GPU Assignment

Kaustav Vats
2016048.

Q-1 (a) Degree of Concurrency

(1) $\frac{N+1}{2} = \frac{2^n - 1 + 1}{2} = 2^{n-1}$ (Max in leaf Node which can work ||).

(2) $\frac{N+1}{2} = \frac{2^n - 1 + 1}{2} = 2^{n-1}$ ["]

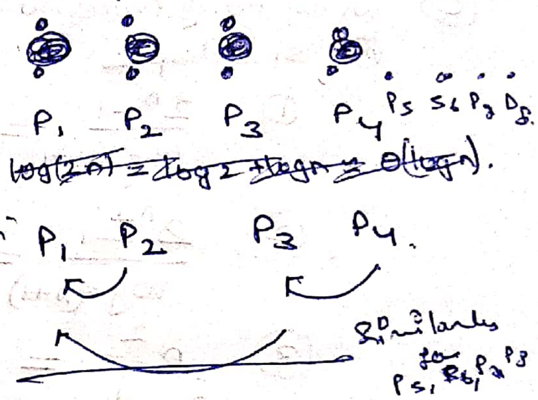
(3) $\sqrt{N} = \sqrt{n^2} = n$ (Diagonal process at 45° in given fig can work in ||).

(4) $n \rightarrow n$ element can work ~~tog~~ in parallel.

(b) $S = \frac{T_S}{T_P}$ each Level takes unit

(1) $T_S = N$ $T_P = \log(N+1)$ ↳ height of the tree

$S = \frac{T_S}{T_P} = \frac{2^n - 1}{\log(n+1)}$ T_P = no. of levels in Tree

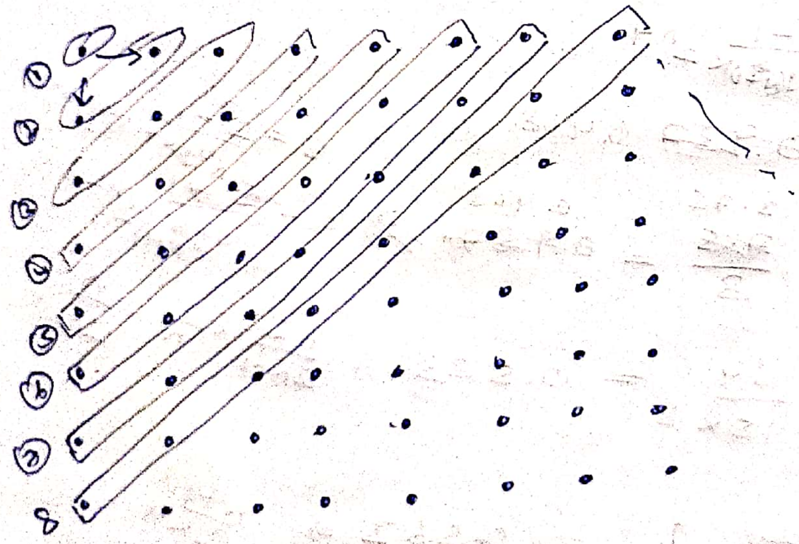


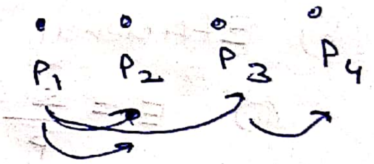
(2) $T_S = N$ $T_P = \cancel{\log n}$ $\log(N+1)$.

$S = \frac{T_S}{T_P} = \frac{2^n - 1}{\log(N+1)}$. →

No of Task are reduced by half.
Time taken for Graph (a) & (b) is ~~O(log n)~~.

