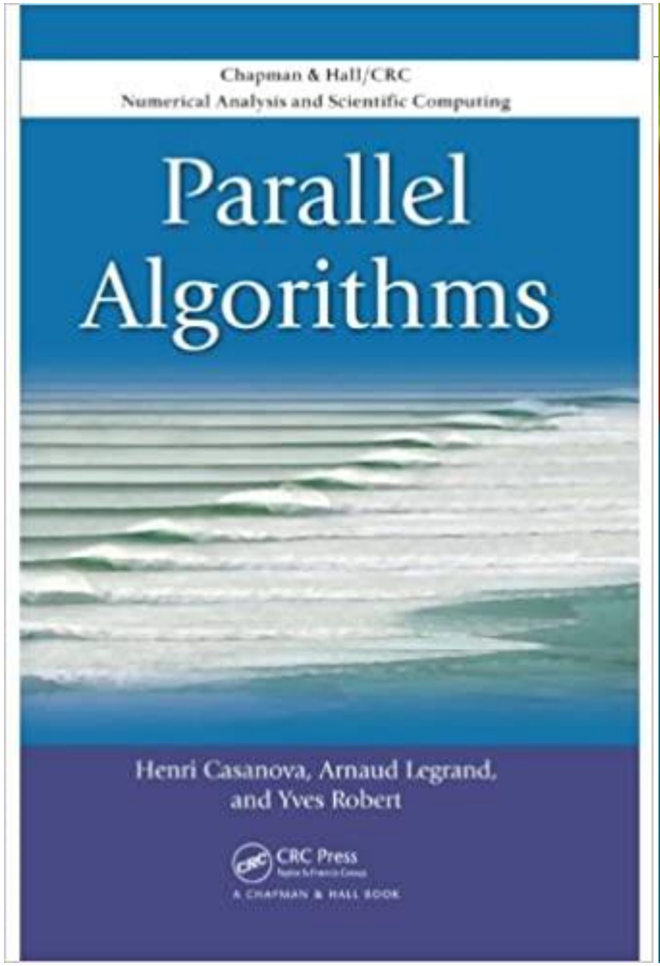
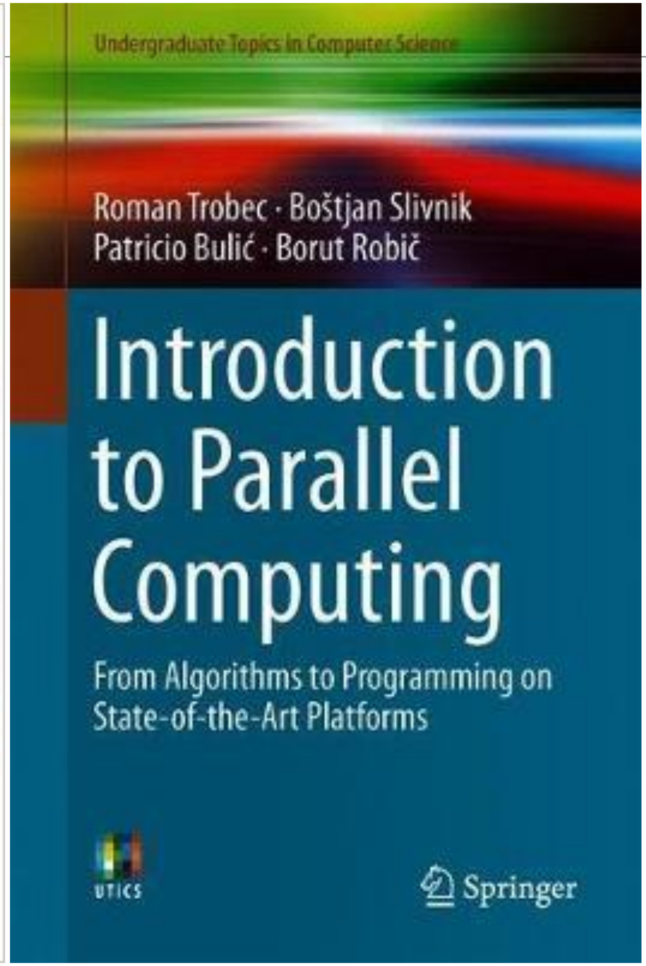
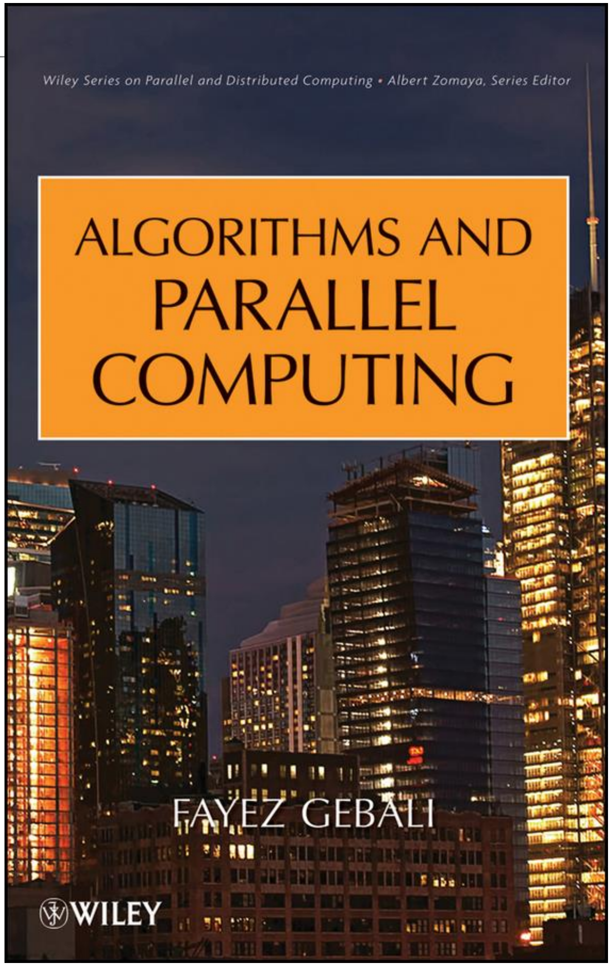
Parallel Programming

