>W9

- Cristian's Algorithm

$$T_s = \text{Server time} + \frac{T_1 - T_0 - I}{2}$$

- Berkley's algorithm

$T_{master} = T_m$ + average time differences

↳ All slaves change time to master $\quad \Delta T_n = T_M - T_n$

- Averaging Algorithm (all process change)

$$\Delta M_0 = \frac{M_0 + M_1 + M_2 + M_3}{4} - M_0$$

- Lamports: Happens before, transitive

- WFG, resource allocation situations

>W10

- MUTEX: Critical section: coordinator, distributed, ring token

request, grant, release — ok signal, check sent vs recv time — only process w/ token

- Deadlock: 1) Mutual Exclusion: Resource held by at most 1 process
  2) Hold and Wait: Process holds resource, waits for another
  3) Non-preemption: Resource cannot be predicted. Must be released
  4) Circular-wait: $P_0 \to P_1 \to \dots \to P_0$

Knots / cycles → one process/ resources
↳ multiple resources

- Deadlock detection: - Ostrich

false deadlock from delays → - Coordinator merge resource graphs
→ - CMH: probe (original, sender, recv)

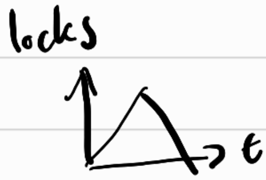
> Election

- Bully: Process sends election msg
  ↳ if higher process, response, new leader
  ↳ cont. until no response / timeout = new leader
- Ring: → Send election msg until reaches sender
  ↳ Higher process ignores msg
  ↓
  coordinate msg sent

> Concurrency control

- Lock based: Shared / Exclusive locks — RW, WW conflicts
  - Shared ↓ read ↓ can be read by other shared/unlocked
  - Exclusive ↓ write ↓ cannot be written by anything

  2-phase locking: 1) Get all locks
  2) Release all locks



- Timestamp: Read: $T \geq x_{\text{write time}}$
  Write: $T \geq x_{\text{write}}$ AND $T \geq x_{\text{read}}$

- Optimistic: Check conflicts @ end — 1) read 2) validate 3) write
  Validate: 1) T_i finish 3 phase $< T_j$
  OR 2) T_i finish $< T_j$ write
  OR 3) T_i finish read $< T_j$ read

7W 11

> Solving Matrix input time computations
  ↳ Row based
  ↳ Tile based

Question Collations

1) a) $24 \times 8 \times 64 + 16 \times 16 \times 12.6 + 6462.7 \times 2$
$= 28439$ GFLOPS

