PARALLEL AND DISTRIBUTED ALGORITHMS

BY
DEBDEEP MUKHOPADHYAY
AND
ABHISHEK SOMANI

http://cse.iitkgp.ac.in/~debdeep/courses_iitkgp/PAlgo/index.htm

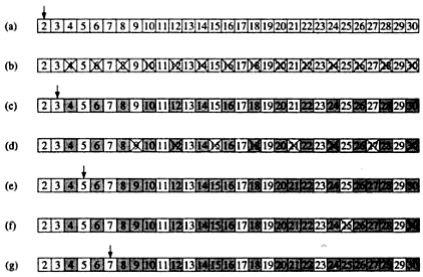
PARALLEL THINKING: THE SIEVE OF ERATOSTHENES

THE SIEVE OF ERATOSTHENES

Classic prime finding algorithm:

- Want to find the number of primes less than or equal to some positive integer n.
- A prime has exactly two factors: itself and one.
- *The Sieve of Eratosthenes begins with a list of natural numbers 2, 3, 4, …, n, and removes composite numbers from the list by striking multiples of 2, 3, 5, and successive primes. The sieve terminates after multiples of the largest prime less than or equal to $\sqrt{n}$ have been struck.*

a) Prime is next unmarked natural number- ie. 2
b) Strike all multiples of 2, starting with 2^2
c) Prime is next unmarked natural number- ie. 3
d) Strike all multiples of 3, starting with 3^2
e) Prime is next unmarked natural number- ie. 5
f) Strike all multiples of 5, starting with 5^2
g) Prime is next unmarked natural number- ie. 7. Since 7^2 is greater than n=30, algorithm terminates. All unmarked natural numbers are also prime.



3

FEW POINTS ON THE SEQUENTIAL ALGORITHM

The Sieve of Eratosthenes is impractical for testing primality of numbers with hundreds of digits.

- The time complexity of the algorithm is Ω(n), and n increases exponentially with the number of digits.
- However modern sieving techniques use the sieving techniques through other suitable manipulations.

A sequential implementation of the Sieve of Eratosthenes manages 3 key data structures:

- An array whose elements correspond to the natural numbers being sieved.
- An integer corresponding to latest prime number found.
- An integer used as a loop index, incremented as multiples of the latest (current) prime are marked as composite.

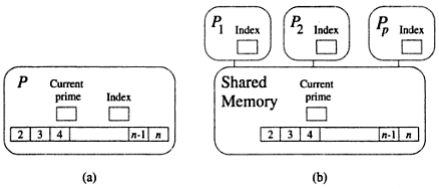
4

SHARED MEMORY MODEL FOR PARALLEL ERATOSTHENES ALGORITHM

Control Parallel Approach:

- Every processor goes through the two step process of finding the next prime number
- Striking from the list multiples of that prime, beginning with its square.
- The processors continue until a prime is found whose value is greater than $\sqrt{n}$

Shared Memory Model:
(a) Sequential algorithm maintains array of natural numbers, variable storing current prime, and index



Parallel Model:
(b) Each processor has its own private loop index and shares access to other variables with all processors.

5

INEFFICIENCIES

Two processors may asynchronously end up using the same prime to sieve.

- The first processor will access the value of the current prime and start sieving with it.
- The second processor will start from the next unmarked cell, which it updates as the current prime.
- If another processor starts before this update then it also starts sieving with the same prime.

Also a processor may end up sieving with composite numbers!

- The first processor starts sieving with multiples of 2.
- Before it marks any cell, a second processor starts sieving with the next prime, which is 3.
- A third processor starts sieving with the next unmarked cell, which is 4 (and has not been marked yet by the first processor)!

Our implementations hence needs to ensure that these time wasting situations do not happen.

6

ESTIMATING THE MAX SPEEDUP

Assumptions:

1. The above situations do not occur.

2. Ignore the time spent finding the next prime, and concentrate on the operations of marking the cells.

First analyze the sequential algorithm:

7

ESTIMATING THE MAX SPEEDUP

Assume it takes one unit of time for a processor to mark a cell.

Suppose there are k primes less than or equal to $\sqrt{n}$

Denote them by $\pi_1, \pi_2, \ldots, \pi_k$. Thus, $\pi_1 = 2, \pi_2 = 3, \pi_3 = 5, \ldots$

The total time required by a single processor is:

$$\left\lceil \frac{(n+1)-\pi_1^2}{\pi_1} \right\rceil + \left\lceil \frac{(n+1)-\pi_2^2}{\pi_2} \right\rceil + \left\lceil \frac{(n+1)-\pi_3^2}{\pi_3} \right\rceil + \cdots + \left\lceil \frac{(n+1)-\pi_k^2}{\pi_k} \right\rceil =$$

$$\left\lceil \frac{n-3}{2} \right\rceil + \left\lceil \frac{n-8}{3} \right\rceil + \left\lceil \frac{n-24}{5} \right\rceil + \cdots + \left\lceil \frac{(n+1)-\pi_k^2}{\pi_k} \right\rceil$$

For n=1000, the sum is 1,411.

