

Q0: 基础实验环境说明:

(1) 操作系统:CentOS6.5 VMWare虚拟机环境

```
→ examUse tree ./
./
├── elephaht_examhos
│   ├── Enmoedu_Hadoop_CentOS_6.5_CDH_5.6.vmdk
│   └── Enmoedu_Hadoop_CentOS_6.5_CDH_5.6.vmx
└── monkey_exam
    ├── Enmoedu_Hadoop_CentOS_6.5_CDH_5.6.vmdk
    └── Enmoedu_Hadoop_CentOS_6.5_CDH_5.6.vmx
```

(2) 局域网环境，网卡设置为桥接模式，主从两节点(elephant/monkey)

```
[enmoedu@bogon ~]$ hostname
elephant
[enmoedu@bogon ~]$ ifconfig | grep 10
    inet addr:10.8.2.157  Bcast:10.8.2.255  Mask:255.255.255.0
    collisions:0 txqueuelen:1000
    RX packets:10 errors:0 dropped:0 overruns:0 frame:0
    TX packets:10 errors:0 dropped:0 overruns:0 carrier:0

[enmoedu@bogon ~]$ hostname
monkey
[enmoedu@bogon ~]$ ifconfig | grep 10
    inet addr:10.8.2.168  Bcast:10.8.2.255  Mask:255.255.255.0
    collisions:0 txqueuelen:1000
```

(3)设置关闭防火墙和SELinux

```
# change to root user
[root@elephant enmoedu]# service iptables stop
iptables: Setting chains to policy ACCEPT: filter      [ OK ]
iptables: Flushing firewall rules:                    [ OK ]
iptables: Unloading modules:                           [ OK ]

# edit selinux configure file
[root@elephant enmoedu]# vim /etc/selinux/config

# change `enforcing` to `disabled`
SELINUX=disabled

# shutdown selinux forever
[root@monkey enmoedu]# setenforce 0

# deny iptables restart when system reboot
[root@monkey enmoedu]# chkconfig iptables off

# check iptables setting
[root@monkey enmoedu]# chkconfig --list | grep iptables
iptables          0:关闭    1:关闭    2:关闭    3:关闭    4:关闭    5:关闭    6:关闭
[root@monkey enmoedu]#
```

Q1: 列举能够查看Linux操作系统 swap 使用情况的命令（3个），截图上传

(1) htop/top

- top 工具中 Swap 信息示意:

```
Swap: 2097144k total, 0k used, 2097144k free, 282936k cached
```

- htop 中有图示的 Swap 消耗信息:

```
1 [|||||87.2%] 5 [|||||11.4%]
2 [||1.4%] 6 [||1.0%]
3 [||1.4%] 7 [||0.9%]
4 [||1.9%] 8 [||1.0%]
Mem[|||||24816/31898MB] Tasks: 114, 437 thr; 3 running
Swp[|||||2814/32447MB] Load average: 0.98 1.37 1.61
Uptime: 88 days, 22:26:26

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
449 root 20 0 267M 0 0 S 0.0 0.0 0:00.00 /usr/sbin/lvmtd -f
718 root 20 0 25960 0 0 S 0.0 0.0 0:00.07 /usr/sbin/atd -f
729 root 20 0 53060 0 0 S 0.0 0.0 0:00.00 /usr/sbin/wpa_supplicant -u -f /var/lo
918 root 20 0 186M 0 0 S 0.0 0.0 0:00.00 /usr/sbin/cupsd -f
1315 root 20 0 106M 0 0 S 0.0 0.0 0:00.00 nginx: master process /usr/sbin/nginx
1321 nginx 20 0 109M 0 0 S 0.0 0.0 4:56.74 nginx: worker process
3711 hongfei 20 0 126M 0 0 S 0.0 0.0 0:00.00 SCREEN -R jupyter
4203 master 20 0 127M 0 0 S 0.0 0.0 0:00.03 SCREEN -R neallee
4204 master 20 0 139M 0 0 S 0.0 0.0 0:00.07 /bin/zsh
5100 master 20 0 140M 0 0 S 0.0 0.0 0:00.05 -zsh
6635 master 20 0 4474M 0 0 S 0.0 0.0 0:01.86 java -Xms1024m -Xmx1024m -XX:ReservedC
6636 master 20 0 4474M 0 0 S 0.0 0.0 0:00.02 java -Xms1024m -Xmx1024m -XX:ReservedC

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

X enmoedu@elephant:~
top - 02:25:39 up 20 min, 3 users, load average: 0.06, 0.02, 0.00
Tasks: 155 total, 1 running, 154 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 0.3%sy, 0.0%ni, 99.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1554840k total, 597228k used, 937612k free, 27384k buffers
Swap: 2097144k total, 0k used, 2097144k free, 280680k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1411 root 20 0 199m 7248 4316 S 0.3 0.5 0:00.45 ManagementAgent
2357 enmoedu 20 0 291m 17m 14m S 0.3 1.2 0:01.23 vmttoolsd
2745 enmoedu 20 0 15036 1268 944 R 0.3 0.1 0:00.09 top
1 root 20 0 19364 1544 1228 S 0.0 0.1 0:01.30 init
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
3 root RT 0 0 0 0 S 0.0 0.0 0:00.01 migration/0
4 root 20 0 0 0 0 S 0.0 0.0 0:00.02 ksoftirqd/0
5 root RT 0 0 0 0 S 0.0 0.0 0:00.00 migration/0
6 root RT 0 0 0 0 S 0.0 0.0 0:00.00 watchdog/0
7 root RT 0 0 0 0 S 0.0 0.0 0:00.30 migration/1
8 root RT 0 0 0 0 S 0.0 0.0 0:00.00 migration/1
9 root 20 0 0 0 0 S 0.0 0.0 0:00.01 ksoftirqd/1
10 root RT 0 0 0 0 S 0.0 0.0 0:00.00 watchdog/1
11 root 20 0 0 0 0 S 0.0 0.0 0:00.79 events/0
12 root 20 0 0 0 0 S 0.0 0.0 0:00.09 events/1
13 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cgroup
```