Lec 2

Books

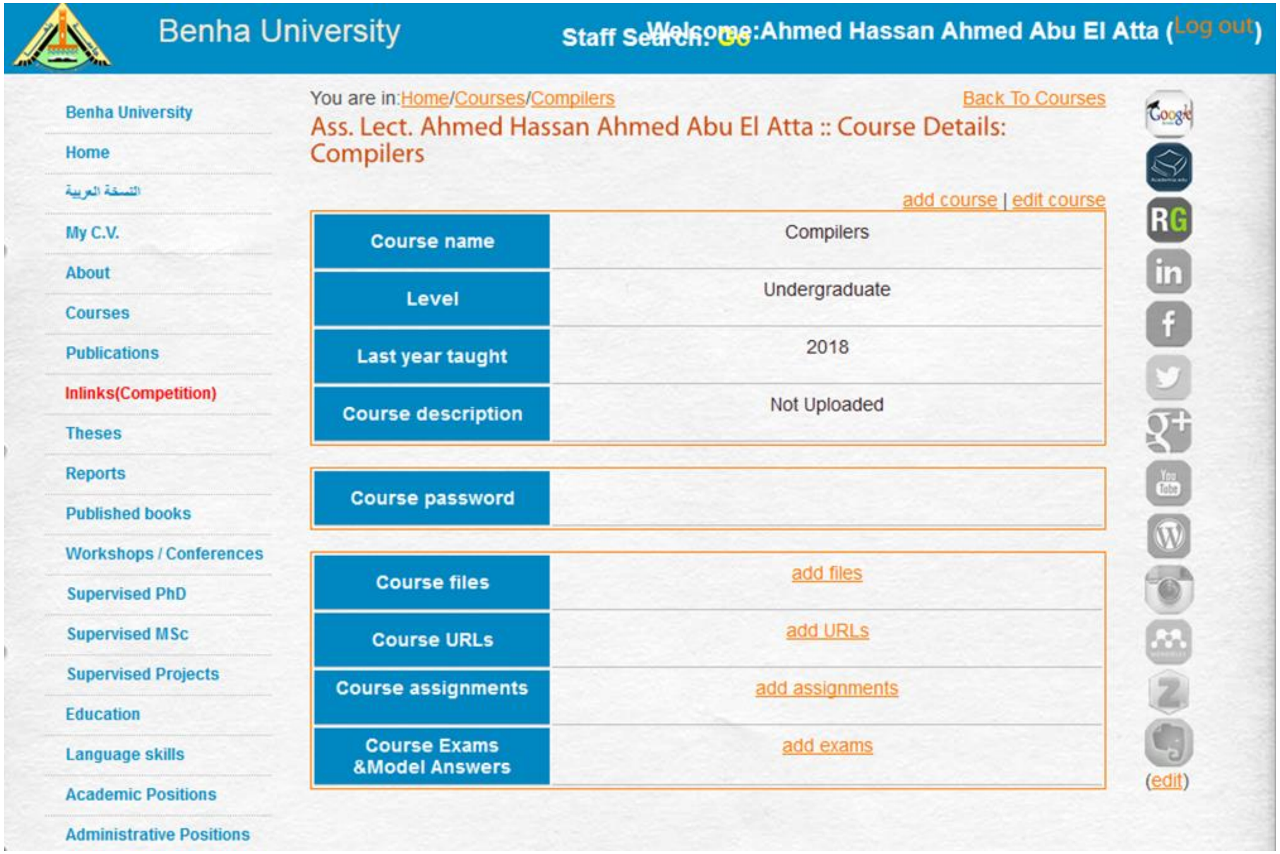

Chapman & Hall/CRC
Numerical Analysis and Scientific Computing
Parallel Algorithms
Henri Casanova, Arnaud Legrand, and Yves Robert
CRC Press
A CHAPMAN & HALL BOOK


Undergraduate Topics in Computer Science
Roman Trobec · Boštjan Slivnik
Patricio Bulić · Borut Robič
Introduction to Parallel Computing
From Algorithms to Programming on State-of-the-Art Platforms
UTiCS
Springer


Wiley Series on Parallel and Distributed Computing • Albert Zomaya, Series Editor
ALGORITHMS AND PARALLEL COMPUTING
FAYEZ GEBALI
WILEY

PowerPoint

http://www.bu.edu.eg/staff/ahmedaboalatah14-courses/14779



Compute the summation of an array of integer numbers

Suppose that we are given the problem P ≡ "add n given numbers."

