

南昌大学实验报告

姓名：陈华豪

学号：6130116238

邮箱地址：6130116238@email.ncu.edu.cn

专业班级：网络工程161班

实验日期：2019.4.28

课程名称：云计算

实验项目名称

Lab 4 - Live Migration Massively

实验目的

- Understanding the basic techniques for VM migration
- Migrate containers to test your migration skill
- Understanding the concept of checkpoint and restore
- Successfully migrate multiple images either one by one or in a batch

实验基础

- After migrate your first VM, you now need to move to a place with rich resource
- Migrating many docker images and doing it lively until your system halts
- Analysis why it halts and understand the idea of resource bottleneck

实验步骤

- Learn to do live migration in docker
- Using Checkpoint and Restore (CRIU) tool in docker to do this job
- https://criu.org/Live_migration is your source to check

- Ask one of our friend/roommate and migrate your Golang job (with your ID or name) to his/her place
- Measure the duration of migration and collect all other runtime statistics
- **New stuff: Write a script to move docker image automatically**

0. 连接云端服务器

```

Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-36-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Sun Apr 28 15:33:38 UTC 2019

System load:  0.0                       Processes:            103
Usage of /:   35.4% of 24.06GB           Users logged in:      0
Memory usage: 52%                       IP address for eth0:  159.65.70.253
Swap usage:   0%                        IP address for docker0: 172.17.0.1

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch

131 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

*** System restart required ***
Last login: Sun Apr 28 14:36:18 2019 from 106.224.1.178
root@vps-sfo181022:~#

```

1. 启动 docker 服务。

```

*** System restart required ***
Last login: Sun Apr 28 15:33:39 2019 from 106.224.1.178
root@vps-sfo181022:~# service docker start
root@vps-sfo181022:~#

```

2. 检查CRIU是否安装成功，正常情况下会显示looks good。

```
root@vps-sfo181022:~# service docker start
root@vps-sfo181022:~# criu check
Looks good.
root@vps-sfo181022:~#
```

3. 编写批量生成 golang 容器及其检查点以及将检查点文件传至目标主机的脚本 vm1.sh。

```
>#!/bin/bash
int=300
while (( $int<=305 ))
do
    sudo docker run -d -i --name go$int golang bash
    sudo docker checkpoint create --checkpoint-dir=/tmp/checkpoint5 go$int cp$int

    echo succeed
    PASS="asd745699887"
    expect -c "
        set timeout -1
spawn -noecho sudo scp -r /tmp/checkpoint5$int root@ccogito.xyz:/home/checkpoint/huahaochen
expect {
    "*password:*" { send \"$PASS\r\n"; exp_continue }
eof { exit }
}
exit"
    int=$(( $int+1 ))
done
```

```
root@vps-sfo181022:~# vi vm1.sh
```



23:43

```
#!/bin/bash
int=200
while (( $int<=205 ))
do
    sudo docker run -d -i --name go$int golang bash
    sudo docker checkpoint create --checkpoint-dir=/tmp/c
checkpoint5 go$int cp$int

    echo succeed
    PASS="981129"
    expect -c "

    set timeout -1

    spawn -noecho sudo scp -r /tmp/checkpoint5$int root@192.168.8.126:/home/yu
ehao/>云计算实验/clone

    expect {
        "*password:*" { send \"\$PASS\r\"; exp_continue }
        eof { exit }
    }
    exit"
    int=$(( $int+1 ))
done
```

"vm1.sh" [New] 19L, 633C written

```
"vm1.sh" [New] 19L, 633C written
root@vps-sfo181022:~# bash vm1.sh
```

```

HD 4G+ 4G 23:49
#!/bin/bash
int=300
while (( $int<=305 ))
do
    sudo docker run -d -i --name go$int golang bash
    sudo docker checkpoint create --checkpoint-dir=/tmp/c
checkpoint5 go$int cp$int

    echo succeed
    PASS="asd745699887"
    expect -c "

    set timeout -1

    spawn -noecho sudo scp -r /tmp/checkpoint5$int root@ccogito.xyz:/home/chec
kpoint/huahaochen

    expect {
        "*password:*" { send \"\$PASS\r\"; exp_continue }
        eof { exit }
    }
    exit"
    int=$(( $int+1 ))
done

-- INSERT --
1,12 All

```

4. 运行脚本 vm1.sh。

[illegible]

```
>#!/bin/bash

echo "请输入对方目录的地址"
read DIR
for f in $(ls $DIR)
do
    int=1
    sudo docker create --name go-clone${int} golang bash
```

```
docker start --checkpoint-dir=$DIR --checkpoint=$f go-clone
int=$((int+1))
done
```

~

6. 运行脚本 vm2.sh

实验数据或结果

在上述步骤中

实验思考

参考资料
