工程实践与科技创新 III-D 作业 2

姓名:侯晟元　　学号:518021910604

要求:

(1)QEMU Source Code Compilation and Installation. (Required)

(2)Using Benchmarks(e.g. sysbench and stream bench) for performance testing in QEMU. (Required)

(3)QEMU Source Code Modification. (Optional)

内容简介

QEMU 是一套由法布里斯·贝拉(Fabrice Bellard)所编写的以 GPL 许可证分发源码的模拟处理器，在 GNU/Linux 平台上使用广泛。Bochs，PearPC 等与其类似，但不具备其许多特性，比如高速度及跨平台的特性，通过 KQEMU 这个闭源的加速器，QEMU 能模拟至接近真实电脑的速度。

-----百度百科

一. QEMU 源代码编译与安装

参考链接:https://www.qemu.org/download/#source

```
root@skywalker:/# sudo apt-get install git libglib2.0-dev libfdt-dev libpixman-1
-dev zlib1g-dev
```

```
root@skywalker:/# sudo apt-get install libibverbs-dev libjpeg8-dev libncurses5-d
ev libnuma-dev
```

按照参考链接中的指导按部就班。下载 QEMU5.1.0 源代码，解压源代码压缩文件

```
root@skywalker:/# wget https://download.qemu.org/qemu-5.1.0.tar.xz
--2020-10-05 13:55:24--  https://download.qemu.org/qemu-5.1.0.tar.xz
Resolving download.qemu.org (download.qemu.org)... 172.99.69.163
Connecting to download.qemu.org (download.qemu.org)|172.99.69.163|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 62911540 (60M) [application/x-xz]
Saving to: ‘qemu-5.1.0.tar.xz’

qemu-5.1.0.tar.xz                   100%[===================================================================================>]  60.00M   418KB/s    in 2m 10s

2020-10-05 13:57:36 (471 KB/s) - ‘qemu-5.1.0.tar.xz’ saved [62911540/62911540]
```

```
root@skywalker:/# tar xvJf qemu-5.1.0.tar.xz
```

```
root@skywalker:/# ls
bin                        dev   initrd.img      lost+found  proc              run   sys  vmhgfs-fuse  vmware-config-tools.pl     vmware-uninstall-tools.pl
boot                       doc   initrd.img.old  media       qemu-5.1.0        sbin  tmp  vmlinuz      vmware-guestproxycerttool  vmware-user
cdrom                      etc   lib             mnt         qemu-5.1.0.tar.xz snap  usr  vmlinuz.old  vmware-hgfsclient          vmware-xferlogs
CentOS-7-x86_64-DVD-2003.iso home lib64           opt         root              srv   var  vm-support   vmware-toolbox-cmd
```

解压后进入 qemu-5.1.0 目录下编译安装(./configure+make+make install)。

```
root@skywalker:/# cd qemu-5.1.0
```

```
root@ubuntu:/qemu-5.1.0# ./configure --enable-kvm --enable-debug --enable-vnc --enable-werror --target-list="x86_64-softmmu"
Install prefix    /usr/local
BIOS directory    /usr/local/share/qemu
firmware path     /usr/local/share/qemu-firmware
binary directory  /usr/local/bin
library directory /usr/local/lib
module directory  /usr/local/lib/qemu
libexec directory /usr/local/libexec
include directory /usr/local/include
config directory  /usr/local/etc
local state directory   /usr/local/var
Manual directory  /usr/local/share/man
ELF interp prefix /usr/gnemul/qemu-%M
Build directory   /qemu-5.1.0
Source path       /qemu-5.1.0
GIT binary        git
GIT submodules
C compiler        cc
Host C compiler   cc
C++ compiler      c++
Objective-C compiler cc
ARFLAGS           rv
CFLAGS            -g
```

```
root@ubuntu:/qemu-5.1.0# make -j4
  GEN     x86_64-softmmu/config-devices.mak.tmp
  GEN     config-host.h
make[1]: Entering directory '/qemu-5.1.0/slirp'
  GEN     /qemu-5.1.0/slirp/src/libslirp-version.h
  CC      /qemu-5.1.0/slirp/src/tcp_timer.o
  CC      cs.o
  GEN     qemu-options.def
  GEN     x86_64-softmmu/config-devices.mak
  GEN     qapi-gen
  CC      /qemu-5.1.0/slirp/src/state.o
  CC      utils.o
  GEN     storage-daemon/qapi/qapi-gen
  CC      SStream.o
  CC      /qemu-5.1.0/slirp/src/ip_input.o
  CC      MCInstrDesc.o
  CC      MCRegisterInfo.o
  CC      arch/ARM/ARMDisassembler.o
  CC      /qemu-5.1.0/slirp/src/dhcpv6.o
  CC      /qemu-5.1.0/slirp/src/ndp_table.o
  CC      /qemu-5.1.0/slirp/src/ip6_icmp.o
  GEN     trace/generated-tcg-tracers.h
  CC      /qemu-5.1.0/slirp/src/ip6_input.o
  GEN     trace/generated-helpers-wrappers.h
```

执行 make 阶段后最终结果如下，未出现 error，指令执行成功。注意，由于指令执行过程过长，因此仅仅在此展示最后执行结束时的状态。

```
  CC      x86_64-softmmu/target/i386/cc_helper.o
  CC      x86_64-softmmu/target/i386/excp_helper.o
  CC      x86_64-softmmu/target/i386/fpu_helper.o
  CC      x86_64-softmmu/target/i386/int_helper.o
  CC      x86_64-softmmu/target/i386/mem_helper.o
  CC      x86_64-softmmu/target/i386/misc_helper.o
  CC      x86_64-softmmu/target/i386/mpx_helper.o
  CC      x86_64-softmmu/target/i386/seg_helper.o
  CC      x86_64-softmmu/target/i386/smm_helper.o
  CC      x86_64-softmmu/target/i386/svm_helper.o
  CC      x86_64-softmmu/target/i386/machine.o
  CC      x86_64-softmmu/target/i386/arch_memory_mapping.o
  CC      x86_64-softmmu/target/i386/arch_dump.o
  CC      x86_64-softmmu/target/i386/monitor.o
  CC      x86_64-softmmu/target/i386/kvm.o
  CC      x86_64-softmmu/target/i386/hyperv.o
  CC      x86_64-softmmu/target/i386/sev.o
  GEN     trace/generated-helpers.c
  CC      x86_64-softmmu/trace/control-target.o
  CC      x86_64-softmmu/softmmu/main.o
  CC      x86_64-softmmu/gdbstub-xml.o
  CC      x86_64-softmmu/trace/generated-helpers.o
  LINK    x86_64-softmmu/qemu-system-x86_64
```

执行 make install 指令后运行结果如下，未出现 error，指令执行成功。注意，由于指令执行过程过长，因此仅仅在此展示最后执行结束时的状态

```
root@ubuntu:/qemu-5.1.0# make install
config-host.mak is out-of-date, running configure
Install prefix    /usr/local
BIOS directory    /usr/local/share/qemu
firmware path     /usr/local/share/qemu-firmware
binary directory  /usr/local/bin
library directory /usr/local/lib
module directory  /usr/local/lib/qemu
libexec directory /usr/local/libexec
include directory /usr/local/include
config directory  /usr/local/etc
local state directory   /usr/local/var
Manual directory  /usr/local/share/man
ELF interp prefix /usr/gnemul/qemu-%M
Build directory   /qemu-5.1.0
Source path       /qemu-5.1.0
GIT binary        git
GIT submodules
C compiler        cc
Host C compiler   cc
C++ compiler      c++
Objective-C compiler cc
```

```
done
for s in 16x16 24x24 32x32 48x48 64x64 128x128 256x256 512x512; do \
        mkdir -p "/usr/local/share/icons/hicolor/${s}/apps"; \
        install -c -m 0644 /qemu-5.1.0/ui/icons/qemu_${s}.png \
                "/usr/local/share/icons/hicolor/${s}/apps/qemu.png"; \
done; \
mkdir -p "/usr/local/share/icons/hicolor/32x32/apps"; \
install -c -m 0644 /qemu-5.1.0/ui/icons/qemu_32x32.bmp \
        "/usr/local/share/icons/hicolor/32x32/apps/qemu.bmp"; \
mkdir -p "/usr/local/share/icons/hicolor/scalable/apps"; \
install -c -m 0644 /qemu-5.1.0/ui/icons/qemu.svg \
        "/usr/local/share/icons/hicolor/scalable/apps/qemu.svg"
mkdir -p "/usr/local/share/applications"
install -c -m 0644 /qemu-5.1.0/ui/qemu.desktop \
        "/usr/local/share/applications/qemu.desktop"
install -d -m 0755 "/usr/local/share/qemu/keymaps"
set -e; for x in da     en-gb  et  fr     fr-ch  is  lt  no  pt-br  sv ar      d
e     en-us  fi  fr-be  hr     it  lv  nl         pl  ru     th de-ch  es     fo
  fr-ca  hu     ja  mk  pt  sl     tr bepo    cz; do \
        install -c -m 0644 /qemu-5.1.0/pc-bios/keymaps/$x "/usr/local/share/qemu
/keymaps"; \
done
install -c -m 0644 /qemu-5.1.0/trace-events-all "/usr/local/share/qemu/trace-eve
nts-all"
```

以上两个指令的运行结果说明了编译 QEMU 源代码和安装成功，下面展示安装结果。

输入 qemu-img -V 命令效果如下。

```
root@skywalker:/# qemu-img -V
qemu-img version 5.1.0
Copyright (c) 2003-2020 Fabrice Bellard and the QEMU Project developers
```

