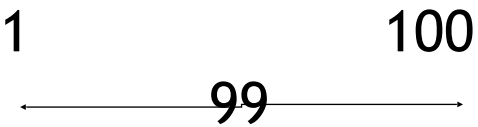
PRAM Algorithm

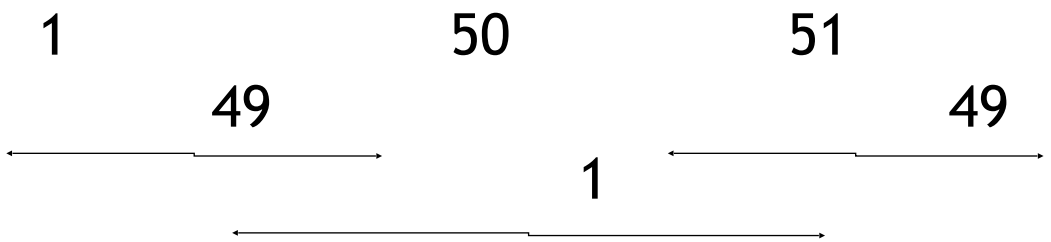
3.2

Introduction

- Let there be 100 numbers to be added
- How much time will a person take to add?



- How much time will two person take to add?



i.e. 50

Asymptotic speed up = 99/50
=1.98
= 2

Asymptotic speed up

- Asymptotic speed up = s(n)/T(n,p)

  Where
  n= number of inputs
  s(n) = sequential (run time of best known sequential algorithm)
  p= number of processors
  T(n,p) = parallel (time taken by p processors using parallel algorithm)

- Total work done O(s(n)) = p * T(n,p)

Efficiency Θ

- Θ = s(n)/(p*T(n,p))

  = s(n)/O(s(n))

- To be work optimal it requires the efficiency to be Θ(1)

Example of speed up, work done and efficiency

- Let a be A n-processor parallel algorithm that sorts n keys in $\Theta(\log n)$ time. Let B be n^2 processor algorithm that also sorts n-keys in $\Theta(\log n)$ time. Find out which algorithm is work optimal
- Assumption: optimal run time to sort n keys = $\Theta(n \log n)$

Speed up of A = = $\Theta(n)$

Speed up of B = = $\Theta(n)$

Work done of A = n. $\Theta(\log n)$
Work done of B = n^2. $\Theta(\log n)$

Example of speed up, work done and efficiency contd…

Efficiency of A = 1

Efficiency of B = 1/n

Since Efficiency of A is 1, we can say that algorithm A is work optimal.

Note: The criticality of any parallel algorithm depends on the calculation of processors and the minimization of idle time of these processors.

Prefix computation algorithm

- Let ∑ be the domain in which binary associative operator ($\oplus$) is defined.
  - Any operator is binary associative if for any of three elements x, y, z from
  - ∑(((x $\oplus$ y) $\oplus$ z = (x $\oplus$ (y $\oplus$ z))) [$\oplus$ can be +,-,/, max, min, avg]

E.g. ∑ = (5, 8, -2, 7, -11, 12)

$\oplus$ = minimum

Output = 5, 5, -2, -2, -11, -11

Prefix computation algorithm contd…

- Statement of prefix computation
  - Let ∑ be the set of input of n elements such that

    $x_1, x_2, \ldots x_n \ \varepsilon \ \sum$

  - The problem is to compute the n element from ∑

x_1, $x_1 \oplus x_2$, $x_1 \oplus x_2 \oplus x_3, \ldots, x_1 \oplus x_2 \oplus x_3 \oplus \ldots \oplus x_n$

These outputs are referred to as prefixes

Prefix computation algorithm contd…

E.g. Let n=8, p=8.

Let the input be Σ = 12, 3, 6, 8, 11, 4, 5, 7

Let $\oplus$ = Additive

Step 1:

processor p1 to p4 computes the prefix sum of 12, 3, 6, 8 to arrive at 12, 15, 21, 29

AND

processor p5 to p8 computes the prefix sum of 11, 4, 5, 7 to arrive at 11, 15, 20, 27

Prefix computation algorithm contd…

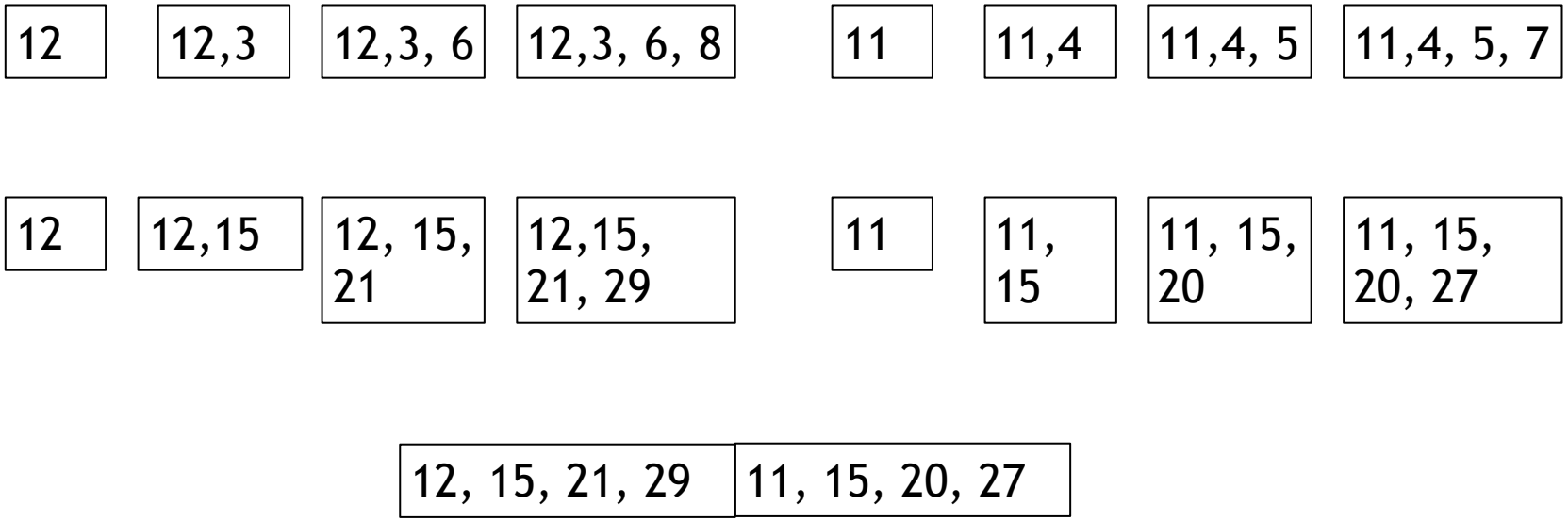
Step 2

processor p1 to p4 sits idle

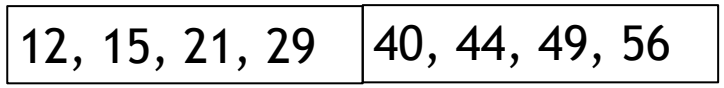
processor p5 to p8 will update their results by adding 29 to every prefix to obtain 40, 44, 49, 56

Prefix computation algorithm contd…

Step 1



Step 2



Prefix computation algorithm contd…

Analysis

Step 1 takes T(n/2)

Step 2 takes o(1)

Recurrence Relation is $T(n) = T(n/2) + O(1)$, $T(1) = 1$

Therefore $T(n) = T(n,p) = O(\log n)$

Task: Calculate Efficiency to see whether it is work optimal or not?