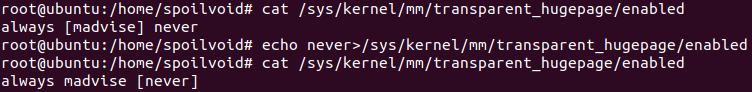
**课程大作业四**

叶增渝 519030910168

1.在VMWare开的Ubuntu虚拟机中关闭transparent\_hugepage



2.由于本虚拟机支持hugepage机制，所以已经存在本地目录，在这里进行挂载

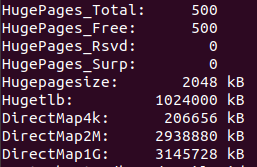
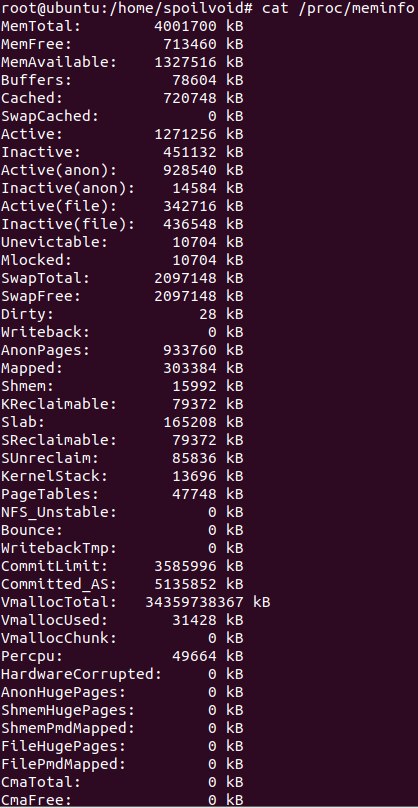


可以看到hugepage的TLB表已经链接到对应的位置上，每个hugepage的大小为2M

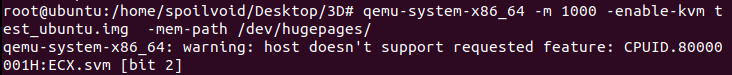
3.设置hugepage的数量为500个，即总共1G的huagepage



4.查看当前的mem配置文件

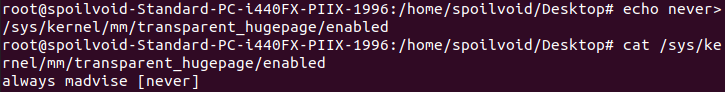


4.我们分配与hugepage大小相同的内存，并且将内存位置指向我们创建hugepage的目录，即host机器allocate hugepage

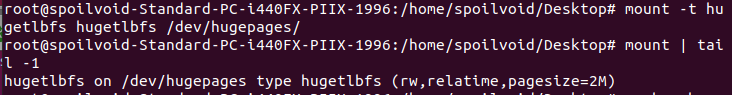


5.在打开的QEMU虚拟机上下载sysbench测试工具，并如上配置hugepage

将transparent\_hugepage关闭



挂载hugepage目录.每个hugepage大小为2M



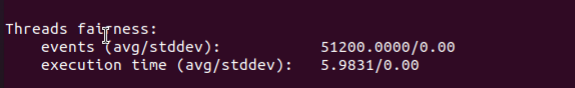
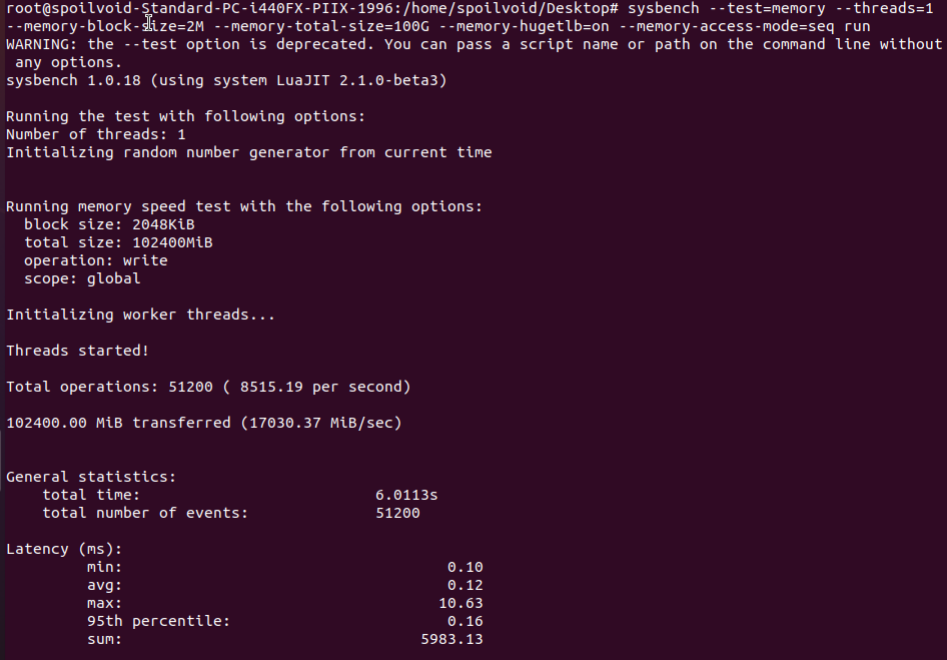
设置hugepage数量为200