可以看到输出版本与安装版本号相同，说明 QEMU 编译与安装成功。

任务二需要对 QEMU 下运行的程序进行测试，因此需要用 virt-manager 开启虚拟机。

检测系统是否支持硬件虚拟化

```
root@skywalker:/# egrep -c '(svm|vmx)' /proc/cpuinfo
2
```

输出大于 0 说明系统支持硬件虚拟化。

```
root@skywalker:/# kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
```

安装相关的依赖包 virt-manager， libvirt-bin, bridge-utils

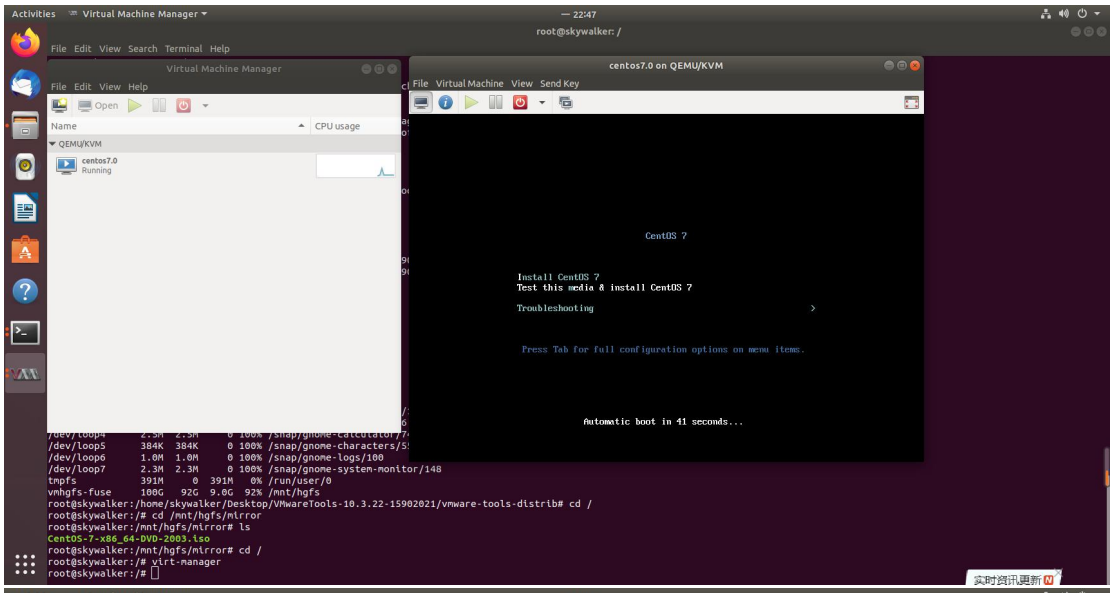
```
root@skywalker:/# sudo apt install virt-manager
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  augeas-lenses bridge-utils cpu-checker dmeventd ebtables
  gir1.2-appindicator3-0.1 gir1.2-gtk-vnc-2.0 gir1.2-libosinfo-1.0
  gir1.2-libvirt-glib-1.0 gir1.2-spiceclientglib-2.0
  gir1.2-spiceclientgtk-3.0 ibverbs-providers ipxe-qemu
  ipxe-qemu-256k-compat-efi-roms libaio1 libaugeas0 libcacard0
  libdevmapper-event1.02.1 libfdt1 libgovirt-common libgovirt2
  libgtk-vnc-2.0-0 libgvnc-1.0-0 libibverbs1 libiscsi7 liblvm2app2.2
  liblvm2cmd2.02 libnetcf1 libnl-route-3-200 libosinfo-1.0-0 libphodav-2.0-0
  libphodav-2.0-common libpython-stdlib librados2 librbd1 librdmacm1
  libreadline5 libsdl1.2debian libspice-client-glib-2.0-8
  libspice-client-gtk-3.0-5 libspice-server1 libusbredirhost1
  libusbredirparser1 libvirt-clients libvirt-daemon
  libvirt-daemon-driver-storage-rbd libvirt-daemon-system libvirt-glib-1.0-0
  libvirt0 libxen-4.9 libxenstore3.0 libxml2-utils lvm2 msr-tools osinfo-db
  python python-asn1crypto python-cairo python-certifi python-cffi-backend
  python-chardet python-cryptography python-dbus python-enum34 python-gi
  python-gi-cairo python-idna python-ipaddr python-ipaddress python-libvirt
  python-libxml2 python-minimal python-openssl python-pkg-resources
  python-requests python-six python-urllib3 python2.7 python2.7-minimal
  qemu-block-extra qemu-kvm qemu-system-common qemu-system-x86 qemu-utils
  seabios sharutils spice-client-glib-usb-acl-helper virt-viewer virtinst
Suggested packages:
  augeas-doc augeas-tools libosinfo-l10n gstreamer1.0-plugins-bad
```

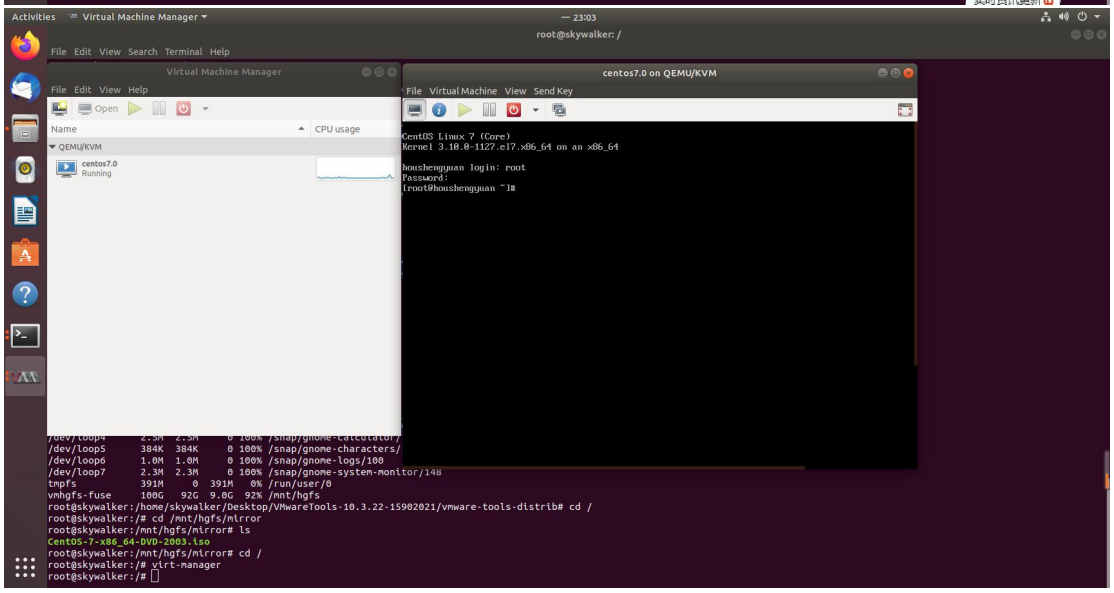
```
root@skywalker:/# sudo apt install bridge-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
bridge-utils is already the newest version (1.5-15ubuntu1).
bridge-utils set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 94 not upgraded.
root@skywalker:/# sudo apt install libvirt-bin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  libvirt-bin
0 upgraded, 1 newly installed, 0 to remove and 94 not upgraded.
Need to get 5,796 B of archives.
After this operation, 121 kB of additional disk space will be used.
Get:1 http://cn.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libvirt-bin
 amd64 4.0.0-1ubuntu8.17 [5,796 B]
Fetched 5,796 B in 3s (2,208 B/s)
Selecting previously unselected package libvirt-bin.
(Reading database ... 167322 files and directories currently installed.)
Preparing to unpack .../libvirt-bin_4.0.0-1ubuntu8.17_amd64.deb ...
Unpacking libvirt-bin (4.0.0-1ubuntu8.17) ...
Setting up libvirt-bin (4.0.0-1ubuntu8.17) ...
```

