COL732 ASSIGNMENT 3
VIRTUALISATION AND CLOUD COMPUTING

K Laxman
2018CS50408

1.Page table benchmarks

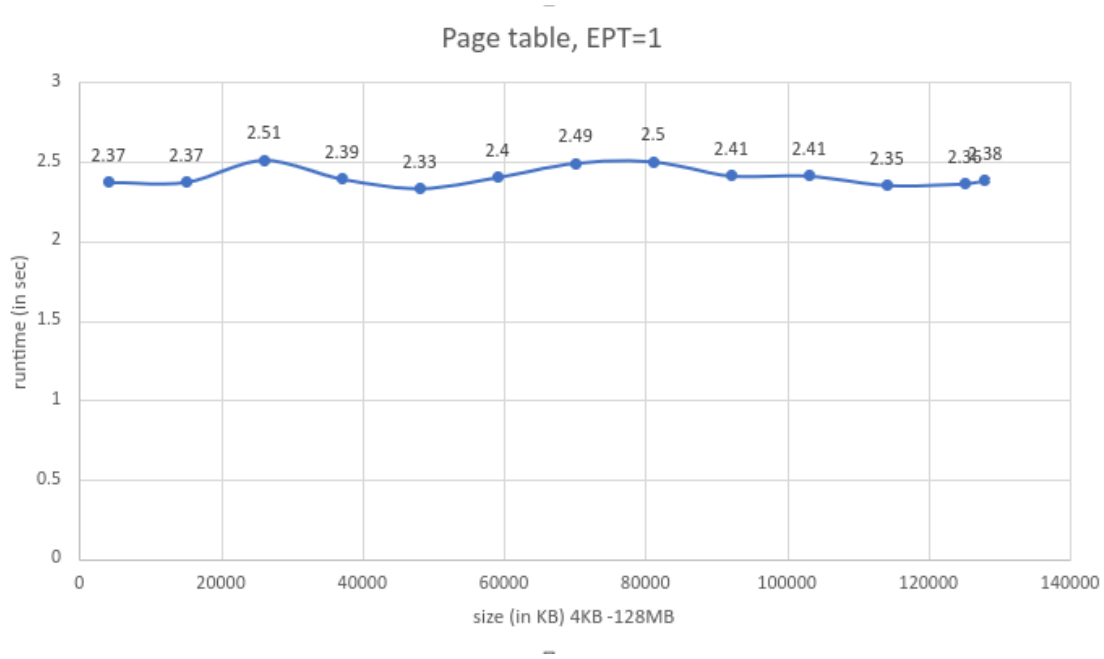
a.Experimental setup. Exact commands/programs that were run for the measurement.

Host : nc -w 3 192.168.241.2 1234 < page.out
Vm : nc -l -p 1234 > page.out

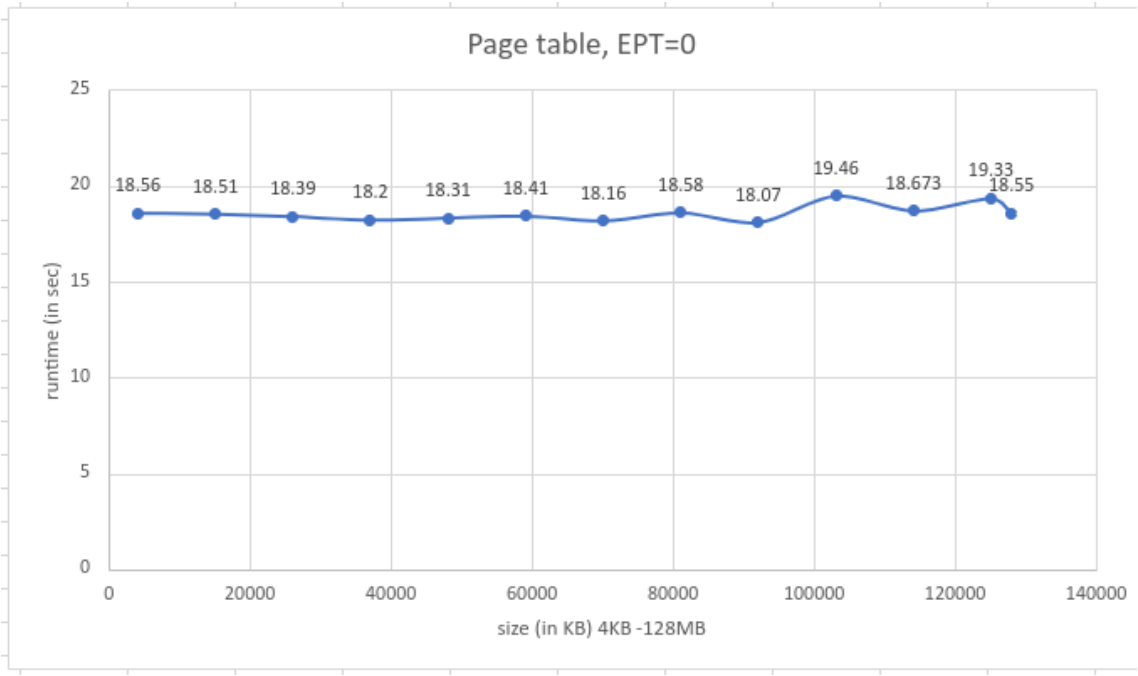
In Vm:

cd home
chmod +x page.out
time ./page.out 4096 here i have given the size as argument in KB with 11MB gap in each one and took total of 13 readings to analyse the benchmark.
For **enable/disable** EPT I have used :
sudo modprobe -r kvm_intel
sudo modprobe kvm_intel ept=0/1

b.Relationship of runtime (with EPT=1) with the size of the resident memory



c.Relationship of runtime (with EPT=0) with the size of the resident memory



d.Justification of observations

Here we have to traverse the whole page table whether we find the cache or not ,which takes the constant time for each traversal

As this is the shadow page table technique , it requires more memory so page table uses more time to fork the processes .

From the above graphs of pagetable we can say that it takes almost constant amount of time for traversal because it need constant time .

The TLB is a cache that holds (likely) recently used pages. It says that the pages referenced in

the TLB are likely to be used again soon. This is the underlying idea for all caching. When these

pages are needed again, it takes minimal time to find the address of the page in the TLB. The

page table itself can be enormous, so walking it to find the address of the needed page can get very expensive.

Therefore ept = 1 is better than shadow pagetable(ept =0) from the above graph 1 ,2

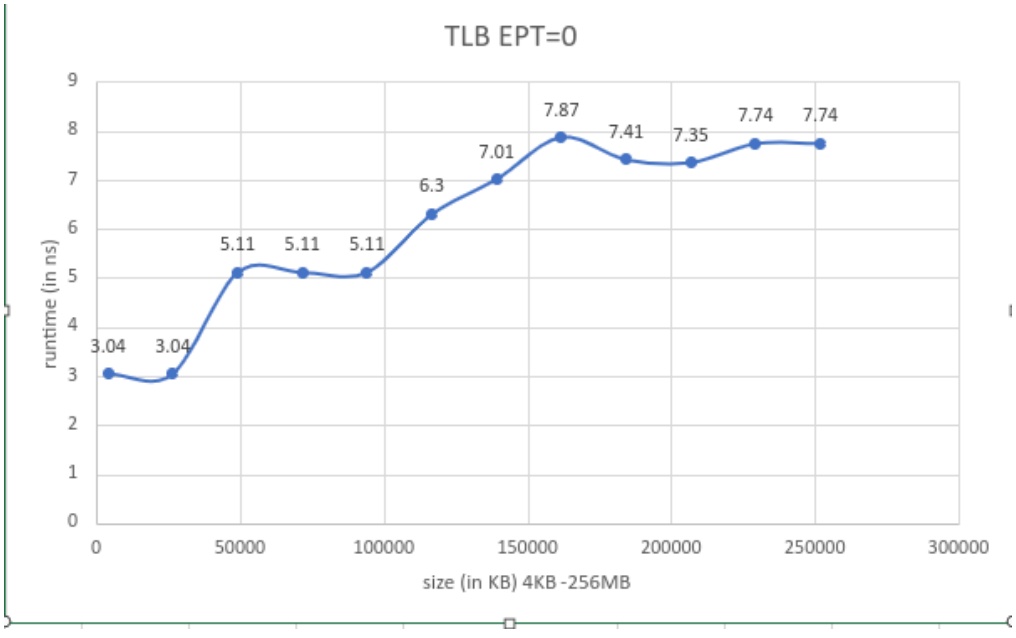
2.TLB benchmark

a.Experimental setup. Exact commands/programs that were run for the measurement.