(3) $T_S = n^2$ ~~T_P~~



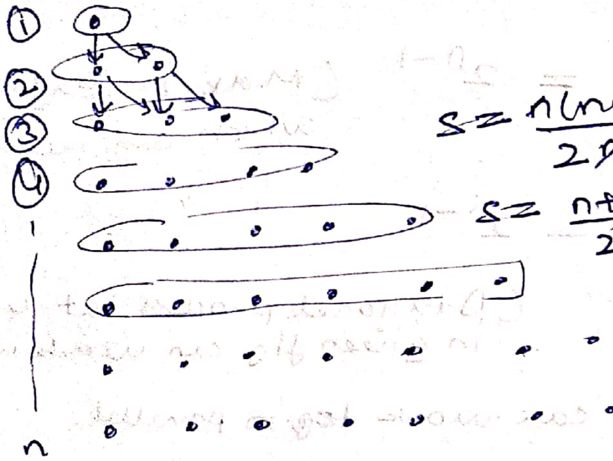


$T_P = n + n - 1$
↙
for first 8.
Then similarly n for other half &
−1 for removing extra count. ~~[illegible]~~

$T_P = 2n - 1$

$S = \frac{n^2}{2n-1}$

(4) $T_s = n(n+1)/2$



$T_p = n$

$S = \frac{n(n+1)}{2\cancel{n}}$

$S = \frac{n+1}{2}$

each layer takes 1unit of time to complete the task. Since ~~da~~ there data dependency between layers, so ~~eve~~ forwards layer can't work first ~~of~~ if previous layers are not completed.

(c) when no. of processor equal to degree of concurr

(a) speedups

① $\frac{2^n-1}{\log_2(N+1)} = \frac{15}{\log_2 16} = \frac{15}{4} = \cancel{3.75}\ 3.75$

② $\frac{2^n-1}{\log_2(n+1)} = \frac{15}{\log_2(16)} = \frac{15}{4} = \cancel{3.75}\ 3.75.$

③ $\frac{n^2}{2n-1} = \frac{64}{15} = 4.26$

④ $\frac{n+1}{2} = \frac{9}{2} = 4.5$

(b) Efficiency

① $E = \frac{S}{P} = \frac{2^n-1}{\log(N+1) \times 2^{n-1}}$

$E = \frac{3.75}{8} = \cancel{0.937}\ 0.468$

② $E = \frac{2^n-1}{P \times \log_2(N+1)} = \frac{3.75}{8} = 0.468$

$E = \frac{2^n-1}{\log(N+1) \times 2^{n-1}}$

③ $E = \frac{n^2}{P \times 2n-1} = \frac{64}{15 \times 8} = 0.533 \Rightarrow \frac{n^2}{2n-1} \times \frac{1}{n} = \frac{n}{2n-1}$

④ $E = \frac{n+1}{2P} = \frac{9}{2 \times 8} = 0.562 \Rightarrow \frac{n+1}{2 \times n}$

(c) Overhead $T_O = pT_p - T_s$

(1) $T_O = p \times \log^{(N+1)} - 2^n - 1 \Rightarrow 2^{n-1} \times \log^{(N+1)} - (2^n - 1)$.

$T_O = 8 \times 4 - 15$

$T_O = 17$

(2) $T_O = p \times \log^{(N+1)} - (2^n - 1) \Rightarrow 2^{n-1} \times \log^{(N+1)} - (2^n - 1)$.

$T_O = 8 \times 4 - 15$

$T_O = 17$

(3) $T_O = \cancel{p \times n^2} - (2n-1)$

$\cancel{T_O = 8 \times 64 - 15}$

$\cancel{T_O = 497}$

$\Rightarrow n \times (2n-1) - n^2$

$T_O = p \times (2n-1) - n^2$

$T_O = 8 \times 15 - 64$

$= 56$

(4) $T_O = p \times n \cancel{(n+1)/2} - \frac{n(n+1)}{2} \Rightarrow n \times n - \frac{n(n+1)}{2}$

$\Rightarrow n^2 - \frac{n^2}{2} - \frac{n}{2}$

$\frac{n^2}{2} - \frac{n}{2} = \frac{n(n-1)}{2}$.

