Wednesday, December 15, 2021 3:18 PM

Making VM fast

Q. What do we have to do for each memory access with virtual memory?

- Load data from disk
- Access the page table in RAM (1)
- Translate the address (2)
- Update the cache
- Have the OS update the pagetable
- Access the data in RAM (3)

Step 1 — Memory Access X
Step 3 — Memory Access
} 2

We have an avg. of 1.33 memory access per instruction

IF, ID, Ex. Mem. WB

MIPS architecture

any context switch (Process Context Swich)

P1 P2

PT PT

Page Table Base Address — PTBR

Software solution to this problem? X

Hardware solutions

Cache Buffer

Principle of locality

i) Temporal locality — W.r.t. Time
ii) Spatial locality — W.r.t. Space

t t+1 t+2

t+δ

loop construct indx = 1 to 100

a a+1 a+2

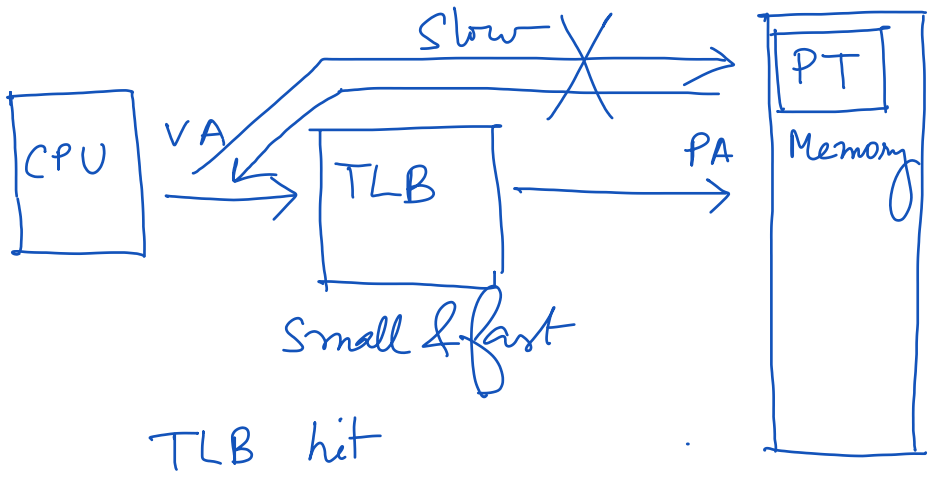
$\frac{a}{t}$ $t+\delta$

ith indx i+1 th indx

Buffer

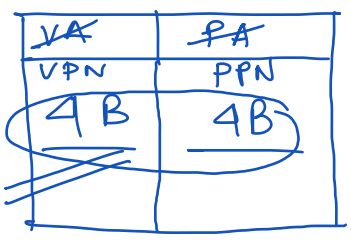
TLB — Translation Look Aside Buffer

Cache to Page Table



TLB hit

TLB Miss PT



32 bit system

32 bit VA

4kB

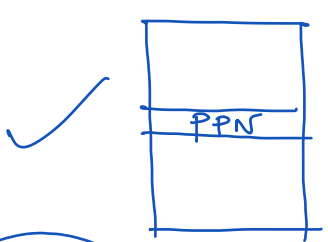
2^{20}

$2^{20} \times (4+4)$

$= 2^{23}$ B

$= 2^3 \times 2^{20}$ B

$= 8$ MB



$2^{20} \times 4B = 4$ MB

TLB 32 or 64 entries

1) Page is in RAM

A) PTE in TLB — TLB Hit

Excellent

B) PTE not in TLB — TLB Miss

— Poor

— 20 → 1000 cycles to load from PTE in RAM

2) Page not in RAM

— PTE in TLB

(Unlikely) ~1 cycle to know its on disk

~ 80 million cycles
to get from disk

— PTE not in TLB
(horrible)
20–1000 cycles to know its on disk
~ 80 m cycles to get it
from disk

Q. How can you make TLB, effectively bigger without slowing them down?

Page Size Larger

64 | 4 KB | = 256 KB
32 | 2 MB | = 64 MB

0 to $2^{32}-1$
0 to $2^{64}-1$ } ??

48 bit Address Virtual address
52 bit Address Physical Address

32 bit Address
32 bit "

47 bit used for userspace
2^{47} B $= 2^7 \times 2^{40}$ B
$= 128$ TB

Virtual address space is sparse
scattered

Add a second TLB that is larger,
but a bit slower

1st TLB 32 or 64 entries
2nd TLB 256 or 512 "

Page Table Size

32 bit system with 4KB pagesize
offset ⇒ 12 bit
VPN/PTI ⇒ 20 bit
PTE ⇒ 2^{20} entries
Each PTE is of size 4 B

$2^{20} \times 4\ B = 4\ MB$

100 programs running

$100 \times 4\ MB = 400\ MB$

Can we swap page tables out to disk? No

Page Tables should be always in RAM.

How can we fix this?

