

## Coordination Tools / Language / Paradigm

- messages
- Object-oriented
- shared data structures

We know about messages

We know about OO (Java RMI, etc)

We know about one data structures,  
shared memory (threads)

Problems to think about

DATABASE QUERY

FIND SHORTEST PATH

CALCULATE GCD (x, y)

SEARCH DNA string for pattern

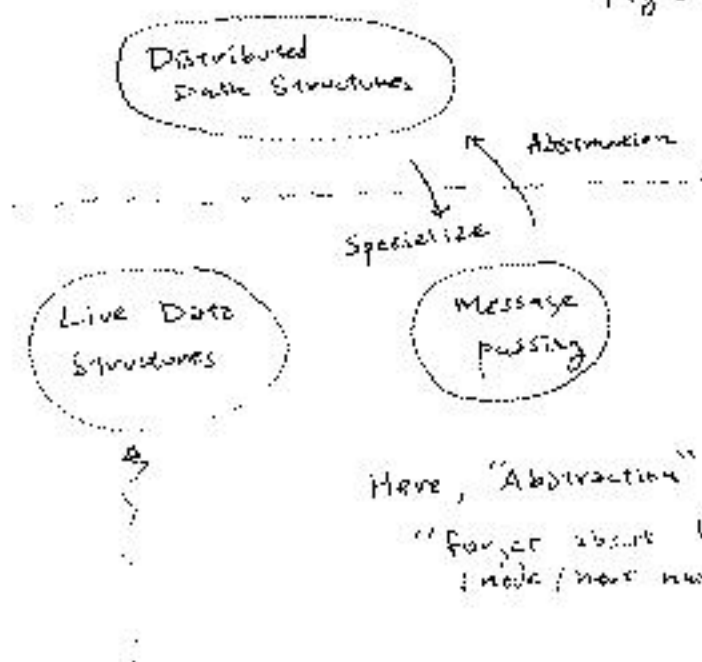
CENSUS OF POPULATION

For vectors  $a^1, a^2, a^3, \dots, a^k$

$$\text{compute } \sum_{i=1}^k x_i^j$$

Author suggests a different three coordination techniques

Fig 27 on p 32



Here, "Abstraction" means  
"forget about location  
{node / host number, etc}"

what: what?

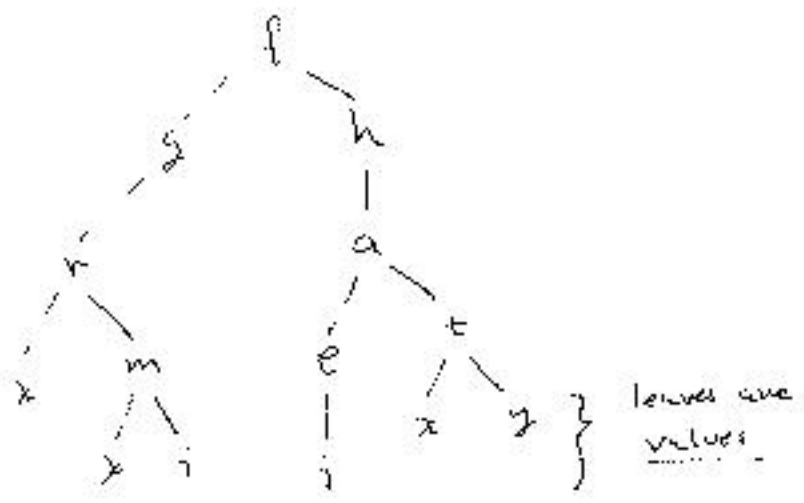
Suppose  
Program is

"compute  $f(g(x), h(y, x))$ "

where  $g(x) = v(x, m(x, i))$

$h(y, x) = a(e(i), t(x, y))$

∴ actually, program has structure like



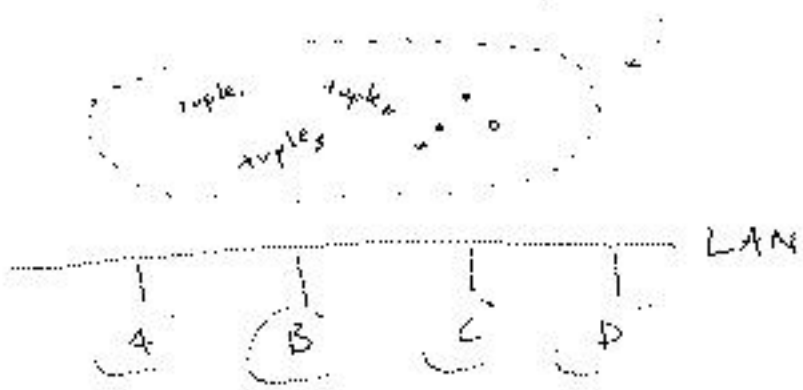
Idea of active structure is :

put a process / thread at each non-leaf in structure. whenever a process finished, it "turns into" a value (the result). Finally, the root (f) becomes a value.

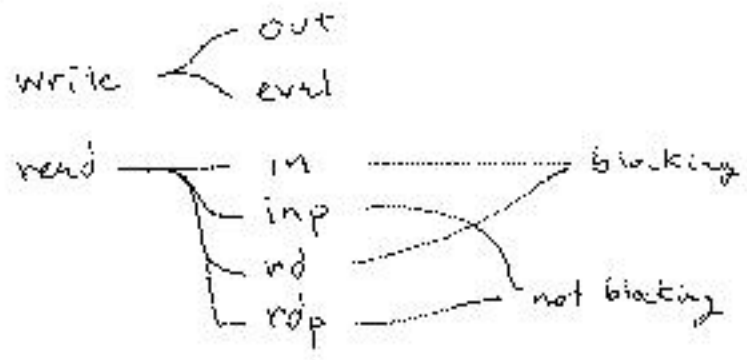
could have lots of parallel execution, but use many, many threads / processes.

Linda = one shared, distributed

data structure called the tuple space



Operations:



in, inp remove a matching tuple

out (tuple) → adds tuple to space

in (pattern) → finds a matching tuple, removes from space, returns tuple