$T_O = 8 \times 8 - \frac{8(9)}{2}$

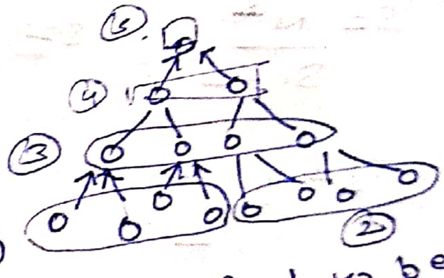
$= 64 - 36 = 28$

II Now $p = \frac{1}{2} \times$ Degree of ~~Degree~~ Concurrency

(a) speedup $\rightarrow$.

(1) $T_s = N = 2^n - 1$

$T_p = \log^{(N+1)} + 1$



It takes one extra unit to solve the problem as only half of last layer Task are required to be solved separately.

$S = \frac{T_s}{T_p} = \frac{2^n - 1}{\log^{(N+1)} + 1}$

$= \frac{15}{5} = 3$

$\neq \cancel{\frac{15}{2+1} = 5}$

(2) $T_s = N = 2^n - 1$

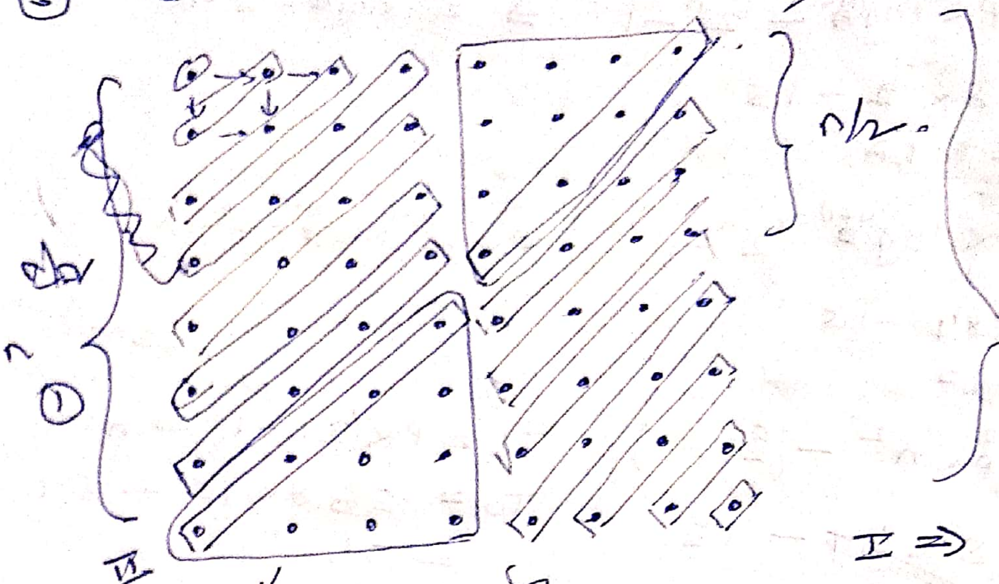
$T_p = \log^{(N+1)} + 1$

same explanation.

$S = \frac{T_s}{T_p} = \frac{2^n - 1}{\log_{(N+1)} + 1} = \cancel{\frac{15}{2+1} = 5}$

$= \frac{15}{5} = 3$

(3) $T_s = N^2$



II

similarly This region is $\frac{n}{2}$ units.

$\text{I} \Rightarrow n$

$\text{II} \Rightarrow \frac{n}{2}$ } have one Layer Common

$\text{III} \Rightarrow \frac{n}{2}$ } subtracting 1 unit

$\text{IV} \Rightarrow n.$

$\cancel{\frac{n}{2} + \frac{n}{2} + }$ $\cancel{n + \frac{n}{2} + \frac{n}{2} - \frac{1}{2}}$

Both Region (I) & Region (II) computer Include same layer twice so we subtract one ~~fr~~ unit from it (Similar for Region III & IV)