6.在host机allocate hugepage的情况下在QEMU虚拟机内use hugepage进行sysbench memory test

(1)host机allocate hugepage，QEMU use hugepage：

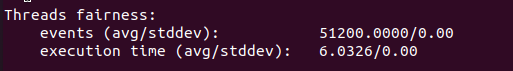
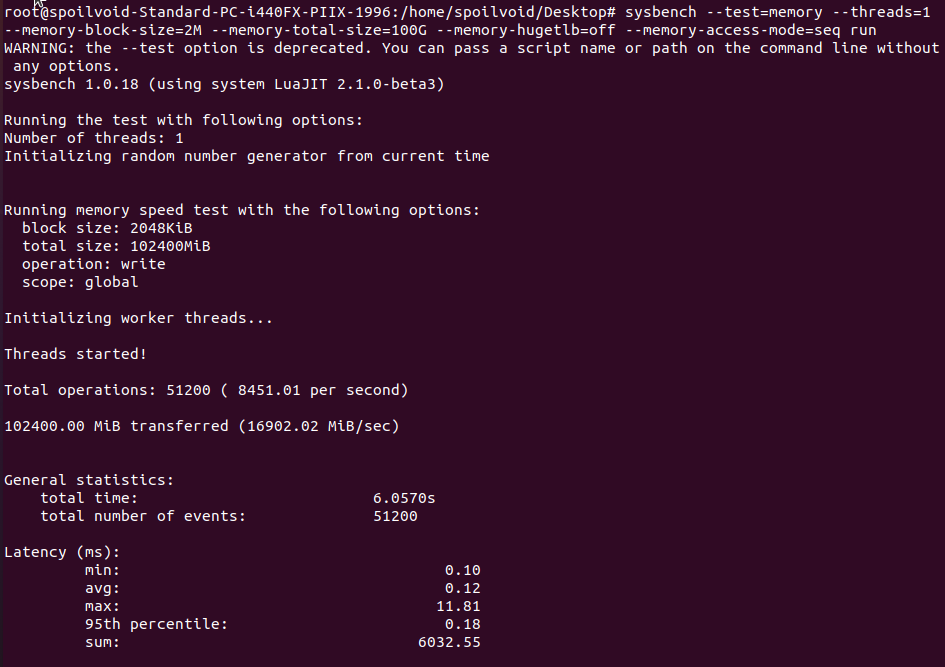
下方命令的含义为进行内存测试，线程数为1，每一个block为2M大小，总测试数据量为100G，从hugetlb即之前hugepage挂载的目录分配内存，进行顺序存储



最终得到transfer rate为17030.37MiB/sec

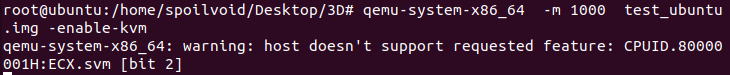
(2) host机allocate hugepage，QEMU not use hugepage：

下方命令的含义为进行内存测试，线程数为1，每一个block为2M大小，总测试数据量为100G，不从hugetlb中分配内存，进行顺序存储



最终得到transfer rate为16902.02MiB/sec

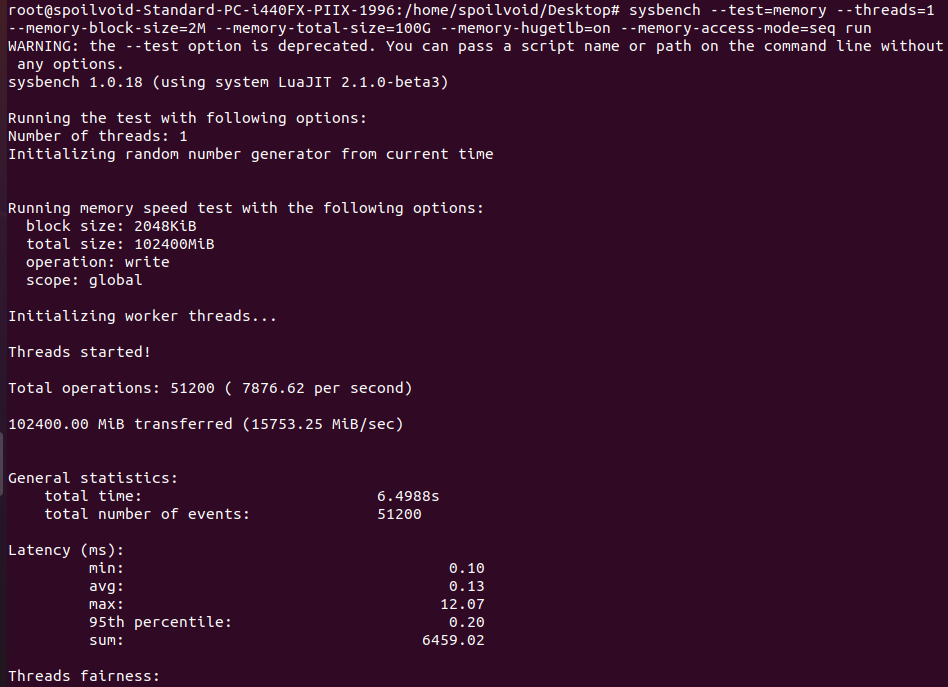
7. host机同样分配大小相同的1000M内存，不使用huagepage直接打开QEMU虚拟机，即host not allocate hugepage

