Contents
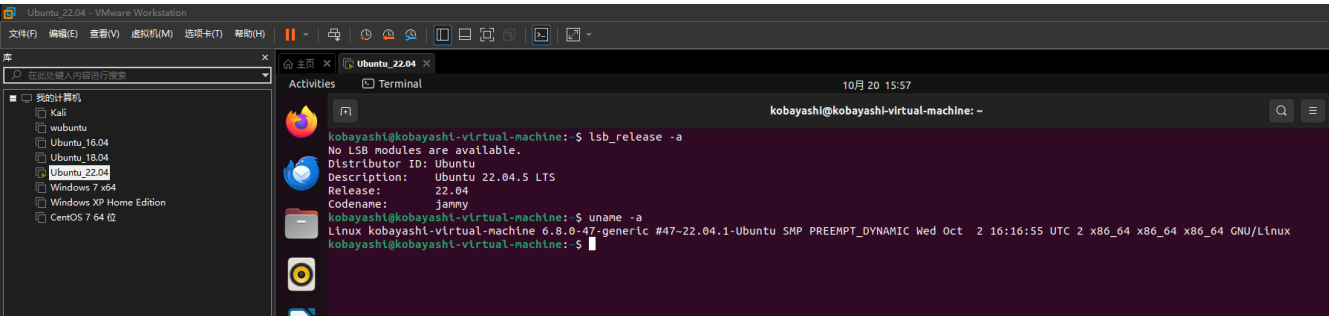
问题描述

CRIU 是一种在用户空间实现的进程或者容器 `checkpoint` 和 `restore` 的方法，从而实现进程或者容器的保存和恢复。请利用 CRIU 实现进程和容器的迁移（迁移种类不限），并测试迁移过程中的性能损耗（如进程停止时间、网络传输时间等）

解决方案

1. 首先，在 VMware 中创建一台 Ubuntu 22.04 的虚拟机

2. 设置 Docker 的官方 apt 仓库，安装 Docker，并测试 Docker

```
# Add Docker's official GPG key:
$ sudo apt-get update
$ sudo apt-get install ca-certificates curl
$ sudo install -m 0755 -d /etc/apt/keyrings
$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
$ sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
$ echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.c
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

```
kobayashi@kobayashi-virtual-machine:~$ sudo apt update
Hit:1 http://cn.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://cn.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://cn.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://cn.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [712 kB]
Get:6 http://cn.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2,108 kB]
Get:7 http://cn.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [103 kB]
Get:8 http://cn.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 DEP-11 Metadata [212 B]
Get:9 http://cn.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,132 kB]
Get:10 http://cn.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [737 kB]
Get:11 http://cn.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [356 kB]
Get:12 http://cn.archive.ubuntu.com/ubuntu jammy-updates/universe DEP-11 48x48 Icons [248 kB]
Get:13 http://cn.archive.ubuntu.com/ubuntu jammy-updates/universe DEP-11 64x64 Icons [401 kB]
Get:14 http://cn.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:15 http://cn.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [5,320 B]
Get:16 http://cn.archive.ubuntu.com/ubuntu jammy-backports/main DEP-11 48x48 Icons [14.6 kB]
Get:17 http://cn.archive.ubuntu.com/ubuntu jammy-backports/main DEP-11 64x64 Icons [19.4 kB]
Get:18 http://cn.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [216 B]
Get:19 http://cn.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [23.1 kB]
Get:20 http://cn.archive.ubuntu.com/ubuntu jammy-backports/universe DEP-11 48x48 Icons [20.1 kB]
Get:21 http://cn.archive.ubuntu.com/ubuntu jammy-backports/universe DEP-11 64x64 Icons [32.0 kB]
Get:22 http://cn.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]
Get:23 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [43.2 kB]
Get:24 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 DEP-11 Metadata [208 B]
Get:25 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [126 kB]
Get:26 http://security.ubuntu.com/ubuntu jammy-security/universe DEP-11 48x48 Icons [83.4 kB]
Get:27 http://security.ubuntu.com/ubuntu jammy-security/universe DEP-11 64x64 Icons [124 kB]
Get:28 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 DEP-11 Metadata [208 B]
Fetched 6,674 kB in 3s (2,414 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
kobayashi@kobayashi-virtual-machine:~$
```