$\Rightarrow \frac{n}{2} + \frac{n}{2} + \frac{n}{2} + n - 1 - 1$

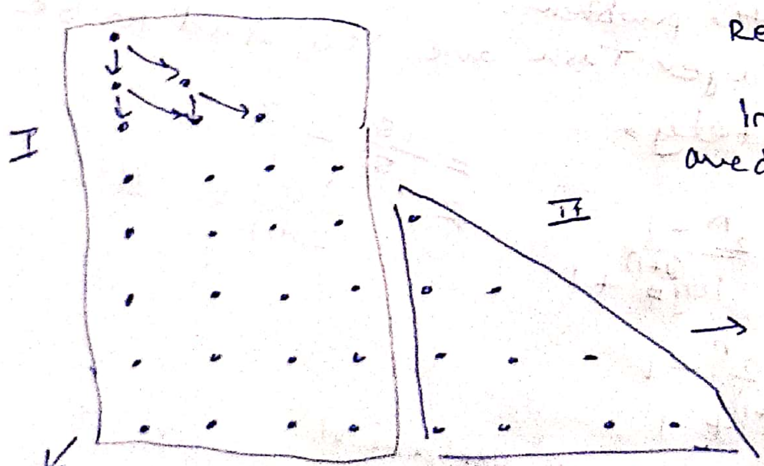
$n + n + n - 2$

$3n - 2$

$T_p = 3n - 2$

$S = \frac{N^2}{3n-2}$ $S = \frac{64}{3 \times 8 - 2} = \frac{64}{22} = 2.909$

(4) $T_s = \frac{n(n+1)}{2}$



Region I fully utilize given processors where as In Region II ~~Same~~ Task layers are dependent on previous layer

$\frac{n}{2}$ units.

n units.

Region
I → Takes n units
II → Takes $\frac{n}{2}$ units

$T_p = n + \frac{n}{2} = \frac{3n}{2}$ units

$T_p = n + \frac{n}{2}$

$= \frac{3n}{2}$

$$S = \frac{T_s}{T_p} = \frac{n(n+1)}{2 \times 3\cancel{n}} \cancel{2} = \frac{(n+1)}{2}$$

$$S = \frac{8+1}{3} = 3$$

(b) Efficiency $\boxed{P = 4}$

(1) $E = \frac{S}{P} = \frac{2^{n}-1}{(\log(N+1)+1) \times \frac{2^{n-1}}{2}}$

$\cancel{E = \frac{2^n - 1}{P \times \log n + 1}}$ $= \frac{3}{4} = 0.75$

$\cancel{E = \frac{S}{4} = }$

(2) $E = \frac{S}{P}$

$E = \frac{2^n - 1}{P \times (\log^{N+1} + 1)} \Rightarrow \frac{2^n - 1}{\frac{2^{n-1}}{2}(\log(N+1)+1)} \Rightarrow \frac{2^n - 1}{2^{n-2}(\log(N+1)+1)}$

$\cancel{E = \frac{15}{3 \times 4}}$ $= \frac{3}{4} = 0.75$

(3) $E = \frac{S}{P}$

$E = \frac{n^2}{(3n-2) \times P}$

$E = \frac{64}{(24-2) \times 4}$

$= \frac{64}{22 \times 4} = \frac{64}{88} = 0.727$

(4) $E = \frac{S}{P}$

$E = \frac{\cancel{n^2}\; n(n+1)\; 2}{(2 \times 3n) \times P}$

$E = \frac{3}{4} = 0.75.$

Scanned by CamScanner

(c) Overhead fnc $T_O = T_p \times P - T_s$

$T_O = P T_p$

(1) $T_O = P \times (\log_{(N+1)} n + 1) - (2^n - 1)$ $\Rightarrow \frac{2^{n-1}}{2} \times (\log (N+1) + 1) - (2^n - 1)$

$= \cancel{64 \times (\log 4 + 1) - 15}$ $\Rightarrow \frac{2^n}{4} \log (N+1) + \frac{2^n}{4} - 2^n + 1$

$= \cancel{4 \times 3 - 15}$ $\frac{2^n}{4} \log (N+1) + 1 - \frac{3 2^n}{4}$

$= \cancel{12 - 15} =$ $\Rightarrow 4 \times 5 - 15 = 5$

(2) $T_O = P \times (\log_n^{N+1} + 1) - (2^n - 1)$ $\Rightarrow \frac{2^n - 1}{2} (\log (N+1) + 1) - (2^n - 1)$

$= \cancel{4 \times (\log 4 + 1) - (2^n - 1)}$.

