Instrumentation and Measurements of Multics

5/21/70

Background on Multics

Work started ~1964

Done by GE, BTL, MIT

Objective: prototype computer utility; sharing of information

Techniques: demand paging, multiprogramming

Background on Instrumentation

Paper in ACM 2nd Symposium Op-Sys.

Will be published in ACM Comm.

Overall types

- software accessible clocks (used everywhere)
- general software packages (less emphasis now)
- specific counters and clock readers (pricing tool)
- PDP-8 monitor/script loader (used daily)

Interest in numbers as well as techniques

(Slide 1) (8 numbers)

Most people recognize that absolute numbers are meaningless

(Slide 2)

But that numbers are equally meaningless - We need to add

- uncertainty
- conditions of measurement
- deep understanding of what the numbers mean.

Start with recent hardware measurements of C.P.U test points.

- Digital counter observing pulses
- 10 second period, average of 25 samples
- Wide ranging load (10 - 35 users)

1. 410,000 virtual memory references / sec. } ± 10%
330,000 instruction executions / sec. }

- supervisor uses segmented address space
- Instructions are fetched in pairs except on transfers

$\frac{410}{330}$ = 1.25 virtual address ~~memory~~ references per instruction
−.5 instruction fetches per instruction
.75 ~~virtual address~~ operand references per instruction

suggests: effective address calculation is a popular pastime in segmented environment

2. 98% — 99.5% } Associative Memory searches are successful.

- 16 registers
- retain recently used SDW's and PTW's
- strict LRU (hardware) algorithm
- ~~glitch~~ does not really use SDW's
- ~~plan to plot as function of size ≤ 16.~~

3. 20,000 Base register bounds base.

- ~~8 pairs~~ 4 paired registers (segment-number)
- Upper bound on improvement if there were more address base registers. $\frac{20}{330} = 6\%$

Measurements related to Demand Paging

Background

- no prediction (we demand even the previous page when we will start)

~~- user loop chain of pages, b~~

- uses hardware bits to detect use
- pages are chained in a circle; select first page with use bit off. (Similar to CP-67 and MTSS)

Average # of steps to find a page is 4.1 ± .5 (min. is 1.0)

independent of
- number of users
- nature of use
- load
- idle time

Does drop to 3.5 when larger core configuration is used.

Running time between page faults

20 ms ⟷ 70 ms

heavy load, small configuration — lighter load, large configuration

So far, no model which predicts what it has been constructed.

Loop time

(Time a page can remain idle before being discovered = 1.5 * loop time.)

~~B160~~ .97 sec. fully loaded.
3-4 sec. under a light load

Implications: - loss of a page while waiting for a page is not common. (Interactive tuning.)

- C/W average time a process is loaded (~1 sec.) suggests his pages stay in core twice as long as strictly necessary. (suggests maximum utility ~~value~~ of purging)

Cost of missing page handling

- time to compute which page is needed and where to put it
- time to switch to another process (if any are ready)
- time spent idling because core is full

page fault + switch time = 3.4 ms (about ~~1000~~ 1200 instructions)

% time in p.f. handling	10-20	under full load
idle time	10-20	under full load
Does not vary much.	25-35	" " "

N. B. Multiprogramming depth (3 with 256K core) is adjusted to minimize that sum.

Paging read/write ~~ration~~ ratio is 2/1
(because of pure procedure)
(no model)

Paging rate is
60 pages/sec to paging drum
12 pages/sec to fixed head disk.
72 p.i.s. under full load

No automatic move mechanisms yet.
Static assignment when file is created.
- session Temporaries on drum
- system commands on drum
- user personal files on disk

$\frac{60}{12} \simeq 5 = \frac{90}{18}$ } ratio of waiting times

~ 50% of page waiting time is for each device.

System Capacity

Criteria

Interactive response falls over cliff
(average > 5 sec.)

True Idle time snuffs out for long periods
Queues build up.

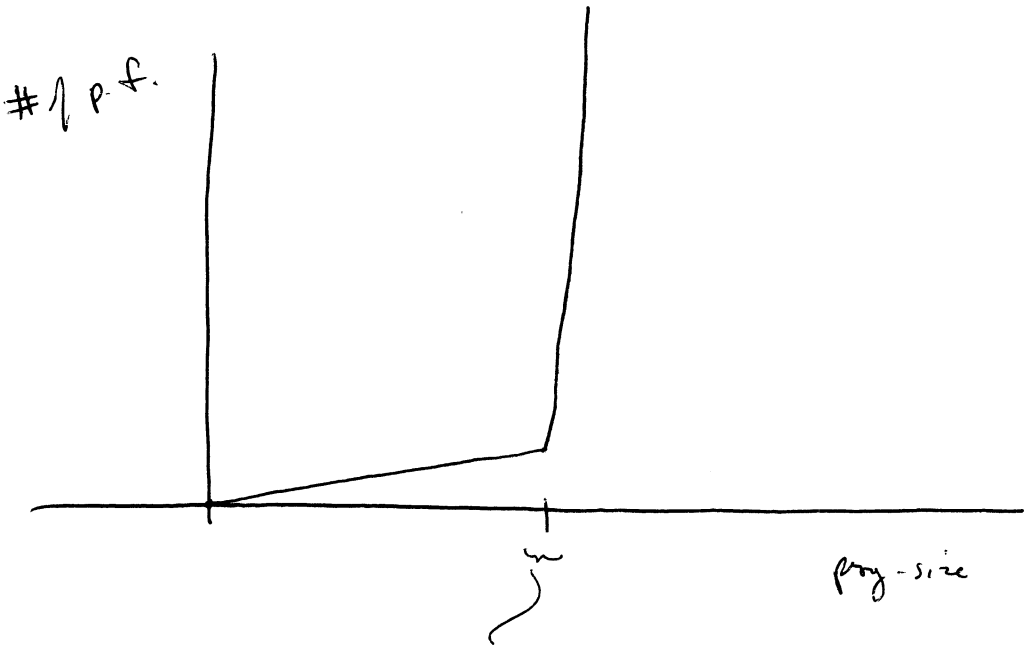
What are the users doing?

256K 1 CPU

12	~~P~~ System Programmer (PL/I)
1	~~Restricted~~ Batch compiler process
2	I/O processes (Backup tape; Printer CR PU)
20	General users (⅓ BASIC, ⅔ PL/I)
35	simultaneous users

With 384K 2 CPU we haven't found the limit -
it is certainly ≥ 40.

Very sharp paging threshold.



value = 85-90 K words fully loaded 256 K system
= 150 K " " 384 K "
= 161 K unloaded 256 K system

35
24
5
---
84 K %
11
---
10 K