启动 libvirtd 服务，并检测服务状态，可以看到 libvirtd 的服务状态已经打开

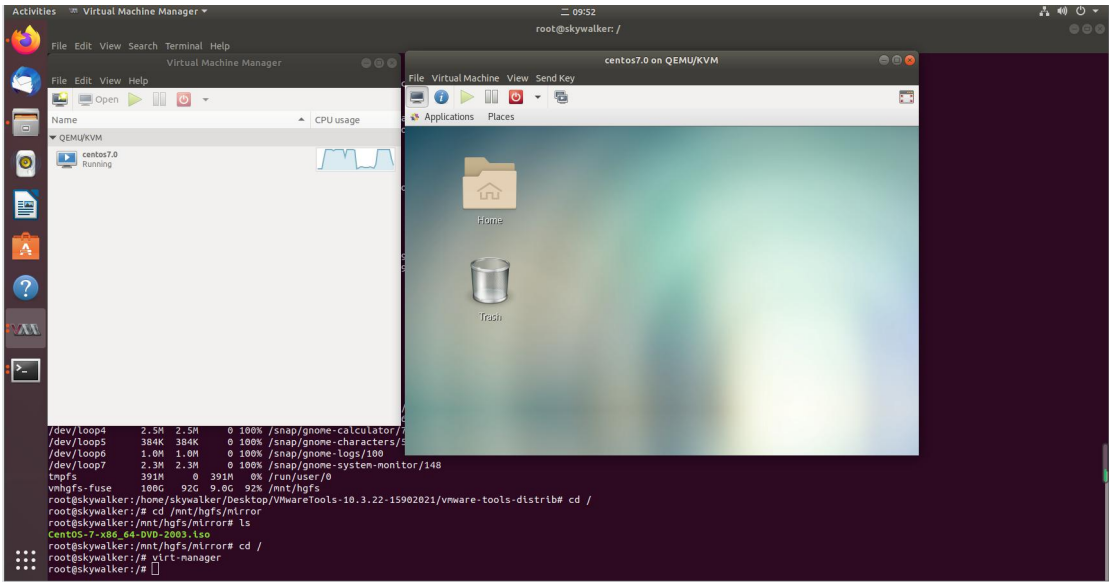
```
root@skywalker:/# sudo service libvirtd start
root@skywalker:/# sudo update-rc.d libvirtd enable
root@skywalker:/# service libvirtd status
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor preset
   Active: active (running) since Mon 2020-10-05 21:33:25 CST; 8min ago
     Docs: man:libvirtd(8)
           https://libvirt.org
 Main PID: 4704 (libvirtd)
    Tasks: 19 (limit: 32768)
   CGroup: /system.slice/libvirtd.service
           ├─4704 /usr/sbin/libvirtd
           ├─5030 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/defaul
           └─5031 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/defaul

10月 05 21:33:25 skywalker systemd[1]: Started Virtualization daemon.
10月 05 21:33:27 skywalker dnsmasq[5030]: started, version 2.79 cachesize 150
10月 05 21:33:27 skywalker dnsmasq[5030]: compile time options: IPv6 GNU-getopt
10月 05 21:33:27 skywalker dnsmasq-dhcp[5030]: DHCP, IP range 192.168.122.2 --
10月 05 21:33:27 skywalker dnsmasq-dhcp[5030]: DHCP, sockets bound exclusively
10月 05 21:33:27 skywalker dnsmasq[5030]: reading /etc/resolv.conf
10月 05 21:33:27 skywalker dnsmasq[5030]: using nameserver 127.0.0.53#53
10月 05 21:33:27 skywalker dnsmasq[5030]: read /etc/hosts - 7 addresses
10月 05 21:33:27 skywalker dnsmasq[5030]: read /var/lib/libvirt/dnsmasq/default
10月 05 21:33:27 skywalker dnsmasq-dhcp[5030]: read /var/lib/libvirt/dnsmasq/de
lines 1-22/22 (END)
```

接下来在 virt-manager 中 QEMU/KVM 环境下开启 CentOS7 虚拟机





接下来安装 CentOS 的图形化界面



虚拟机安装成功，登陆成功，任务一完成

二. 使用 sysbench 和 streambench 两套基准程序对 QEMU 进行性能测试

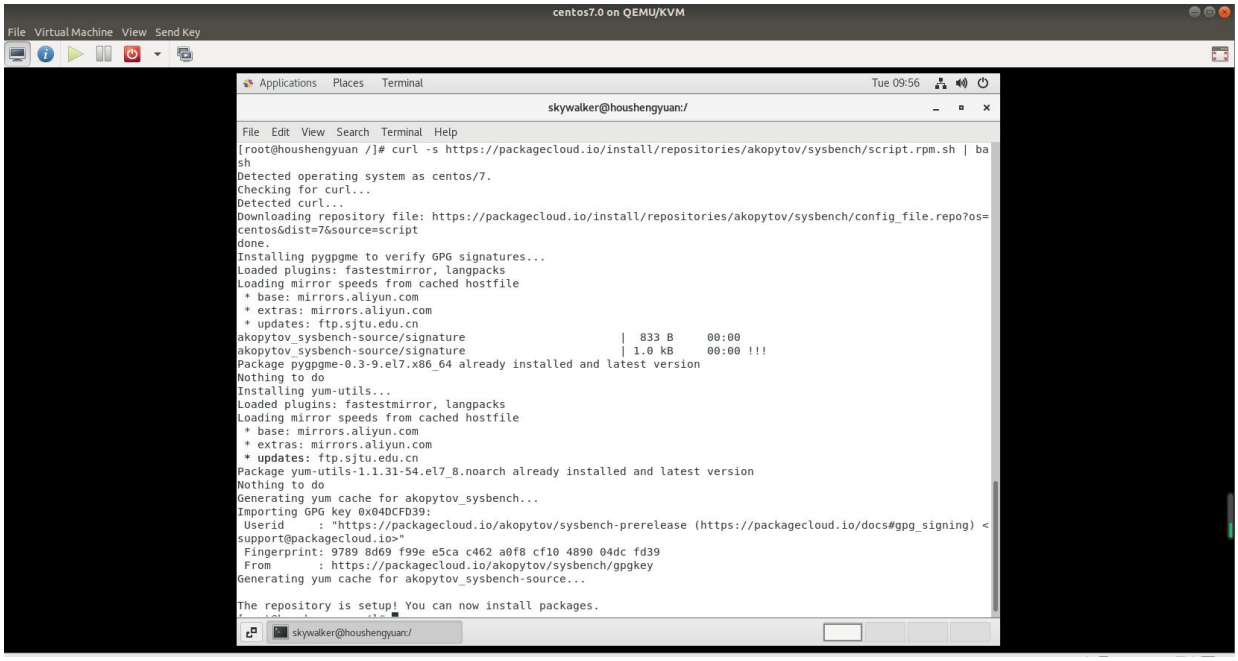
（一）首先要下载 sysbench 和 streambench,编译安装后在 CentOS 上对其进行测试。

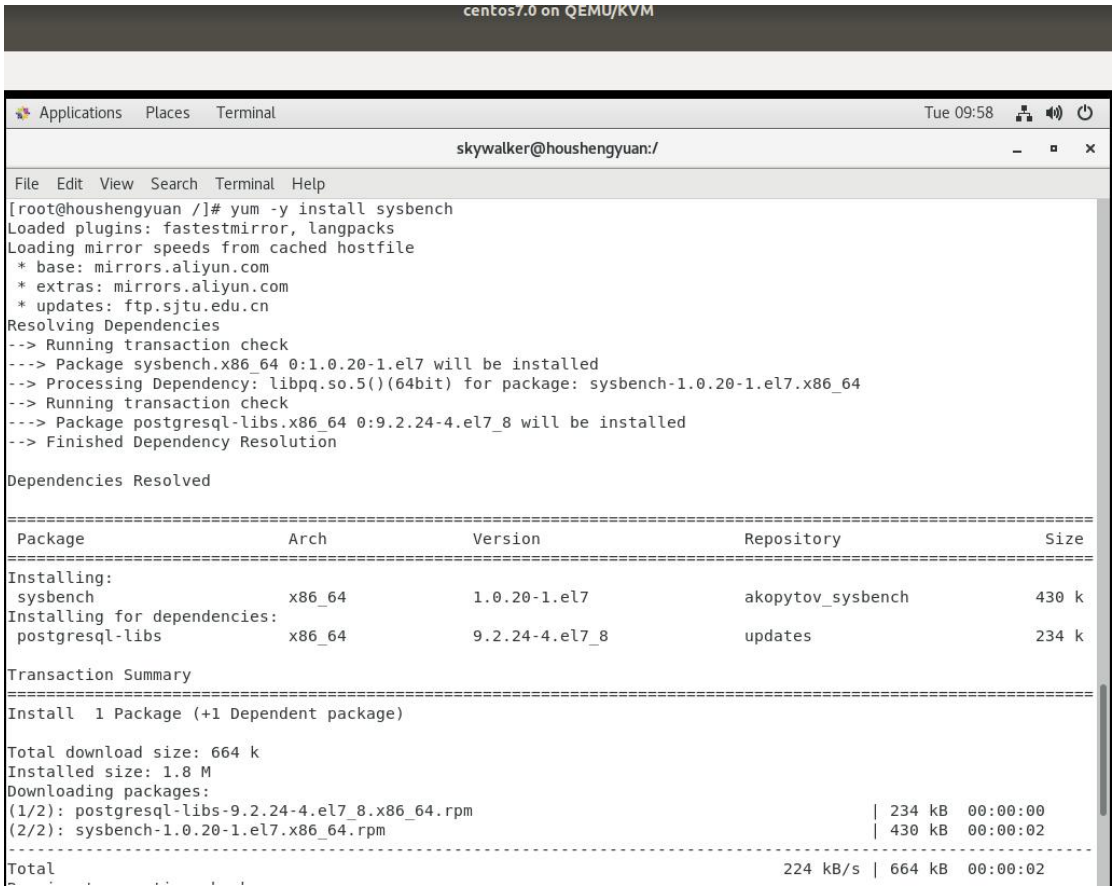
(1)sysbench 下载

参考链接:

https://www.jianshu.com/p/38ce94bf0dbf

(1) sysbench 二进制包安装





```
Total                                                                                    224 kB/s | 664 kB  00:00:02
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : postgresql-libs-9.2.24-4.el7_8.x86_64                                                              1/2
  Installing : sysbench-1.0.20-1.el7.x86_64                                                                       2/2
  Verifying  : sysbench-1.0.20-1.el7.x86_64                                                                       1/2
  Verifying  : postgresql-libs-9.2.24-4.el7_8.x86_64                                                              2/2

Installed:
  sysbench.x86_64 0:1.0.20-1.el7

Dependency Installed:
  postgresql-libs.x86_64 0:9.2.24-4.el7_8

Complete!
```

验证 sysbench 已经下载成功

```
[root@houshengyuan /]# sysbench --version
sysbench 1.0.20
```