```
kobayashi@kobayashi-virtual-machine:~$ sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203~22.04.1).
ca-certificates set to manually installed.
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 194 kB of archives.
After this operation, 455 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://cn.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.18 [194 kB]
Fetched 194 kB in 0s (1,132 kB/s)
Selecting previously unselected package curl.
(Reading database ... 201888 files and directories currently installed.)
Preparing to unpack .../curl_7.81.0-1ubuntu1.18_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.18) ...
Setting up curl (7.81.0-1ubuntu1.18) ...
Processing triggers for man-db (2.10.2-1) ...
kobayashi@kobayashi-virtual-machine:~$ ls /etc/apt/keyrings/
docker.asc
kobayashi@kobayashi-virtual-machine:~$
```

```
kobayashi@kobayashi-virtual-machine:~$ echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
Hit:1 http://cn.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://cn.archive.ubuntu.com/ubuntu jammy-updates InRelease
Get:3 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Hit:4 http://cn.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [40.7 kB]
Fetched 89.5 kB in 1s (81.8 kB/s)
Reading package lists... Done
kobayashi@kobayashi-virtual-machine:~$
```

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

```
kobayashi@kobayashi-virtual-machine:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras git git-man liberror-perl libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin git git-man liberror-perl libslirp0 pigz
  slirp4netns
0 upgraded, 12 newly installed, 0 to remove and 3 not upgraded.
Need to get 127 MB of archives.
After this operation, 463 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://cn.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://cn.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:3 http://cn.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.11 [955 kB]
Get:4 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.7.22-1 [29.5 MB]
Get:5 http://cn.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.11 [3,165 kB]
Get:6 http://cn.archive.ubuntu.com/ubuntu jammy/main amd64 libslirp0 amd64 4.6.1-1build1 [61.5 kB]
Get:7 http://cn.archive.ubuntu.com/ubuntu jammy/universe amd64 slirp4netns amd64 1.0.1-2 [28.2 kB]
Get:8 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-buildx-plugin amd64 0.17.1-1~ubuntu.22.04~jammy [30.3 MB]
Get:9 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:27.3.1-1~ubuntu.22.04~jammy [15.0 MB]
Get:10 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce amd64 5:27.3.1-1~ubuntu.22.04~jammy [25.6 MB]
Get:11 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-rootless-extras amd64 5:27.3.1-1~ubuntu.22.04~jammy [9,589 kB]
Get:12 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-compose-plugin amd64 2.29.7-1~ubuntu.22.04~jammy [12.7 MB]
Fetched 127 MB in 22s (5,891 kB/s)
Selecting previously unselected package pigz.
(Reading database ... 201895 files and directories currently installed.)
Preparing to unpack .../00-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
```

```
$ docker --version
$ sudo docker pull hello-world
```

```
kobayashi@kobayashi-virtual-machine:~$ docker --version
Docker version 27.3.1, build ce12230
kobayashi@kobayashi-virtual-machine:~$ sudo docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:d211f485f2dd1dee407a80973c8f129f00d54604d2c90732e8e320e5038a0348
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
kobayashi@kobayashi-virtual-machine:~$
```

3. 开启 Docker 的实验模式

```
sudo vi /etc/docker/daemon.json
```

- 将下面内容添加到 daemon.json 文件中

```
{
  "experimental": true
}
```

- 重启 Docker 服务

```
sudo systemctl daemon-reload
sudo systemctl restart docker
```

4. 添加 criu 的 apt 仓, 安装 criu, 并检查 criu