(2) free -m

Swap 行信息中 total 表示swap区总量, used 代表消耗量, free 代表剩余量:

```
[enmoedu@elephant ~]$ free -m
```

	total	used	free	shared	buffers	cached
Mem:	1498	583	915	0	26	274
-/+ buffers/cache:		282	1216			
Swap:	2047	0	2047			

```
[enmoedu@elephant ~]$ free -m
```

	total	used	free	shared	buffers	cached
Mem:	1498	583	915	0	26	274
-/+ buffers/cache:		282	1216			
Swap:	2047	0	2047			

```
[enmoedu@elephant ~]$
```

(3) cat /proc/meminfo

在 `/proc/meminfo` 文件中也提供了关于 `Swap` 的三个指标：

```
[enmoedu@elephant ~]$ cat /proc/meminfo | grep Swap
```

SwapCached:	0 kB
SwapTotal:	2097144 kB
SwapFree:	2097144 kB

```
[enmoedu@elephant ~]$ cat /proc/meminfo | grep Swap
```

SwapCached:	0 kB
SwapTotal:	2097144 kB
SwapFree:	2097144 kB

```
[enmoedu@elephant ~]$
```

Q2. 解释vm.swappiness 内核参数的作用 / 取值范围 / 默认值，永久修改该值为0

1. `vm.swappiness` 内核参数的作用是控制内核从物理内存中移到交换空间的频度。值越大，将引起越多内存页发生交换，充分利用 `Swap` 空间；值越小，就有越多的进程驻留在内存中，尽快保证 `Swap` 空间是空闲的。
2. `vm.swappiness` 内核参数取值范围是 `0 ~ 100`
3. `vm.swappiness` 内核参数默认值 `60`

永久设置 `vm.swappiness=0` 需要修改系统配置 `/etc/sysctl.conf`

```
[root@slave04 master]# vim /etc/sysctl.conf
```

```
# add or modiy `vm.swappiness` equal 0
```

```
vm.swappiness=0
```

```
[root@elephant enmoedu]# cat /etc/sysctl.conf | grep vm
vm.swappiness = 0
```

Q3. Linux 操作系统虚拟内存的作用，如果没有MMU，32bit操作系统支持的最大内存数量为多少？

1. 虚拟内存用于扩展物理内存,将物理内存地址转为虚拟内存地址并加以管理。
2. 如果没有MMU,32bit操作系统支持的最大内存数量只能有 $2^{32} = 4\text{GB}$ 。

Q4. 如何查看Linux 操作系统查看某个进程内存空间使用情况 (2种方法)

- `/proc/[pid]/status` 进入/proc/然后找到相应的PID看status信息(`VMPeak/VMSize..`)

```
[root@elephant enmoedu]# ps aux | grep bash
enmoedu    2445    0.0   0.1 108340   1788 pts/0    Ss+  18:05   0:00 bash
enmoedu    7139    0.0   0.1 108336   1768 pts/1    Ss   23:12   0:00 -bash
enmoedu    7156    0.0   0.1 108340   1780 pts/1    S    23:12   0:00 bash
root       7178    0.0   0.1 108340   1804 pts/1    S    23:12   0:00 bash
root       7223    0.0   0.0 103260    848 pts/1    S+   23:19   0:00 grep bash

[root@elephant enmoedu]#
[root@elephant enmoedu]#
[root@elephant enmoedu]# cat /proc/7178/status | grep Vm
VmPeak:    108340 kB
VmSize:    108340 kB
VmLck:      0 kB
VmHWM:     1804 kB
VmRSS:     1804 kB
VmData:    336 kB
VmStk:      88 kB
VmExe:     848 kB
VmLib:    1880 kB
VmPTE:      76 kB
VmSwap:      0 kB
```