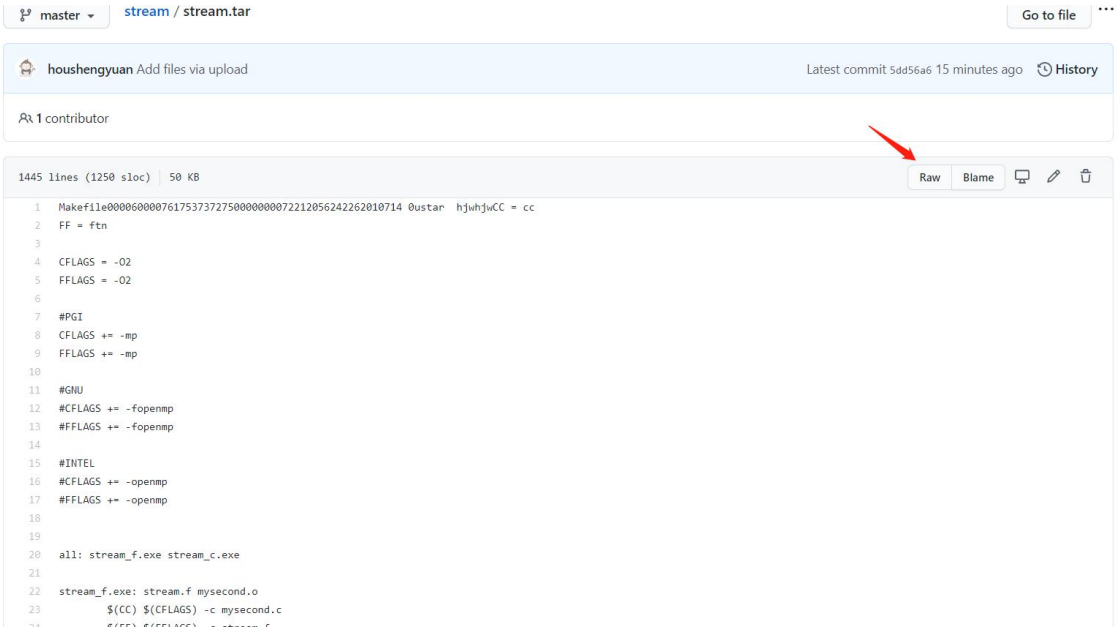
可以看到，指令执行效果与链接中所给效果相同，说明安装成功。

(2)streambench 下载

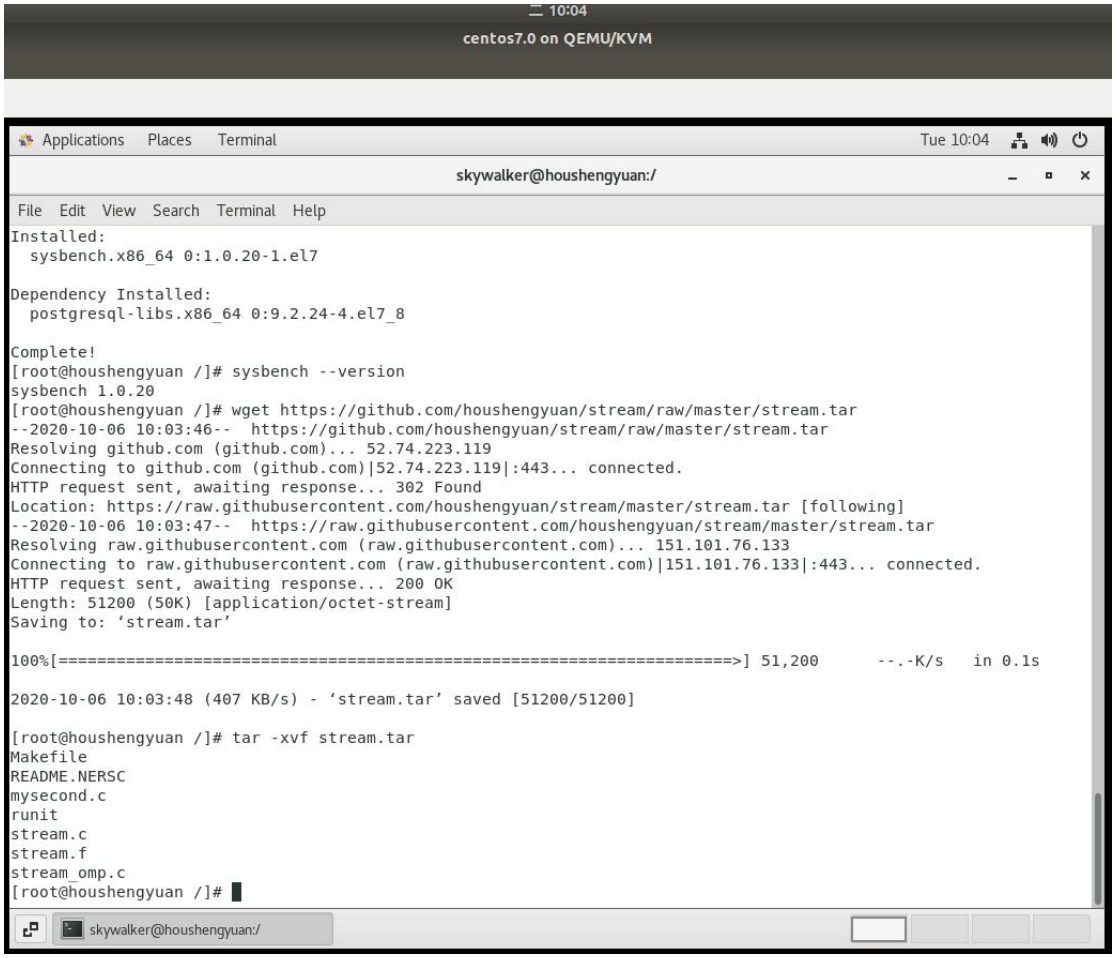
参考链接:

https://blog.csdn.net/liudong124521/article/details/101205119

注意，由于该网站所给的下载链接已经失效，所以我先从百度网盘中下载，而后将压缩包存入 github 中，通过 github 中的链接作为下载的网页链接(红色箭头指向部分右键点击复制链接地址即可)



下载和解压缩所用命令行如下



(3) 使用 streambench 进行性能测试

参考链接:https://blog.csdn.net/liudong124521/article/details/101205119

streambench 会分配一个大小为 100000000B 的数组进行各种操作，编译指令如下

gcc -O -mcmodel=medium -DSTREAM_ARRAY_SIZE=100000000 -mcmodel=large -DNTIME=20 stream.c -o stream.o

其中重要参数的含义如下:

-DSTREAM_ARRAY_SIZE 测试数组大小，默认是 10000000

-DNTIMES 测试时间，默认是 10

-OFFSET 调节数组的内存对齐，默认为 0，一般不用修改

-STREAM_TYPE 测试数字的数组类型

-openMP 多线程支持 添加-fopenmp 选项，icc 为-openmp，pgcc 为-mp，Open64 的 opencc 为-openmp

共有四种测试:

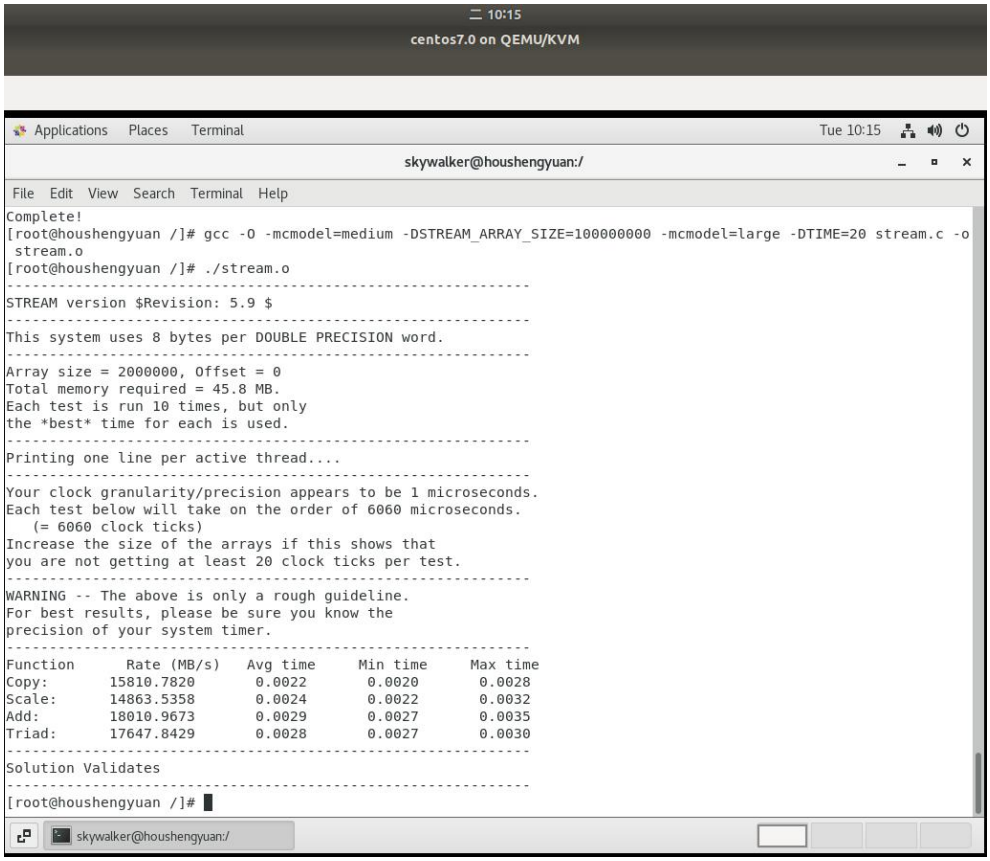
(1)Copy-数组的复制

(2)Scale-数组的尺度变换

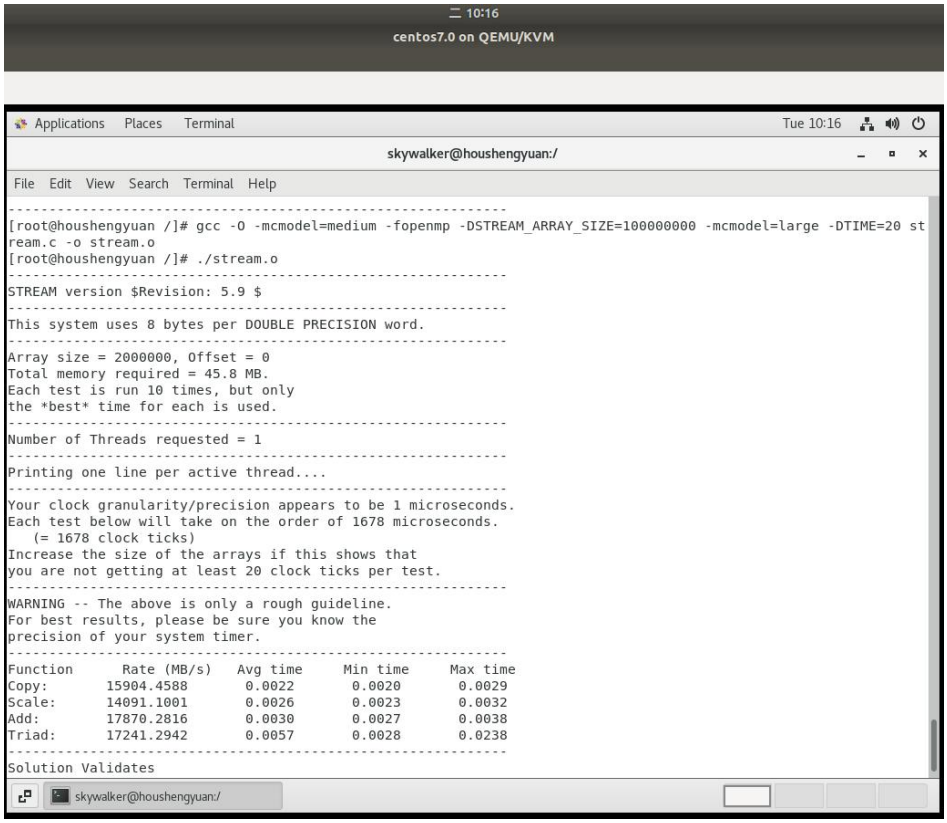
(3)Add-数组的矢量求和

(4)Triad-数组的复合矢量求和

使用单线程进行编译测试，结果如下



使用多线程进行编译测试，结果如下



由以上两个”validate”可知数组的复制(Copy)，数组的尺度变换(Scale)，数组的矢量求和(Add),

数组的复合矢量求和(Triad)三个测试均最终获得了成功。其中多线程测试的平均时间和最小时间整体上分别要低于单线程测试的平均时间和最小时间，这可能是由于多线程减少了重复类型操作的开销。而对于最大时间单线程和多线程在不同的测试程序上表现各有优劣，这可能是由于线程数较少带来的收益过少，维护多线程本身(创建，调用，回收.....)的一些操作开销超过了分摊同类型操作所带来的收益。多线程的内存访问速率普遍高于单线程。

(4)使用 sysbench 进行性能测试

参考链接:https://www.jianshu.com/p/38ce94bf0dbf

(1) CPU 性能测试

最大质数发生器测试(最大质数不超过 20000，线程数量为 20)

```
[root@houshengyuan /]# sysbench cpu --cpu-max-prime=20000 --threads=20 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 20
Initializing random number generator from current time


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:   341.14

General statistics:
    total time:                          10.0263s
    total number of events:              3421

Latency (ms):
         min:                                    2.42
         avg:                                   57.76
         max:                                  545.88
         95th percentile:                      227.40
         sum:                                197605.11

Threads fairness:
    events (avg/stddev):           171.0500/2.52
    execution time (avg/stddev):   9.8803/0.09

[root@houshengyuan /]#
```

打开 debug 选项，输出该测试线程的详细信息

```
二 10:21
centos7.0 on QEMU/KVM

Applications  Places  Terminal                                   Tue 10:21
skywalker@houshengyuan:/
File  Edit  View  Search  Terminal  Help
[root@houshengyuan /]# sysbench cpu --cpu-max-prime=20000 --threads=20 --debug=on --events=10000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 20
Debug mode enabled.

Initializing random number generator from current time


Doing CPU performance benchmark

Prime numbers limit: 20000

Initializing worker threads...

DEBUG: Worker thread (#11) started
DEBUG: Worker thread (#11) initialized
DEBUG: Worker thread (#13) started
DEBUG: Worker thread (#13) initialized
DEBUG: Worker thread (#14) started
DEBUG: Worker thread (#14) initialized
DEBUG: Worker thread (#16) started
DEBUG: Worker thread (#16) initialized
DEBUG: Worker thread (#19) started
DEBUG: Worker thread (#19) initialized
DEBUG: Worker thread (#12) started
DEBUG: Worker thread (#12) initialized
DEBUG: Worker thread (#8) started
DEBUG: Worker thread (#8) initialized
DEBUG: Worker thread (#4) started
DEBUG: Worker thread (#4) initialized
DEBUG: Worker thread (#5) started
DEBUG: Worker thread (#5) initialized
DEBUG: Worker thread (#0) started
skywalker@houshengyuan:/
```

二 10:22

centos7.0 on QEMU/KVM

Applications Places Terminal Tue 10:22

skywalker@houshengyuan:/

File Edit View Search Terminal Help

```
DEBUG: Worker thread (#0) initialized
DEBUG: Worker thread (#1) started
DEBUG: Worker thread (#1) initialized
DEBUG: Worker thread (#2) started
DEBUG: Worker thread (#2) initialized
DEBUG: Worker thread (#6) started
DEBUG: Worker thread (#6) initialized
DEBUG: Worker thread (#7) started
DEBUG: Worker thread (#7) initialized
DEBUG: Worker thread (#10) started
DEBUG: Worker thread (#10) initialized
DEBUG: Worker thread (#15) started
DEBUG: Worker thread (#15) initialized
DEBUG: Worker thread (#17) started
DEBUG: Worker thread (#17) initialized
DEBUG: Worker thread (#3) started
DEBUG: Worker thread (#3) initialized
DEBUG: Worker thread (#9) started
DEBUG: Worker thread (#9) initialized
DEBUG: Worker thread (#18) started
DEBUG: Worker thread (#18) initialized
Threads started!

Time limit exceeded, exiting...
(last message repeated 19 times)
Done.

CPU speed:
    events per second:   343.63

General statistics:
    total time:                          10.0498s
    total number of events:              3454

Latency (ms):
```

skywalker@houshengyuan:/

二 10:22

centos7.0 on QEMU/KVM

Applications Places Terminal Tue 10:22

skywalker@houshengyuan:/

File Edit View Search Terminal Help

```
Latency (ms):
         min:                                    2.42
         avg:                                   57.38
         max:                                  542.61
         95th percentile:                      231.53
         sum:                                198203.08

Threads fairness:
    events (avg/stddev):           172.7000/1.98
    execution time (avg/stddev):   9.9102/0.07

DEBUG: Verbose per-thread statistics:

DEBUG:     thread #  0: min: 0.0025s  avg: 0.0578s  max: 0.2765s  events: 171
DEBUG:                  total time taken by event execution: 9.8762s
DEBUG:     thread #  1: min: 0.0024s  avg: 0.0569s  max: 0.2428s  events: 174
DEBUG:                  total time taken by event execution: 9.8923s
DEBUG:     thread #  2: min: 0.0025s  avg: 0.0588s  max: 0.2550s  events: 168
DEBUG:                  total time taken by event execution: 9.8707s
DEBUG:     thread #  3: min: 0.0025s  avg: 0.0555s  max: 0.2476s  events: 177
DEBUG:                  total time taken by event execution: 9.8319s
DEBUG:     thread #  4: min: 0.0025s  avg: 0.0574s  max: 0.3204s  events: 173
DEBUG:                  total time taken by event execution: 9.9237s
DEBUG:     thread #  5: min: 0.0025s  avg: 0.0571s  max: 0.2421s  events: 174
DEBUG:                  total time taken by event execution: 9.9311s
DEBUG:     thread #  6: min: 0.0025s  avg: 0.0574s  max: 0.2649s  events: 172
DEBUG:                  total time taken by event execution: 9.8667s
DEBUG:     thread #  7: min: 0.0025s  avg: 0.0572s  max: 0.2505s  events: 172
DEBUG:                  total time taken by event execution: 9.8347s
DEBUG:     thread #  8: min: 0.0025s  avg: 0.0581s  max: 0.2641s  events: 171
DEBUG:                  total time taken by event execution: 9.9342s
DEBUG:     thread #  9: min: 0.0025s  avg: 0.0560s  max: 0.2610s  events: 175
DEBUG:                  total time taken by event execution: 9.7986s
DEBUG:     thread # 10: min: 0.0025s  avg: 0.0571s  max: 0.2489s  events: 172
DEBUG:                  total time taken by event execution: 9.8240s
```