Then "add numbers 1, 2, 3, 4, 5, 6, 7, 8" is an instance of size = n = 8 of the problem P.

Let us now focus on all instances of size 8, that is, instances of the form "add numbers $a_1;a_2;a_3;a_4;a_5;a_6;a_7;a_8$."

The fastest sequential algorithm for computing the sum

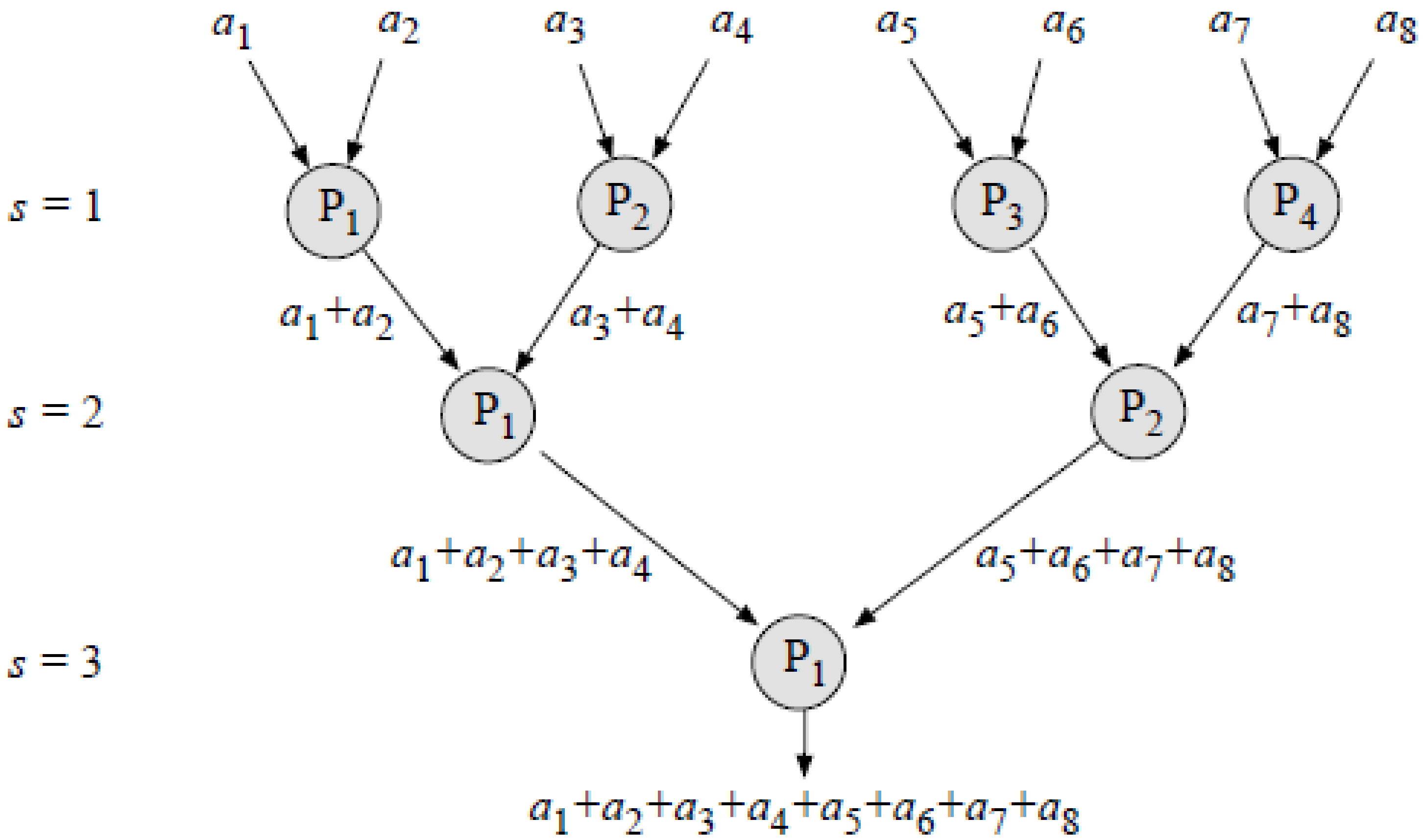
Sum = 0

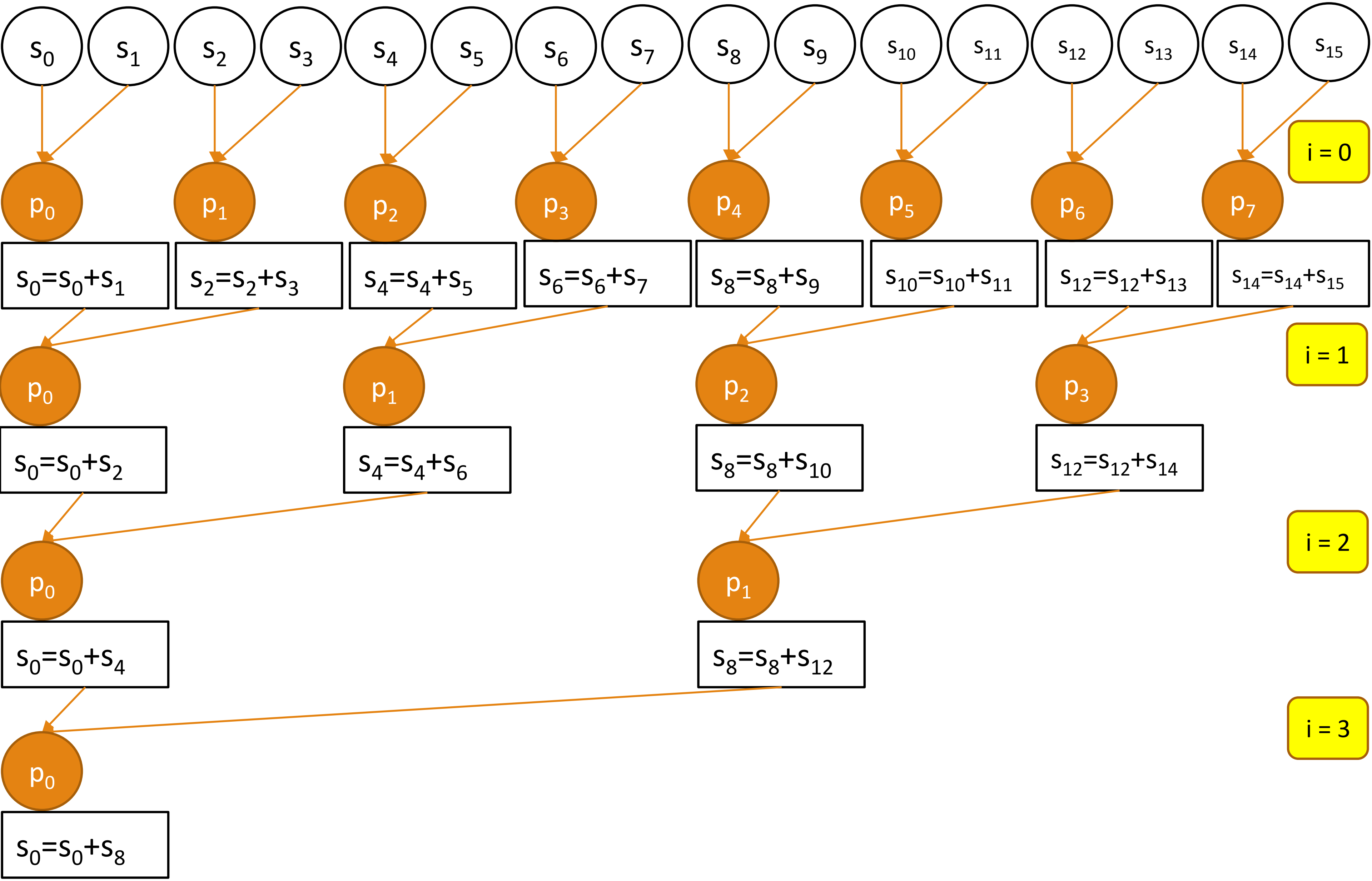
for (i = 1; i ≤ 8; i++)

    sum += a_i

requires $T_{seq}(8) = 7$ steps → $T_{seq}(n) = O(n)$

Compute the summation of an array of integer numbers





s0 s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 s14 s15
i = 0
p0 p1 p2 p3 p4 p5 p6 p7
s0=s0+s1
s2=s2+s3
s4=s4+s5
s6=s6+s7
s8=s8+s9
s10=s10+s11
s12=s12+s13
s14=s14+s15
i = 1
p0 p1 p2 p3
s0=s0+s2
s4=s4+s6
s8=s8+s10
s12=s12+s14
i = 2
p0 p1
s0=s0+s4
s8=s8+s12
i = 3
p0
s0=s0+s8

Compute the summation of an array of integer numbers

```
Sum = 0
For j = 0 to j < n/2 do parallel
    For i = 0 to i < 2 do
        s[j*2+i] = a[j*2+i]
For i = 0 to i<log(n) do
    For j = 0 to j <n/(2^(i+1)) do in parallel
        s[j * 2^(i+1)] += s[j * 2^(i+1)+2^i]
sum = s[0]
```

Compute the summation of an array of integer numbers

In general, instances of size n of P can be solved in parallel time $T_{par} = O(\log n)$

speedup is $S(n) = T_{seq}(n) / T_{par}(n) = O(n/\log n)$.