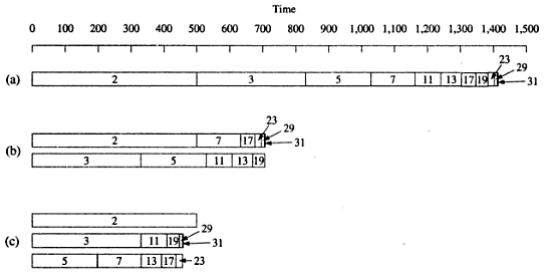
8

TIME TAKEN BY THE PARALLEL ALGORITHM

Time reduction with addition of processors (n=1000):

a) Single Processor strikes out all composite numbers in 1,411 units of time.
b) Two processors reduce the execution time to 706 units of time. This corresponds to a speedup of 1411/706=2
c) Three processors reduce the time to 499 time units, which leads to speedup of 2.83.

Note adding more processors does not help here, because with more than 2 processors the time required to sieve all multiples of 2 determine the parallel execution time.



9

DATA PARALLEL APPROACH

Let us consider another approach.

In this case, the approach is data parallel: that is different processor elements perform the same operation on different data sets.

- Each processor will be responsible for a segment of the array representing the natural numbers.
- All the processors perform the same operation (ie. strikes off multiples of the same prime) on its own segment of data.

Analyzing the speedup is straight-forward and is left as an exercise.

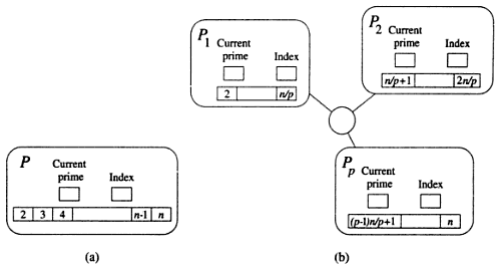
10

MODEL WITH NO SHARED MEMORY: MESSAGE PASSING PARADIGM

Consider a different model for parallel computing:

- There is no shared memory
- **Processors interact by message passing**

Shared Memory Model:
(a) Sequential algorithm maintains array of natural numbers, variable storing current prime, and index



Parallel Model:
(b) Each processor has its own copy of the variables containing the current prime and the loop index. Processor 1 finds prime and communicates them to other processors. Each processor iterates through its own portion of the array.

Assume the number of processors p<<$\sqrt{n}$. Thus the list controlled by the first processor has all primes less than $\sqrt{n}$ and the first prime greater than $\sqrt{n}$. Termination of the program happens when processor 1 reaches a prime greater than $\sqrt{n}$

11

ANALYSIS

We need to consider the time spent communicating the value of the current prime from processor 1 to all other processors.

Assume it takes χ time units for a processor to mark a multiple of a prime as being a composite number.

Suppose there are k primes as before, less than or equal to $\sqrt{n}$.

Computation Time:

The total time a processor spends striking out composite numbers is:

$$\left(\left\lceil\frac{\lceil(\frac{n}{p})\rceil}{2}\right\rceil+\left\lceil\frac{\lceil(\frac{n}{p})\rceil}{3}\right\rceil+\left\lceil\frac{\lceil(\frac{n}{p})\rceil}{5}\right\rceil+\cdots+\left\lceil\frac{\lceil(\frac{n}{p})\rceil}{\pi_k}\right\rceil\right)\chi$$

Communication Time: Assume each time processor 1 finds a new prime it communicates the value to each of the (p-1) processors in turn.

If processor 1 spends λ amount of time it passes a number to another process, total communication time for k primes is $k(p-1)\lambda$.

12

ANALYSIS FOR N=1,000,000

It turns out there are 168 primes less than 1,000 (square root of 10^6).

The largest is 997.

Therefore maximum computation time:

$$\left(\left\lceil\frac{\lceil\left(\frac{1,000,000}{p}\right)\rceil}{2}\right\rceil+\left\lceil\frac{\lceil\left(\frac{1,000,000}{p}\right)\rceil}{3}\right\rceil+\left\lceil\frac{\lceil\left(\frac{1,000,000}{p}\right)\rceil}{5}\right\rceil+\cdots+\left\lceil\frac{\lceil\left(\frac{1,000,000}{p}\right)\rceil}{997}\right\rceil\right)\chi$$

Total Communication Time:168(p-1)λ

Assume $\lambda = 100\chi$ and lets plot the speedup.

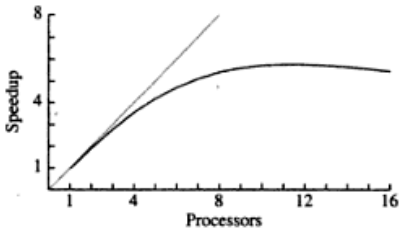
13

ESTIMATED SPEEDUP

Note that speedup is not directly proportional to the number of processors used.