```
$ sudo add-apt-repository ppa:criu/ppa
$ sudo apt install criu
$ criu --version
$ sudo criu check
$ sudo criu check --all
```

```
kobayashi@kobayashi-virtual-machine:~$ sudo add-apt-repository ppa:criu/ppa
Repository: 'deb https://ppa.launchpadcontent.net/criu/ppa/ubuntu/ jammy main'
More info: https://launchpad.net/~criu/+archive/ubuntu/ppa
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/criu-ubuntu-ppa-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/criu-ubuntu-ppa-jammy.list
Adding key to /etc/apt/trusted.gpg.d/criu-ubuntu-ppa.gpg with fingerprint 4E2A48715C45AEEC077B48169B29EEC9246B6CE2
Hit:1 http://cn.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:3 http://cn.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://cn.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://ppa.launchpadcontent.net/criu/ppa/ubuntu jammy InRelease [18.0 kB]
Get:7 https://ppa.launchpadcontent.net/criu/ppa/ubuntu jammy/main amd64 Packages [532 B]
Get:8 https://ppa.launchpadcontent.net/criu/ppa/ubuntu jammy/main Translation-en [208 B]
Fetched 18.8 kB in 2s (9,099 B/s)
Reading package lists... Done
```

```
kobayashi@kobayashi-virtual-machine:~$ sudo apt install criu
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libnet1 libprotobuf-c1
The following NEW packages will be installed:
  criu libnet1 libprotobuf-c1
0 upgraded, 3 newly installed, 0 to remove and 3 not upgraded.
Need to get 836 kB of archives.
After this operation, 3,521 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://cn.archive.ubuntu.com/ubuntu jammy/main amd64 libnet1 amd64 1.1.6+dfsg-3.1build3 [46.9 kB]
Get:2 http://cn.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libprotobuf-c1 amd64 1.3.3-1ubuntu2.1 [20.3 kB]
Software Updater  a.launchpadcontent.net/criu/ppa/ubuntu jammy/main amd64 criu amd64 3.19-1ppa1.24.04 [768 kB]
Fetched 836 kB in 3s (260 kB/s)
Selecting previously unselected package libnet1:amd64.
(Reading database ... 203139 files and directories currently installed.)
Preparing to unpack .../libnet1_1.1.6+dfsg-3.1build3_amd64.deb ...
Unpacking libnet1:amd64 (1.1.6+dfsg-3.1build3) ...
Selecting previously unselected package libprotobuf-c1:amd64.
Preparing to unpack .../libprotobuf-c1_1.3.3-1ubuntu2.1_amd64.deb ...
Unpacking libprotobuf-c1:amd64 (1.3.3-1ubuntu2.1) ...
Selecting previously unselected package criu.
Preparing to unpack .../criu_3.19-1ppa1.24.04_amd64.deb ...
Unpacking criu (3.19-1ppa1.24.04) ...
Setting up libnet1:amd64 (1.1.6+dfsg-3.1build3) ...
Setting up libprotobuf-c1:amd64 (1.3.3-1ubuntu2.1) ...
Setting up criu (3.19-1ppa1.24.04) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
kobayashi@kobayashi-virtual-machine:~$
```

```
kobayashi@kobayashi-virtual-machine:~$ criu --version
Version: 3.19
kobayashi@kobayashi-virtual-machine:~$
```

```
kobayashi@kobayashi-virtual-machine:~$
Looks good.
kobayashi@kobayashi-virtual-machine:~$
sudo: mon_handle_sigchld: waitpid: No c
Warn  (criu/cr-check.c:1346): Nftables
Looks good but some kernel features are
which, depending on your process tree,
dump or restore failure.
kobayashi@kobayashi-virtual-machine:~$
```

5. 测试 checkpoint

- 这里方便起见, 我直接使用一个 busybox 容器, 并在其上运行一个简单的计数脚本

```
$ sudo docker run --security-opt=seccomp:unconfined --name cr -d busybox /bin/sh -c 'i=0; while true; do e
$ sudo docker ps
```

