5. 配置Nginx的负载均衡文件，并进行合理设置，实现负载均衡能力
 - 增加upstream块，列出可用的业务服务器
 - 增加server块给出负载服务器参数



2.1 Nginx集群负载均衡实验

■ 5. 配置Nginx的负载均衡文件，并进行合理设置，实现负载均衡能力

— e.g. web01 服务器的配置(web02类似)

```
server {  
    listen    8080;  
    server_name 192.168.3.37;  
    location / {  
        root /etc/nginx/node; //自行设置目录  
        index index.html index.htm;  
    }  
}
```

— e.g. web01 服务器的简单web服务(web02类似)

```
mkdir /etc/nginx/node  
cd /etc/nginx/node  
创建index.html文件('I'm web01 node! ')
```

— 重启nginx

2.1 Nginx集群负载均衡实验

- 5. 配置Nginx的负载均衡文件，并进行合理设置，实现负载均衡能力
 - e.g. lb负载服务器的配置

```
http {  
    ...  
    upstream nodes {  
        server 192.168.3.37:8080;  
        server 192.168.3.38:8080;  
    }  
    server {  
        listen      8080;  
        server_name 192.168.3.36;  
        location / {  
            proxy_pass http://node; //请求转向nodes定义的服务器列表  
            include proxy_params; #需手动创建此文件  
        }  
    }  
}
```

- 重启nginx

2.1 Nginx集群负载均衡实验

- 5. 配置Nginx的负载均衡文件，并进行合理设置，实现负载均衡能力
 - e.g. lb负载服务器配置的proxy_params文件(可选)
 - 文件中存放代理的请求头相关参数

```
proxy_set_header Host $http_host;  
proxy_set_header X-Real-IP $remote_addr;  
proxy_set_header X-Forwarded-For  
$proxy_add_x_forwarded_for;
```

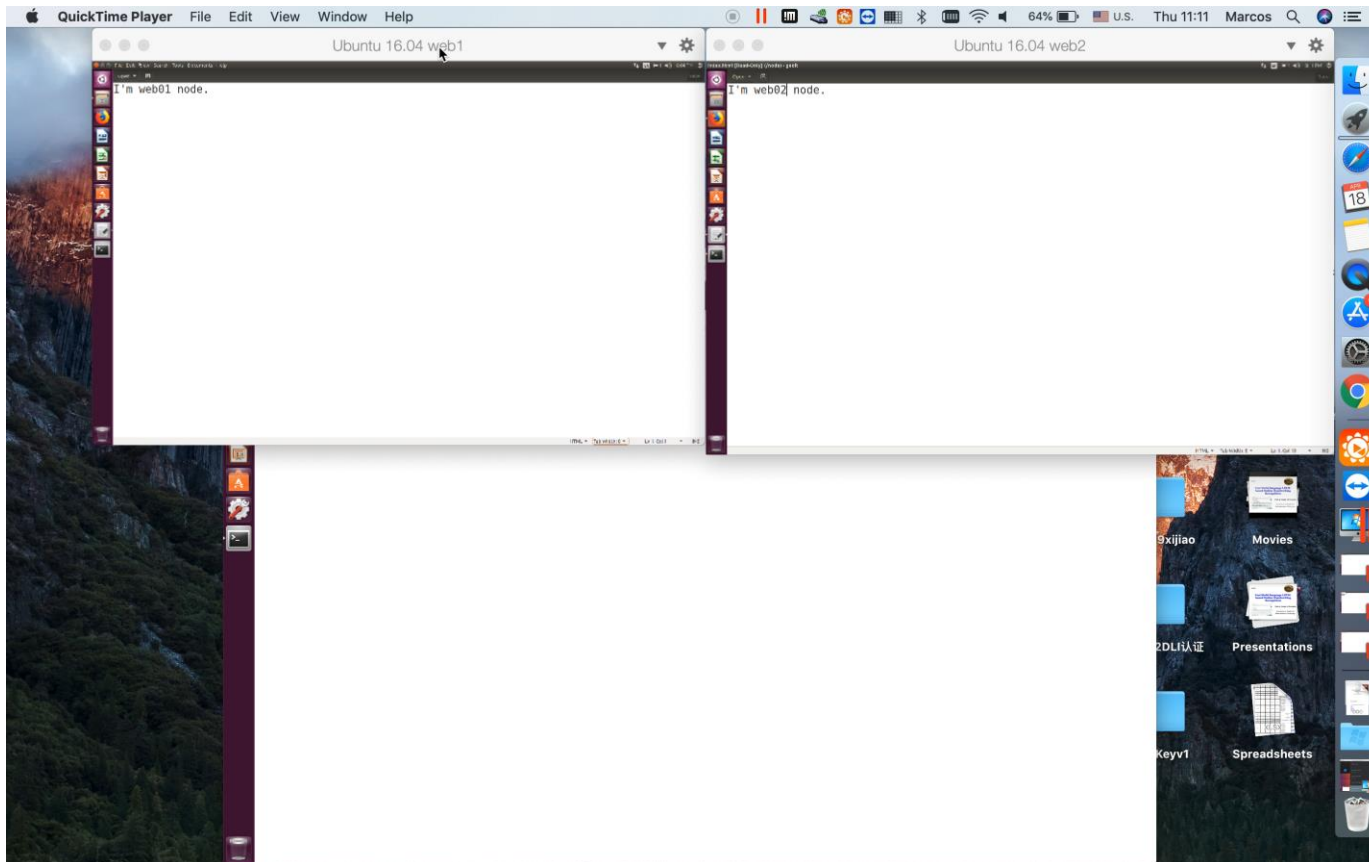
```
proxy_connect_timeout 30;  
proxy_send_timeout 60;  
proxy_read_timeout 60;
```

```
proxy_buffering on;  
proxy_buffer_size 32k;  
proxy_buffers 4 128k;
```