$= \cancel{4 \times 3 - 15}$ $\Rightarrow 4 \times 5 - 15 = 5$.

$= \cancel{12 - 15}$

(3) $T_O = P \times (3n - 2) - n^2$

$= 4 \times (3 \times 8 - 2) - 64$

$= 4 \times 22 - 64$

$= 88 - 64 = 24$

(4) $T_O = P \times \left(\frac{3n}{2}\right) - \frac{n(n+1)}{2}$

$= \cancel{4}^{2} \times \frac{3 \times 8}{\cancel{2}} - \frac{\cancel{8}^{4}(9)}{\cancel{2}}$

$= 48 - 36$

$= 12$

Q-2 (a) $T_p = \left(\frac{n}{p} - 1\right) + 11 \log p$

$T_p \leq 512$

$\frac{n}{p} - 1 + 11 \log p \leq 512$

$\frac{n}{p} \leq 512 - 11 \log p + 1$

$n \leq (513 - 11 \log p) \times p$

$\log_2(\omega)$ → Base 2.

$p = 1$

$n \leq (513 - 11 \log(1)) \times 1$

$n \leq 513$

$\boxed{n = 513}$

$p = 4$

$n \leq (513 - 11 \log(4)) \times 4$

$n \leq (513 - 22) \times 4$

$n \leq 1964$

$\boxed{n = 1964}$

$p = 16$

$n \leq (513 - 11 \log(16)) \times 16$

$n \leq (513 - 44) \times 16$

$\boxed{n = 7504}$

$p = 64$

$n \leq (513 - 11 \log 64) \times 64$

$n \leq (513 - 66) \times 64$

$\boxed{n = 28603}$

$p = 256$

$n \leq (513 - 11 \times \log 256) \, 256$

$n \leq (513 - 88) \times 256$

$n \leq 108800$

$p = 1024$

$n \leq (513 - 11 \times \log 1024) \, 1024$

$n \leq (513 - 110) \times 1024$

$\boxed{n = 412672}$

$p = 4096$

$n \leq (513 - 11 \log(4096)) \, 4096$

$n \leq (513 - 132) \, 4096$

$\boxed{n = 1560576}$

Scanned by CamScanner

Q-2 (b) No, its not possible to solve a problem in finite amount of time with infinite resources.

Rate at which the problem size must increase wrt no. of processing elements to keep efficiency fixed determine the scalability of the system, slower rate are better.

$$W = \frac{E}{1-E} T_o(W, P)$$

Function is called isoefficient func. determines the ease at which a parallel system can maintain a constant efficiency and achieve speedup increase.

Problem size can be increased linearly with the the number of processors for cost optimal function and also maintains a fixed execution time. ∴ iso efficient funct is $\Theta(P)$

Q-3 Standard speedup

$$S = \frac{T_s}{T_p} = \frac{W}{\left(\frac{W}{P} - 1\right) + \log P \times 11}$$

