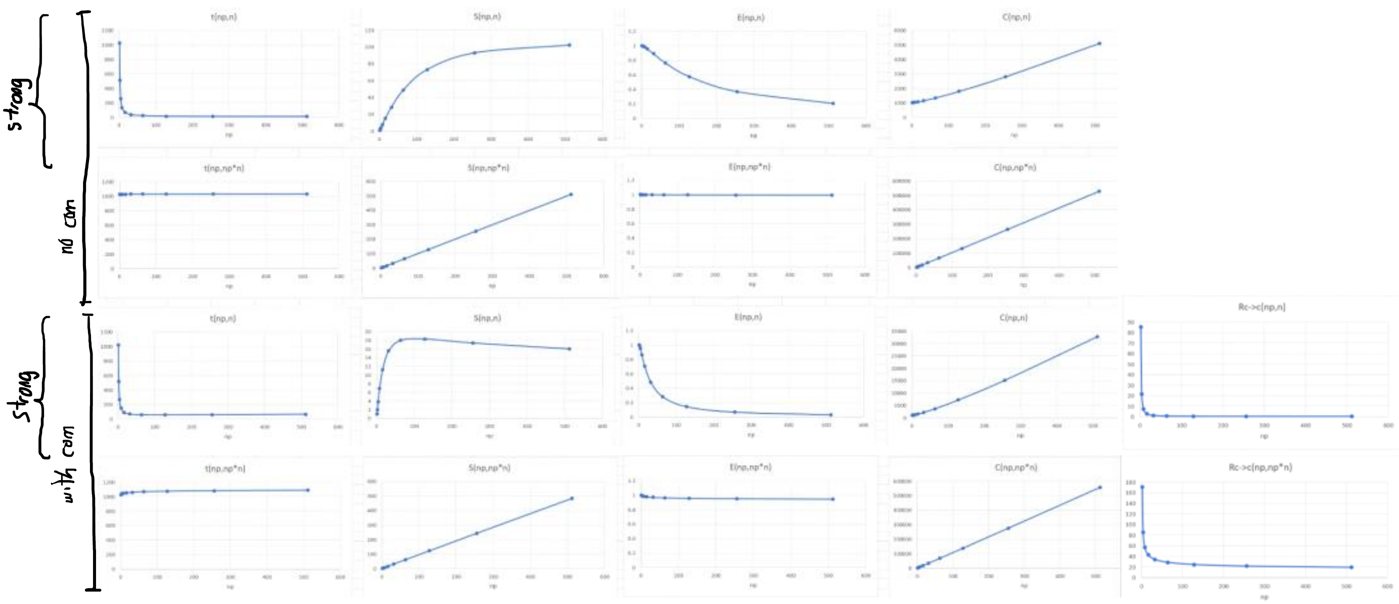
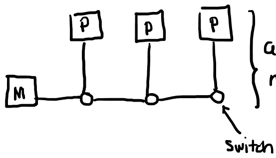
Performance

jueves, 10 de marzo de 2022 06:49 a. m.



Note: $(np^*n) \to$ size of the problem, and thus all plots depend on np

→ Now, our example network will be 'shared' type of network:



all processors connect through memory using the same line of communication: if not available, waiting time is needed.

=

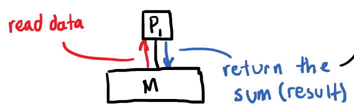
only one processor at a time has access to memory

Problem: $\vec{x} = (-1,2,4,1,6,0,-1,0)$

The serial case:

Assume: 3 units of time for max time of communication

$kp = 0 \to 2^{kp} = 2^0 = 1$ $\quad \vec{x} = (-1,2,4,1,6,0,-1,0)$



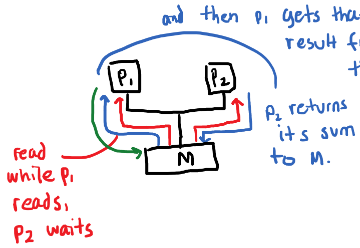
$t = 3 + 1(3) + 0(2)(3)$ $\quad$ 0 [○

$\underbrace{3}_{\text{return to M the sum}}$

Parallel cases:

