

# 课程大作业4

519021910366 窦铨明

## 1 在主机上准备hugepage

### 1.1 开启hugepage

使用如下代码打开主机hugepage：

```
1 sudo gedit /etc/libvirt/qemu.conf
2 # 取消hugetlbfs_mount = "/dev/hugepages"注释
3 sudo gedit /etc/default/grub
4 # 添加一行：GRUB_CMDLINE_LINUX="transparent_hugepage=never
   default_hugepagesz=1G hugepagesz=1G hugepages=5"
5 sudo update-grub
6 sudo reboot
```

### 1.2 查看hugepage配置

重启完成后，输入如下代码查看hugepage的配置：

```
1 cat /proc/meminfo | grep -i hugepage
```

输出如下：

```
dodo@ubuntu:/proc$ cat /proc/meminfo | grep -i hugepage
AnonHugePages:      0 kB
ShmemHugePages:     0 kB
FileHugePages:      0 kB
HugePages_Total:    5
HugePages_Free:      5
HugePages_Rsvd:      0
HugePages_Surp:      0
Hugepagesize:       1048576 kB
```

可以看出主机上开启了5个大小为1GB的hugepage

## 2 虚拟机安装

### 2.1 创建qcow2文件

使用如下代码创建一个新qcow2文件：

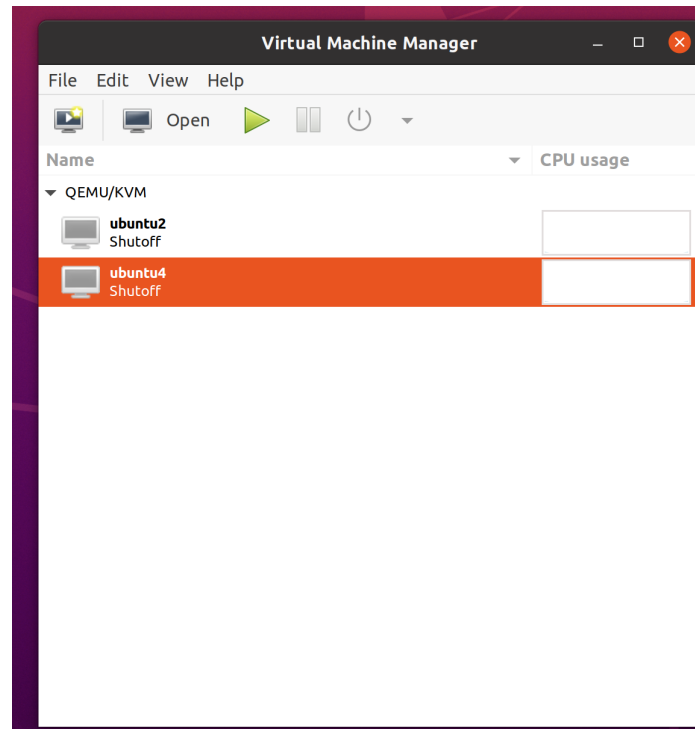
```
1 qemu-img create -f qcow2 ubuntu.img 30G
```

### 2.2 创建并启动虚拟机

使用三种方法启动虚拟机

## 2.2.1 使用virt manager启动

使用virt-manager创建并启动虚拟机：



## 2.2.2 使用virsh启动

首先使用virt-manager配置系统，然后在其系统概况中打开xml文件，复制进本地的xml文件

然后输入以下命令创建并运行虚拟机：

```
1 # create
2 sudo virsh create ./ubuntu.xml
3 # start
4 virsh start --domain ubuntu
5 # undefine
6 virsh undefine --domain ubuntu
```

