

Chipyard 实验环境配置方法

Chipyard 实验环境配置方法

- 使用配置好的 VM 环境 (Ubuntu18.04)
- 自己在虚拟机或者其他 Linux 环境下安装 (推荐)
 - 安装并配置虚拟机
 - 安装环境依赖
 - 更换 apt 源:
 - 安装 clash 并配置代理文件
 - 安装依赖并下载 verilator
 - 下载 Chipyard 及其子模块
 - 搭建工具链
 - 修改环境变量

使用配置好的 VM 环境 (Ubuntu18.04)

1. 安装 VMware Workstation
2. 下载 ova 文件

ip: 10.212.43.214
用户名: ftpguest
密码: ftpguest

由于 ftp 不支持下载16G这么大的文件（也可能是因为二十年前的服务器比较老旧），可以用 scp 协议进行下载：

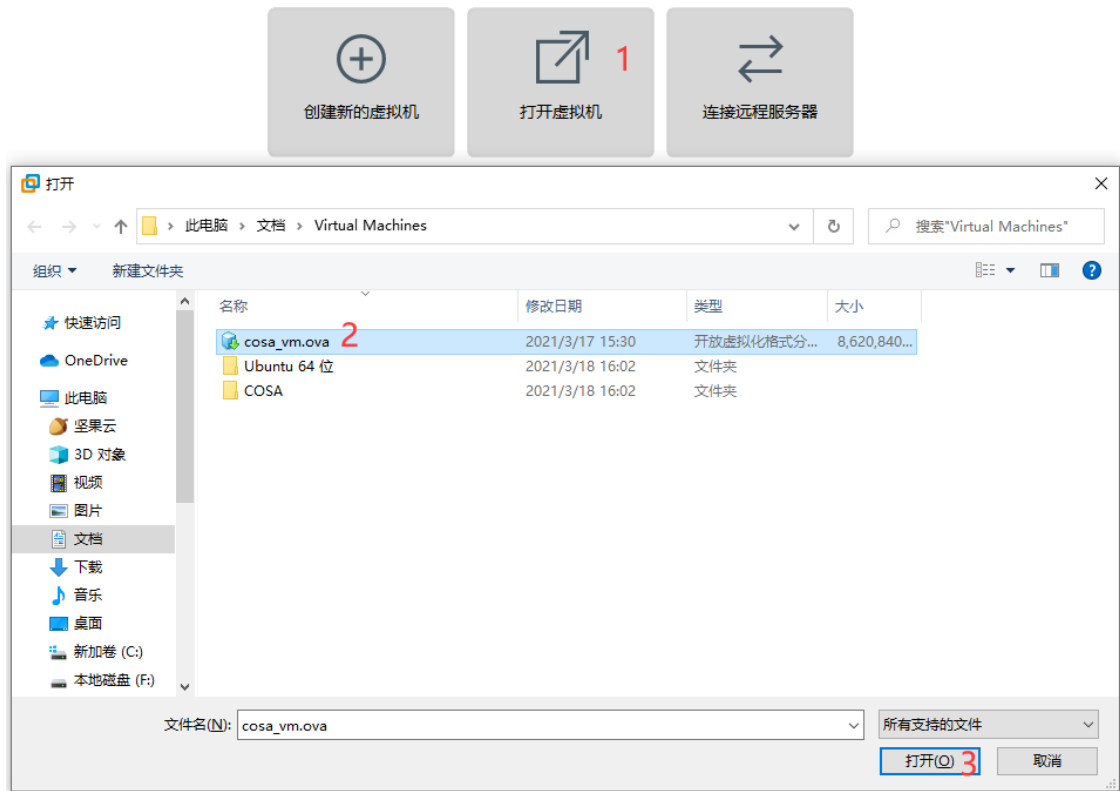
```
scp ftpguest@10.212.43.214:/ftphome/Course/2021-Computer-Organization/cosa_vm.ova 目标路径
```

```
C:\Users\leizongfei>scp ftpguest@10.212.43.214:/ftphome/Course/2021-Computer-Organization/cosa_vm.ova H:\
ftpguest@10.212.43.214's password:
cosa_vm.ova                               17% 2829MB  6.7MB/s   33:18 ETA
```

