Concurrent Programming

Session 11: Simple Problems and Tips

Computer Engineering Department
Iran University of Science and Technology
Tehran, Iran

Lecturer: Nima Ghaemian
Distributed Systems Lab.
Computer Engineering Department,
Iran University of Science and Technology,
nima@comp.iust.ac.ir

Reduction of an Array

A ***⊗-reduction*** of an array $x[1\,..\,n]$, where $\otimes$ is an associative operator, is the value

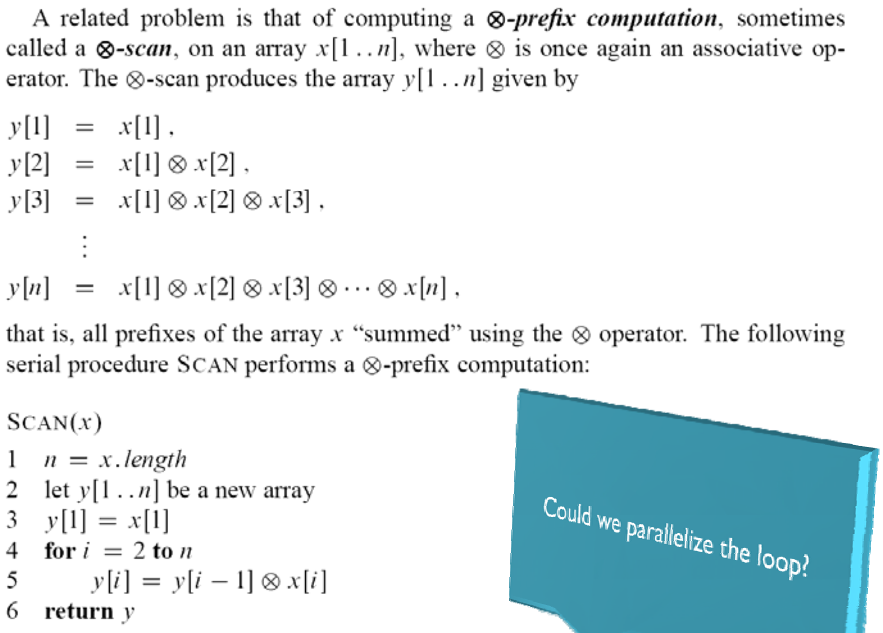
$$y = x[1] \otimes x[2] \otimes \cdots \otimes x[n]\ .$$

The following procedure computes the $\otimes$-reduction of a subarray $x[i\,..\,j]$ serially.

```
REDUCE(x, i, j)
y = x[i]
for k = i + 1 to j
    y = y ⊗ x[k]
return y
```

Use nested parallelism to implement a multithreaded algorithm P-REDUCE, which performs the same function with $\Theta(n)$ work and $\Theta(\lg n)$ span. Analyze your algorithm.

Prefix (Scan)



A related problem is that of computing a **⊗-*prefix computation***, sometimes called a **⊗-*scan***, on an array $x[1\,..\,n]$, where $\otimes$ is once again an associative operator. The $\otimes$-scan produces the array $y[1\,..\,n]$ given by

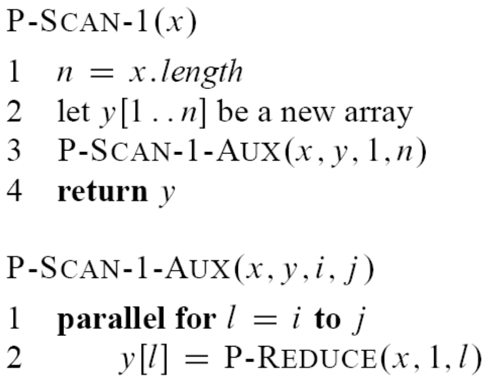
$$\begin{aligned} y[1] &= x[1]\,, \\ y[2] &= x[1] \otimes x[2]\,, \\ y[3] &= x[1] \otimes x[2] \otimes x[3]\,, \\ &\vdots \\ y[n] &= x[1] \otimes x[2] \otimes x[3] \otimes \cdots \otimes x[n]\,, \end{aligned}$$

that is, all prefixes of the array x "summed" using the $\otimes$ operator. The following serial procedure SCAN performs a $\otimes$-prefix computation:

```
SCAN($x$)
$n = x.length$
let $y[1\,..\,n]$ be a new array
$y[1] = x[1]$
for $i = 2$ to $n$
    $y[i] = y[i-1] \otimes x[i]$
return $y$
```

Could we parallelize the loop?