- 重启nginx

2.1 Nginx集群负载均衡实验

- 6. 验证集群的基本负载均衡系统
 - 访问lb节点的Nginx，反复刷新
 - 你看到了什么？



2.1 Nginx集群负载均衡实验

■ 7. 集群负载均衡算法验证

- 通过upstream块可以配置不同的负载均衡算法
- 测试并对比至少三种不同负载均衡算法（策略）的效果
- 某个机器宕机，服务是否不可用？为什么？
- 请根据本步骤的实验对比，给出你对负载均衡算法的理解和认识

2.1 Nginx集群负载均衡实验

- 8. 应用负载均衡技术改造遗留的“进销存”系统
 - 支持海量用户的在线高并发请求场景
 - 请给出设计细节并分析负载均衡前后的区别

2.1 Nginx集群负载均衡实验

■ 9. 迁移到真实集群环境(选做)

- 以上实验均是在虚拟集群下进行
- 请迁移到真实集群下重做负载均衡实验
- 请给出实验过程中需要注意的问题
- 比较并分析虚拟集群和真实集群下的异同

2.2 Hadoop分布式计算实验

■ 1. 建立虚拟机环境(同2.1)

■ 2. 安装JDK

- 下载jdk二进制文件，建议jdk-11.0.10-linux-x64_bin.tar.gz
- 解压下载后的文件到某个目录(如: /sam/jdk-11.0.10)
- 修改/etc/profile文件，配置JDK信息。然后执行source /etc/profile命令生效。

```
#在文件最后添加
export JAVA_HOME=/sam/jdk-11.0.10
export CLASSPATH=.:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
export PATH=$PATH:$JAVA_HOME/bin
```