Add one more level of indirection

Multi level Page Tables

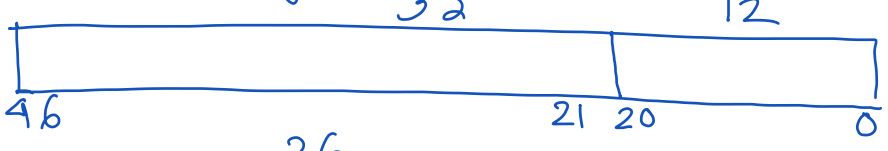
0 to $2^{47} - 1$

Page Size 2 MB $= 2 \times 2^{20}$ B

4kB

Page offset $\Rightarrow$ 21 bit / 12

35 | 12



$2^{26} \times 8\ B$

$= 2^{26} \times 2^{3}\ B$

$= 2^{29}\ B$

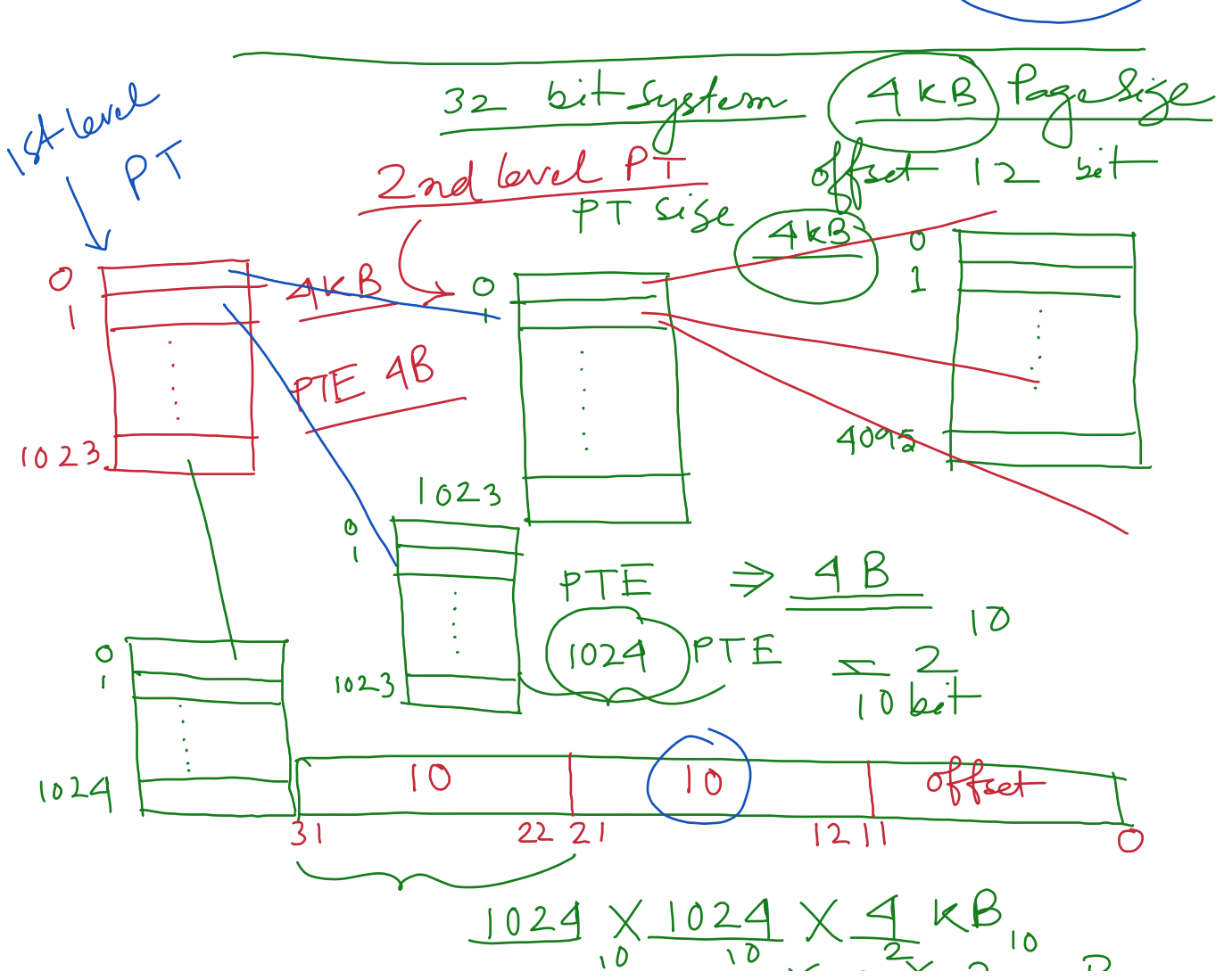
512 MB

$2^{35} \times 8\ B = 2^{38}\ B$

$= 256\ GB$

32 bit system 4kB Page Size

offset 12 bit



$1024 \times 1024 \times 4\ kB$

$10 \quad 10 \quad 2^{10}$

$= 2 \times 2 \times 2^{\wedge} 2$ B

$= 2^{32}$ B 4GB

9. With 2-level pagetables what is the smallest amount of pagetables we need to keep in memory?

— 4 MB + 4KB (always need the whole PT + the 1st level page)

— 4 MB (Always need to whole PT in RAM)

— ~~4KB + 4KB~~ (need the 1st level PT and one 2nd level PT)

— 4 KB (just need the 1st level PT)

Intel x-86 architecture for 32 bit system

4 MB 8 KB

400 MB ✓ 800 KB ✓