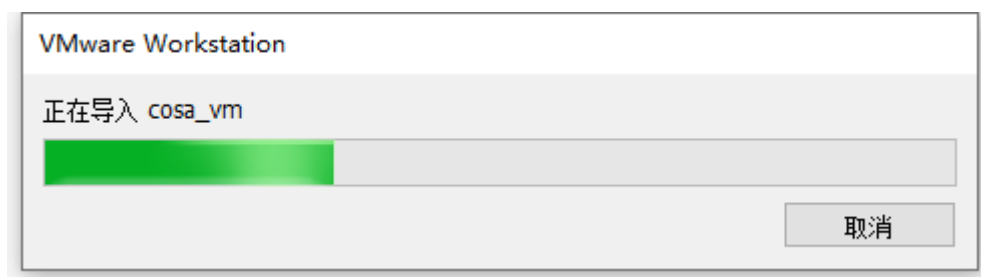
关于 scp 的详细使用方法可以参考：[Linux系统操作（18）：scp（服务器之间的文件传输）](#)

3. 在 VMware Workstation 中打开下载好的 ova 文件

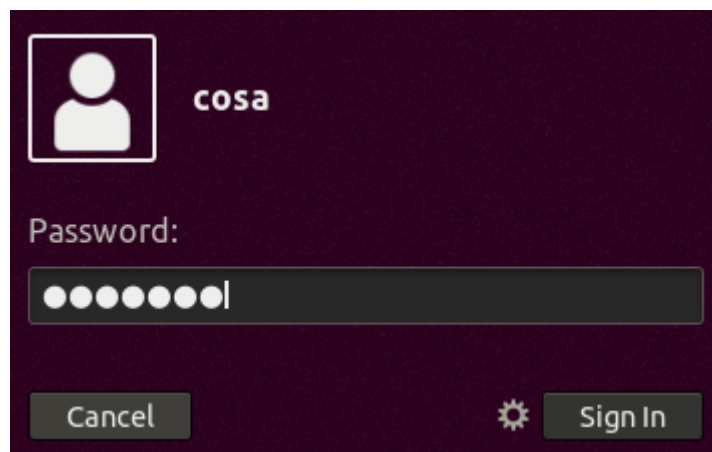
WORKSTATION 15.5 PRO™



若提示导入失败，点击 **重试** 即可。



4. 登录虚拟机



用户名：cosa

密码：cosa

5. chipyard 安装在 `~/code/chipyard` 路径下

```
cosa@ubuntu: ~/code/chipyard
File Edit View Search Terminal Help
cosa@ubuntu:~$ cd code
cosa@ubuntu:~/code$ ls
chipyard  verilator
cosa@ubuntu:~/code$ cd chipyard/
cosa@ubuntu:~/code/chipyard$ ls
build.sbt          init-submodules-no-riscv-tools.log  target
CHANGELOG.md      LICENSE                             tests
common.mk         LICENSE.SiFive                       toolchains
CONTRIBUTING.md  project                             tools
docs              README.md                           variables.mk
env-riscv-tools.sh riscv-tools-install                 vcs.mk
env.sh            scripts                             vlsi
fpga              sims
generators        software
cosa@ubuntu:~/code/chipyard$
```

6. 环境变量

```
source ./env.sh
```

可以把其内容复制到 `.bashrc` 文件中，或者每次新开一个终端都重新运行一次。

自己在虚拟机或者其他 Linux 环境下安装（推荐）

安装并配置虚拟机

1. 安装 `VMware Workstation`
2. 下载虚拟机镜像文件并安装

安装客户机操作系统

虚拟机如同物理机，需要操作系统。您将如何安装客户机操作系统？

安装来源：

☐ 安装程序光盘(D):

DVD RW 驱动器 (E:)

☒ 安装程序光盘映像文件(iso)(M):

C:\Users\leizongfei\Downloads\ubuntu-18.04.5-desktop- ▾

浏览(R)...

☐ 已检测到 Ubuntu 64 位 18.04.5。
该操作系统将使用简易安装。 [\(这是什么?\)](#)

☐ 稍后安装操作系统(S)。

创建的虚拟机将包含一个空白硬盘。

帮助

< 上一步(B)

下一步(N) >

取消

3. 在设置硬盘容量的时候请设置的大一些（这里设置的是50G），毕竟仅安装就已经占了30多个G了。

容量

当前大小: 33.7 GB

系统可用空间: 57.5 GB

最大大小: 50 GB

安装环境依赖

更换 apt 源:

```
# 首先备份源列表
sudo cp /etc/apt/sources.list /etc/apt/sources.list_backup
# 用 vi 编辑 sources.list 文件
sudo vi /etc/apt/sources.list
```

在文件最前面添加阿里云镜像源：

```
# 阿里源
deb http://mirrors.aliyun.com/ubuntu/ bionic main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ bionic-security main restricted universe
multiverse
deb http://mirrors.aliyun.com/ubuntu/ bionic-updates main restricted universe
multiverse
deb http://mirrors.aliyun.com/ubuntu/ bionic-proposed main restricted universe
multiverse
deb http://mirrors.aliyun.com/ubuntu/ bionic-backports main restricted universe
multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ bionic main restricted universe
multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ bionic-security main restricted
universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ bionic-updates main restricted
universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ bionic-proposed main restricted
universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ bionic-backports main restricted
universe multi
```

```
# 刷新列表
sudo apt-get update
```

