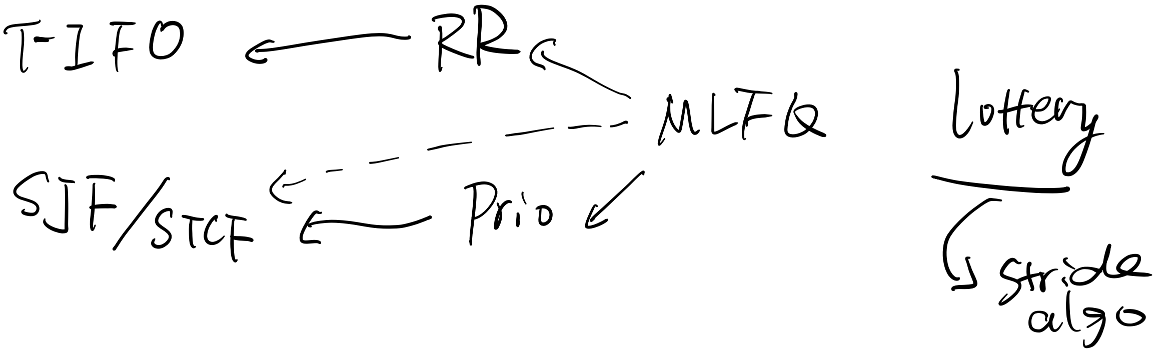
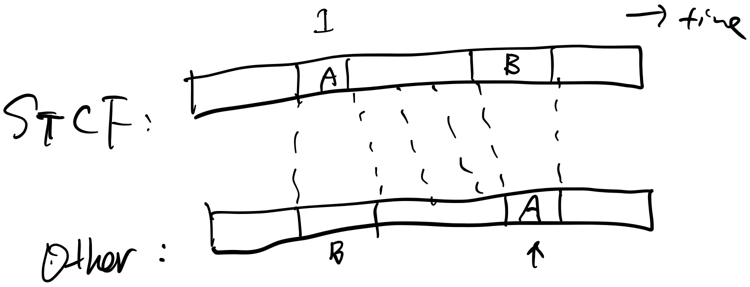
1. Last time
2. More about scheduling
3. Scheduling problem today
4. Lessons and conclusions
5. Threads
6. Intro to concurrency

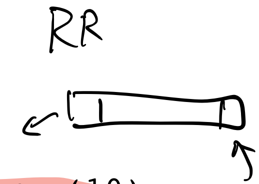
GPU + MIG

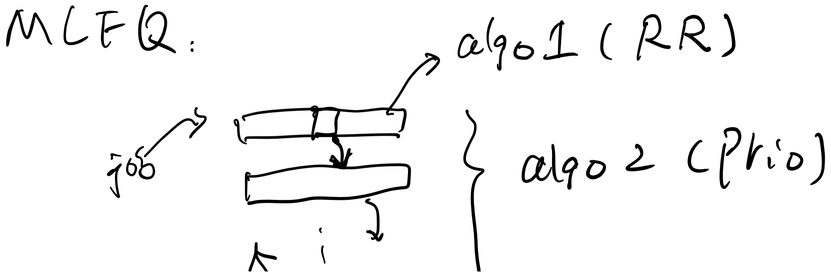




Votes last time:
(with candidate >=5 votes)

"Best Turnaround Time":	STCF (48)
"Best Response Time":	RR (34), MLFQ (10)
"Best Fairness":	RR (33), MLFQ (10), lottery (10)
"Most popular algorithm":	MLFQ (23), lottery (14), STCF (5), Prio (5)





random.

- randomized algo
- SGD
- PoW, PoS

P1 P2 P1 P1 P2

Stride:

