Principles for parallel performance analysis

Tuesday, August 28, 2018 11:01 AM

Principles

ω : serial work — serial time $= \omega$

$T(n, p)$: Parallel runtime on an input size of n & processor size of p

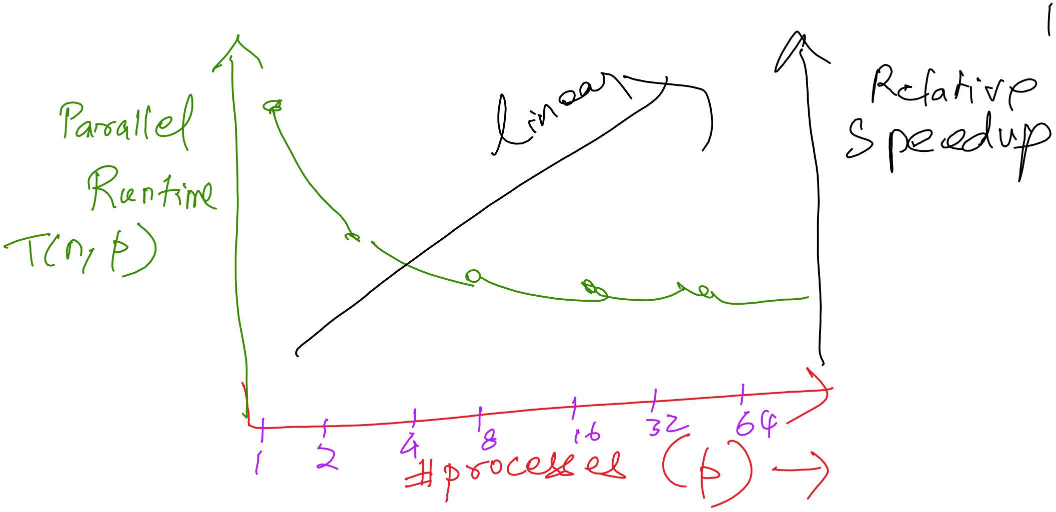
$T(n, 1)$: parallel runtime on 1 process

$T(n, 1) = \Omega(\omega)$

Speedup: $S(n,p)$ = ratio of serial time and parallel time

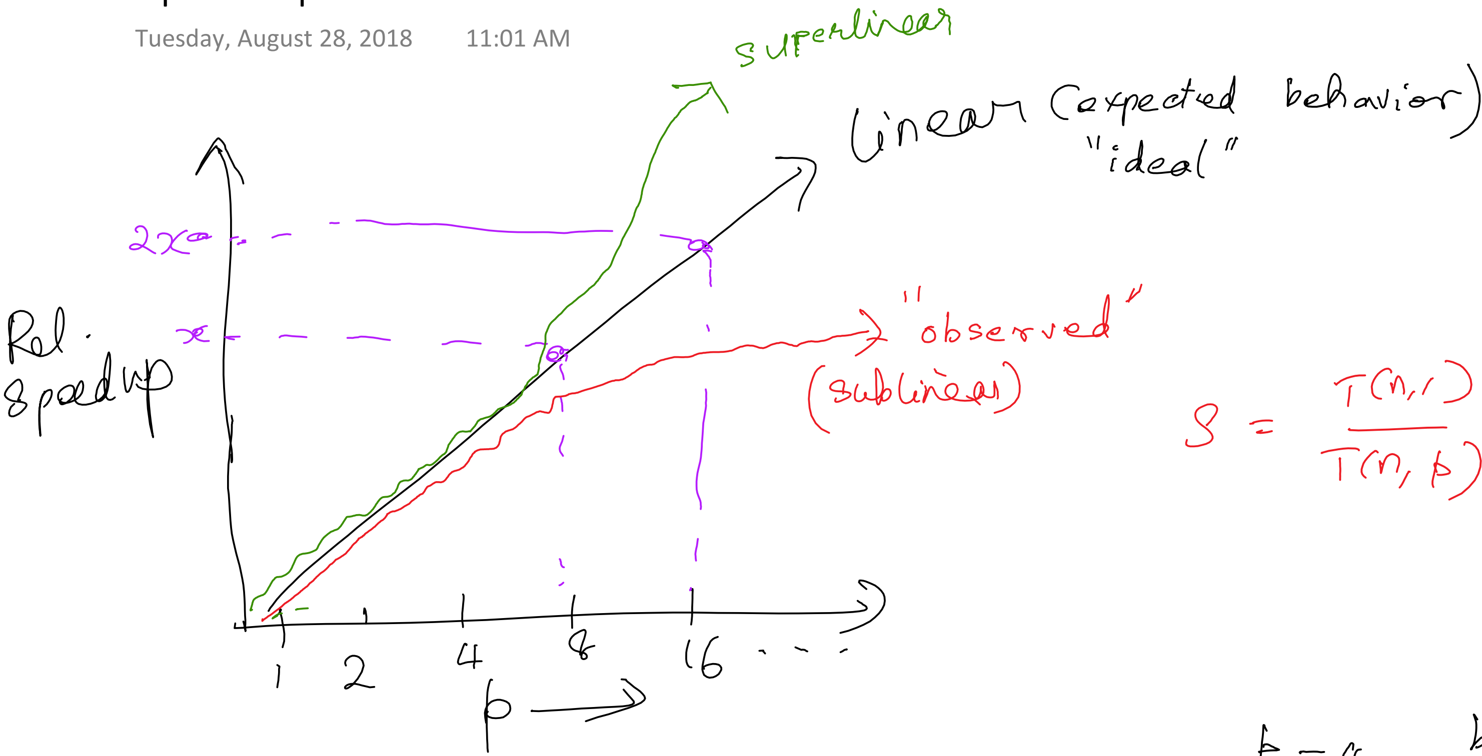
Relative speedup: $S(n,p) = \dfrac{T(n,1)}{T(n,p)}$