## 2.2.3 使用C语言程序启动

如1.2.2中介绍，首先使用virt-manager配置好系统并复制xml文件至本地后，编译并运行以下程序：

```
1 #include <libvirt.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4 char *fileToChar(const char *fileName)
5 {
6     FILE *fp = fopen(fileName, "r");
7     if (!fp)
8     {
9         fprintf(stderr, "Error in open file %s\n", fileName);
10        return NULL;
11    }
12    fseek(fp, 0, SEEK_END);
13    long fileSize = ftell(fp);
14    char *words = (char *)malloc(fileSize + 1);
15    fseek(fp, 0, SEEK_SET);
```

```

16     fread(words, fileSize, sizeof(char), fp);
17     fclose(fp);
18     return words;
19 }
20 virDomainPtr defineDomainFromXML(virConnectPtr conn, const char
    *xmlFileName)
21 {
22     char *xml = fileToChar(xmlFileName);
23     if (!xml)
24     {
25         return NULL;
26     }
27     const int flags[] = {
28         VIR_DOMAIN_NONE,
29         VIR_DOMAIN_START_PAUSED,
30         VIR_DOMAIN_START_AUTODESTROY,
31         VIR_DOMAIN_START_BYPASS_CACHE,
32         VIR_DOMAIN_START_FORCE_BOOT,
33         VIR_DOMAIN_START_VALIDATE};
34     virDomainPtr vm = virDomainCreateXML(conn, xml, flags[2]);
35     free(xml);
36     if (!vm)
37     {
38         fprintf(stderr, "Error in create machine\n");
39         return NULL;
40     }
41     return vm;
42 }
43 int main()
44 {
45     virConnectPtr conn = virConnectOpen("qemu:///system");
46     const char *XML_FILE = "ubuntu-1.xml";
47     virDomainPtr vm = defineDomainFromXML(conn, XML_FILE);
48     char c;
49     while (1)
50     {
51         scanf("%c", &c);
52         if (c == 'c')
53             break;
54     }
55     virDomainFree(vm);
56     return 0;
57 }

```

## 3 使用sysbench进行测试

### 3.1 sysbench安装

使用如下代码进行sysbench安装:

```
1 | sudo apt install sysbench
```

### 3.2 在虚拟机上执行测试

在不同机器上，统一使用如下代码进行测试：

```
1 sysbench --test=memory --threads=${threads} --memory-block-size=${memory-  
  block-size} --memory-total-size=100G --memory-access-mode=${mode} --memory-  
  hugetlb=[on|off] run
```

#### 3.2.1 虚拟机不使用主机的hugepage，虚拟机内部也不开启hugepage

测试结果如下：

threads	memory-block-size	mode	speed (MB/s)
1	1024K	seq	21983.28
2	1024K	seq	42659.60
1	16M	seq	19504.11
2	16M	seq	33104.29
1	1024K	rnd	2830.81
2	1024K	rnd	4815.39
1	16M	rnd	1366.86
2	16M	rnd	2542.66

#### 3.2.2 虚拟机使用主机的hugepage，但虚拟机内部不使用hugepage

在虚拟机的xml文件中添加如下配置，使用主机上开启的hugepage：

```
1 <memoryBacking>  
2   <hugepages>  
3     <page size="1" unit="G"/>  
4   </hugepages>  
5 </memoryBacking>
```

测试结果如下：

threads	memory-block-size	mode	speed (MB/s)
1	1024K	seq	21348.37
2	1024K	seq	41695.79
1	16M	seq	19752.54
2	16M	seq	34424.87
1	1024K	rnd	2660.92
2	1024K	rnd	4782.11
1	16M	rnd	1480.05
2	16M	rnd	2733.41

3.2.3 虚拟机不使用主机的hugepage，但虚拟机内部开启并使用hugepage

使用与主机相同的步骤在虚拟机中打开hugepage（本实验中开启了500个大小为2MB的hugepage）  
测试结果如下：

threads	memory-block-size	mode	speed (MB/s)
1	1024K	seq	20301.86
2	1024K	seq	39974.49
1	16M	seq	17391.45
2	16M	seq	28173.63
1	1024K	rnd	2821.32
2	1024K	rnd	5014.83
1	16M	rnd	1680.50
2	16M	rnd	3045.02