- 检测是否成功安装

```
java -version
```

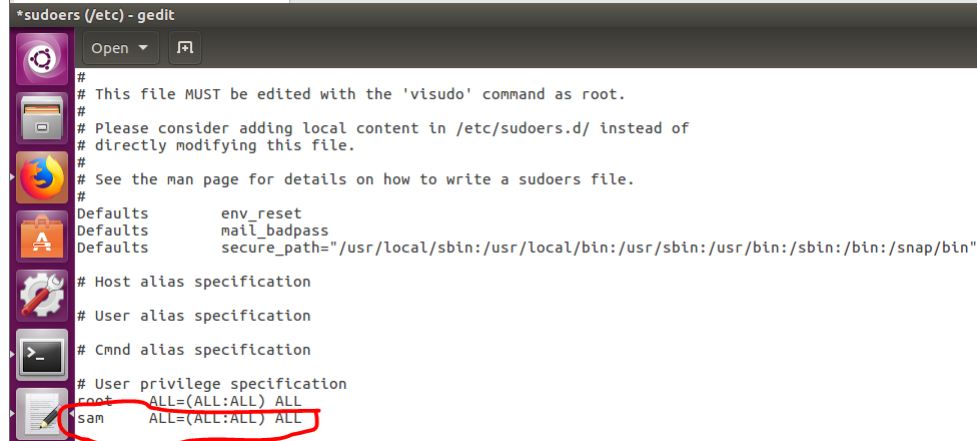
```
sam@ubuntu:/sam$ source /etc/profile
sam@ubuntu:/sam$ java -version
java version "1.8.0_211"
Java(TM) SE Runtime Environment (build 1.8.0_211-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)
sam@ubuntu:/sam$
```


2.2 Hadoop分布式计算实验

■ 3. 前期配置

- 权限设置，为sam用户添加权限

```
sudo gedit /etc/sudoers
```



2.2 Hadoop分布式计算实验

■ 3. 前期配置

— 设置主机名

- 修改 /etc/hostname 文件：master节点的主机设置为 master ，其他两个虚拟机分别设置为slave1、slave2
- 修改 /etc/hosts文件如下：

```
192.168.190.128 master  
192.168.190.129 slave1  
192.168.190.131 slave2
```

2.2 Hadoop分布式计算实验

■ 3. 前期配置

— 配置SSH免密码登录

- 确认安装了ssh，否则通过`sudo apt-get install ssh`命令安装
- 确保当前用户目录下存在`.ssh`目录，否则自己创建
- 为当前用户`sam`赋予拥有权限：`sudo chown -R sam .ssh`
- 生成RSA密匙对：`ssh-keygen -t rsa`
- 集群内共享密匙：`cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`

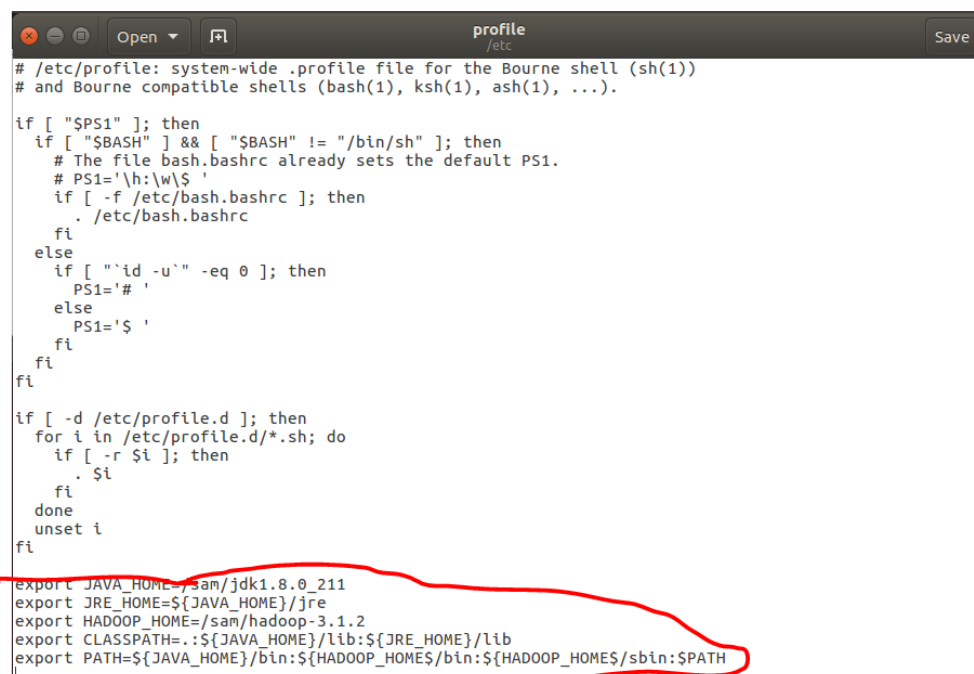
```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACner14R2BuBf/I5FUD8RXqQz
UMGuqBV5oXpRVKdhzLGxkJxFCnBKnbzZo0WtfKC1I1CLf5wSuUi4lm5Zg
XYbCRB3KRWI8rjbWa8fkCyqH5bKns4i/jUt9ftQmK91IEAdRLEYvG8mDFJU
pHdMQ81XgfAMbWrlLsmNpX49jDyNAqVhcoRY6Vmtvkfglt/rKRfaljSub1jKd
7R9bjXmiutf8PidQRDipJs/CBtwGBhVw45rn+gcVHalFCEOxCYQ3tfsOrMgfz
L1ca2xFgSmCCTjQtCj8KRpgJdRUATWrCQ+xN7RgHma7nXXmGNP8MkUz
xzvPzZdww1V6JAFwfdwMQmURR sam@master
```