Cost $C(n) = n*O(\log n) = O(n\log n)$

$E(n) = T_{seq}(n) / C(n) = O(n / (n\log n)) = O(1/\log n) < 1$

Reducing the Processors Number to Reach to More Efficient Parallel Algorithm

$E_p(n) = C_s(n)/C_p(n) = 1$

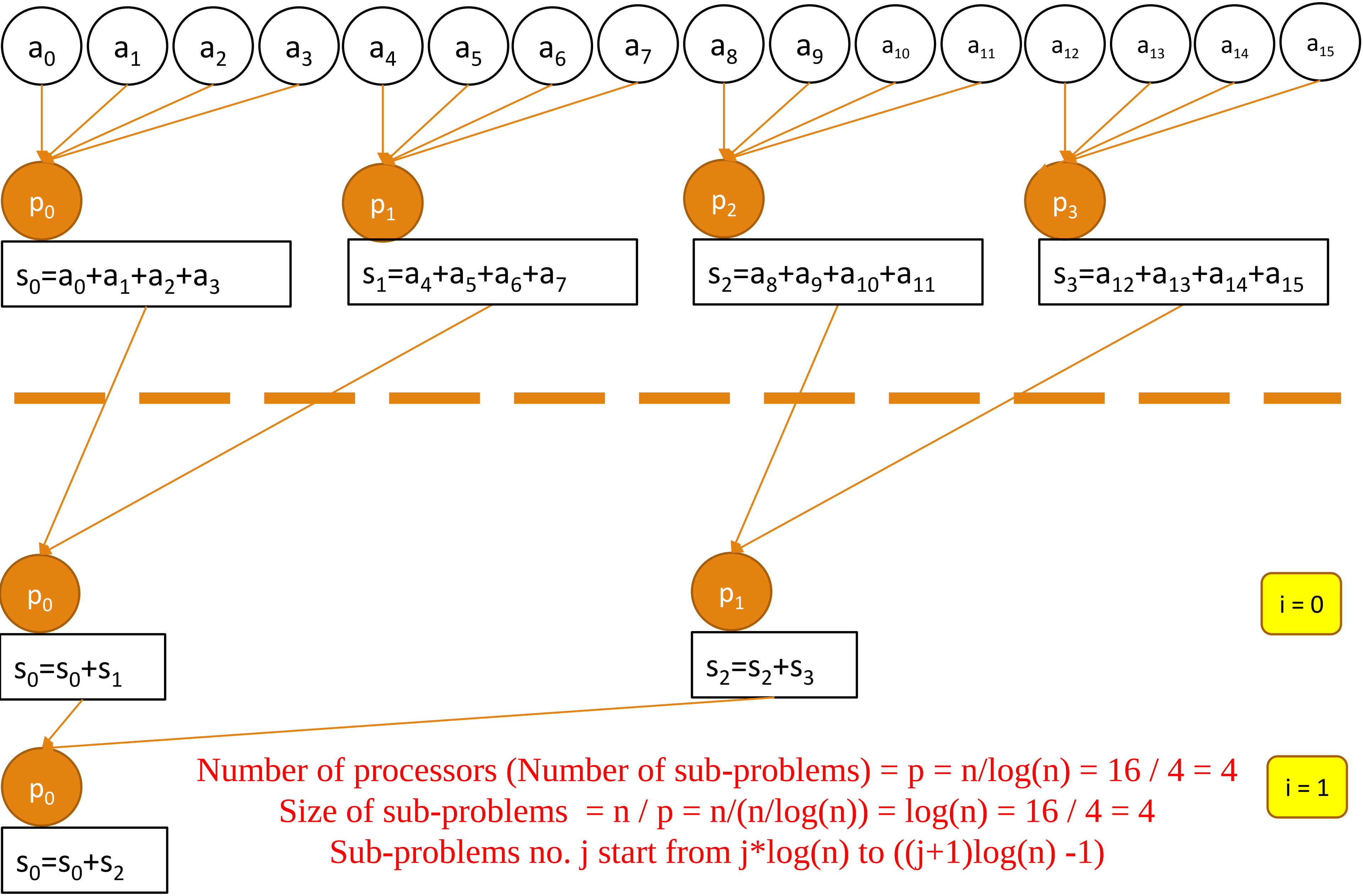
$C_s(n)/C_p(n) = 1 \quad \rightarrow \quad C_s(n) = C_p(n)$

$1*T_s(n) = p*T_p(n) \quad \rightarrow \quad p = T_s(n)/T_p(n)$

In the summation problem:

$p = T_s(n)/T_p(n) = n/\log(n)$

$$p = n/\log(n)$$



Number of processors (Number of sub-problems) = p = n/log(n) = 16 / 4 = 4

Size of sub-problems = n / p = n/(n/log(n)) = log(n) = 16 / 4 = 4

Sub-problems no. j start from j*log(n) to ((j+1)log(n) -1)

More Efficient Algorithm

```
Sum = 0
For j = 0 to j < n/log(n) do in parallel
        s[j] = 0
        For i = j*log(n) to i < ((j+1)*log(n)) do
                s[j] += a[i]
        End For
End par
For i = 0 to i< log(n/log(n)) do
        For j = 0 to j <n/(2^(i+1)) do in parallel
                s[j * 2^(i+1)] += s[j * 2^(i+1)+2^i]
        End par
End For
sum = s[0]
```

More Efficient Algorithm

In general, instances of size n of P can be solved in parallel time $T_{par} = O(\log n)$ with number of processors equals $p = n/\log(n)$

speedup is $S(n) = T_{seq}(n) / T_{par}(n) = O(n/\log n)$.

Cost $C(n) = (n/\log(n))*O(\log n) = O(n)$

$E(n) = T_{seq}(n) / C(n) = O(n / n) = 1$

Prefix Sums

Given the sequence of elements $X = \{x_0, x_1, x_2, \ldots, x_{n-1}\}$ and an associative operation $\oplus$, assume s_j is defined as:

$$S_j = x_0 \oplus x_1 \oplus x_2 \oplus \ldots \oplus x_j$$

The sums $S_j = x_0 \oplus x_1 \oplus x_2 \oplus \ldots \oplus x_j$ are called the prefix sums in which j =0,2, ... ,n-1. In other words, $S_j = S_{j-1} \oplus x_j$ for i = 1, ... ,n-1 with $S_0 = X_0$.

