Parallel Algorithms
So far we have assumed the following model: RAM model

| Program | Memory with size n |

unit time ops

\( +, \cdot, \div, (\text{log}_n \text{ bits}) \)

(read/write into memory)

A central model to describe Graph Algorithms.

Other models:
1) Agents/ants
2) Pointer machines
RAM is unrealistic as n goes to infinity.

1) Speed of light (large size machines)

2) Quantum effects (small size machines)

Bottom line: RAM

1) Many important algorithms were found using this model.

2) Most algorithms are coded in a RAM like language.

eg C
Parallel Models

Fixed connection machines
machines = infinite state machine
RAM

A) Cellular Arrays 1D, 2D, 3D

1940's von Neumann
60's, 70's algorithms for CA.
Alvy Ray Smith 1974
Highly connected models

1) Hypercube $\equiv (V, E)$ 1980's

$V = \{(a_1, \ldots, a_m) \mid a_i \in \{0, 1\}\}$  $m = \log n$

$(a_1, \ldots, a_m), (a_1, \ldots, \bar{a}_i, \ldots, a_m) \in E$

2) Shuffle-exchange graph 1980's

$V = \{(a, \ldots, a_m) \mid a_i \in \{0, 1\}\}$

$((a_1, \ldots, a_m), (\bar{a}, a_2, \ldots, a_m)) \in E$

$((a_1, \ldots, a_m), (a_m a_1, \ldots, a_{m-1})) \in E$

3) Randomly connected graphs possible models of the brain!
Shared memory models

1) PRAM (Parallel Random Access Machine)

- Processors

Unit time ops:
+ , x , \div
read/write

ER    EW
CR    CW
PRAM issues:

Penalty for CR on an ER machine?

1) Machine crashes!
2) Garbage read!

How is synchronization handled?

1) After each unit of time!
2) None!
Circuit Model
Inputs in either bits or words

node

1) Constant fan in.
2) Arbitrary fan out.

nodes: \(\land, \lor, \neg\) gates

1) \(\land, \lor, \neg\) gates
2) Arithmetic ops

\[\text{Work} = \# \text{ nodes}\]
\[\text{Time} = \text{longest path from input to output}\]
Naive Matrix Multiply

in the Circuit Model

\[ C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj} \quad \text{and} \quad C = A \cdot B \]

Input

\[ A_{nn} \rightarrow A_{nm} \quad B_{nn} \rightarrow B_{mn} \quad 2n^2 \]

\[ N^3 \text{ nodes} \]

\[ \text{Depth} \quad \text{size} \quad \log n \quad 2n \]

\[ C_{ii} \rightarrow C_{ij} \rightarrow C_{mn} \]
Totals: Work: $O(n^3)$
Time: $O(\log n)$

Naive MM on PRAM

$P = \#\text{ processor}$ \hspace{1cm} $T = \text{Parallel time}$

1) CREW $P = O(n^3)$ \hspace{1cm} One processor/node
$T = O(\log n)$

Note: fan out = reads
fan in = writes

2) CREW $P = O(n^3/\log n)$
$T = O(\log n)$

Each process does the work of $\log n$ virtual process.

3) EREW $P = n^3/\log n$ \hspace{1cm} $T = O(\log n)$
**Def** Work = PT

One must pay for each process for life of the run.

**Naive MM:** $O(n^3)$ work, $O(n^2 \log n)$ time

Claim: If we have $P < n^3/\log n$ processes, then the time $= W/P$.

---

**Strassen's Alg**

Recall: recurrence

$$MM(n) = 7 \cdot MM(n/2) + O(n^2)$$

$\uparrow$

7 recursive cells

$\downarrow$

matrix additions

$A + B$
Matrix addition

$O(n^2)$ work

$O(1)$ time

Time:

$T(n) = T\left(\frac{n}{2}\right) + O(1)$

$\leftarrow$ parallel calls

$O(\log n)$

Work:

$w(n) = 7w\left(\frac{n}{2}\right) + Ch^2$

$\leftarrow$ we must pay for each call!