The First Solution



```
P-SCAN-1($x$)
$n = x.length$
let $y[1\,..\,n]$ be a new array
P-SCAN-1-AUX($x, y, 1, n$)
return $y$

P-SCAN-1-AUX($x, y, i, j$)
parallel for $l = i$ to $j$
    $y[l] =$ P-REDUCE($x, 1, l$)
```

Inefficient!

A Better Solution

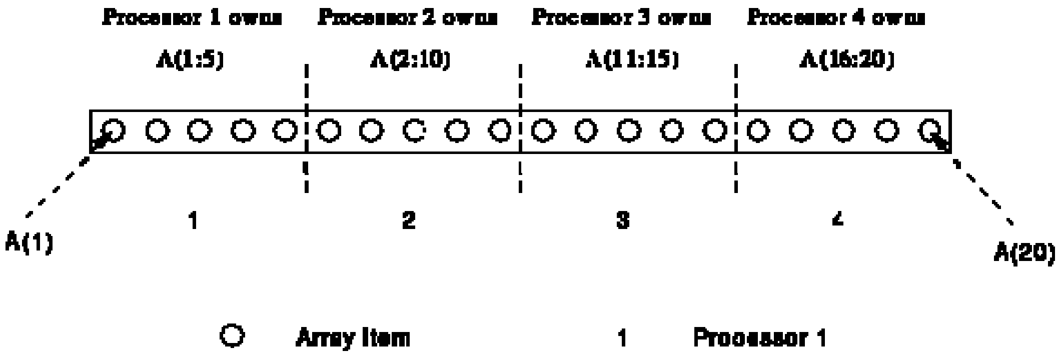
```
P-SCAN-2($x$)
$n = x.length$
let $y[1 \,..\, n]$ be a new array
P-SCAN-2-AUX($x, y, 1, n$)
return $y$

P-SCAN-2-AUX($x, y, i, j$)
if $i == j$
    $y[i] = x[i]$
else $k = \lfloor (i + j)/2 \rfloor$
    spawn P-SCAN-2-AUX($x, y, i, k$)
    P-SCAN-2-AUX($x, y, k + 1, j$)
    sync
    parallel for $l = k + 1$ to $j$
        $y[l] = y[k] \otimes y[l]$
```

Find a Better One!

Data Distributions

- Block Distribution

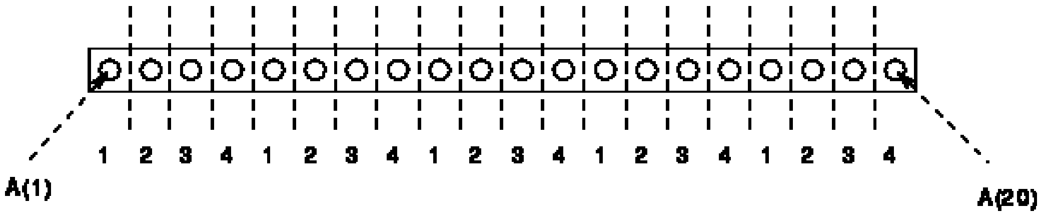


Data Distributions

- Cyclic Distribution

Processor 1 owns A(1::4)   Processor 3 owns A(3::4)

Processor 2 owns A(2::4)   Processor 4 owns A(4::4)



○ Array Item   1 Processor 1

Data Distributions

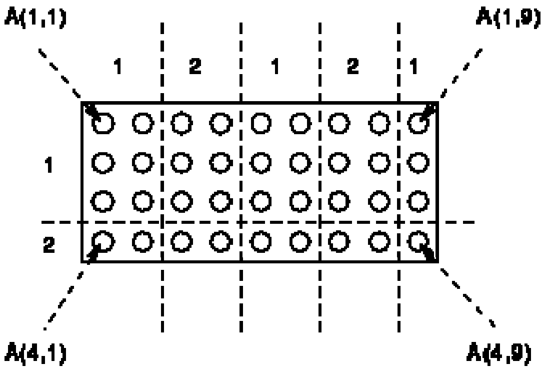
2D Distribution

Processor P(1,1) owns A(1:3,1:2), A(1:3,5:6) and A(1:3,9:9)

Processor P(2,1) owns A(4:4,1:2), A(4:4,5:6) and A(4:4,9:9)

Processor P(1,2) owns A(1:3,3:4) and A(1:3,7:8)

Processor P(2,2) owns A(4:4,3:4) and A(4:4,7:8)



○ Array Item

1 Processor 1