- 配置 `slave1` 和 `slave2` 节点可以通过 `ssh` 无密码互相访问：将`authorized_keys`复制到`slave1`和`slave2`中的`.ssh/`目录中
 - `scp authorized_keys sam@slave1:~/.ssh/`
- 确认是否可以免密登录，例如`ssh slave1`(首次执行时需要确认`yes`，之后则不需要)

2.2 Hadoop分布式计算实验

■ 4. 安装并运行Hadoop

- 下载hadoop二进制文件
 - hadoop-3.1.4.tar.gz
- 解压hadoop到某目录(如/sam)
- 配置 hadoop 的环境变量
 - `sudo gedit /etc/profile`
 - `source /etc/profile`



```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "$PS1" ]; then
  if [ "$BASH" ] && [ "$BASH" != "/bin/sh" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\S '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi

export JAVA_HOME=/sam/jdk1.8.0_211
export JRE_HOME=${JAVA_HOME}/jre
export HADOOP_HOME=/sam/hadoop-3.1.2
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:${HADOOP_HOME}/bin:${HADOOP_HOME}/sbin:${PATH}
```

根据自己计算机中实际情况修改→

2.2 Hadoop分布式计算实验

■ 4. 安装并运行Hadoop

— 配置三台主机的Hadoop文件

- 修改**Hadoop安装目录/etc/hadoop/目录**下的Hadoop-env.sh文件，添加

```
export JAVA_HOME=/sam/jdk-11.0.10
export HADOOP_HOME=/sam/hadoop-3.1.4
PATH=${JAVA_HOME}/bin:${HADOOP_HOME}/bin:${HADOOP_HOME}/sbin:$PATH
```

- **core-site.xml**中主要内容(Hadoop的配置文件)

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://master:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/tmp</value>
  </property>
</configuration>
```

2.2 Hadoop分布式计算实验

■ 4. 安装并运行Hadoop

— 配置三台主机的Hadoop文件

- **hdfs-site.xml**中主要内容(HDFS的配置文件)

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
</configuration>
```

- **mapred-site.xml**中主要内容(MapReduce的配置文件)

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>master:9001</value>
  </property>
</configuration>
```

2.2 Hadoop分布式计算实验

■ 4. 安装并运行Hadoop

— 配置三台主机的Hadoop文件

- **masters**文件，如果没有手动添加一个，配置如下

```
master
```

- **workers**文件

```
slave1  
slave2
```

2.2 Hadoop分布式计算实验

■ 4. 安装并运行Hadoop

- 向 slave1 和 slave2 节点复制 hadoop-3.1.4 整个目录至相同的位置
- 进入sam@master节点hadoop目录下使用

```
scp -r hadoop-3.1.4 sam@slave1:~/
scp -r hadoop-3.1.4 sam@slave2:~/
```

- 注：如果采用克隆虚拟机的方式创建多个节点，则上述步骤可以省略
- 在master节点上执行hdfs namenode -format
- 在master节点上执行start-all.sh验证正确性

- 注：步骤2、3、4中除个别环节外，对各节点(master,slave1,slave2)的配置是相同的，建议在一个节点上完成2-4环节的配置后，采用克隆的方式复制出其他节点虚拟机，以节省时间。

2.2 Hadoop分布式计算实验

■ 5. hdfs基本操作和wordcount程序

- 进入hadoop安装目录中的样例程序目录：
 - /sam/hadoop-3.1.4/share/hadoop/mapreduce
- 列出当前路径下的文件
 - 其中带有example字样的为样例程序