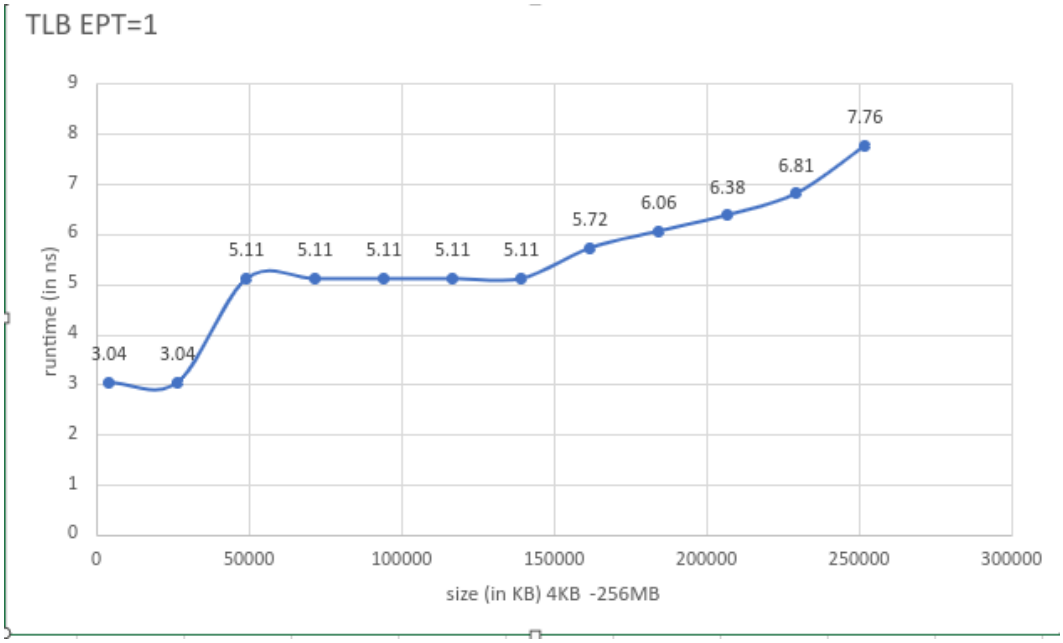
Host : nc -w 3 192.168.241.2 1234 < a.out
Vm : nc -l -p 1234 > a.out

In Vm:
cd home
chmod +x a.out
./a.out 4096 here i have given the size as argument in KB with 22MB gap in each one and took a total 12 readings to make a analysis .
For **enable/disable** EPT I have used in host:
sudo modprobe -r kvm_intel
sudo modprobe kvm_intel ept=0
cat /sys/module/kvm_intel/parameters/ept used this for checking whether ept is Y or N

b.Relationship of runtime (with EPT=0) with the map size (4KB – 256M)



c.Relationship of runtime (with EPT=1) with the map size (4KB – 256M)



d.Justification of observations:

If we get hit

From Graph 1:
It is increasing drastically because we have two pagetable as we visit ,we need to visit both

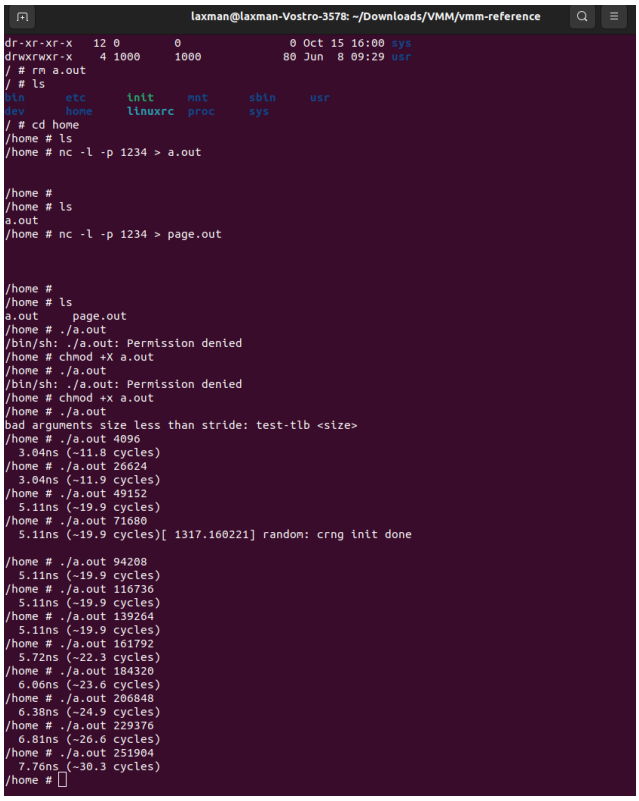
pagetables so there are chances of higher cache miss rate.