$O(n^{2.81})$
All-Prefix-Sums / Prescan

Let $\oplus$ be an associative binary op

\[ a_0, a_1, \ldots, a_n \]

**Def. All-Prefix-Sums**

**input:** $[a_0, a_1, \ldots, a_{n-1}]$

**output:** $[a_0, a_0 \oplus a_1, a_0 \oplus a_1 \oplus a_2, \ldots, a_0 \oplus \cdots \oplus a_{n-1}]$

**Prescan**

**output:** $[a_0, a_0 \oplus a_1, \ldots, a_0 \oplus \cdots \oplus a_{n-2}]$
Prescan

Input: \((3, 1, 7, 0, 4, 1, 6, 3)\) (addition)

Alg
1) Compute tree of partial sums
2) set root to zero
3) Down!

3) Down:
   a) Right-Child \(\leftarrow\) Parent \(\oplus\) Left-Child
   b) Left-Child \(\leftarrow\) Parent
$T(n) = O(n \log n)$

$W(n) = O(n)$?
List Ranking

Input: linked list
Output: a mark on each node s.t.
mark = distance from head or
= distance to tail

Head
0 ———> 1 ———> 2 ———> 3 ———> 4
Tail
4 3 2 1 0

Assume:
1) pointers are in consecutive memory
2) we know location of head & tail
3) pointers in arbitrary order.
Wyllie's Alg

Inll rank(!) := 1 ; rank(tail) := 0
Inll while succ(head) ≠ nil do
    if succ(!) ≠ nil do
        rank(!) := rank(!) ∪ rank(succ(!))
        succ(!) := succ(succ(!))
    fi
fi

#$processors = n$

Time = O(\log n)
Work = O(n \log n)
CREW model

Goal

O(\log n)
O(n)
Wyllie's Alg

Initial

Rank

After one round

After two rounds

H

nil

H

nil

H

nil
Random-Mate

1) Contraction Phase

1) Each live node randomly picks a sex

2) If $F^a \rightarrow M^b \rightarrow X$ then $a + b$

$$
\begin{align*}
F & \rightarrow M \\
M & \rightarrow X
\end{align*}
$$
dies

3) Stop when head points to nil. (only head is live)
Thm: The contraction phase stops in $C \log n$ rounds with high prob.

Let $P_i$ = Event that node $i$ is still live after one round.

Note: node $i$ not head then $\text{Prob}[P_i] = \frac{3}{4}$

Let $P_i^K$ = Event that node $i$ still live after $K$ rounds.

Note: $\text{Prob}[P_i^K] = \left(\frac{3}{4}\right)^K$ if not head.

Set $K = C \log_{4/3} n$.

$$
\text{Prob}[P_i^K] = \left(\frac{3}{4}\right)^K \leq \left(\frac{4}{3}\right)^{-C \log_{4/3} n} = \frac{1}{n^C}
$$
Let $P^k = \text{Event that some non-head node is still live.}$

Assume that node $0$ is the head.

$$P^k = P_1^k U P_2^k U \cdots U P_n^k$$

$$\text{Prob} [P^k] = \text{Prob} \left[ P_1^k U \cdots U P_n^k \right]$$

$$\leq \text{Prob} [P_1^k] + \cdots + \text{Prob} [P_n^k]$$

$$\leq n \cdot \frac{1}{n^c} = \frac{1}{n^{c-1}}$$

If we set $c=2$ then the contraction phase stops with prob $\leq \frac{1}{n}$. 

In the expansion phase we run contraction phase "backwards".

\[ \text{live} \quad \text{dead} \quad \text{live} \]
\[ \text{dist} = d \]

\[ \text{live} \quad \text{live} \quad \text{live} \]
\[ \text{dist} = b + c \]