安装 clash 并配置代理文件

直接使用 git 经常会遇到诸如 `gnutls_handshake() failed: Error in the pull function` 的报错信息，可以通过在 Linux 下安装代理，例如使用 **clash**：

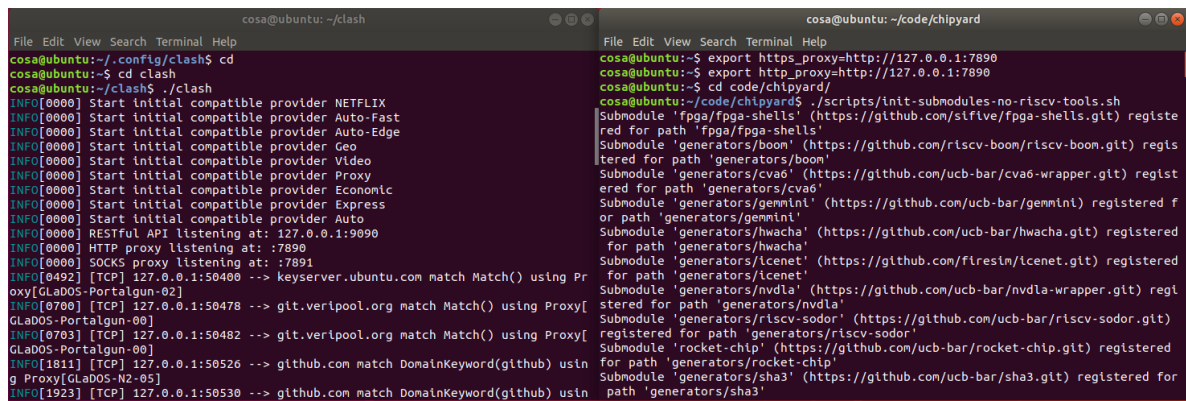
1. 下载 <https://github.com/Dreamacro/clash> 的 release 中的 linux-amd64 版本
2. 解压缩并运行 clash

```
sudo gunzip ***.gz #解压缩
chmod +x clash*** #添加权限
./clash***

# 运行clash，这一步会自动在 ~/.config/下创建clash文件夹，里面有config.yaml 与
Country.mmdb 2个文件，config.yaml 为空文件，Country.mmdb为正在下载的文件，等待
Country.mmdb下载完成之后关闭clash程序
```
3. 将你的配置文件替换掉 `~/.config/config.yaml`
4. 再次打开clash

```
./clash***
```
5. 设置 系统代理 或者 新开一个terminal的代理为 clash 开放的代理端口(7890/7891).

```
export https_proxy=http://127.0.0.1:7890
export http_proxy=http://127.0.0.1:7890
```



```
cosa@ubuntu: ~/clash
File Edit View Search Terminal Help
cosa@ubuntu:~/config/clash$ cd
cosa@ubuntu:~/config/clash$ cd
cosa@ubuntu:~/config/clash$ ./clash
INFO[0000] Start initial compatible provider NETFLIX
INFO[0000] Start initial compatible provider Auto-Fast
INFO[0000] Start initial compatible provider Auto-Edge
INFO[0000] Start initial compatible provider Geo
INFO[0000] Start initial compatible provider Video
INFO[0000] Start initial compatible provider Proxy
INFO[0000] Start initial compatible provider Economic
INFO[0000] Start initial compatible provider Express
INFO[0000] Start initial compatible provider Auto
INFO[0000] RESTful API listening at: 127.0.0.1:9090
INFO[0000] HTTP proxy listening at: :7890
INFO[0000] SOCKS proxy listening at: :7891
INFO[0492] [TCP] 127.0.0.1:50400 --> keyserver.ubuntu.com match Match() using Proxy[GLAD05-Portalgun-02]
INFO[0700] [TCP] 127.0.0.1:50478 --> git.veripool.org match Match() using Proxy[GLAD05-Portalgun-00]
INFO[0703] [TCP] 127.0.0.1:50482 --> git.veripool.org match Match() using Proxy[GLAD05-Portalgun-00]
INFO[1011] [TCP] 127.0.0.1:50526 --> github.com match DomainKeyword(github) using Proxy[GLAD05-N2-05]
INFO[1923] [TCP] 127.0.0.1:50530 --> github.com match DomainKeyword(github) using Proxy[GLAD05-N2-05]
```

```
cosa@ubuntu: ~/code/chipyard
File Edit View Search Terminal Help
cosa@ubuntu:~/code/chipyard$ export https_proxy=http://127.0.0.1:7890
cosa@ubuntu:~/code/chipyard$ export http_proxy=http://127.0.0.1:7890
cosa@ubuntu:~/code/chipyard$ cd code/chipyard/
cosa@ubuntu:~/code/chipyard$ ./scripts/init-submodules-no-riscv-tools.sh
Submodule 'fpga/fpga-shells' (https://github.com/sifive/fpga-shells.git) registered for path 'fpga/fpga-shells'
Submodule 'generators/boom' (https://github.com/riscv-boom/riscv-boom.git) registered for path 'generators/boom'
Submodule 'generators/cva6' (https://github.com/ucb-bar/cva6-wrapper.git) registered for path 'generators/cva6'
Submodule 'generators/gemmini' (https://github.com/ucb-bar/gemmini) registered for path 'generators/gemmini'
Submodule 'generators/hwacha' (https://github.com/ucb-bar/hwacha.git) registered for path 'generators/hwacha'
Submodule 'generators/icenet' (https://github.com/firesin/icenet.git) registered for path 'generators/icenet'
Submodule 'generators/nvdl' (https://github.com/ucb-bar/nvdl-wrapper.git) registered for path 'generators/nvdl'
Submodule 'generators/riscv-sodor' (https://github.com/ucb-bar/riscv-sodor.git) registered for path 'generators/riscv-sodor'
Submodule 'rocket-chip' (https://github.com/ucb-bar/rocket-chip.git) registered for path 'generators/rocket-chip'
Submodule 'generators/sha3' (https://github.com/ucb-bar/sha3.git) registered for path 'generators/sha3'
```