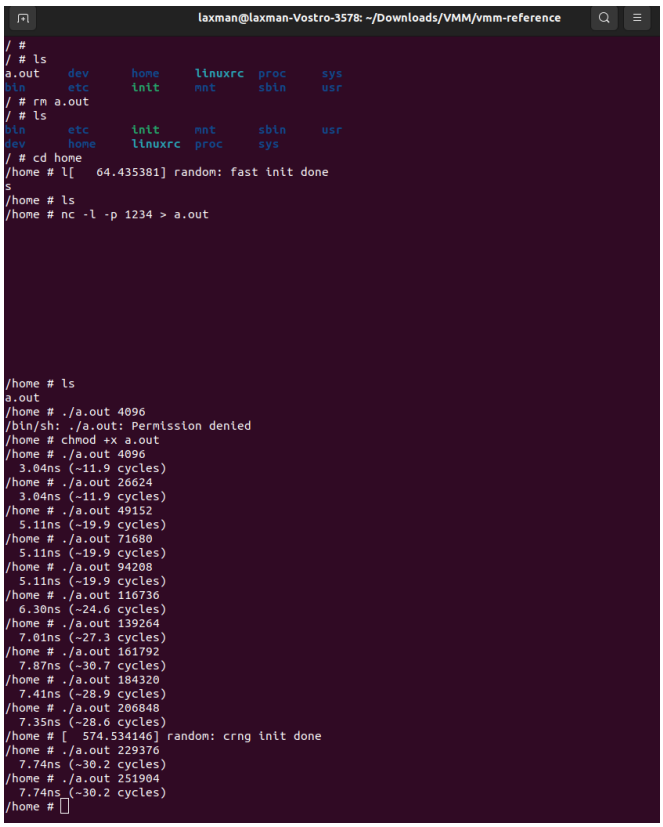
In the case of tlb it is increasing drastically because if the translational lookaside buffer

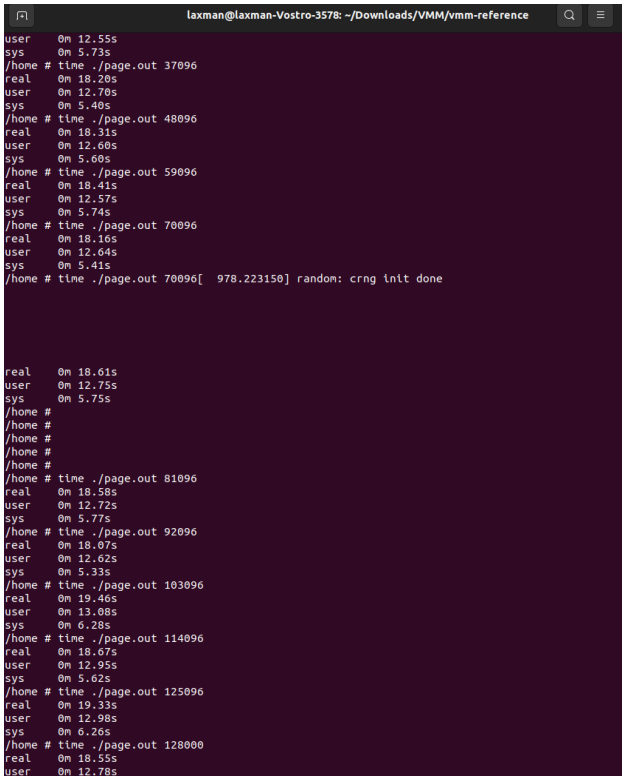
doesnt get any cache hits and with high miss rates .It takes time to search in the already filled tlb.
which makes the time complexity higher comparatively.

