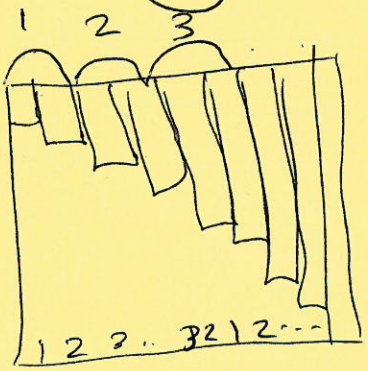
13 Nov 2019

model of computation : real-RAM
(in this class, typically)

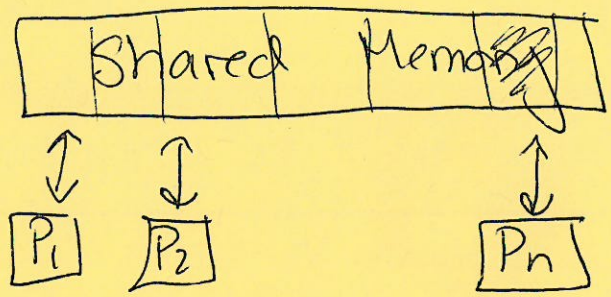
parallel MOC
↳ no universally-accepted model!

What comes into play for parallel programs?

1. CPU vs GPU
2. race conditions / deadlock / starvation
3. livelock
4. memory: shared or distributed
5. mutual exclusion
6. resource allocation
7. thread scheduling / # threads / # processors
8. communication protocal
9. static - vs- dynamic threading
10. load balancing

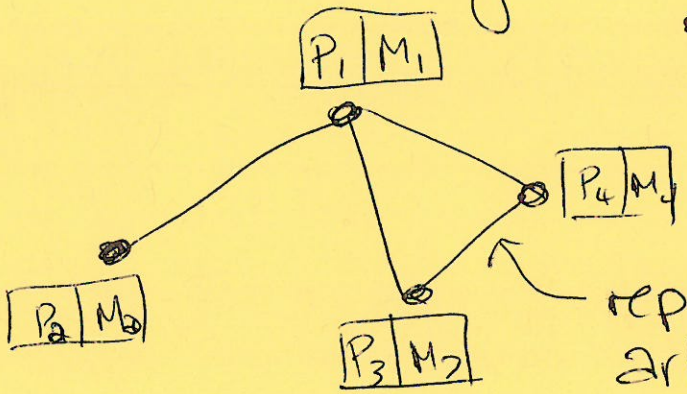


Looking at MOC - parallel:

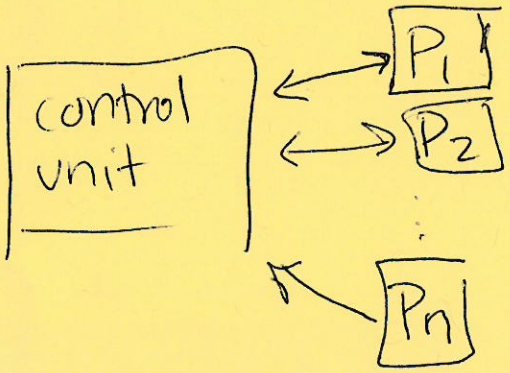
1. 

- locks and/or other communication
- design to work on different parts of memory

2. Distributed Memory



- distributed or replicated memory

3. 

- shared or distributed memory here too

Book / Class:

shared memory, dynamic multithreading on an ideal parallel computer

$F(0) = F(1) = 1$, $F(2) = F(1) + F(0)$, $F(k) = F(k-1) + F(k-2)$
$= 1 + 1 = 2$

SEQ-CODE:

```
FIB(n)
if n ≤ 1
    return 1
else
    X = FIB(n-1)
    Y = FIB(n-2)
    return X + Y
end if
```

```
PARALLEL-FIB(n)
if n ≤ 1                          } ①
    return 1
else
    X = SPAWN FIB(n-1)            } ②
    Y = FIB(n-2)                  } ③
    SYNC
    return X + Y
end if
```

Exercise: $n=3$ recursion tree
recurrence relation

$T(n) = T(n-1) + T(n-2) + \Theta(1)$
$= \Theta(\phi^n)$

$T(n) \leq 2T(n-1) + \Theta(1)$

$T(n) \geq 2T(n-2) + \Theta(1)$

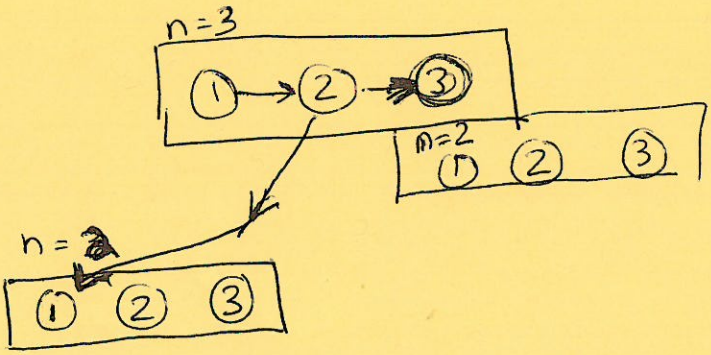
note: ϕ is the golden ratio
$\phi = (1+\sqrt{5})/2$

Note: DP is better...

TASKS:

① What is the recurrence relation now?

② Draw DAG for $n=3$.



Q: How long is the longest path? (asympt.)

new recurrence relation?

$$T(n) = \max\{T(n-1), T(n-2)\} + \Theta(1)$$

$$= T(n-1) + \Theta(1) = \Theta(n)$$

T_1 = "work" = amount of time on one processor
= sequential runtime

T_∞ = "span" = length of the path in DAG, assuming we have as many processors as needed

T_P = time on P processors.

Work Law: $T_P \geq T_1$ ← also true!

$$\downarrow$$

$$T_P \geq T_1 / P$$

Span Law: $T_P \geq T_\infty$

{QUESTION: Parallel -vs- Concurrent!}

Keywords for pseudocode:

- SPAWN = start a new thread for this operation
- SYNC = wait until threads are all done.