```
kobayashi@kobayashi-virtual-machine:~$ sudo docker run --security-opt=seccomp:unconfined --name cr -d busybox /bin/sh -c 'i=0; while true; do echo $i; i=$(expr $i
+ 1); sleep 1; done'
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
a46fbb00284b: Pull complete
Digest: sha256:768e5c6f5cb6db0794eec98dc7a967f40631746c32232b78a3105fb946f3ab83
Status: Downloaded newer image for busybox:latest
d999ba5177fc7891fb62193c93b1d47ed1cb60f0e4a5eb2c2c3c9dad24d10f06
kobayashi@kobayashi-virtual-machine:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND              CREATED        STATUS        PORTS     NAMES
d999ba5177fc   busybox   "/bin/sh -c 'i=0; wh…"   3 seconds ago   Up 3 seconds            cr
```

- 创建 checkpoint，并记录此时的容器输出状态

```
$ sudo docker checkpoint create cr checkpoint1
$ sudo docker ps -a
$ sudo docker logs cr
```

```
kobayashi@kobayashi-virtual-machine:~$ sudo docker checkpoint create cr checkpoint1
checkpoint1
kobayashi@kobayashi-virtual-machine:~$ sudo docker logs cr
0
1
2
3
4
kobayashi@kobayashi-virtual-machine:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND     CREATED     STATUS     PORTS     NAMES
kobayashi@kobayashi-virtual-machine:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND              CREATED         STATUS                      PORTS     NAMES
d999ba5177fc   busybox   "/bin/sh -c 'i=0; wh…"   2 minutes ago   Exited (137) 2 minutes ago            cr
kobayashi@kobayashi-virtual-machine:~$
```

- 通过 checkpoint 恢复容器状态

```
$ sudo docker start --checkpoint checkpoint1 cr
$ sudo docker ps
$ sudo docker stop cr
$ sudo docker logs cr
```

```
kobayashi@kobayashi-virtual-machine:~$ sudo docker start --checkpoint checkpoint1 cr
kobayashi@kobayashi-virtual-machine:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND              CREATED         STATUS         PORTS     NAMES
d999ba5177fc   busybox   "/bin/sh -c 'i=0; wh…"   4 minutes ago   Up 3 seconds             cr
kobayashi@kobayashi-virtual-machine:~$ sudo docker stop cr
cr
kobayashi@kobayashi-virtual-machine:~$ sudo docker logs cr
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
kobayashi@kobayashi-virtual-machine:~$
```

可见容器成功恢复 checkpoint1 中的状态，从 checkpoint1 中的位置开始计数

6. 实现跨容器迁移

- 重建一个相同的容器 cr-clone，并记录创建该容器所使用的时间

```
$ time sudo docker create --security-opt=seccomp:unconfined --name cr-clone busybox /bin/sh -c 'i=0; while
```

```
kobayashi@kobayashi-virtual-machine:~$ time sudo docker create --security-opt=seccomp:unconfined --name cr-clone busybox /bin/sh -c 'i=0; while true; do echo $i;
i=$(expr $i + 1); sleep 1; done'
5b0ed947276a187e00c8709446fcfa62928e04f964ebc264f861c7b614691483

real    0m0.155s
user    0m0.017s
sys     0m0.005s
kobayashi@kobayashi-virtual-machine:~$ sudo docker container ls -a
CONTAINER ID   IMAGE     COMMAND                CREATED          STATUS                      PORTS     NAMES
5b0ed947276a   busybox   "/bin/sh -c 'i=0; wh…"  56 seconds ago   Created                               cr-clone
d999ba5177fc   busybox   "/bin/sh -c 'i=0; wh…"  33 minutes ago   Exited (137) 28 minutes ago           cr
kobayashi@kobayashi-virtual-machine:~$
```

- 将 cr 容器的 checkpoint1 文件复制到 cr-clone 容器中