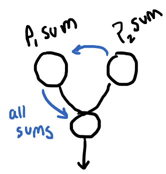
$kp = 1 \to 2^{kp} = 2^1 = 2$

and then P_1 gets that partial result from M, sums the two



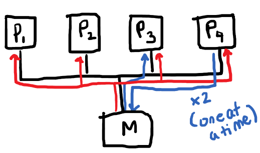
$t = 3 + 2(3) + 2(3)$

return of final sum to M; both read; P_1 reads the sum, P_2 puts sum in M



half

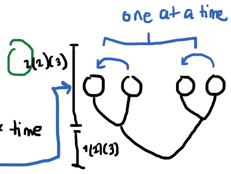
$kp = 2 \to 2^{kp} = 2^2 = 4$



$t = 3 + 4(3) + (2)(2)(3)$

return to rom; all 4 read; two times; r/w; max time

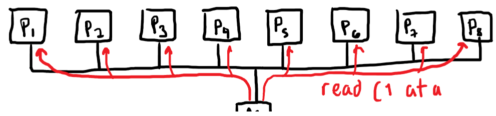
one at a time



half

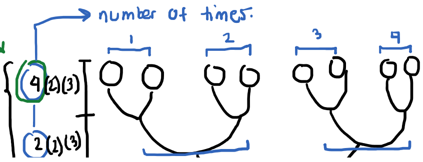
$+ (1)(2)(3)$

$kp = 3 \to 2^{kp} = 2^3 = 8$

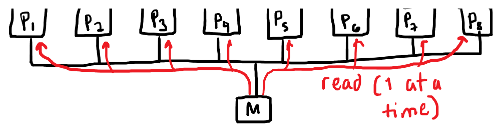


$t = 3 + \underbrace{8(3)}_{8\ \text{ps read}} +$

→ number of times.

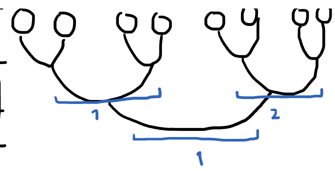


$Kp = 3 \rightarrow 2^{Kp} = 2^3 = 8$



$t = 3 + \underbrace{8(3)}_{8\ ps\ read} + \underbrace{\left\{ \begin{matrix} 4(2)(3) \\ 2(2)(3) \\ 1(2)(3) \end{matrix} \right.}_{\text{time in transference of partial sums}}$

$4(2)(3) + 2(2)(3)$
$+ 1(2)(3)$



In general form, the 3 parallel cases are:

$t = 3 + 3np + (2)(3)[\underbrace{1+2+4}_{2^0 + 2^1 + 2^2}]$ $\quad 2 \rightarrow Kp-1$

$t = 3 + 3np + (2)(3)[\underbrace{1+2}_{2^0 + 2^1}]$

$t = 3 + 3np + (2)(3)[\underbrace{1}_{2^0}]$

Thus,

$$t = 3 + 3np + (2)(3)\underbrace{\sum_{i=0}^{(Kp-1)} 2^i}$$

Geometric series:

$$\sum_{n=0}^{N} ax^n = a\left[\frac{1 - x^{N+1}}{1 - x}\right], \quad \underbrace{x \neq 1}$$

since $x = 1$ gives ∞

Here, we can substitute:

$x = 2$
$n = i$
$a = 1$
$N = Kp - 1$

Therefore

$$\sum_{i=0}^{(Kp-1)} 2i = (1)\left[\frac{1 - 2^{(Kp-1)+1}}{1 - 2}\right], \quad 2 \neq 1$$

$$= \left[\frac{1 - 2^{Kp}}{-1}\right] = 2^{Kp} - 1$$

Plugging into (1) the term $2^{Kp} - 1$ where $\sum_{i=0}^{Kp-1} 2^i$,

$$t_c(np, n) = 3 + 3np + 2(3)[2^{Kp} - 1]$$

$$= 3 + 3np + 6[2^{Kp} - 1]$$

$$= 3 + 3np + 6(2^{Kp}) - 6$$

$$= 3np + 6(2^{Kp}) - 3, \quad \text{where } Kp = \log_2(np)$$

$$= 3np + 6(2^{\log_2(np)}) - 3$$

cancel log

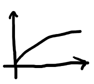
$$= 3np + 6np - 3$$

$t_c(np, n) = \boxed{9np - 3}$ Time the network takes to communicate

it is linear, and that's why shared types are limiting scabity

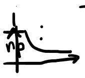
Note:

$\log_b np$:

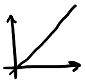


$\sqrt{np}$:

$\frac{1}{np}$:



np :
(linear)



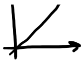
We need to add this t_c to computation time $t(np, n)$,

$$t(np, n) = \left[\frac{n}{np} + 1\right] + Kp + 9np - 3$$

It is suggested to have $\log_b np$, $\sqrt{np}$ and $1/np$. If we try to plugged some in,

$$t(np, n) = \left[\frac{n}{\ } + 1\right] + Kp + 9np - 3$$

np :
(linear)



to plugged some in,

$$t(np,n) = \left[\frac{n}{np}+1\right] + kp + 9np - 3$$

$$= \left[\frac{n}{np}+1\right] + \log_2 np + 9np - 3$$

$$R_{c\to c} = \frac{\left[\frac{n}{np}-1\right] + \log_2 np}{9np - 3}$$

if $np \to \infty$, $\log_2 np$ increases slower than $9np - 3$, so the overall term is not so urgently directed to ∞, as desired.

In excel file: $t_c \to 9*np - 3$ } $t_c(np,n)$

We conclude from strong scalab. that this works only for a small amount of processors.

We can say from weak scalab. that as np grows, speed is softened and $R_{c\to c}$ approaches zero.

But overall, this network is not so good at parallelism. But, if we got this network, look for the <u>weak</u> <u>scalability</u>.