COLO_FT 测试

0. 设置追踪时间和增加追踪点

1. 主节点启动脚本

```
imagefolder="XXX"

qemu-system-x86_64 -enable-kvm -cpu qemu64,+kvmclock -m 16G -smp 8 -qmp stdio \
   -device piix3-usb-uhci -device usb-tablet -name primary \
   -netdev tap,id=hn0,vhost=off,helper=/usr/lib/qemu/qemu-bridge-helper \
   -device rtl8139,id=e0,netdev=hn0 \
   -chardev socket,id=mirror0,host=0.0.0.0,port=9003,server,nowait \
   -chardev socket,id=compare1,host=0.0.0.0,port=9004,server,wait \
   -chardev socket,id=compare0,host=127.0.0.1,port=9001,server,nowait \
   -chardev socket,id=compare0-0,host=127.0.0.1,port=9001 \
   -chardev socket,id=compare_out,host=127.0.0.1,port=9005,server,nowait \
   -chardev socket,id=compare_out0,host=127.0.0.1,port=9005 \
   -object filter-mirror,id=m0,netdev=hn0,queue=tx,outdev=mirror0 \
   -object filter-redirector,netdev=hn0,id=redire0,queue=rx,indev=compare_out \
   -object filter-redirector,netdev=hn0,id=redire1,queue=rx,outdev=compare0 \
   -object iothread,id=iothread1 \
   -object colo-compare,id=comp0,primary_in=compare0-0,secondary_in=compare1,\
outdev=compare_out0,iothread=iothread1 \
   -drive if=ide,id=colo-disk0,driver=quorum,read-pattern=fifo,vote-
threshold=1,\
children.0.file.filename=$imagefolder/primary.qcow2,children.0.driver=qcow2 -S
```

2. 设置从节点启动脚本

```
imagefolder="XXX"
primary_ip=XXX.XXX.XXX.XXX

# qemu-img create -f qcow2 $imagefolder/secondary-active.qcow2 10G

# qemu-img create -f qcow2 $imagefolder/secondary-hidden.qcow2 10G

qemu-system-x86_64 -enable-kvm -cpu qemu64,+kvmclock -m 16G -smp 8 -qmp stdio \
   -device piix3-usb-uhci -device usb-tablet -name secondary \
   -netdev tap,id=hn0,vhost=off,helper=/usr/lib/qemu/qemu-bridge-helper \
   -device rtl8139,id=e0,netdev=hn0 \
   -chardev socket,id=red0,host=$primary_ip,port=9003,reconnect=1 \
   -chardev socket,id=red1,host=$primary_ip,port=9004,reconnect=1 \
   -object filter-redirector,id=f1,netdev=hn0,queue=tx,indev=red0 \
   -object filter-redirector,id=f2,netdev=hn0,queue=rx,outdev=red1 \
   -object filter-rewriter,id=rew0,netdev=hn0,queue=all \
   -drive if=none,id=parent0,file.filename=$imagefolder/primary-
copy.qcow2,driver=qcow2 \
   -drive
if=none,id=childs0,driver=replication,mode=secondary,file.driver=qcow2,\
top-id=colo-disk0,file.file.filename=$imagefolder/secondary-active.qcow2,\
```

```
file.backing.driver=qcow2,file.backing.file.filename=$imagefolder/secondary-
hidden.qcow2,\
file.backing.backing=parent0 \
   -drive if=ide,id=colo-disk0,driver=quorum,read-pattern=fifo,vote-
threshold=1,\
children.0=childs0 \
   -incoming tcp:0.0.0.0:9998
```

3. 启动脚本打印信息

先启动主节点在启动从节点

```
# primary
----- output message -----
imagefolder = /home/data/mengsen/colo-test
binfolder = /home/data/mengsen/qemu/qemu-5.0.0/x86_64-softmmu/
host_ip_inner = 0.0.0.0
host_ip_out = 127.0.0.1

# secondary
----- output message -----
imagefolder = /home/data/mengsen/colo-test
binfolder =
primary_ip = localhost
```