b) $8 \times 64 + 6462.7 = 6974.7$ ✓

1) a) $24 \times 8 \times 10 + 12^2 \times 5.6 + 9322.46 \times 2$
$= 21371.32$

b) $8 \times 10 + 9322.46 = 9402.46$ ✓

2) $10^{10} \times 32$ bits

$$t = \frac{10^{10} \times 32}{8 \times 1024^3 \times 256} = 0.146s$$ ✓

2) $24000^2 \times 16$ bits

$$t = \frac{24000^2 \times 16}{8 \times 1024^3 \times 32} = 0.0335s$$ ✓

3) Cristian's

S

T_0 T_1

$$T_s = S + \frac{T_1 - T_0 - I}{2}$$

$$= 15:55 + \frac{16:21 - 15:05 - 0.05}{2}$$

$\frac{75.95}{2} = 37.975$

$15:55 + 37.975$
$= 16:32.975$

$16:21 \rightarrow 16:32.975$
$= +11.975s$

✓

3) $= 27:11 + \frac{27:21 - 26:15 - 0.04}{2}$

32.98

$= 27:43.980$

$43.98 - 21 = +22.98\,s$

4) Averaging Algorithm

$$\Delta M_1 = \frac{M_1 + M_2 + M_3 + M_4}{4} - M_1$$

$$1 = \frac{-20 - 10 + M_4}{4}$$

$4 + 30 = M_4$ $\qquad M_4 = +34$

$M_4 = 12:00:34$

4) $$5 = \frac{-5:14 - 2:10 + M_4}{4}$$

$$5 = \frac{-444 + M_4}{4}$$

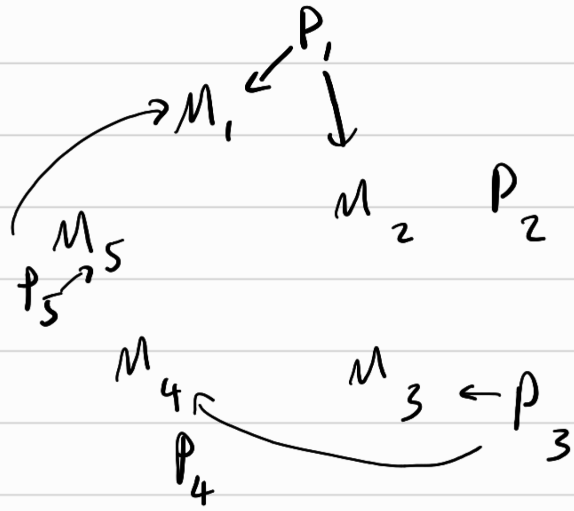
$M_4 = +464\,s$

$+7:44$

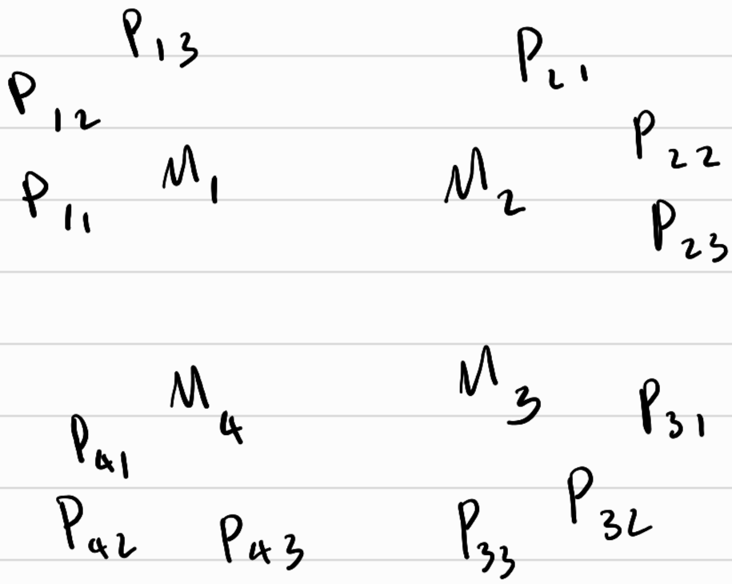
$M_4 = 10:07:44$

5) Ring Topology

4



5)

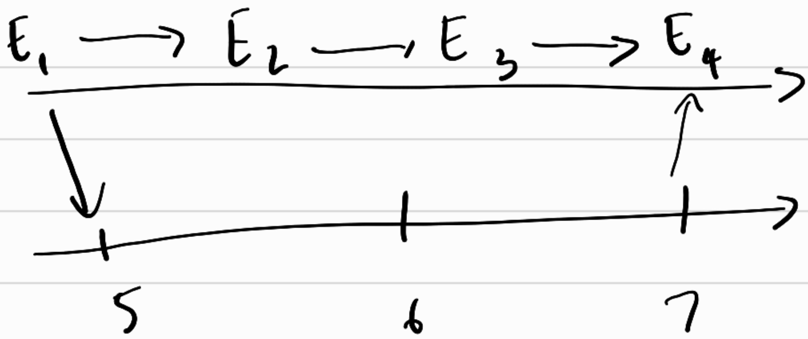


4

> W10

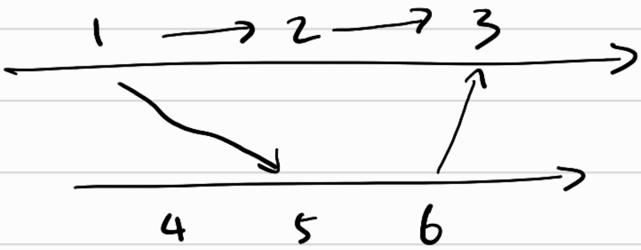
Lamport's

1)



1 2

1)



2 4 !

Elections

2) 4

2) 1

3) Optimistic CC

$T_1 \xrightarrow{r} x_1 \xrightarrow{w} x_2$

$T_2 \xrightarrow{r} x_2 \xrightarrow{w} x_3$

$T_3 \xrightarrow{r} x_3 \xrightarrow{w} x_1$

for read

$T \geq x_{write}$

for write

$T \geq x_{write}$

$T \geq x_{read}$

3

3) 3 4 5 6

4) Ahmdal's w/ comm

$$S = \frac{1}{0.5 + \frac{0.5}{2}} = 1.33$$

$orig_{time} = 80s$

40s	20s	20s
parallel	serial	comm

1) $\frac{40}{2} + 20 + 10$

$= 50s$

2) $\frac{40}{2} + 20 + 5$

$= 45s$

3) $40 + 20 + 10$

$= 70s$

9) $$S = \frac{1}{0.5 + \frac{0.5}{2}}$$

$= 1.33$

$orig_t = 100s$

50s	25s	25s
p	s	comm

a) $\frac{50}{2} + 25 + \frac{25}{2^2}$

$= 56.25s$

b) $\frac{50}{2} + 25 + \frac{25}{8}$

$= 53.125s$

c) $50 + 25 + \frac{25}{4}$

$= 81.25s$

> Mutex etc.

10)

t_4 A t_1

D B

t_3 C t_2

(3) (4) (6) (7)

(8)

> W11

1) $\frac{600}{2^x} \le 50$ $x = 4$

MPI

2) a) $\frac{100}{20} = 5$ rows matrix still needs whole column to

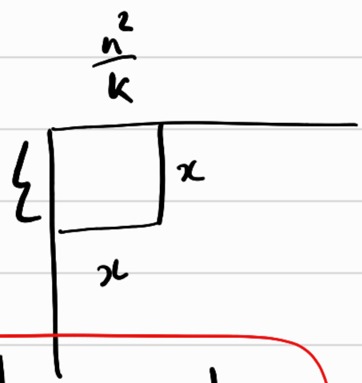
100 columns ↗ perform multiplication

105 sends × 100 = 10500 ms

b) 105 × 100 × 19 = 199500 ms

machines

3) a) tile size = $\frac{100^2}{25}$ = 400



$x = \sqrt{\frac{n^2}{k}}$

for matrix, we need x rows and x columns

10 rows, 20 columns

×100 ×100 elements ?

2000 + 2000 = 4000 ms

4000 × 24 = 96000 ms

7 W12 +

11) input vector = 40000 x 15000