skywalker@houshengyuan:/

```
DEBUG:     thread # 11: min: 0.0025s  avg: 0.0589s  max: 0.2669s  events: 170
DEBUG:                  total time taken by event execution: 10.0061s
DEBUG:     thread # 12: min: 0.0024s  avg: 0.0580s  max: 0.3787s  events: 172
DEBUG:                  total time taken by event execution: 9.9764s
DEBUG:     thread # 13: min: 0.0025s  avg: 0.0576s  max: 0.4628s  events: 174
DEBUG:                  total time taken by event execution: 10.0227s
DEBUG:     thread # 14: min: 0.0025s  avg: 0.0580s  max: 0.3560s  events: 172
DEBUG:                  total time taken by event execution: 9.9702s
DEBUG:     thread # 15: min: 0.0025s  avg: 0.0565s  max: 0.3231s  events: 174
DEBUG:                  total time taken by event execution: 9.8363s
DEBUG:     thread # 16: min: 0.0025s  avg: 0.0578s  max: 0.3128s  events: 173
DEBUG:                  total time taken by event execution: 9.9991s
DEBUG:     thread # 17: min: 0.0025s  avg: 0.0561s  max: 0.2892s  events: 175
DEBUG:                  total time taken by event execution: 9.8177s
DEBUG:     thread # 18: min: 0.0025s  avg: 0.0585s  max: 0.2559s  events: 171
DEBUG:                  total time taken by event execution: 10.0008s
DEBUG:     thread # 19: min: 0.0025s  avg: 0.0574s  max: 0.5426s  events: 174
DEBUG:                  total time taken by event execution: 9.9897s

[root@houshengyuan /]#
```

(2)内存性能测试

向内存中传输 2G 的数据，每个块的大小为 8K。

```
二 10:26
centos7.0 on QEMU/KVM

Applications   Places   Terminal                                   Tue 10:26
skywalker@houshengyuan:/
File  Edit  View  Search  Terminal  Help
[root@houshengyuan /]# sysbench memory --memory-block-size=8K --memory-total-size=2G run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Running memory speed test with the following options:
  block size: 8KiB
  total size: 2048MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 262144 (1154348.13 per second)

2048.00 MiB transferred (9018.34 MiB/sec)


General statistics:
    total time:                          0.2238s
    total number of events:              262144

Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                   23.62
         95th percentile:                        0.00
         sum:                                  181.84

Threads fairness:
```

```
Threads fairness:
    events (avg/stddev):           262144.0000/0.00
    execution time (avg/stddev):   0.1818/0.00

[root@houshengyuan /]#
```

内存分配性能测试(设置 12 个线程，向内存中传输 100G 的数据，块的大小为 8K)

顺序分配性能测试如下

二 10:30
centos7.0 on QEMU/KVM

Applications Places Terminal Tue 10:30

skywalker@houshengyuan:/

File Edit View Search Terminal Help

```
[root@houshengyuan /]# sysbench --threads=12 --events=10000 --test=memory --memory-block-size=8K --memory-total-
size=100G --memory-access-mode=seq run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any opt
ions.
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 12
Initializing random number generator from current time


Running memory speed test with the following options:
  block size: 8KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 13107192 (1351323.16 per second)

102399.94 MiB transferred (10557.21 MiB/sec)


General statistics:
    total time:                          9.6976s
    total number of events:              13107192

Latency (ms):
         min:                                    0.00
         avg:                                    0.01
         max:                                  193.53
         95th percentile:                        0.00
```

```
Latency (ms):
         min:                                    0.00
         avg:                                    0.01
         max:                                  193.53
         95th percentile:                        0.00
         sum:                                92544.63

Threads fairness:
    events (avg/stddev):           1092266.0000/0.00
    execution time (avg/stddev):   7.7121/0.40

[root@houshengyuan /]#
```

随机分配性能测试如下

二 10:32
centos7.0 on QEMU/KVM

Applications Places Terminal Tue 10:32

skywalker@houshengyuan:/

File Edit View Search Terminal Help

```
[root@houshengyuan /]# sysbench --threads=12 --events=10000 --test=memory --memory-block-size=8K --memory-total-
size=100G --memory-access-mode=rnd run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any opt
ions.
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 12
Initializing random number generator from current time


Running memory speed test with the following options:
  block size: 8KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 2389649 (238893.18 per second)

18669.13 MiB transferred (1866.35 MiB/sec)


General statistics:
    total time:                          10.0013s
    total number of events:              2389649

Latency (ms):
         min:                                    0.00
         avg:                                    0.05
         max:                                  288.59
         95th percentile:                        0.00
```

skywalker@houshengyuan:/

```
Threads fairness:
    events (avg/stddev):           199137.4167/7496.97
    execution time (avg/stddev):   9.4564/0.18

[root@houshengyuan /]#
```

比较上述两种分配策略可以发现，顺序分配的延迟时间要少于随机分配，

(3)文件 IO 性能测试

```
[root@houshengyuan /]# sysbench fileio help
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

fileio options:
  --file-num=N                  number of files to create [128]
  --file-block-size=N           block size to use in all IO operations [16384]
  --file-total-size=SIZE        total size of files to create [2G]
  --file-test-mode=STRING       test mode {seqwr, seqrewr, seqrd, rndrd, rndwr, rndrw}
  --file-io-mode=STRING         file operations mode {sync,async,mmap} [sync]
  --file-async-backlog=N        number of asynchronous operatons to queue per thread [128]
  --file-extra-flags=[LIST,...] list of additional flags to use to open files {sync,dsync,direct} []
  --file-fsync-freq=N           do fsync() after this number of requests (0 - don't use fsync()) [100]
  --file-fsync-all[=on|off]     do fsync() after each write operation [off]
  --file-fsync-end[=on|off]     do fsync() at the end of test [on]
  --file-fsync-mode=STRING      which method to use for synchronization {fsync, fdatasync} [fsync]
  --file-merged-requests=N      merge at most this number of IO requests if possible (0 - don't merge) [0]
  --file-rw-ratio=N             reads/writes ratio for combined test [1.5]

[root@houshengyuan /]#
```

1.准备阶段:首先生成需要测试的文件，生成的小文件都存放在当前目录下

二 10:37

centos7.0 on QEMU/KVM

Applications Places Terminal Tue 10:37

skywalker@houshengyuan:/

File Edit View Search Terminal Help

```
[root@houshengyuan /]# sysbench fileio --threads=2 --file-total-size=1G --file-test-mode=rndrw prepare
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

128 files, 8192Kb each, 1024Mb total
Creating files for the test...
Extra file open flags: (none)
Creating file test_file.0
Creating file test_file.1
Creating file test_file.2
Creating file test_file.3
Creating file test_file.4
Creating file test_file.5
Creating file test_file.6
Creating file test_file.7
Creating file test_file.8
Creating file test_file.9
Creating file test_file.10
Creating file test_file.11
Creating file test_file.12
Creating file test_file.13
Creating file test_file.14
Creating file test_file.15
Creating file test_file.16
Creating file test_file.17
Creating file test_file.18
Creating file test_file.19
Creating file test_file.20
Creating file test_file.21
Creating file test_file.22
Creating file test_file.23
Creating file test_file.24
Creating file test_file.25
Creating file test_file.26
Creating file test_file.27
Creating file test_file.28
```

skywalker@houshengyuan:/

```
Creating file test_file.95
Creating file test_file.96
Creating file test_file.97
Creating file test_file.98
Creating file test_file.99
Creating file test_file.100
Creating file test_file.101
Creating file test_file.102
Creating file test_file.103
Creating file test_file.104
Creating file test_file.105
Creating file test_file.106
Creating file test_file.107
Creating file test_file.108
Creating file test_file.109
Creating file test_file.110
Creating file test_file.111
Creating file test_file.112
Creating file test_file.113
Creating file test_file.114
Creating file test_file.115
Creating file test_file.116
Creating file test_file.117
Creating file test_file.118
Creating file test_file.119
Creating file test_file.120
Creating file test_file.121
Creating file test_file.122
Creating file test_file.123
Creating file test_file.124
Creating file test_file.125
Creating file test_file.126
Creating file test_file.127
1073741824 bytes written in 6.20 seconds (165.06 MiB/sec).
[root@houshengyuan /]#
```

skywalker@houshengyuan:/

可以看到，生成文件操作共耗时 6.20s

2.运行阶段(测试随机读写大小总共 1G 的文件，线程数量为 2)

```
[root@houshengyuan /]# sysbench fileio --threads=2 --file-total-size=1G --file-test-mode=rndrw run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!


File operations:
    reads/s:                      1008.25
    writes/s:                     672.17
    fsyncs/s:                     2166.85

Throughput:
    read, MiB/s:                  15.75
    written, MiB/s:               10.50

General statistics:
    total time:                          10.0552s
    total number of events:              38436
```

skywalker@houshengyuan:/

```
Latency (ms):
         min:                                    0.00
         avg:                                    0.52
         max:                                   24.94
         95th percentile:                        1.52
         sum:                                19938.95

Threads fairness:
    events (avg/stddev):           19218.0000/47.00
    execution time (avg/stddev):   9.9695/0.00

[root@houshengyuan /]#
```

3.运行测试完毕后需要将临时文件清理回收

```
[root@houshengyuan /]# sysbench fileio --threads=2 --file-total-size=1G --file-test-mode=rndrw cleanup
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Removing test files...
[root@houshengyuan /]#
```

