Day 20 3/30/98 "It's just better if I start over."

1 Parallel Algorithm Design

1.1 Sequential model (Random Access Memory)

Standard algorithm design has several unit time operations:

$$+, *, \div$$

(all with $log(n)$ bits)

Also memory read and writes occur in unit time.

1.2 Parallel RAM

Similar operations except it is necessary to worry about whether reads and writes are concurrent or exclusive.

$ExclusiveRead$	$ExclusiveWrite$
$ConcurrentRead$	$ConcurrentWrite$

1.3 Circuit model

Consider a circuit arranged as a DAG from inputs to outputs. Define:

$$work = \#gates$$

$$time = depth$$

1.4 example: naive matrix multiply

For a matrix multiply, the obvious method uses n^3 processors and has $O(log(n))$ depth indicating total work around $W = n^3 log(n)$

By using only $n^3/log(n)$ processors it is possible to make $W = n^3$ while increasing the time by a factor of 3.

1.5 example: strassen matrix multiply

adding matrices: $time = O(1)$, $work = n^2$, $\#processors = n^2$.

All recursive calls are done in parallel so:

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

$W(n) = 7W(\frac{n}{2}) + cn^2 = O(n^{2.81\cdots})$

1.6 example: all-prefix-sum

$\bigoplus$ = binary associative operator
input: $[a_0, ...a_{n-1}]$
output: $[a_0, a_0 \bigoplus a_1, ..., a_0 \bigoplus ... \bigoplus a_{n-1}]$
Change the desired output slightly to be:
prescan = $[I, a_0, a_0 \bigoplus a_1, ..., a_0 \bigoplus ... \bigoplus a_{n-2}]$

1.6.1 algorithm.

The prescan can be computed by adding pairs in a tree, then doing a 'shift and subtract' from the root towards the leaves.

1.6.2 cost

if $\#P = n$ time $= O(log(n))$ then $W(n) = O(n * log(n))$
if $\#P = n/\log(n)$ then the time is still $O(log(n))$ but $W(n) = O(n)$.

1.7 example: list-ranking

The goal is to associate a number with each node in a list = the nodes depth in the list. We want to do this in parallel.

Wiley's algorithm: At each stage, the node points to a neighbor and n = number of steps to reach neighbor. General step: grab the neighbor of your neighbor and add $n + neignbor(n)$. In $log(n)$ steps you are done. The algorithm is $EREW$.

1.7.1 Random-mate version

Assign a random bit to each node ('female' or 'male').

Rule: if 'female' points to 'male', have the 'female' move to the neighbor of the 'male'

Reassign random bits and continue.

Analysis Probability that a node is removed in a step = $\frac{1}{4}$.

With high probability the algorithm runs in $O(log(n))$.