		cost
P1	(20)	1
P2	(10)	2

P1

0 +1 +1 +1

P2

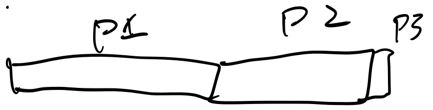
0 +2 +2

- I/O

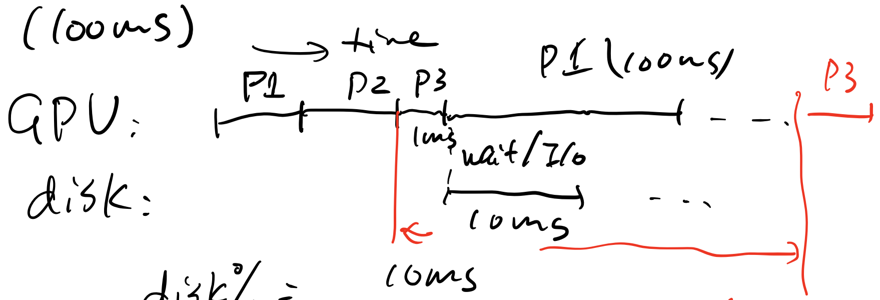
3 jobs
  P1, P2: both CPU bound, run for a week
  P3: I/O bound, loop
    (1 ms of CPU, 10 ms of disk I/O)

process	arrival	running
P1	0	1 week
P2	0 +ε	1 week
P3	0 +2ε	30 sec (with 300sec I/O)

FIFO:

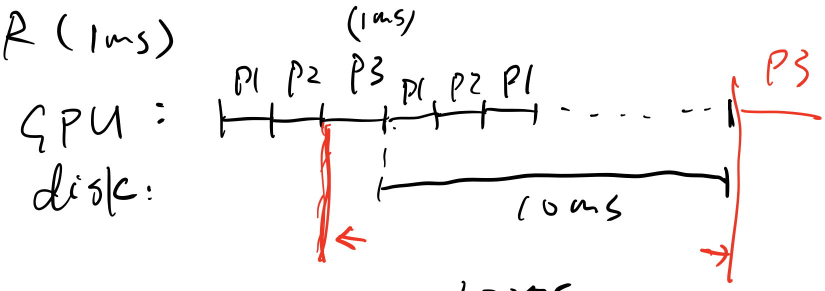


RR (100ms)



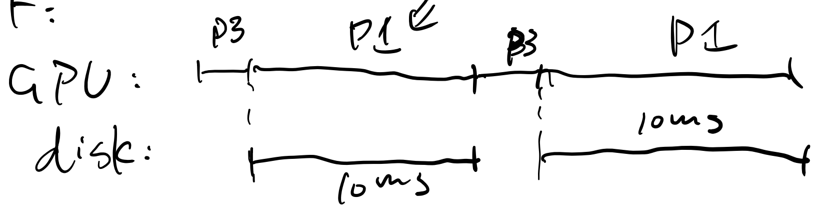
$$\text{disk\%} = \frac{10\text{ms}}{2\times 100\text{ms} + 1\text{ms}} \approx 5\%$$

RR (1ms)



$$\text{disk\%} = \frac{10\text{ms}}{11\text{ms}} \approx 90\%$$

STCF:



disk% ≈ 90%

- predict future

EWMA

$t_n \leftarrow$ nth running

$$\tau_{n+1} = \alpha \cdot t_n + (1-\alpha)\tau_n$$

$\alpha \in (0, 1]$

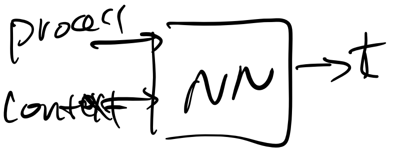
$\alpha = 1 : \quad \tau_{n+1} = t_n$

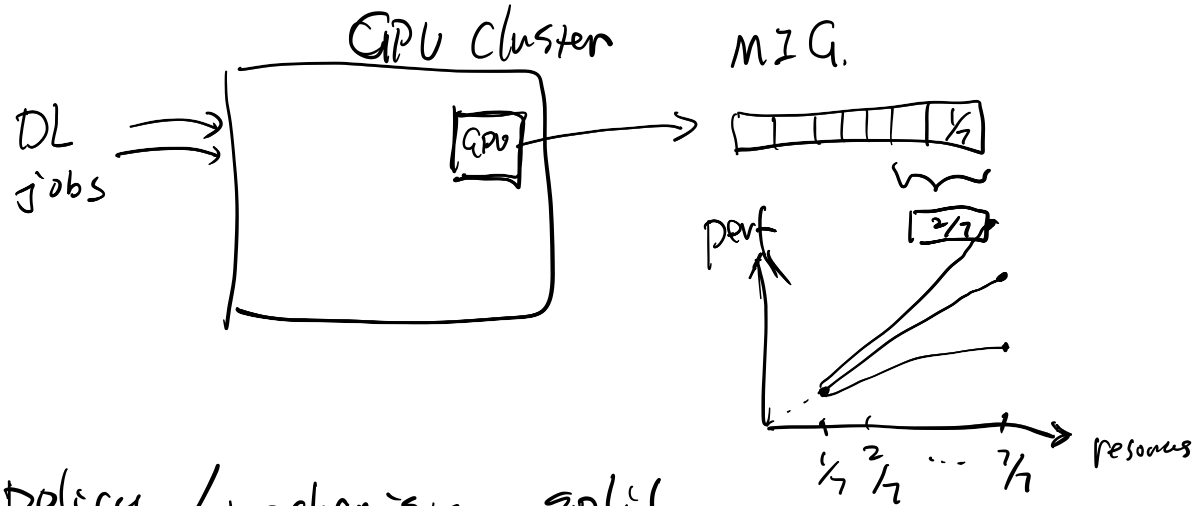
$\alpha = \frac{1}{2} : \quad \tau_{n+1} = \frac{1}{2} t_n + \frac{1}{2} \cdot \tau_n$

