L6 2/23/04

(0) m

Performance

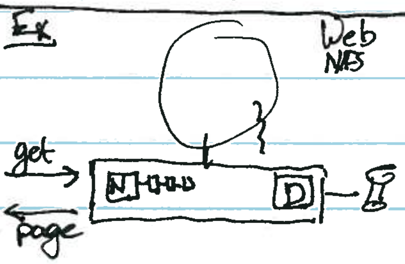
- caching } today
- concurrency
- batching } wed.
- speculative I/O
- I/O scheduling

(replication/parallelity)

↳ flashcrowds.

(1) F

Ex.



(show slide 1)

m

Simple:

latency: disk limited 10msec

throughput: 100/s

⇒ 8 M/day

(not all pages are likely to be accessed

↓ slide 2.

(4) m

Cache (6.004).

$A = H \times A_R + (1-H) D_D$

0.99 × 10nsec + 0.01 × 10ms

100 μsec ⇒ 10,000/s.

0.999 → 19.9 μsec

locality of reference

↳ web not exactly like computer programs, but zipf conditions hold.

(5) F

Replacement

LRU ↩ 2 3 4 MRU

used by flash

LRU		MRU	
1	4	1	1
2	3	2	2
3	2	3	4

⇒ 1/second

(1) what is the best page to throw out?

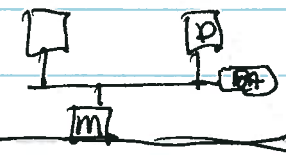
HW: replacement strategy dictated by cache structure
direct-mapped
N-way associative

SW: more flexibility

— How good are we doing?
block request on miss!
⇒ multiple threads
(show code)

(6) m

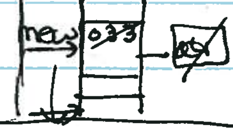
T1: A, B, C, D 6.004
T2: A, B, C, D 6.033



(7) F

Race condition

$A_1 A_2 B_2 B_1$
$C_1 C_2 D_2 D_1$



(difficult to debug
therac-25
mod ↦ passwd file
preemptive scheduling
in our threadpackage

(8) **Isolation** lock all

whenever thread write-share a variable

acquire(lock)
atomic
release(lock)

programmer convention ⇒
skip modularity
show code
(still wrong)

(9) m

acquire(lock) {
while (lock==1);
lock ← 1
}

1: taken
0: free

(10) F

TSL mem, R
R ← [mem]
[mem] ← 1
(bus arbiter)
(show code)

small atomic → big atomic