4. 输入QMP命令启动COLO

先输入从节点再输入主节点

```
// secondary
{'execute':'qmp_capabilities'}
{'execute': 'nbd-server-start', 'arguments': {'addr': {'type': 'inet', 'data':
{'host': '0.0.0.0', 'port': '9999'} } } }
{'execute': 'nbd-server-add', 'arguments': {'device': 'parent0', 'writable':
true } }

// primary
{'execute':'qmp_capabilities'}
// peer ip
{'execute': 'human-monitor-command', 'arguments': {'command-line': 'drive_add -n
buddy
driver=replication,mode=primary,file.driver=nbd,file.host=20.21.22.71,file.port=
9999,file.export=parent0,node-name=replication0'}}
{'execute': 'x-blockdev-change', 'arguments':{'parent': 'colo-disk0', 'node':
'replication0' } }
{'execute': 'migrate-set-capabilities', 'arguments': {'capabilities': [
{'capability': 'x-colo', 'state': true } ] } }
{'execute': 'migrate', 'arguments': {'uri': 'tcp:20.21.22.71:9998' } }
```

```
// secondary
{'execute':'qmp_capabilities'}
{'execute': 'nbd-server-start', 'arguments': {'addr': {'type': 'inet', 'data':
{'host': '0.0.0.0', 'port': '9999'} } } }
{'execute': 'nbd-server-add', 'arguments': {'device': 'parent0', 'writable':
true } }

// primary
{'execute':'qmp_capabilities'}
{'execute': 'human-monitor-command', 'arguments': {'command-line': 'drive_add -n
buddy
driver=replication,mode=primary,file.driver=nbd,file.host=127.0.0.2,file.port=99
99,file.export=parent0,node-name=replication0'}}
{'execute': 'x-blockdev-change', 'arguments':{'parent': 'colo-disk0', 'node':
'replication0' } }
{'execute': 'migrate-set-capabilities', 'arguments': {'capabilities': [
{'capability': 'x-colo', 'state': true } ] } }
{'execute': 'migrate', 'arguments': {'uri': 'tcp:127.0.0.2:9998' } }
```

5. 开启追踪时间设置输出文件

```
// 这条命令借用了 QEMU Monitor 的逻辑，理论上在 Monitor 能用的命令这里都可以用
{'execute': 'human-monitor-command', 'arguments': {'command-line': 'logfile
/home/data/mengsen/log/test_SVM.log'}}
// 这条命令是QMP提供的
{'execute': 'trace-event-set-state', 'arguments': {'name': 'colo_send_message',
'enable': true} }
{'execute': 'trace-event-set-state', 'arguments': {'name': 'colo_send_message',
'enable': true} }
{'execute': 'trace-event-set-state', 'arguments': {'name':
'colo_vm_state_change', 'enable': true} }
{'execute': 'trace-event-set-state', 'arguments': {'name': 'VM_state', 'enable':
false} }
{'execute': 'trace-event-set-state', 'arguments': {'name':
'colo_checkpoint_user_log_secondary', 'enable': true} }
```

6. 开启VNC运行脚本

```
{'execute': 'human-monitor-command', 'arguments': {'command-line': 'change vnc
0.0.0.0:1'}}
{'execute': 'human-monitor-command', 'arguments': {'command-line': 'change vnc
0.0.0.0:1'}}
human-monitor-command command-line='change vnc 0.0.0.0:1'
```

7. 一些QMP常用命令

```
{'execute': 'human-monitor-command', 'arguments': {'command-line': 'info cpus'}}
{"return": "* CPU #0: thread_id=21001\r\n  CPU #1: thread_id=21004\r\n  CPU #2:
thread_id=21006\r\n  CPU #3: thread_id=21008\r\n  CPU #4: thread_id=21010\r\n
CPU #5: thread_id=21012\r\n  CPU #6: thread_id=21014\r\n  CPU #7:
thread_id=21015\r\n"}

{"execute": "query-memdev" }
{"return": [{"prealloc": false, "host-nodes": [], "size": 17179869184, "merge":
true, "dump": true, "policy": "default", "id": "pc.ram"}]}
```

8. 一些结论

1. 对于这两个模拟的设备在同一主机上，网络延迟几乎可以忽略
2. 越是模拟高性能设备（可能和`ram`大小以及`cpu cache`大小有关），同步时间越长
3. 观察log文件得出大概每17秒需要同步一次，一次同步时间大概3秒
4. `colo_do_checkpoint_transaction`这个函数在`while`中一直运行，退出的时候也不清理，利用`qemu_sem_wait`进行时间控制，直到结束才会清理
5. 对于`colo_incoming_process_checkpoint`能看到他走到函数末端，然后进入等待直到收到`CHECKPOINT_REQUES`消息唤醒
6. ```
   firewall-cmd --query-port=xxx/tcp => no
   firewall-cmd --add-port=xxx/tcp => success
   ps -ef |grep main.py |awk '{print $2}'|xargs kill -9
   kill -9 $(ps -ef|grep 进程名关键字|gawk '$0 !~/grep/ {print $2}' |tr -s '\n' '
   ')
   ```
7. 当主从节点配置不一致时会报错误

```
qemu-system-x86_64: failed to save SaveStateEntry with id(name): 2(ram)
```
```