```
root@kobayashi-virtual-machine:/var/lib/docker/containers# tree -L 3
├── 5b0ed947276a187e00c8709446fcfa62928e04f964ebc264f861c7b614691483
│   ├── checkpoints
│   ├── config.v2.json
│   └── hostconfig.json
├── d999ba5177fc7891fb62193c93b1d47ed1cb60f0e4a5eb2c2c3c9dad24d10f06
│   ├── checkpoints
│   │   └── checkpoint1
│   ├── config.v2.json
│   ├── d999ba5177fc7891fb62193c93b1d47ed1cb60f0e4a5eb2c2c3c9dad24d10f06-json.log
│   ├── hostconfig.json
│   ├── hostname
│   ├── hosts
│   ├── mounts
│   ├── resolv.conf
│   └── resolv.conf.hash

6 directories, 9 files
root@kobayashi-virtual-machine:/var/lib/docker/containers# time cp d999ba5177fc7891fb62193c93b1d47ed1cb60f0e4a5eb2c2c3c9dad24d10f06/checkpoints/checkpoint1/ -rf 5
b0ed947276a187e00c8709446fcfa62928e04f964ebc264f861c7b614691483/checkpoints/

real    0m0.008s
user    0m0.005s
sys     0m0.003s
root@kobayashi-virtual-machine:/var/lib/docker/containers# tree -L 3
├── 5b0ed947276a187e00c8709446fcfa62928e04f964ebc264f861c7b614691483
│   ├── checkpoints
│   │   └── checkpoint1
│   ├── config.v2.json
│   └── hostconfig.json
├── d999ba5177fc7891fb62193c93b1d47ed1cb60f0e4a5eb2c2c3c9dad24d10f06
│   ├── checkpoints
│   │   └── checkpoint1
│   ├── config.v2.json
│   ├── d999ba5177fc7891fb62193c93b1d47ed1cb60f0e4a5eb2c2c3c9dad24d10f06-json.log
│   ├── hostconfig.json
│   ├── hostname
│   ├── hosts
│   ├── mounts
│   ├── resolv.conf
│   └── resolv.conf.hash

7 directories, 9 files
```

- 通过 checkpoint1 中启动 cr-clone 容器

```
kobayashi@kobayashi-virtual-machine:~$ sudo docker start --checkpoint checkpoint1 cr-clone
kobayashi@kobayashi-virtual-machine:~$ sudo docker ps
CONTAINER ID    IMAGE     COMMAND              CREATED         STATUS        PORTS      NAMES
5b0ed947276a    busybox   "/bin/sh -c 'i=0; wh…"  14 minutes ago  Up 3 seconds             cr-clone
kobayashi@kobayashi-virtual-machine:~$ sudo docker stop cr-clone
cr-clone
kobayashi@kobayashi-virtual-machine:~$ sudo docker logs cr
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
kobayashi@kobayashi-virtual-machine:~$
```

```
kobayashi@kobayashi-virtual-machine:~$ sudo docker logs cr-clone
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
```

可以看见，cr-clone 容器成功恢复了 cr 容器的 checkpoint1 状态，从 checkpoint1 的位置，也就是从 5 开始计数

实验结果

- 迁移过程中的性能损耗

由于我在本次实验中测试的是本地的容器迁移，因此，迁移过程中的性能损耗主要取决于新容器的创建时间以及 checkpoints

的复制等。而由于我在本次实验中，使用的是一个较为轻量的容器，且运行的脚本本身消耗不大，所以可以从上面的实验中看出，实际迁移所损耗的时间相当小。

遇到的问题及解决方法

本次实验中我遇到的问题主要如下：

- criu check 报错：这个问题主要在 wsl 中出现，我在更换到 VMware 中后，问题解决。
- checkpoint 恢复失败：这个问题是我在使用 criu 3.18 时遇到的，我在将 criu 版本更换到 3.19 后，问题解决。