Parallel Algorithms

So far we have assumed the following model: RAM model

A central model to describe Graph Algorithms.

Other models:

1) 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 = finite state machine
≈ RAM

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

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

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

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

$m = \log n$

$(a_i, \neg a_m), (a_i, \neg \overline{a_i}, \neg a_m) \in E$

2) Shuffle-exchange graph 1980's

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

$((a_1, \ldots, a_m), (\overline{a_i}, a_2, \ldots, a_m)) \in E$

$((a_1, \ldots, a_m), (a_m a_1, \ldots, a_m)) \in E$

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

1) PRAM (Parallel Random Access Machine)

Processors

Unit time ops

Read

Write

Exclusive Concurrent

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

\[ \text{node} \] \rightarrow \text{nodes: } \land, \lor, \neg \text{ gates}

1) Constant fan in
2) Arbitrary fan out

\text{DAG} \quad \text{output}

\text{Work} = \# \text{nodes}

\text{Time} = \text{longest path from input to output}
Naive Matrix Multiply

in the Circuit Model

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

input \( A_{nn} \ldots A_{mn} \quad B_{nn} \ldots B_{mn} \quad 2n^2 \)

\( A_{nn} \times B_{nn} \)

\( A_{mn} \times B_{mn} \)

\( A_{in} B_{nj} \)

\( C_{11} \ldots C_{ij} \ldots C_{mn} \)

\( n^3 \) nodes

depth size\( \log n \) \( 2n \)
Totals: Work: $O(n^3)$
        Time: $O(\log n)$

Naive MM on PRAM

$P = \# \text{processor} \quad T = \text{Parallel time}$

1) CREW  $P = O(n^3)$  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  $P = n^3/\log n \quad T = O(\log n)$
Def Work = P x T

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

Naive MM: $O(n^3)$ work $O(n^3)$ time

Claim: If we have $P < \frac{n^3}{\log n}$ processes
then the time $\approx \frac{W}{P}$.

Strassen's Alg

Recall: recurrence

$$MM(n) = 7 \cdot MM\left(\frac{n}{2}\right) + Cn^2$$

7 recurrent
cells

matrix
additions

$A + B$
Note: Matrix addition
\[ O(n^3) \text{ work} \]
\[ O(1) \text{ time} \]

Time: \[ T(n) = T(n^{1/2}) + O(1) \]
parallel calls

\[ O(\log n) \]

Work: \[ w(n) = 7w(n^{1/2}) + Cn^2 \]
we must pay for each call!

\[ O(n^{3/8}) \ldots \]
All-Prefix-Sums / Prescan

Let \( \oplus \) be an associative binary op

e.g. \( + \), \( \cdot \)

**Def:** All-Prefix-Sums

- **input:** \([a_0, \ldots, a_{n-1}]\)
- **output:** \([a_0, a_0 \oplus a, a_0 \oplus a \oplus a_2, \ldots, a_0 \oplus \cdots \oplus a_{n-1}]\)

**Prescan**

- **output:** \([I, a_0, a_0 \oplus a_1, \ldots, a_0 \oplus \cdots \oplus a_{n-2}]\)
### Application

#### Packing Memory

<table>
<thead>
<tr>
<th>location</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>contents</td>
<td>Ø</td>
<td>*</td>
<td>Ø</td>
<td>Ø</td>
<td>*</td>
<td>*</td>
<td>Ø</td>
<td>*</td>
</tr>
</tbody>
</table>

![Diagram](image)
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!

```
       250
       /|
     11 14
    /|
  4  7  5  9
 /|   /|
3 1 7 0 4 1 6 3
```

3) DOWN!
   a) Right-Child ← Parent ⊕ Left-Child
   b) Left-Child ← Parent
\[ T(n) = O(h_0 n) \]
\[ W(n) = O(n) ? \]
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

Init rank(!) := 1 : rank(tail) = 0
Init while succ(head) ≠ nil do
  if succ(!) ≠ nil do
    rank(!) := rank(!) ⊕ rank(succ(!))
    succ(!) := succ(succ(!))
  fi
fi

# processors = n
Time = O(Mn)
Work = O(nMn)
CREW model

Goal
O(Mn)
O(n)
Random-Mate

1) Contraction Phase

1) Each live node randomly picks a sex

2) If $F \xrightarrow{a} M \xrightarrow{b} X$ then

   $a + b$

   \[ F \quad M \quad D \quad X \]
   \[ \text{dies} \]

3) Stop when head points to nil.
   (only head is live)
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 that $P_i$ not head then $\text{Prob}[P_i] = \frac{3}{4}$

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

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

Set $K = c \log \frac{1}{\frac{1}{3}n}$

$$\text{Prob}[P_i^K] = \frac{1}{(\frac{4}{3})^K} \leq \frac{1}{(\frac{4}{3})^{c \log \frac{1}{\frac{1}{3}n}}} = \frac{1}{n^c}$$
Let $P^k$ be the event that some non-head node is still live.

Assume that node $o$ is the head.

$$P^k = P_1^k \cup P_2^k \cup \ldots \cup P_n^k$$

$$\text{Prob}[P^k] = \text{Prob}[P_1^k \cup \ldots \cup P_n^k]$$

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

$$\leq n \cdot \frac{1}{hc} = \frac{1}{n \leq 1}$$

If we set $c=2$, then the contraction phase stops with prob $\leq \frac{1}{n}$.
In the expansion phase we run the contraction phase "backwards".

\[ \text{live} \quad \rightarrow \quad \text{dead} \quad \rightarrow \quad \text{live} \]

\[ \text{dist} = d \]

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

\[ \text{live} \quad \rightarrow \quad \text{live} \quad \rightarrow \quad \text{live} \]