Working Set Model, VM Review

ECE595
Mar 1

Y. Charlie Hu

[week6] Virtual Memory

- Definition: *Virtual memory* permits a process to run with only some of its virtual address space loaded into physical memory
- VM is typically implemented by demand paging
  - Virtual address space translated to either
    - Physical memory (small, fast) or
    - Disk (backing store), large but slow
- Objective:
  - To produce the illusion of memory as big as necessary

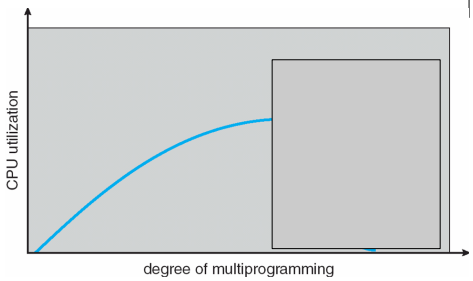
Quiz

- Who is this person?

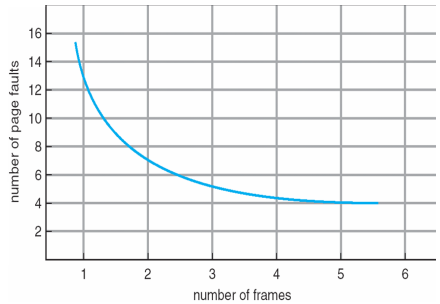
Improving CPU utilization in multiprogramming

- In multiprogramming, when OS sees the CPU utilization is low,
  - It thinks most processes are waiting for I/O
  - it needs to increase the degree of multiprogramming (actual behavior of early paging systems)
  - It adds/loads another process to the system
    - Assume I/O capacity is large, every job spends 50% of time performing I/O, how many such jobs are needed to keep CPU 100% utilized?

Towards Improving CPU utilization



Page faults versus number of physical pages for a process



When there are not enough page frames

- Suppose many processes are making frequent references to 50 pages, memory has 49
- Assuming LRU
  - Each time one page is brought in, another page, whose content will soon be referenced, is thrown out
- What is the average memory access time?
- The system is spending most of its time paging!
- The progress of programs makes it look like "*memory access is as slow as disk*", rather than "*disk being as fast as memory*"

Thrashing

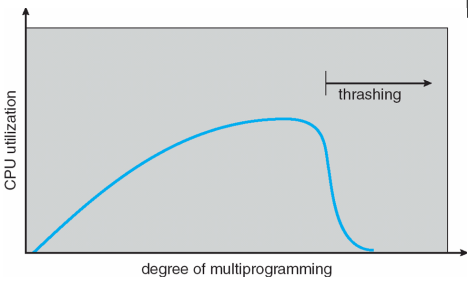
- **Thrashing** ≡ a process is busy swapping pages in and out

Thrashing can lead to vicious cycle

- If a process does not have "enough" pages, the page-fault rate is very high. This leads to:
  - low CPU utilization
  - OS thinks that it needs to increase the degree of multiprogramming (actual behavior of early paging systems)
  - another process added to the system
  - page fault rate goes even higher

Vicious Cycle

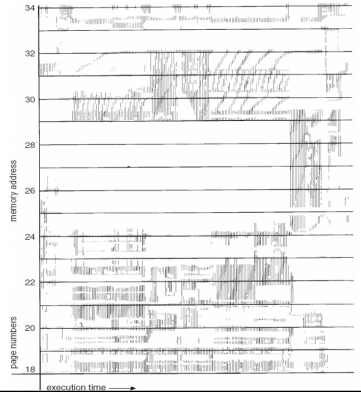
Thrashing (Cont.)



Demand paging and thrashing

- Why does demand paging work?
  - Data reference exhibits locality
- Why does thrashing occur?
  - Σ size of locality > total memory size

Locality in a memory-reference pattern (OSC, 8th ed, Fig 9.19)

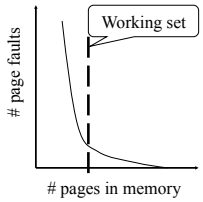


Intuitively, what to do about thrashing?