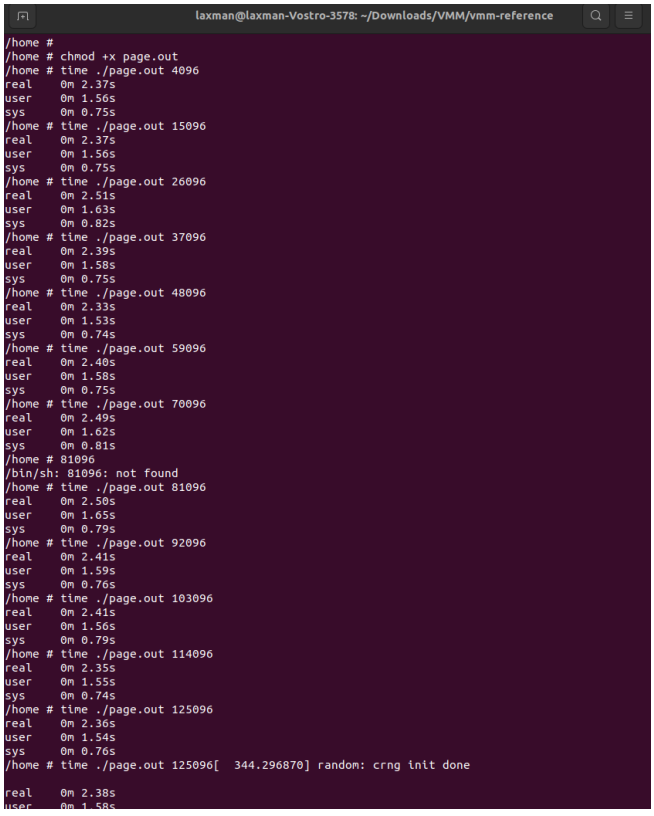
In this case we can see from above graph that as we are using EPT =0 which is shadow page table,it will be using single pagetable there are less chances of cache miss rate .

Screenshots :






laxman@laxman-Vostro-3578: ~/Downloads/VMM/vmm-reference
user 0m 12.55s
sys 0m 5.73s
/home # time ./page.out 37096
real 0m 18.20s
user 0m 12.70s
sys 0m 5.40s
/home # time ./page.out 48096
real 0m 18.31s
user 0m 12.60s
sys 0m 5.60s
/home # time ./page.out 59096
real 0m 18.41s
user 0m 12.57s
sys 0m 5.74s
/home # time ./page.out 70096
real 0m 18.16s
user 0m 12.64s
sys 0m 5.41s
/home # time ./page.out 70096[978.223150] random: crng init done
real 0m 18.61s
user 0m 12.75s
sys 0m 5.75s
/home #
/home #
/home #
/home #
/home #
/home # time ./page.out 81096
real 0m 18.58s
user 0m 12.72s
sys 0m 5.77s
/home # time ./page.out 92096
real 0m 18.07s
user 0m 12.62s
sys 0m 5.33s
/home # time ./page.out 103096
real 0m 19.46s
user 0m 13.08s
sys 0m 6.28s
/home # time ./page.out 114096
real 0m 18.67s
user 0m 12.95s
sys 0m 5.62s
/home # time ./page.out 125096
real 0m 19.33s
user 0m 12.98s
sys 0m 6.26s
/home # time ./page.out 128000
real 0m 18.55s
user 0m 12.78s


laxman@laxman-Vostro-3578: ~/Downloads/VMM/vmm-reference
/home #
/home # chmod +x page.out
/home # time ./page.out 4096
real 0m 2.37s
user 0m 1.56s
sys 0m 0.75s
/home # time ./page.out 15096
real 0m 2.37s
user 0m 1.56s
sys 0m 0.75s
/home # time ./page.out 26096
real 0m 2.51s
user 0m 1.63s
sys 0m 0.82s
/home # time ./page.out 37096
real 0m 2.39s
user 0m 1.58s
sys 0m 0.75s
/home # time ./page.out 48096
real 0m 2.33s
user 0m 1.53s
sys 0m 0.74s
/home # time ./page.out 59096
real 0m 2.40s
user 0m 1.58s
sys 0m 0.75s
/home # time ./page.out 70096
real 0m 2.49s
user 0m 1.62s
sys 0m 0.81s
/home # 81096
/bin/sh: 81096: not found
/home # time ./page.out 81096
real 0m 2.50s
user 0m 1.65s
sys 0m 0.79s
/home # time ./page.out 92096
real 0m 2.41s
user 0m 1.59s
sys 0m 0.76s
/home # time ./page.out 103096
real 0m 2.41s
user 0m 1.56s
sys 0m 0.79s
/home # time ./page.out 114096
real 0m 2.35s
user 0m 1.55s
sys 0m 0.74s
/home # time ./page.out 125096
real 0m 2.36s
user 0m 1.54s
sys 0m 0.76s
/home # time ./page.out 125096[344.296870] random: crng init done
real 0m 2.38s
user 0m 1.58s