```
sam@ubuntu:/sam/hadoop-3.1.2/share/hadoop/mapreduce$ ls
hadoop-mapreduce-client-app-3.1.2.jar
hadoop-mapreduce-client-common-3.1.2.jar
hadoop-mapreduce-client-core-3.1.2.jar
hadoop-mapreduce-client-hs-3.1.2.jar
hadoop-mapreduce-client-hs-plugins-3.1.2.jar
hadoop-mapreduce-client-jobclient-3.1.2.jar
hadoop-mapreduce-client-jobclient-3.1.2-tests.jar
hadoop-mapreduce-client-nativetask-3.1.2.jar
hadoop-mapreduce-client-shuffle-3.1.2.jar
hadoop-mapreduce-client-uploader-3.1.2.jar
hadoop-mapreduce-examples-3.1.2.jar
jdiff
lib
lib-examples
sources
```

2.2 Hadoop分布式计算实验

■ 5. hdfs基本操作和wordcount程序

— 新建文本文件words作为输入

- 内容输入如下，然后使用命令上传到hdfs文件系统中

- `hadoop fs -put words hdfs://localhost:9000/input`

```
hello tom  
hello kitty  
hello world  
hello tom
```

— 在命令行中敲入：

- `hadoop jar hadoop-mapreduce-examples-3.1.4.jar wordcount hdfs://localhost:9000/input hdfs://localhost:9000/output`

(根据自己的hadoop版本和路径相应的调整)

2.2 Hadoop分布式计算实验

■ 5. hdfs基本操作和wordcount程序

- 打开页面: `http://master:9870/dfshealth.jsp`
- 点击图中的Browse the filesystem, 跳转到文件系统界面
- 继续点击图中output/part-r-00000, 看到程序最终运行的结果

NameNode 'localhost:9000' (active)

Started:	Wed Dec 24 04:10:38 PST 2014
Version:	2.2.0, 1529768
Compiled:	2013-10-07T06:28Z by hortonmu from branch-2.2.0
Cluster ID:	CID-62e83bd8-b186-489a-b177-7af7e7c5f969
Block Pool ID:	BP-961320506-127.0.1.1-1419422409011

[Browse the filesystem](#)
[NameNode Logs](#)

Cluster Summary

Security is **OFF**
 3 files and directories, 2 blocks = 5 total.
 Heap Memory used 46.15 MB is 60% of Committed Heap Memory 75.89 MB.
 Max Heap Memory is 966.69 MB.
 Non Heap Memory used 18.72 MB is 99% of Committed Non Heap Memory
 18.81 MB. Max Non Heap Memory is 96 MB.

Configured Capacity	18.94 GB
---------------------	----------

Contents of directory /

Goto: go

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
out	dir				2014-12-24 05:05	rw-r--r--	gyouxu	super
tmp	dir				2014-12-24 05:04	rw-r--r--	gyouxu	super
usr	file	104.17 MB	1	128 MB	2014-12-24 04:40	rw-r--r--	gyouxu	super
words	file	44 B	1	128 MB	2014-12-24 04:39	rw-r--r--	gyouxu	super

[Go back to DFS home](#)

Local logs

File: /out/part-r-00000

Goto: go

[Go back to dir listing](#)
[Advanced view/download options](#)

```
hello 4
kitty 1
tom 2
world 1
```

- 或者用命令: `hadoop fs -cat /output/part-r-00000`直接查看结果

2.2 Hadoop分布式计算实验

- 6. 分布式计算应用于遗留的“进销存”系统
 - 支持海量用户的log日志分析
 - 请模拟输入数据，从1万个用户共100万次操作记录中计算每个用户的访问次数

2.3 ActiveMQ异步消息推送实验

- <https://activemq.apache.org> 自行下载学习



The end

April 21, 2021