- If a single process's locality too large for memory, what can OS do?
- If the problem arises from the sum of several processes?
  - Figure out how much memory each process needs – "locality"
  - What can we do?
    - Can limit effects of thrashing using local replacement
    - Or, bring a process' working set before running it
    - Or, wait till there is enough mem for a process's need

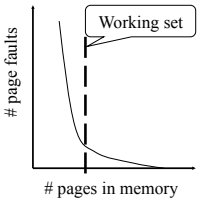
Key observation

- Locality in memory references
  - Spatial and temporal
- Want to keep a set of pages in memory that would avoid a lot of page faults
  - "Hot" pages
- Can we formalize it?



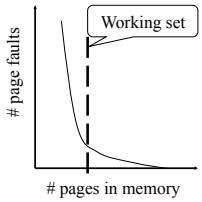
Working Set Model – by Peter Denning (Purdue CS Head, 79-83)

- An informal definition:
  - Working set: The collection of pages that a process is working with, and which must thus be resident if the process is to avoid thrashing
- But how to turn the concept/theory into practical solutions?
  1. Capture the working set
  2. Influence the scheduler or replacement algo



Working Set Model – by Peter Denning (Purdue CS head, 79-83)

- Usage idea: use recent needs of a process to predict its future needs
  - Choose δ, the WS parameter
  - At any given time, all pages referenced by a process in its last δ seconds comprise its working set
  - Don't execute a process unless there is enough mem to fit its working set
- Needs a companion replacement algo



Working Set Clock algorithm (1)

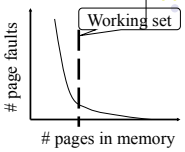
- Main idea
  - *Take advantage of reference bits*
  - *Variation of FIFO with 2nd chance*
- An algorithm (assume reference bit)
  - On a page fault, scan through all pages of the process
  - If the reference bit is 1, clear the bit, record the current time for the page (approx. last us time)
  - If the reference bit is 0, check the "last use time"
    - If the page has not been used within δ, replace the page
    - Otherwise, go to the next page

Working Set Clock Algorithm (2)

(assume reference bit + modified bit)

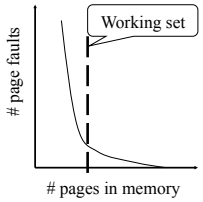
- Upon page fault, follow the clock hand
- If the reference bit is 1, set reference bit to 0, set the current time for the page and go to the next
- If the reference bit is 0, check "last use time"
  - If page used within δ, go to the next
  - If page not used within δ and modify bit is 1
    - Schedule the page for page out (then reset modify bit) and go to the next
  - If page not used within δ and modified bit is 0
    - Replace this page

Challenges with WS algorithm implementation



- What should δ be?
  - What if it is too large?
  - What if it is too small?
- How many jobs need to be scheduled in order to keep CPU busy?
  - Too few → cannot keep CPU busy if all doing I/O
  - Too many → their WS may exceed memory

More Challenges with Capturing Working Set



- Working set may not be static
- There often isn't a single "working set"
  - e.g., Multiple plateaus in previous curve (L1 $, L2 $, etc)
  - Program coding style affects working set
- Working set is often hard to gauge
  - What's the working set of an interactive program?
  - How to calculate WS if pages are shared?

Virtual Memory Review (1/3)

- Page fault handling (mechanism)
- Paging algorithms (policy)
  - Optimal
  - FIFO
  - FIFO with 2nd chance
  - Clock: a simple FIFO with 2nd chance
  - LRU
  - Approximate LRU
  - NFU

Virtual Memory Review (2/3)

- Important questions
  - What is the use of optimal algo?
  - If future is unknown, what make us think there is a chance for doing a good job?
  - Without addi. hardware support, the best we can do?
  - What is the minimal hardware support under which we can do a decent job?
  - Why is it difficult to implement exact LRU? Exact anythg
  - For a fixed replacement algo, more page frames → less page faults?
  - What are stack algorithms?
  - How can we move page-out out of critical path?

Virtual Memory Review (3/3)

- Per-process vs. global page replacement
- Thrashing
- What causes thrashing?
- What to do about thrashing?
- What is working set?

Reading Assignment

- Chapter 8