$\left(\frac{1}{2} t_{n-1} + \frac{1}{2} \tau_{n-1}\right)$

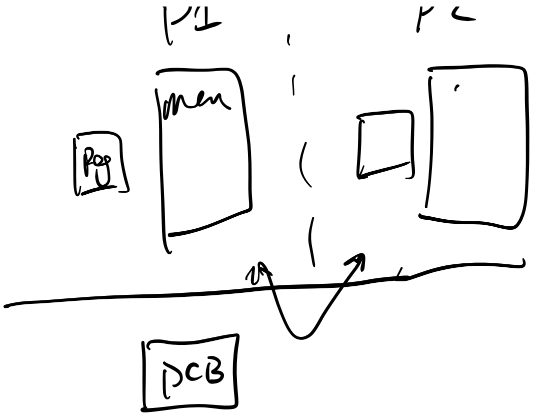
$= \frac{1}{2} t_n + \frac{1}{4} t_{n-1} + \frac{1}{4} \tau_{n-1}$

$+ \frac{1}{8} t_{n-2} + \frac{1}{8} \tau_{n-1}$



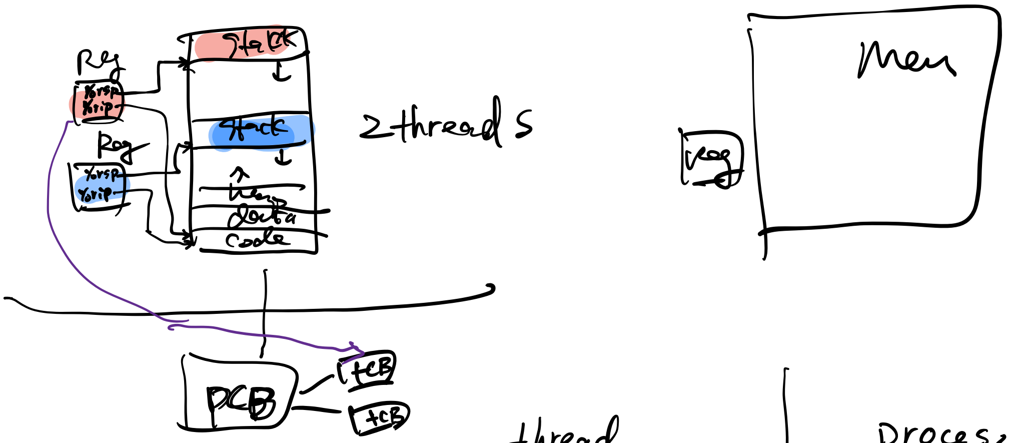


policy / mechanism split



multiple CPU

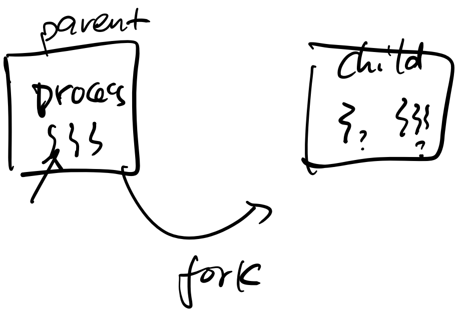
Thread ⟷ Proeess



thread	process
thread_create(void *fn, ...);	fork() ; ProcessCreate()
thread_exit ( );	exit()
thread-join (tid);	wait()

A. multiple threads

B. single thread ⇐


parent
process
child
fork