3.2.4 虚拟机使用主机的hugepage，虚拟机内部也开启并使用hugepage

在虚拟机的xml文件中添加如下配置，使用主机上开启的hugepage：

```
1 <memoryBacking>
2   <hugepages>
3     <page size="1" unit="G"/>
4   </hugepages>
5 </memoryBacking>
```

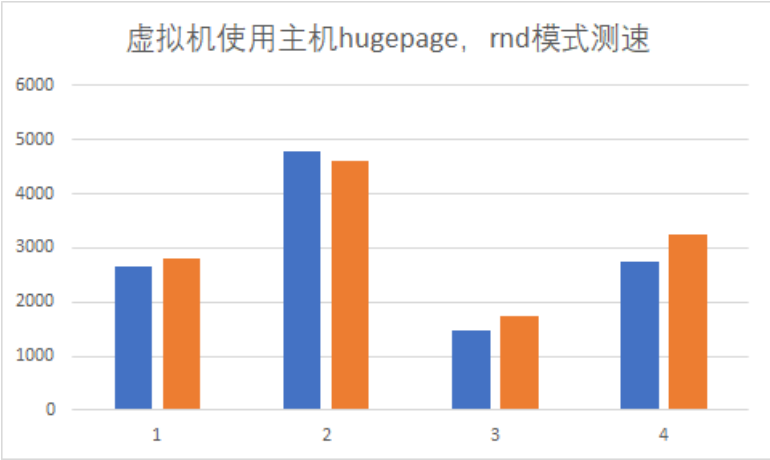
使用与主机相同的步骤在虚拟机中打开hugepage（本实验中开启了500个大小为2MB的hugepage）  
测试结果如下：

threads	memory-block-size	mode	speed (MB/s)
1	1024K	seq	21362.78
2	1024K	seq	41248.30
1	16M	seq	18676.12
2	16M	seq	32338.47
1	1024K	rnd	2790.03
2	1024K	rnd	4618.27
1	16M	rnd	1736.93
2	16M	rnd	3239.40

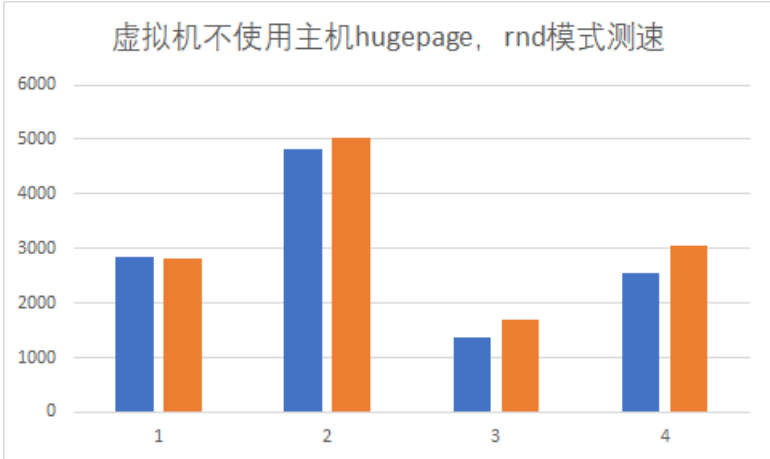
#### 4 测试结果图表（图表中蓝色代表虚拟机不开启hugepage，橙色代表虚拟机开启hugepage）