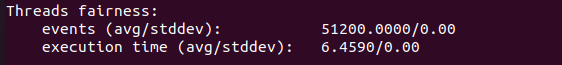


如上第5步配置虚拟机hugepage并关闭transparent\_hugepage分配200个2M大小的hugepage

(1)host机not allocate hugepage，QEMU use hugepage：

下方命令的含义为进行内存测试，线程数为1，每一个block为2M大小，总测试数据量为100G，从hugetlb即之前hugepage挂载的目录分配内存，进行顺序存储

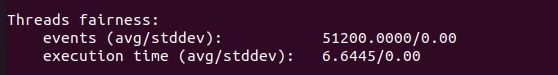
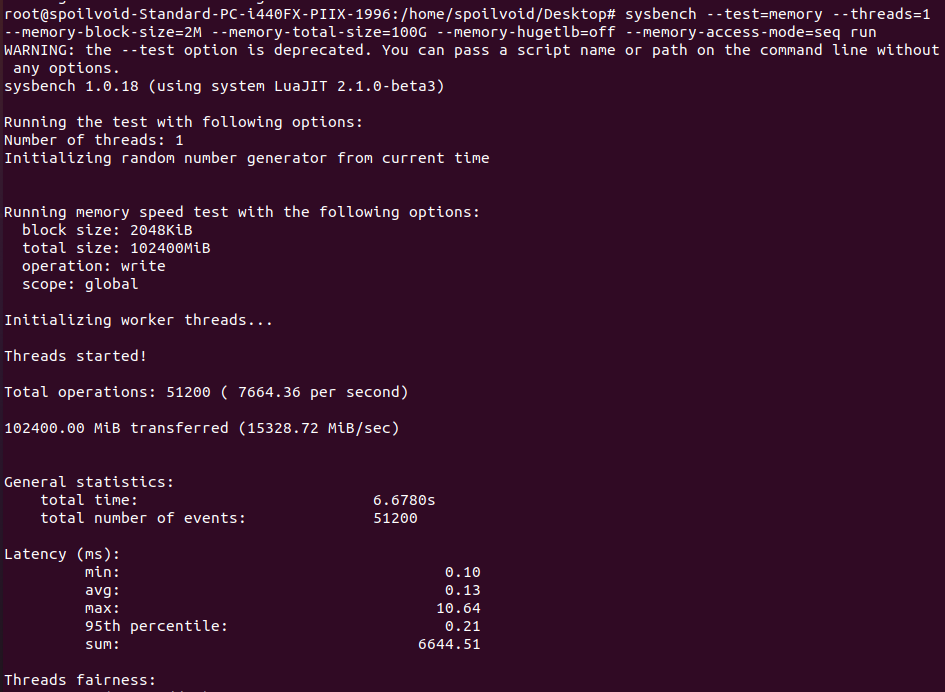




最终得到transfer rate为15735.25MiB/sec

(2) host机not allocate hugepage，QEMU not use hugepage：

下方命令的含义为进行内存测试，线程数为1，每一个block为2M大小，总测试数据量为100G，不从hugetlb中分配内存，进行顺序存储



最终得到transfer rate为15328.72MiB/sec

8.实验结果总结

|  |  |  |
| --- | --- | --- |
| Transfer rate | Host allocate hugepage | Host not allocate hugepage |
| QEMU use hugepage | 17030.37MiB/sec | 15735.25MiB/sec |
| QEMU not use hugepage | 17705.43MiB/sec | 15328.72 MiB/sec |

可以看到在host机allocate hugepage的时候，相比起不allocate hugepage，transfer rate有较大提升，在QEMU虚拟机中使用hugepage确实能提高一定transfer rate，但是效果不是很明显。

可能的解释：使用hugepage使得TLB表项减少，在查询真实地址时的时间减少，从而提升了transfer rate，而在QEMU虚拟机内由于本身由host机分配内存小，所以造成区别不大