In general, the prefix sums problem is to compute the n quantities with the following properties:

$$S_1 = x_1$$
$$S_2 = x_1 \oplus x_2$$
$$S_3 = x_1 \oplus x_2 \oplus x_3$$
$$\ldots$$
$$S_n = x_1 \oplus x_2 \oplus x_3 \oplus \ldots \oplus x_n$$

Sequential Prefix Sums

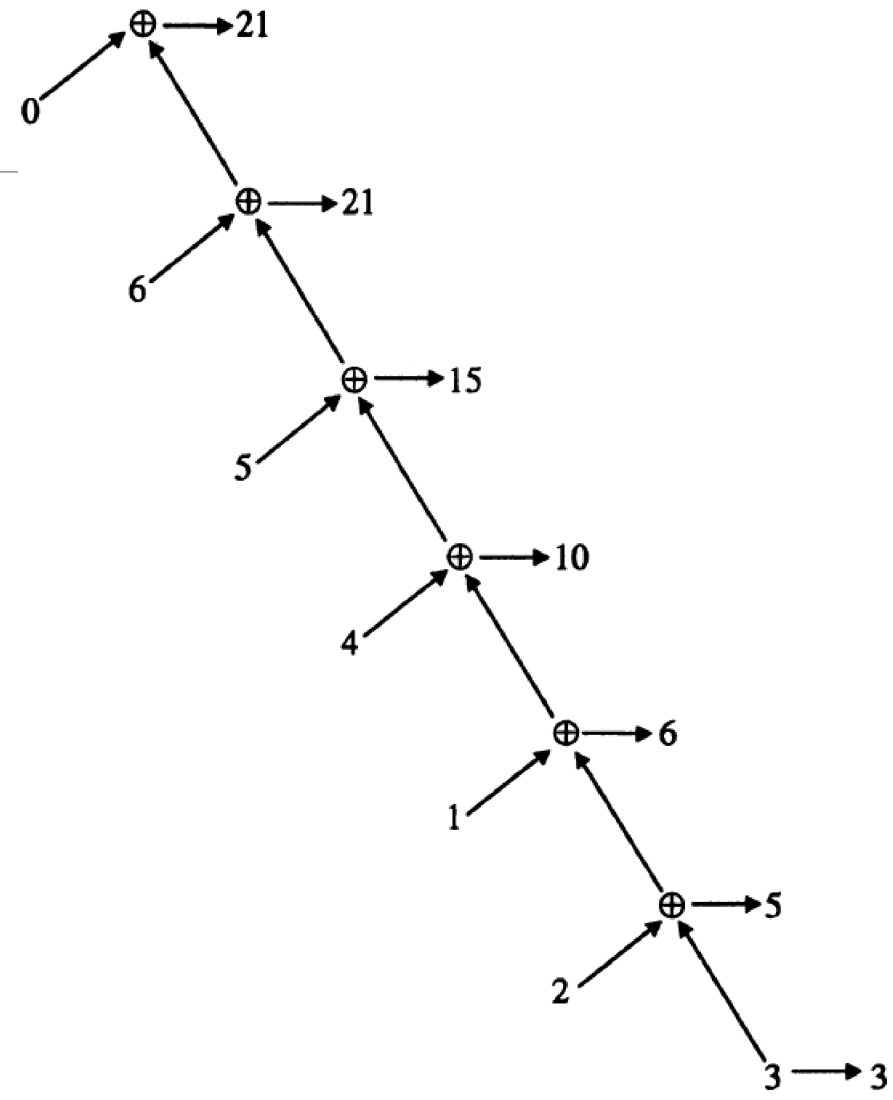
Example:

Given the operation + and the set

X = {3,2,1,4,5,6,0} the prefix sums of X will be

S = {3,5,6,10,15,21,21} as illustrated in the figure.

This represents the process of carrying out 7 additions.