将上述测试结果绘制成柱状图：

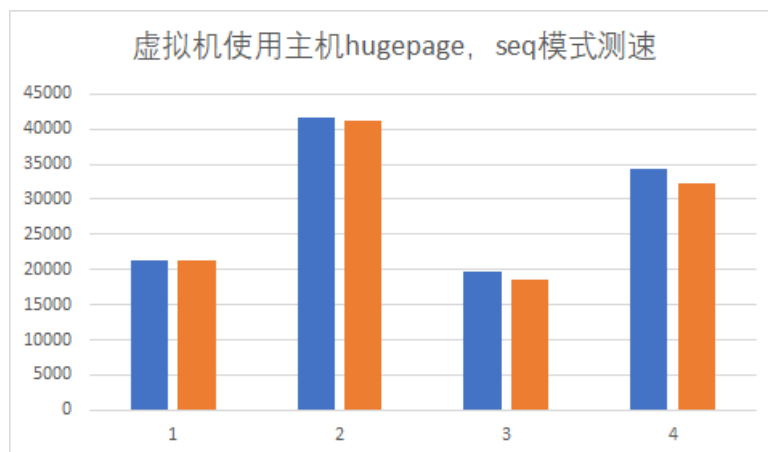
##### 4.1 虚拟机使用主机hugepage，rnd模式测速，比较虚拟机开或不开hugepage的性能差异



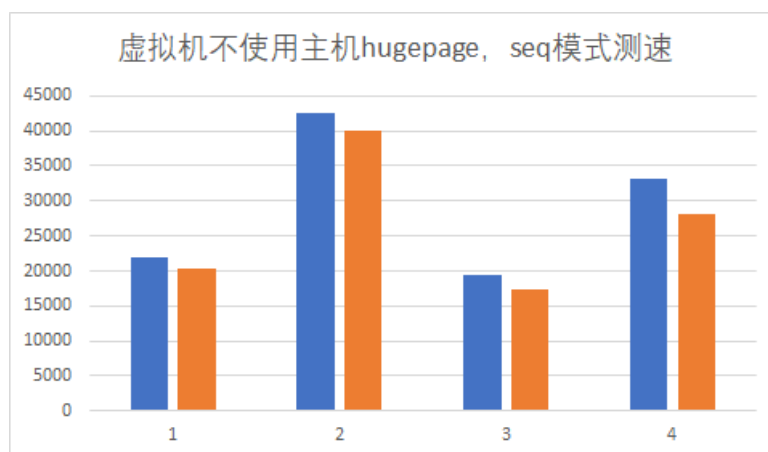
##### 4.2 虚拟机不使用主机hugepage，rnd模式测速，比较虚拟机开或不开hugepage的性能差异



### 4.3 虚拟机使用主机hugepage，seq模式测速，比较虚拟机开或不开hugepage的性能差异



### 4.4 虚拟机不使用主机hugepage，seq模式测速，比较虚拟机开或不开hugepage的性能差异



## 5 结果分析

从以上表格以及柱状图中可以得出如下结论：

1. 对比虚拟机使用或不使用主机的hugepage：控制其他条件相同时，虚拟机是否使用主机的hugepage对测速结果影响很小
2. 对比虚拟机使用或不使用虚拟机中开启的hugepage：
  1. 在rnd模式中，虚拟机使用自己开启的hugepage后，速度会有增加；尤其当mem-block-size设置为16MB时，开启hugepage对速度的提升非常明显（mem-block-size设置为1024KB时，开启hugepage对速度影响不大）
  2. 在seq模式中，虚拟机使用自己开启的hugepage后，对速度几乎没有影响，甚至速度会有小幅度下降

我认为上述结果的原因如下：

当mem-block-size的大小设置为较大的16MB时，如果本机没有使用hugepage，则会在rnd模式中引起非常频繁的page-fault，进而增加开销；若开启了hugepage，那么page-fault的发生概率大大减小，进而增加了内存读写速度。

## 6 致谢

感谢管海兵老师为我们讲解的云计算、虚拟机等相关内容知识，为我们拓宽了视野，同时教会了我们很多生活、科研中的思想；也要感谢侍硕助教不论是课上还是课外时间细致耐心的讲解、答疑、解惑，收获颇丰！