Speedup is highest at 11 processors.

Why does the decline in speedup happen?



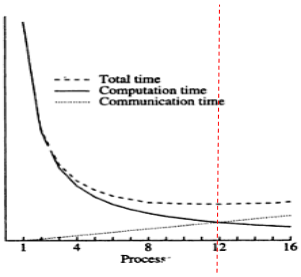
14

COMPUTATION TIME, COMMUNICATION TIME AND PROCESSORS

Computation time is inversely proportional with the number of processors.

Communication time increases linearly with the number of processors.

After 11 processors, increase in communication time is greater than the decrease in computation time.

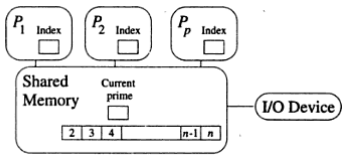


15

DATA PARALLEL APPROACH WITH I/O

The algorithms also need to store and print their results before termination.

Let us consider the data parallel implementation of the sieving method with an output on the shared memory model for parallel computation.

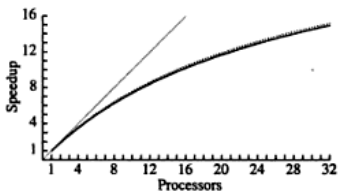


Let $i\beta$ denote the time required for a processor to transmit i prime numbers to that device.

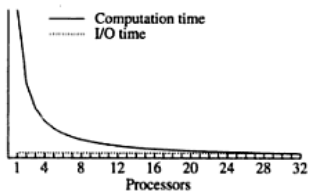
There are 78,498 primes less than 1,000,000.
Thus the time for the I/O is $78{,}498\beta$.

16

SPEEDUP ANALYSIS



Assuming, $\beta = \chi$ we plot the speedup.
The plot shows the variation of speedup for 1,2, …, 32 processors.
There is a damping effect on the speedup.
This is because the output to the I/O device must be performed sequentially.



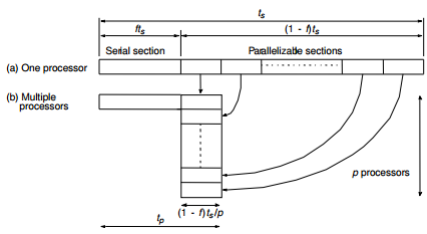
I/O time is a part of the operation which does not depend on the number of processors

17

AMDAHL'S LAW

Let, f be the fraction of operations in a computation that must be performed sequentially, where 0≤f≤1

Maximum speedup S achievable by a parallel computer with p processors is: $\frac{1}{f+(1-f)/p}$



18

APPLYING AMDAHL'S LAW

When n=1,000,000 the sequential algorithm marks 2,122,048 cells and outputs 78,498 primes.

Assuming both these operations take same amount of time, total time required is 2,122,048+78,498=2,200,546.

Thus, f=78,498/2,200,546=0.0357.

Thus, the upper bound on the speedup with p processors is:

$$\frac{1}{.0357+.9643/p}$$

The dotted curve in the speedup plot, shows this upper bound for different values of p.

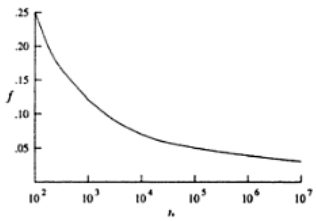
19

AMDAHL EFFECT

As the size of the problem increases, the fraction f of inherently sequential operation decreases.

- This phenomenon is called as Amdahl Effect.

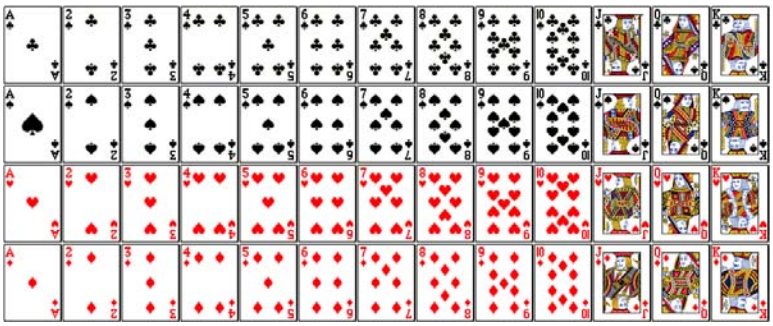
<u>An ameliorating fact:</u> This makes the problem more amenable for parallelization.



Plot of f with n for the data-parallel sieve algorithm with output, assuming $\beta = \chi$.

20

QUESTION



Shuffle a deck of cards and then determine how long it takes to sort the cards as above. Assume it is faster to sort the cards initially by suit, and then by an insertion sort to arrange each suit.

1. How long does it take for p people to sort p decks of shuffled cards?
2. How long does it take p people to sort one deck of cards?

21