Sequential Prefix Sums

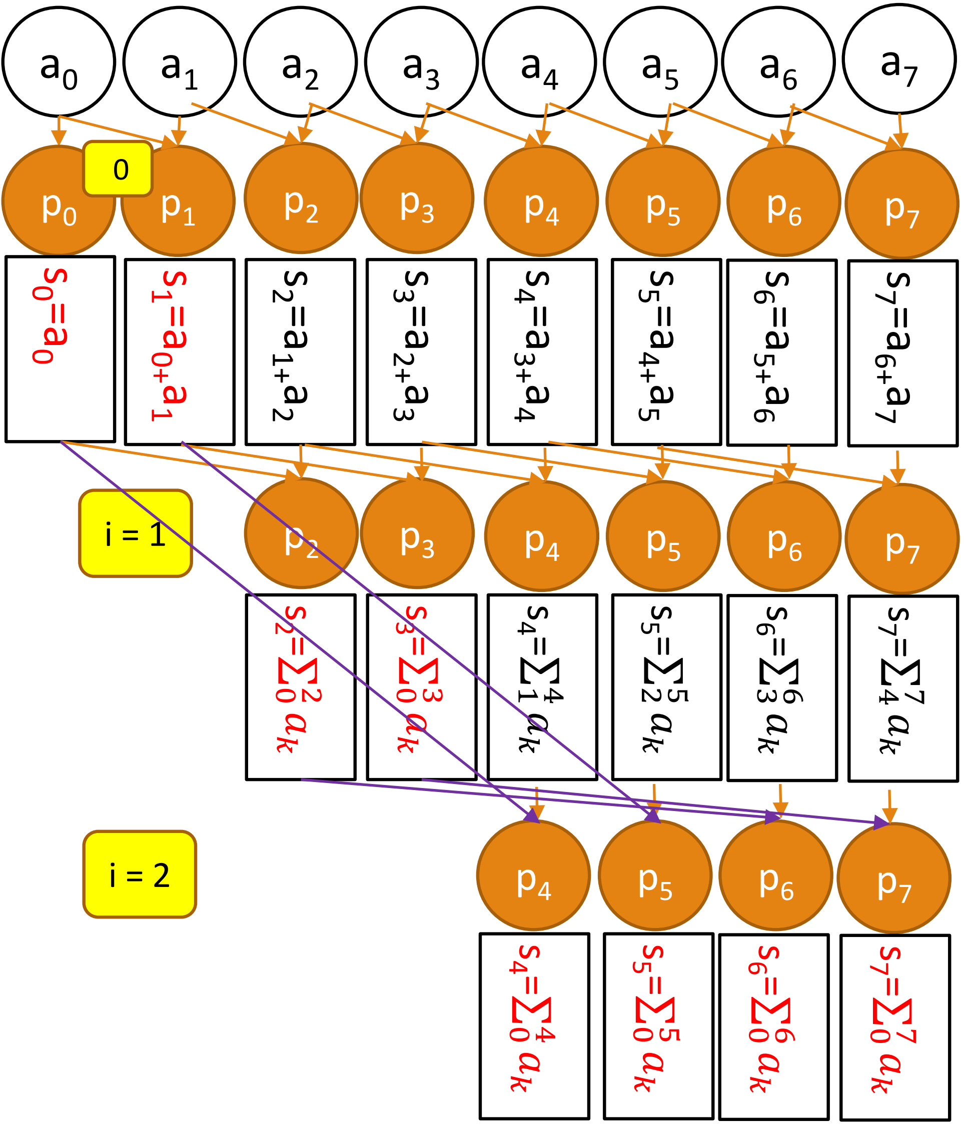
$s_0 = a_0$

For i = 1 to n-1 do

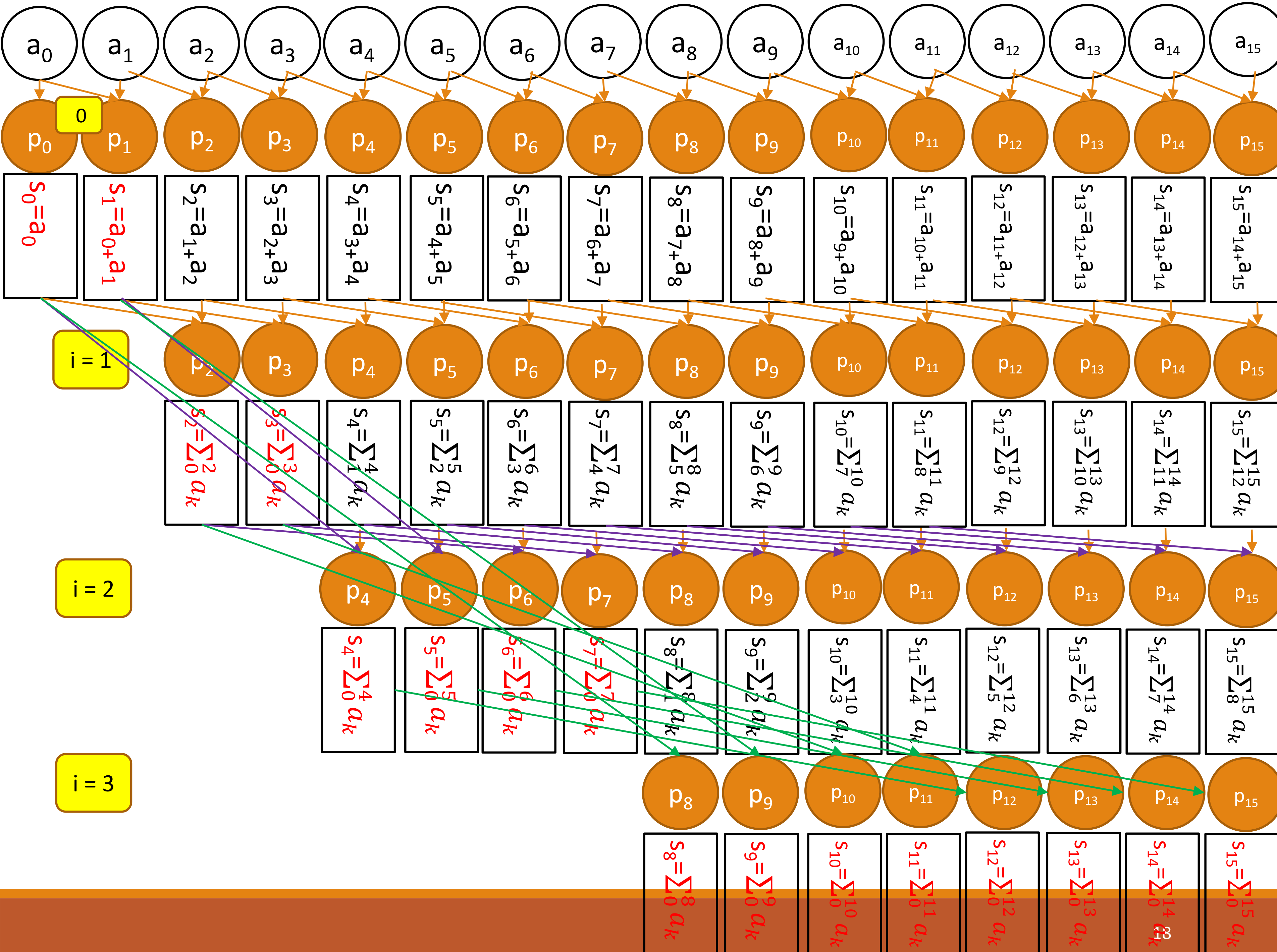
$s_i = s_{i-1} + a_i$

$T_s(n) = O(n)$

Parallel Prefix Sums Algorithm



a0 a1 a2 a3 a4 a5 a6 a7
0
p0 p1 p2 p3 p4 p5 p6 p7
s0=a0
s1=a0+a1
s2=a1+a2
s3=a2+a3
s4=a3+a4
s5=a4+a5
s6=a5+a6
s7=a6+a7
i = 1
p2 p3 p4 p5 p6 p7
$s_2=\sum_0^2 a_k$
$s_3=\sum_0^3 a_k$
$s_4=\sum_1^4 a_k$
$s_5=\sum_2^5 a_k$
$s_6=\sum_3^6 a_k$
$s_7=\sum_4^7 a_k$
i = 2
p4 p5 p6 p7
$s_4=\sum_0^4 a_k$
$s_5=\sum_0^5 a_k$
$s_6=\sum_0^6 a_k$
$s_7=\sum_0^7 a_k$



a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} a_{11} a_{12} a_{13} a_{14} a_{15}
0
p_0 p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 p_{10} p_{11} p_{12} p_{13} p_{14} p_{15}
$S_0=a_0$
$S_1=a_{0+}a_1$
$S_2=a_{1+}a_2$
$S_3=a_{2+}a_3$
$S_4=a_{3+}a_4$
$S_5=a_{4+}a_5$
$S_6=a_{5+}a_6$
$S_7=a_{6+}a_7$
$S_8=a_{7+}a_8$
$S_9=a_{8+}a_9$
$S_{10}=a_{9+}a_{10}$
$S_{11}=a_{10+}a_{11}$
$S_{12}=a_{11+}a_{12}$
$S_{13}=a_{12+}a_{13}$
$S_{14}=a_{13+}a_{14}$
$S_{15}=a_{14+}a_{15}$
i = 1
p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 p_{10} p_{11} p_{12} p_{13} p_{14} p_{15}
$S_2=\sum_0^2 a_k$
$S_3=\sum_0^3 a_k$
$S_4=\sum_1^4 a_k$
$S_5=\sum_2^5 a_k$
$S_6=\sum_3^6 a_k$
$S_7=\sum_4^7 a_k$
$S_8=\sum_5^8 a_k$