P = 1

$$S = \frac{256}{(256 - 1)} = \frac{256}{255} = 1.0039$$

P = 4

$$S = \frac{256}{\left(\frac{256}{4} - 1\right) + 11 \times \log_2 4}$$

$$= \frac{256}{63 + 22} = \frac{256}{85} = 3.011$$

P = 16

$$S = \frac{256}{\left(\frac{256}{16} - 1\right) + 11 \log_2 16}$$

$$S = \frac{256}{15 + 11 \times 4}$$

$$= \frac{256}{59} = 4.3389$$

P = 64

$$S = \frac{256}{\left(\frac{256}{64} - 1\right) + 11 \times \log 64}$$

$$S = \frac{256}{3 + 11 \times 6} = 3.7101$$

P = 256

$$S = \frac{256}{\left(\frac{256}{256} - 1\right) + 11 \times \log 256}$$

$$S = \frac{256}{88} = 2.909$$

Scaled speedup

$$S = \frac{PW}{W - 1 + 11 \log P}$$

P = 1

$$S = \frac{256}{255 + 11 \times 0}$$

$$S = 1.0039$$

P = 4

$$S = \frac{256 \times 4}{255 + 22} = \frac{1024}{277} = 3.6967$$

P = 16

$$S = \frac{16 \times 256}{255 + 44} = \frac{4096}{299} = 13.6989$$

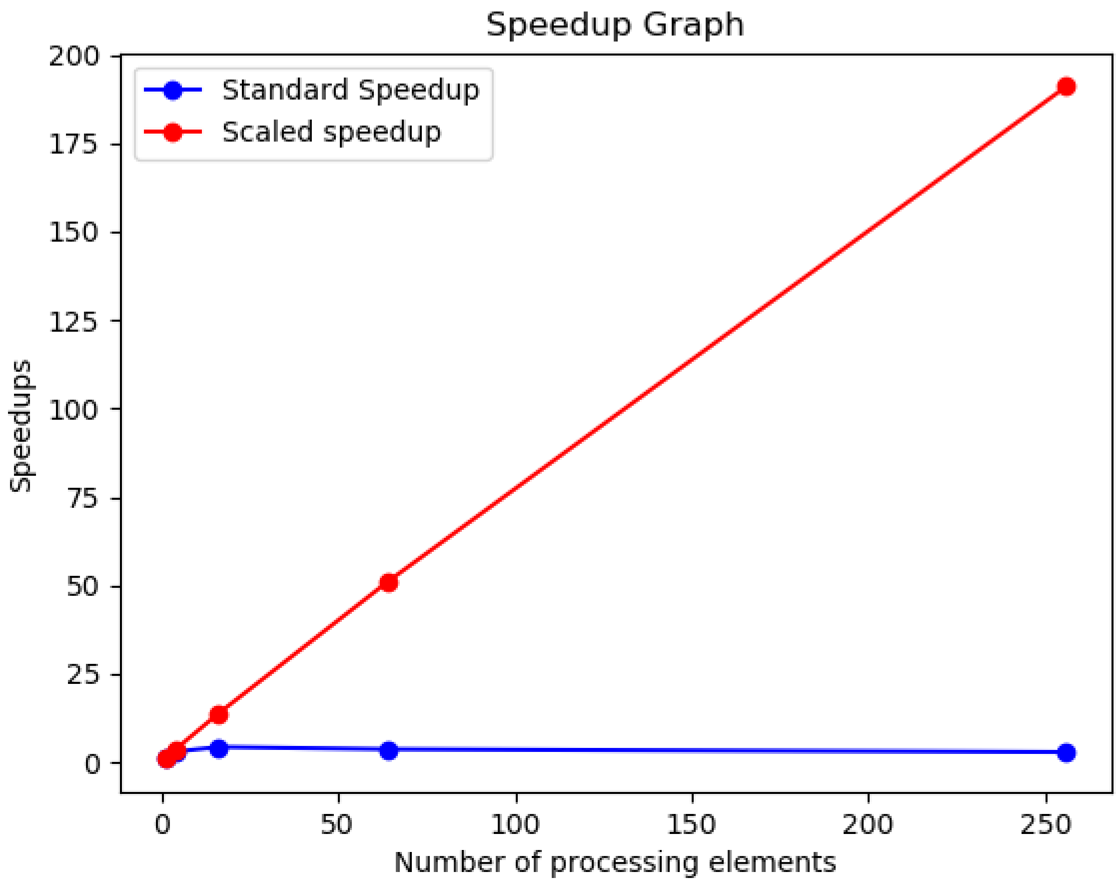
P = 64

$$S = \frac{64 \times 256}{255 + 66} = \frac{16384}{321} = 51.040$$

P = 256

$$S = \frac{256 \times 256}{255 + 88} = \frac{65536}{343}$$

$$= 191.067$$



Speedup Graph
Standard Speedup
Scaled speedup
Speedups
Number of processing elements
0
25
50
75
100
125
150
175
200
0
50
100
150
200
250