Real speedup: $S(n,p) = \dfrac{\omega}{T(n,p)}$

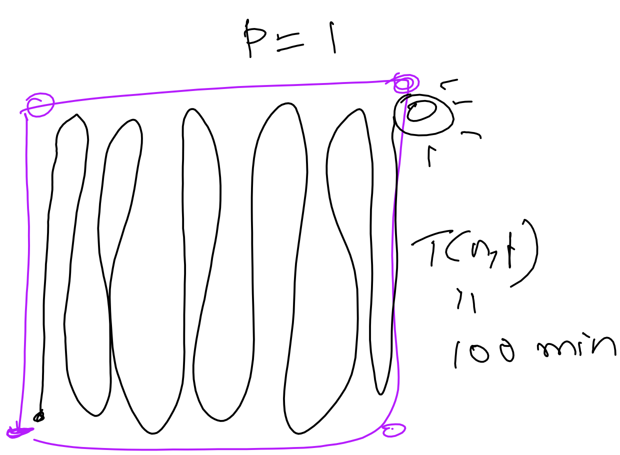


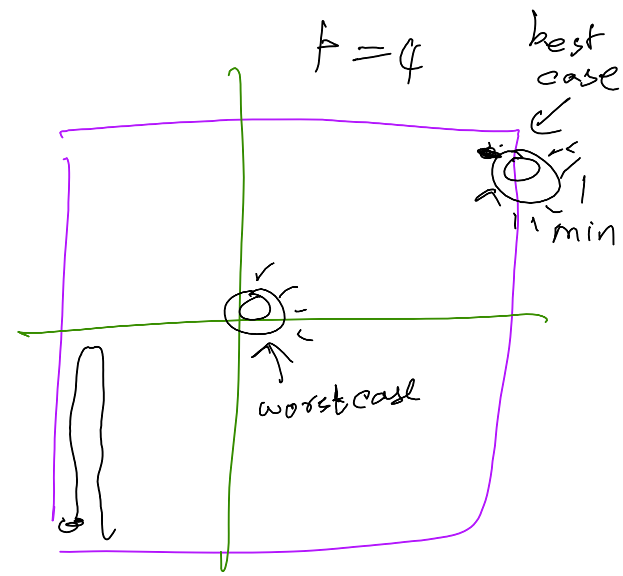
Speedup

Tuesday, August 28, 2018 11:01 AM



$$S = \frac{T(n,1)}{T(n,p)}$$





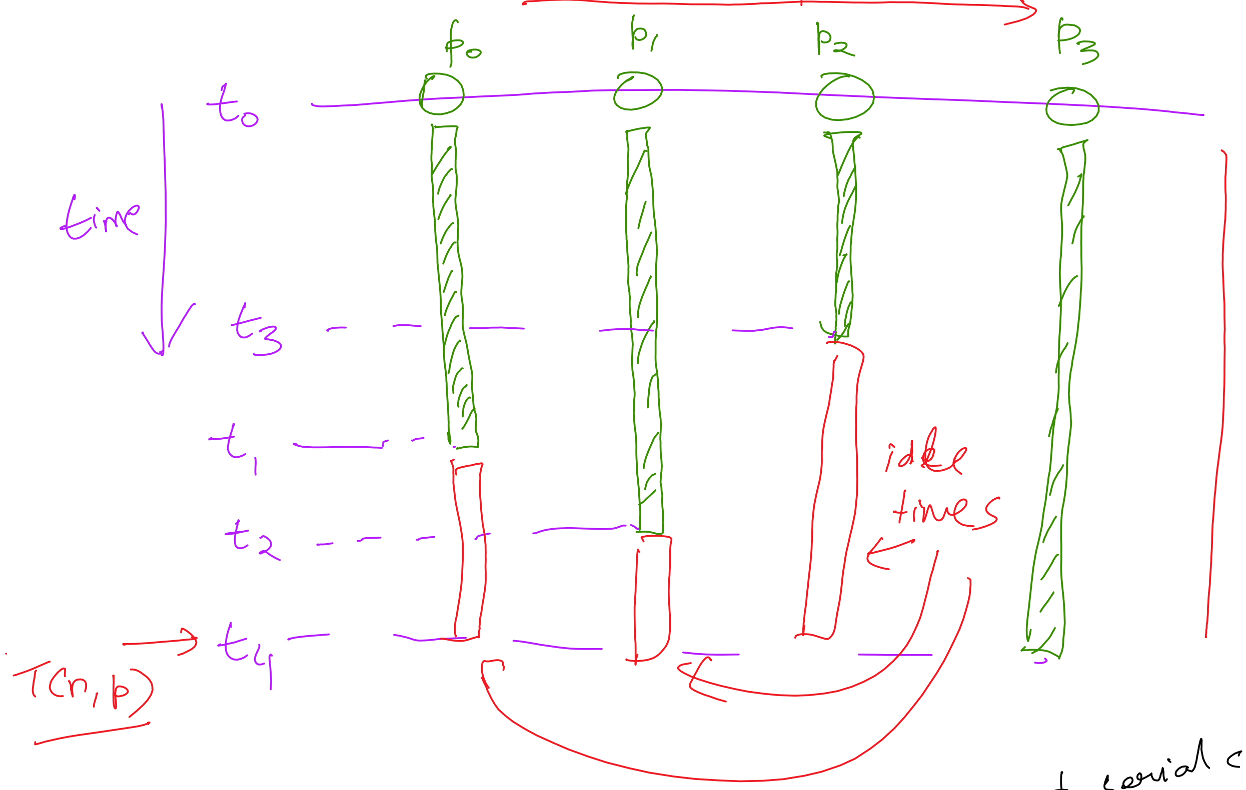
Super linear speedups are possible in scenarios such as the above search application.

Efficiency

Tuesday, September 4, 2018 10:53 AM

Efficiency: $E(n,p) :=$ ratio between serial work and "parallel work"

→ "Parallel work" $= p \times T(n,p)$



$$E(n,p) = \frac{\omega}{p \times T(n,p)}$$

ω → best serial code as "baseline"

$$E(n,p) = \frac{T(n,1)}{p \times T(n,p)}$$

→ relative form

$$E = \frac{S(n,p)}{p}$$

$0\% < E \leq 100\%$

Efficiency

Tuesday, September 4, 2018 10:53 AM

Observation: It is easier to maintain (if not improve) efficiency by reducing the number of processors. (but at the risk of increasing runtime).