(4)互斥锁性能测试

线程数量为 2，互斥数组的总大小 4096，每个线程互斥锁的数量为 50000，互斥锁外部的空循环数量为 10000，模拟所有线程在同一时刻并发运行。

```
centos7.0 on QEMU/KVM
Applications  Places  Terminal                                   Tue 10:45
skywalker@houshengyuan:/
File  Edit  View  Search  Terminal  Help
[root@houshengyuan /]# sysbench mutex help
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

mutex options:
  --mutex-num=N   total size of mutex array [4096]
  --mutex-locks=N number of mutex locks to do per thread [50000]
  --mutex-loops=N number of empty loops to do outside mutex lock [10000]

[root@houshengyuan /]# sysbench mutex --threads=2 --mutex-num=4096 --mutex-locks=50000 --mutex-loops=10000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time


Initializing worker threads...

Threads started!


General statistics:
    total time:                          0.4073s
    total number of events:              2

Latency (ms):
         min:                                    381.95
         avg:                                    392.93
         max:                                    403.91
         95th percentile:                        404.61
         sum:                                    785.86

Threads fairness:
    events (avg/stddev):           1.0000/0.00
    execution time (avg/stddev):   0.3929/0.01
```

(5)POSIX 多线程测试

线程数量为 2，每个请求产生 100 个线程，每个线程拥有锁的数量为 4

```
二 10:47
centos7.0 on QEMU/KVM
Applications  Places  Terminal                                   Tue 10:47
skywalker@houshengyuan:/
File  Edit  View  Search  Terminal  Help
[root@houshengyuan /]# sysbench threads help
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

threads options:
  --thread-yields=N number of yields to do per request [1000]
  --thread-locks=N  number of locks per thread [8]

[root@houshengyuan /]# sysbench threads --threads=2 --thread-yields=100 --thread-locks=4 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time


Initializing worker threads...

Threads started!


General statistics:
    total time:                          10.0001s
    total number of events:              136642

Latency (ms):
         min:                                    0.04
         avg:                                    0.15
         max:                                   30.33
         95th percentile:                        0.23
         sum:                                19931.88

Threads fairness:
    events (avg/stddev):           68321.0000/40.00
    execution time (avg/stddev):   9.9659/0.00
```

三.QEMU 源代码修改调试

写一个线程调用 CPU 运行状态的监视器，输出线程调用 CPU 的相关信息和内核态用户态的切换信息。打开 QEMU 的源代码文件/qemu-5.1.0/accel/kvm/kvm-all.c，添加三行语句如下，输出线程调用 CPU 信息(保存在 CPUState 结构体中),以及 ring 0 和 ring3 模式的切换信息。

```
root@ubuntu: /qemu-5.1.0
File  Edit  View  Search  Terminal  Help
        return EXCP_HLT;
    }

    qemu_mutex_unlock_iothread();
    cpu_exec_start(cpu);
    printf("thread %d is calling the CPU(core %d,event fd %d)\n",cpu->thread
_id,cpu->nr_cores,cpu->kvm_fd);
    do {
        MemTxAttrs attrs;

        if (cpu->vcpu_dirty) {
            kvm_arch_put_registers(cpu, KVM_PUT_RUNTIME_STATE);
            cpu->vcpu_dirty = false;
        }

        kvm_arch_pre_run(cpu, run);
        if (atomic_read(&cpu->exit_request)) {
            DPRINTF("interrupt exit requested\n");
            /*
             * KVM requires us to reenter the kernel after IO exits to compl
ete
             * instruction emulation. This self-signal will ensure that we
             * leave ASAP again.
             */
            kvm_cpu_kick_self();
        }

        /* Read cpu->exit_request before KVM_RUN reads run->immediate_exit.
         * Matching barrier in kvm_eat_signals.
         */
        smp_rmb();
        printf("thread %d enter KVM (ring3 to ring0)\n",cpu->thread_id);
        run_ret = kvm_vcpu_ioctl(cpu, KVM_RUN, 0);
        printf("thread %d exit from KVM (ring 0 to ring 3)\n",cpu->thread_id
);
                                                          2465,11       76%
```

重新设置配置文件./configure，进行编译 make,最后重新 make install 安装

```
root@ubuntu:/qemu-5.1.0# ./configure --enable-kvm --target-list="x86_64-soft
mmu"
Install prefix    /usr/local
BIOS directory    /usr/local/share/qemu
firmware path     /usr/local/share/qemu-firmware
binary directory  /usr/local/bin
library directory /usr/local/lib
module directory  /usr/local/lib/qemu
libexec directory /usr/local/libexec
include directory /usr/local/include
config directory  /usr/local/etc
local state directory   /usr/local/var
Manual directory  /usr/local/share/man
ELF interp prefix /usr/gnemul/qemu-%M
Build directory   /qemu-5.1.0
Source path       /qemu-5.1.0
GIT binary        git
GIT submodules
C compiler        cc
Host C compiler   cc
C++ compiler      c++
Objective-C compiler cc
ARFLAGS           rv
CFLAGS            -O2 -U_FORTIFY_SOURCE -D_FORTIFY_SOURCE=2 -g
QEMU_CFLAGS       -I/usr/include/pixman-1   -pthread -I/usr/include/glib-2.0
 -I/usr/lib/x86_64-linux-gnu/glib-2.0/include -fPIE -DPIE -m64 -mcx16 -D_GNU
_SOURCE -D_FILE_OFFSET_BITS=64 -D_LARGEFILE_SOURCE -Wstrict-prototypes -Wred
undant-decls -Wall -Wundef -Wwrite-strings -Wmissing-prototypes -fno-strict-
aliasing -fno-common -fwrapv -std=gnu99  -Wold-style-declaration -Wold-style
-definition -Wtype-limits -Wformat-security -Wformat-y2k -Winit-self -Wignor
ed-qualifiers -Wempty-body -Wnested-externs -Wendif-labels -Wexpansion-to-de
fined -Wno-missing-include-dirs -Wno-shift-negative-value -Wno-psabi -fstack
-protector-strong -I/usr/include/libpng16 -I$(SRC_PATH)/capstone/include
```

```
root@ubuntu:/qemu-5.1.0# make -j4
  GEN     x86_64-softmmu/config-devices.mak.tmp
  GEN     config-host.h
make[1]: Entering directory '/qemu-5.1.0/slirp'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/qemu-5.1.0/slirp'
  GEN     trace/generated-tcg-tracers.h
  GEN     trace/generated-helpers.h
  GEN     trace/generated-helpers-wrappers.h
  GEN     x86_64-softmmu/config-devices.mak
  GEN     trace/generated-helpers.c
  GEN     module_block.h
  GEN     trace-root.h
  GEN     accel/kvm/trace.h
  GEN     accel/tcg/trace.h
  GEN     backends/trace.h
  GEN     backends/tpm/trace.h
  GEN     crypto/trace.h
  GEN     monitor/trace.h
  GEN     authz/trace.h
  GEN     block/trace.h
  GEN     io/trace.h
  GEN     nbd/trace.h
  GEN     scsi/trace.h
  GEN     audio/trace.h
  GEN     chardev/trace.h
  GEN     hw/9pfs/trace.h
  GEN     hw/acpi/trace.h
  GEN     hw/alpha/trace.h
  GEN     hw/arm/trace.h
  GEN     hw/audio/trace.h
  GEN     hw/block/trace.h
  GEN     hw/block/dataplane/trace.h
  GEN     hw/char/trace.h
  GEN     hw/dma/trace.h
  GEN     hw/hppa/trace.h
```