$S_9=\sum_6^9 a_k$
$S_{10}=\sum_7^{10} a_k$
$S_{11}=\sum_8^{11} a_k$
$S_{12}=\sum_9^{12} a_k$
$S_{13}=\sum_{10}^{13} a_k$
$S_{14}=\sum_{11}^{14} a_k$
$S_{15}=\sum_{12}^{15} a_k$
i = 2
p_4 p_5 p_6 p_7 p_8 p_9 p_{10} p_{11} p_{12} p_{13} p_{14} p_{15}
$S_4=\sum_0^4 a_k$
$S_5=\sum_0^5 a_k$
$S_6=\sum_0^6 a_k$
$S_7=\sum_0^7 a_k$
$S_8=\sum_1^8 a_k$
$S_9=\sum_2^9 a_k$
$S_{10}=\sum_3^{10} a_k$
$S_{11}=\sum_4^{11} a_k$
$S_{12}=\sum_5^{12} a_k$
$S_{13}=\sum_6^{13} a_k$
$S_{14}=\sum_7^{14} a_k$
$S_{15}=\sum_8^{15} a_k$
i = 3
p_8 p_9 p_{10} p_{11} p_{12} p_{13} p_{14} p_{15}
$S_8=\sum_0^8 a_k$
$S_9=\sum_0^9 a_k$
$S_{10}=\sum_0^{10} a_k$
$S_{11}=\sum_0^{11} a_k$
$S_{12}=\sum_0^{12} a_k$
$S_{13}=\sum_0^{13} a_k$
$S_{14}=\sum_0^{14} a_k$
$S_{15}=\sum_0^{15} a_k$

Parallel Prefix Sums Algorithm

```
For j = 0 to j < n do in parallel
 Pj:   s[j] = a[j]
End par


For i = 0 to i < log(n) do
     For j = 2^i to j < n do in parallel
       Pj:        s[j] = s[j-2^i ]+ s[j]
     End par
End For
```

Parallel Prefix Sums Algorithm

In general, instances of size n of P can be solved in parallel time $T_{par} = O(\log n)$

speedup is $S(n) = T_{seq}(n) / T_{par}(n) = O(n/\log n)$.

Cost $C(n) = n*O(\log n) = O(n\log n)$

$E(n) = T_{seq}(n) / C(n) = O(n / (n\log n)) = O(1/\log n) < 1$