- `pmap` 查看详尽的内存占用情况

```
[root@elephant enmoedu]# pmap 7178
7178:  bash
0000000000400000      848K r-x--  /bin/bash
000000000006d3000      40K rw---  /bin/bash
000000000006dd000      20K rw---  [ anon ]
000000000008dc000      36K rw---  /bin/bash
00000000001302000     264K rw---  [ anon ]
0000000333520000     128K r-x--  /lib64/ld-2.12.so
0000000333541f000       4K r----  /lib64/ld-2.12.so
00000003335420000       4K rw---  /lib64/ld-2.12.so
00000003335421000       4K rw---  [ anon ]
0000000333560000       8K r-x--  /lib64/libdl-2.12.so
00000003335602000    2048K -----  /lib64/libdl-2.12.so
00000003335802000       4K r----  /lib64/libdl-2.12.so
00000003335803000       4K rw---  /lib64/libdl-2.12.so
00000003335a0000    1580K r-x--  /lib64/libc-2.12.so
00000003335b8b000    2044K -----  /lib64/libc-2.12.so
00000003335d8a000      16K r----  /lib64/libc-2.12.so
00000003335d8e000       4K rw---  /lib64/libc-2.12.so
00000003335d8f000      20K rw---  [ anon ]
0000000333fe0000     116K r-x--  /lib64/libtinfo.so.5.7
0000000333fe1d000    2048K -----  /lib64/libtinfo.so.5.7
0000000334001d000      16K rw---  /lib64/libtinfo.so.5.7
00007eff04248000      48K r-x--  /lib64/libnss_files-2.12.so
00007eff04254000    2048K -----  /lib64/libnss_files-2.12.so
00007eff04454000       4K r----  /lib64/libnss_files-2.12.so
00007eff04455000       4K rw---  /lib64/libnss_files-2.12.so
00007eff04456000   96836K r----  /usr/lib/locale/locale-archive
00007eff0a2e7000      12K rw---  [ anon ]
00007eff0a2ee000       8K rw---  [ anon ]
00007eff0a2f0000      28K r--s-  /usr/lib64/gconv/gconv-modules.cache
00007eff0a2f7000       4K rw---  [ anon ]
00007fffe6f3c000     84K rw---  [ stack ]
00007fffe6fff000       4K r-x--  [ anon ]
fffffffffff60000       4K r-x--  [ anon ]
total                108340K
```

Q5. 学习Linux内存结构、Java JVM内存结构对大数据的帮助？

万变不离其宗，JVM构建于Linux系统内存之上，整个大数据技术栈其实也是Java技术栈，是基于JVM运作的。

了解Linux内存结构，就可以从系统层开始优化内存。了解JVM的内存结构，可以根据生产环境的实际情况对大数据程序进行内存优化。

Q6. 解释JVM Heap 内存区域？说明minor GC，Full GC的触发条件？

XXX JVM的HEAP区域是整个java程序的堆内存区域，类被创建后在heap区内存活待命。整个HEAP区域被拆分成多块结构，有新到老，最终被GC。当程序数据信息填满 eden 区域，就会触发minorGC，将所有信息挪到old区域，也称Tenured区。当Tenured被填满的时候，就触发了FullGC。

(1) Heap（堆）是JVM的内存数据区。当对象的实例被创建后，Heap将分配相应的空间。

要解释清楚 GC 过程，首先需要明白JVM的 Heap 分为三个部分：- Young: 存放新生的对象实例，内部又分为Eden区，Survivor区(对应semi-Spaces：和Survivor做Copying collection) - Tenured: 在新生代多次回收没有被清除的对象实例 - Perm: 存放类库级对象实例

(2) Minor GC 触发条件：当从新生代发生内存回收。

(3) FullGC 触发条件：清理整个堆空间—包括年轻代和永久代。

Q7. 创建文件testfile 对test组可读，新建用户test1和test2属于test组，请配置testfile权限对test1用户可读，test2用户不可读。

```
touch testfile
chgrp test testfile

useradd test1 -g test
useradd test2 -g test

setfacl -m user:test1:r-- testfile
```

Q8. 永久关闭操作系统SELinux，以及防火墙配置

```
# change to root user
[root@elephant enmoedu]# service iptables stop
iptables: Setting chains to policy ACCEPT: filter      [ OK ]
iptables: Flushing firewall rules:                    [ OK ]
iptables: Unloading modules:                           [ OK ]

# edit selinux configure file
[root@elephant enmoedu]# vim /etc/selinux/config

# change `enforcing` to `disabled`
SELINUX=disabled

# shutdown selinux forever
[root@monkey enmoedu]# setenforce 0

# deny iptables restart when system reboot
[root@monkey enmoedu]# chkconfig iptables off

# check iptables setting
[root@monkey enmoedu]# chkconfig --list | grep iptables
iptables          0:关闭    1:关闭    2:关闭    3:关闭    4:关闭    5:关闭    6:关闭
```