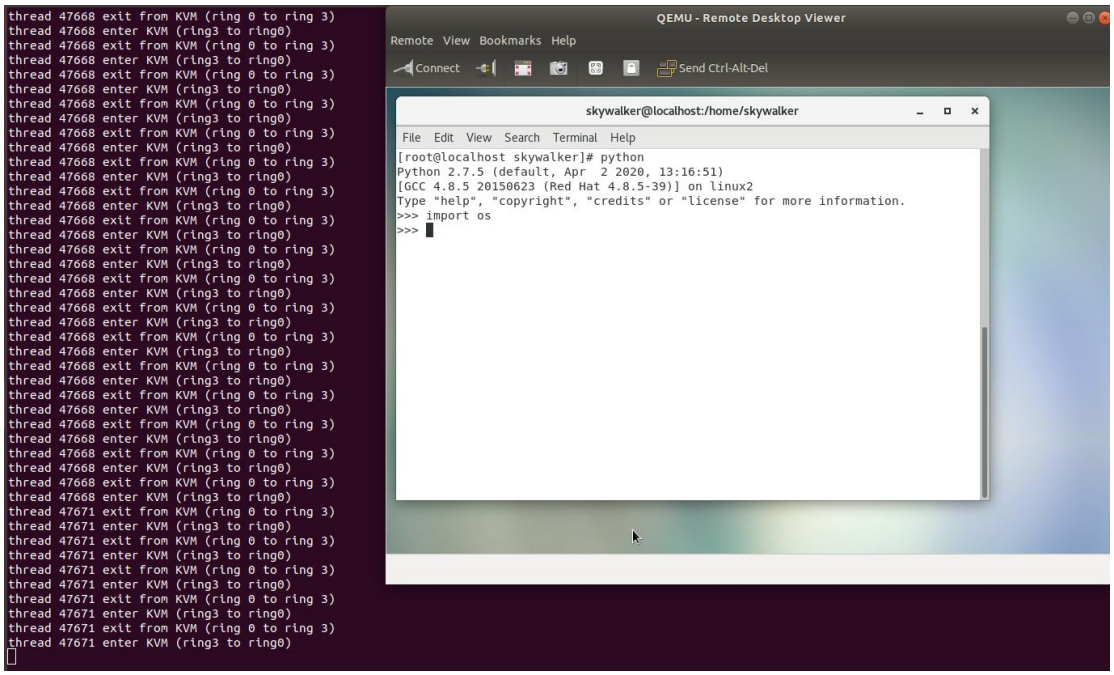
```
root@ubuntu:/qemu-5.1.0# make install
make[1]: Entering directory '/qemu-5.1.0/slirp'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/qemu-5.1.0/slirp'
install -d -m 0755 "/usr/local/share/qemu"
install -d -m 0755 "/usr/local/var"/run
install -d -m 0755 "/usr/local/include"
install -d -m 0755 "/usr/local/bin"
install -c -m 0755 qemu-ga ivshmem-client ivshmem-server qemu-nbd qemu-storage-daemon qemu-img qemu-io qemu-edid  "/usr/local/bin"
strip "/usr/local/bin/qemu-ga" "/usr/local/bin/ivshmem-client" "/usr/local/bin/ivshmem-server" "/usr/local/bin/qemu-nbd" "/usr/local/bin/qemu-storage-daemon" "/usr/local/bin/qemu-im
g" "/usr/local/bin/qemu-io" "/usr/local/bin/qemu-edid"
install -d -m 0755 "/usr/local/libexec"
install -c -m 0755 scsi/qemu-pr-helper qemu-bridge-helper "/usr/local/libexec"
strip "/usr/local/libexec/qemu-pr-helper" "/usr/local/libexec/qemu-bridge-helper"
set -e; for x in bios.bin bios-256k.bin bios-microvm.bin sgabios.bin vgabios.bin vgabios-cirrus.bin vgabios-stdvga.bin vgabios-vmware.bin vgabios-qxl.bin vgabios-virtio.bin vgabios-
ramfb.bin vgabios-bochs-display.bin vgabios-ati.bin openbios-sparc32 openbios-sparc64 openbios-ppc QEMU,tcx.bin QEMU,cgthree.bin pxe-e1000.rom pxe-eepro100.rom pxe-ne2k_pci.rom pxe-
pcnet.rom pxe-rtl8139.rom pxe-virtio.rom efi-e1000.rom efi-eepro100.rom efi-ne2k_pci.rom efi-pcnet.rom efi-rtl8139.rom efi-virtio.rom efi-e1000e.rom efi-vmxnet3.rom qemu-nsis.bmp ba
mboo.dtb canyonlands.dtb petalogix-s3adsp1800.dtb petalogix-ml605.dtb multiboot.bin linuxboot.bin linuxboot_dma.bin kvmvapic.bin pvh.bin s390-ccw.img s390-netboot.img slof.bin skibo
ot.lid palcode-clipper u-boot.e500 u-boot-sam460-20100605.bin qemu_vga.ndrv edk2-licenses.txt hppa-firmware.img opensbi-riscv32-sifive_u-fw_jump.bin opensbi-riscv32-virt-fw_jump.bin
 opensbi-riscv64-sifive_u-fw_jump.bin opensbi-riscv64-virt-fw_jump.bin; do \
        install -c -m 0644 /qemu-5.1.0/pc-bios/$x "/usr/local/share/qemu"; \
done
set -e; for x in pc-bios/edk2-i386-secure-code.fd pc-bios/edk2-i386-code.fd pc-bios/edk2-arm-vars.fd pc-bios/edk2-x86_64-code.fd pc-bios/edk2-arm-code.fd pc-bios/edk2-aarch64-code.f
d pc-bios/edk2-i386-vars.fd pc-bios/edk2-x86_64-secure-code.fd; do \
        install -c -m 0644 $x "/usr/local/share/qemu"; \
done
install -d -m 0755 "/usr/local/share/qemu/firmware"
set -e; tmpf=$(mktemp); trap 'rm -f -- "$tmpf"' EXIT; \
for x in 50-edk2-i386-secure.json 50-edk2-x86_64-secure.json 60-edk2-aarch64.json 60-edk2-arm.json 60-edk2-i386.json 60-edk2-x86_64.json; do \
        sed -e 's,@DATADIR@,/usr/local/share/qemu,' \
                "/qemu-5.1.0/pc-bios/descriptors/$x" > "$tmpf"; \
        install -c -m 0644 "$tmpf" \
                "/usr/local/share/qemu/firmware/$x"; \
done
for s in 16x16 24x24 32x32 48x48 64x64 128x128 256x256 512x512; do \
        mkdir -p "/usr/local/share/icons/hicolor/${s}/apps"; \
        install -c -m 0644 /qemu-5.1.0/ui/icons/qemu_${s}.png \
                "/usr/local/share/icons/hicolor/${s}/apps/qemu.png"; \
done; \
mkdir -p "/usr/local/share/icons/hicolor/32x32/apps"; \
install -c -m 0644 /qemu-5.1.0/ui/icons/qemu_32x32.bmp \
        "/usr/local/share/icons/hicolor/32x32/apps/qemu.bmp"; \
mkdir -p "/usr/local/share/icons/hicolor/scalable/apps"; \
install -c -m 0644 /qemu-5.1.0/ui/icons/qemu.svg \
        "/usr/local/share/icons/hicolor/scalable/apps/qemu.svg"
```

安装完成后使用命令行(qemu-system-x86_64)创建并开启虚拟机，得到输出信息如下。

```
root@ubuntu: /qemu-5.1.0
File  Edit  View  Search  Terminal  Help
root@ubuntu:/qemu-5.1.0# qemu-system-x86_64 -m 3500 -smp 4 --enable-kvm  -bo
ot order=cd -hda /var/lib/libvirt/images/centos7.img -cdrom /var/lib/libvirt
/images/CentOS-7-x86_64-DVD-2003.iso
VNC server running on 127.0.0.1:5900
thread 44049 is calling the CPU(core 1,event fd 14)
thread 44049 enter KVM (ring3 to ring0)
thread 44049 exit from KVM (ring 0 to ring 3)
thread 44049 is calling the CPU(core 1,event fd 14)
thread 44049 enter KVM (ring3 to ring0)
thread 44051 is calling the CPU(core 1,event fd 16)
thread 44051 enter KVM (ring3 to ring0)
thread 44051 exit from KVM (ring 0 to ring 3)
thread 44051 is calling the CPU(core 1,event fd 16)
thread 44051 enter KVM (ring3 to ring0)
thread 44048 is calling the CPU(core 1,event fd 13)
thread 44048 enter KVM (ring3 to ring0)
thread 44048 exit from KVM (ring 0 to ring 3)
thread 44048 is calling the CPU(core 1,event fd 13)
thread 44048 enter KVM (ring3 to ring0)
thread 44048 exit from KVM (ring 0 to ring 3)
thread 44048 enter KVM (ring3 to ring0)
thread 44048 exit from KVM (ring 0 to ring 3)
thread 44048 enter KVM (ring3 to ring0)
thread 44048 exit from KVM (ring 0 to ring 3)
thread 44048 enter KVM (ring3 to ring0)
thread 44050 is calling the CPU(core 1,event fd 15)
thread 44050 enter KVM (ring3 to ring0)
thread 44050 exit from KVM (ring 0 to ring 3)
thread 44050 is calling the CPU(core 1,event fd 15)
thread 44050 enter KVM (ring3 to ring0)
thread 44048 exit from KVM (ring 0 to ring 3)
thread 44048 enter KVM (ring3 to ring0)
thread 44048 exit from KVM (ring 0 to ring 3)
thread 44048 enter KVM (ring3 to ring0)
thread 44048 exit from KVM (ring 0 to ring 3)
thread 44048 enter KVM (ring3 to ring0)
```

可以看到虚拟机初始化时共有四个初始化线程(线程号 44049,44051,44048,44050)调用 CPU 运行，最终线程 44048 进入稳定的运行状态，调用 CPU 执行特权指令，不断地在 ring 0 和 ring 3 模式之间切换(用户态和内核态的不断切换)。

开启图形界面 VNC 后运行中间状态输出如下所示(由于每次开启虚拟机分配的线程号不同，因此此处分两次截屏，得到的线程号与前一张图片不太相同)。可以看出虚拟机运行过程中线程 47668 与 47671 交替调用 CPU。



四.实验问题及感想

(1)实验过程中得到以下几点不错的经验:

1.make 编译 qemu 源代码的过程中使用命令行选项 --target-list=”x86_64-softmmu”这样可以大大加快编译速度，因为此选项指定仅编译 x86 平台内容。

2.使用 VMware 的快照技术可以大大减少试错时间，提高实验效率。

3.环境安装最好不要嵌套进行，此次实验我一开始在安装了 Xen 的 Ubuntu 下安装 QEMU 后来发现 Ubuntu 此时已不支持虚拟化，浪费了很长的时间。

4.和同学多多交流，会进步很快。

(2)感想:本门课程真的刷新了我对虚拟机认知的下限，感觉整个过程里有 90%的时间都在进行虚拟机的调整操作，感谢这门课让我熟练掌握并复习了大量的虚拟机基本命令行操作和配置操作，这也是我在此门课程中的最大收获，不过当然也对 QEMU 虚拟化技术有了一定的初步了解。非常非常感谢助教和老师的付出，为我解答了大量的问题！！！！！