(More formally)

Lemma: $E(n, p_1) \geq E(n, p_2)$, if $p_1 < p_2$.

Proof:

$$E(n, p_1) = \frac{T(n,1)}{p_1 T(n, p_1)} \qquad E(n, p_2) = \frac{T(n,1)}{p_2 T(n, p_2)}$$

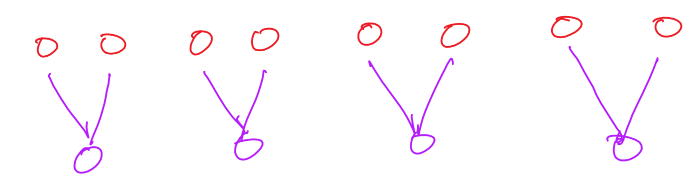
Q) What is the relationship between $T(n, p_1)$ & $T(n, p_2)$?

A) Since $p_1 < p_2$, divide the p_2 processors into $\left\lceil \frac{p_2}{p_1} \right\rceil$ groups, such that:

the work done by each group is given to a single processor of the p_1 group.

E.g: let $p_1 = 4$ and $p_2 = 8$

⇒



$$\Rightarrow T(n, p_1) \leq \left\lceil \frac{p_2}{p_1} \right\rceil T(n, p_2) \longrightarrow (1)$$

$$\Rightarrow E(n, p_1) = \frac{T(n,1)}{p_1 T(n, p_1)}$$

$$\geq \frac{T(n,1)}{p_1 \left(\frac{p_2}{p_1}\right) T(n, p_2)} \qquad \text{// substituting from (1)}$$

$$= E(n, p_2)$$

∎