如上图，左边的终端保持开启，如果其他终端的命令走了代理，左边的终端都会记录下来。

【2】是 CC98 上一个免费梯子的教程，【3】为为命令行设置代理的详细说明。

安装依赖并下载 verilator

```
# 新建code文件夹
mkdir code
cd code
```

将下面内容保存为 `env.sh` 文件，然后运行。

```
vi env.sh # vi 会自动新建文件，按i进入编辑模式，编辑结束后按Esc，然后输入:wq回车退出
source env.sh # 用 bash 运行这段代码
```

```
#!/bin/bash

sudo apt-get install -y build-essential bison flex
sudo apt-get install -y libgmp-dev libmpfr-dev libmpc-dev zlib1g-dev vim git
default-jdk default-jre
# install sbt: https://www.scala-sbt.org/release/docs/Installing-sbt-on-Linux.html
echo "deb https://dl.bintray.com/sbt/debian /" | sudo tee -a
/etc/apt/sources.list.d/sbt.list
curl -sL "https://keyserver.ubuntu.com/pks/lookup?
op=get&search=0x2EE0EA64E40A89B84B2DF73499E82A75642AC823" | sudo apt-key add
sudo apt-get update
sudo apt-get install -y sbt
sudo apt-get install -y texinfo gengetopt
sudo apt-get install -y libexpat1-dev libusb-dev libncurses5-dev cmake
# deps for poky
sudo apt-get install -y python3.6 patch diffstat texi2html texinfo subversion
chrpath git wget
# deps for qemu
sudo apt-get install -y libgtk-3-dev gettext
# deps for firemarshal
sudo apt-get install -y python3-pip python3.6-dev rsync libguestfs-tools expat
ctags
# install DTC
sudo apt-get install -y device-tree-compiler

# install verilator
git clone http://git.veripool.org/git/verilator
```

```
cd verilator
git checkout v4.034
autoconf && ./configure && make -j$(nproc) && sudo make install
```

下载 Chipyard 及其子模块

下载前请确保当前终端的 http/https 代理端口为7890（见上一步）

```
git clone https://github.com/ucb-bar/chipyard.git
cd chipyard
./scripts/init-submodules-no-riscv-tools.sh
```

运行结束后，会在当前路径下生成一个 `env.sh` 文件。

搭建工具链

```
./scripts/build-toolchains.sh riscv-tools # for a normal risc-v toolchain
```

运行结束后，会在当前路径下生成一个 `env-riscv-tools.sh` 文件。

修改环境变量

```
source ./env.sh
```

可以把其内容复制到 `.bashrc` 文件中，或者每次新开一个终端都重新运行一次。

注：

- 【1】[Ubuntu 18.04 LTS 更换国内源——解决终端下载速度慢的问题](#)
- 【2】98上有一个[免费的代理Glados](#)
- 【3】[Linux 让终端走代理的方法](#)
- 【4】其余细节可重点参考 Chipyard 官网的 [安装指南](#)