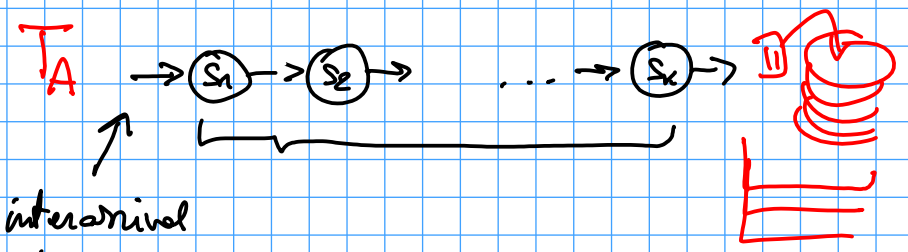
$T_S = \max\{T_{S_i}\}$

"pipeline alone in isolation"



interarrival time

$T_S' = \max\{T_A, T_S\}$

T_S of the pipeline alone

2) "true streams" ∞ their alone speed ⇒ T_A

stream of images from a camera

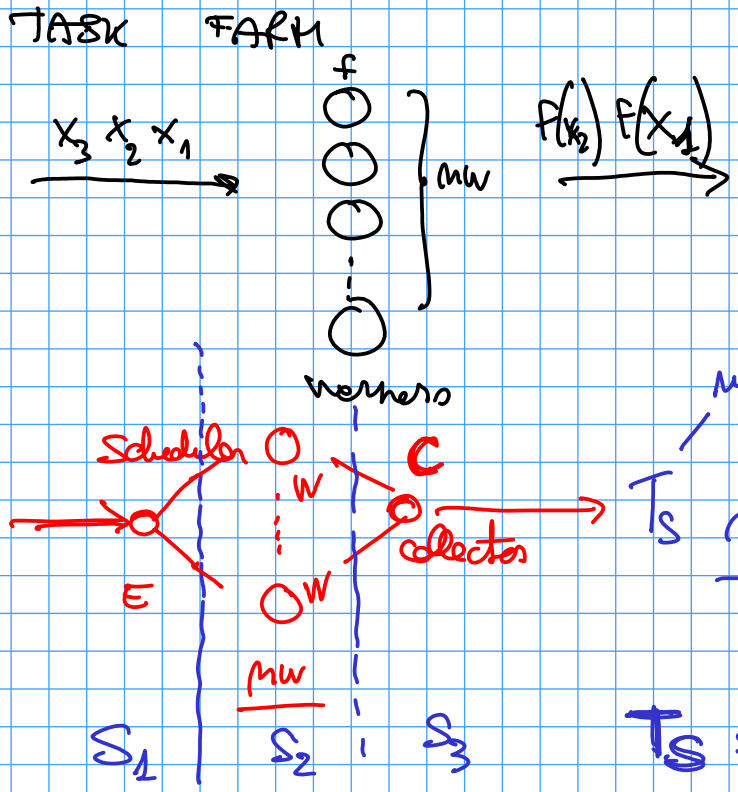
b) for( ) { f; g; h; }

T_A "less critical"

produce a stream

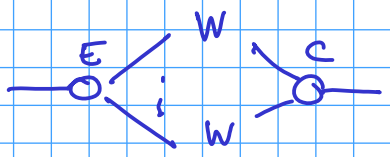
pipeline f; g; h

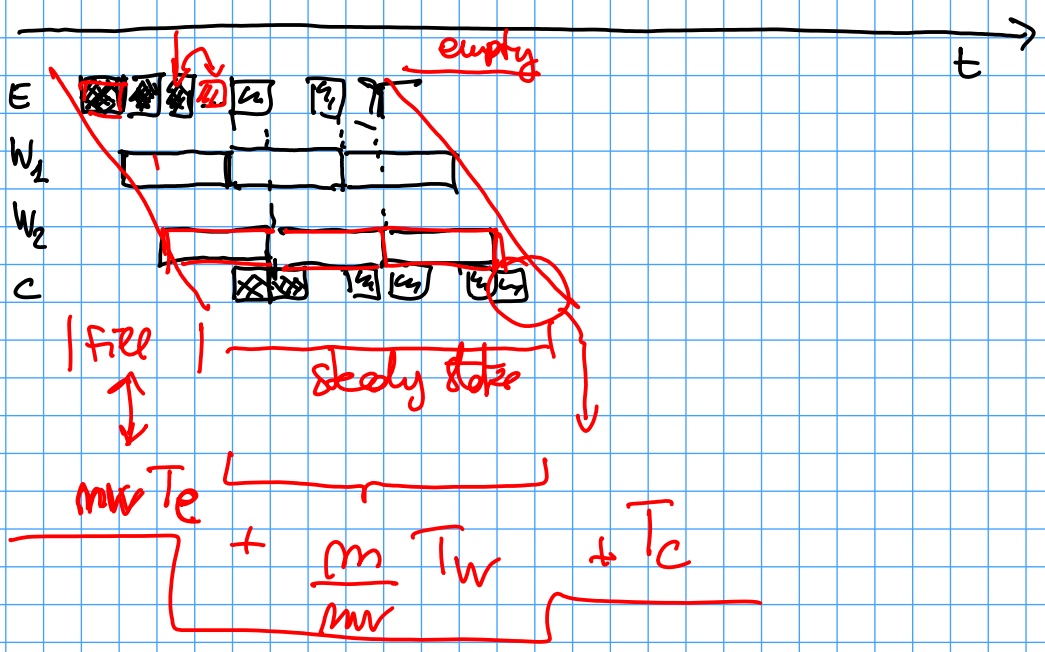
model embarrassingly stream parallel computations



$$T_S \frac{\max\{nw\,T_e, T_w, n\,T_c\}}{nw} \quad (nw)$$

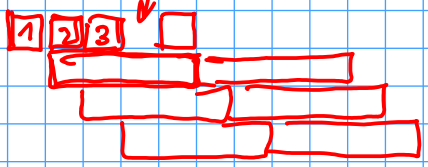
$$T_S = \max\left\{T_e, T_c, \frac{T_w}{nw}\right\}$$





$T_C(nw) \simeq$ $nw\,T_e$ + $\frac{m}{nw}\,T_w$ + T_C

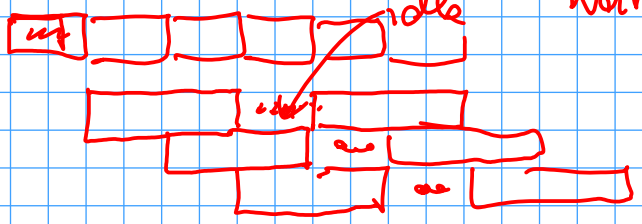
1° idle time of the emitter



emitter "waiting"

$nw\,T_e < T_w$

2° $nw\,T_e > T_w$



Kanten

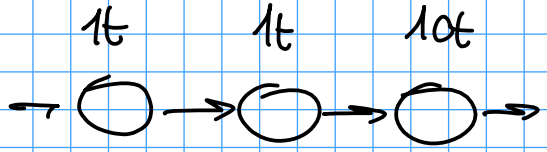
decide "good"

$nw\,T_e \gtrsim T_w$

$T_e = \frac{T_w}{nw}$

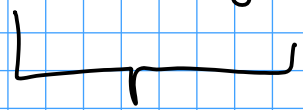
$nw = \frac{T_w}{T_e}$

$T_{seq} = 12t$

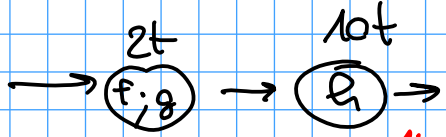


$T_s = 10t$

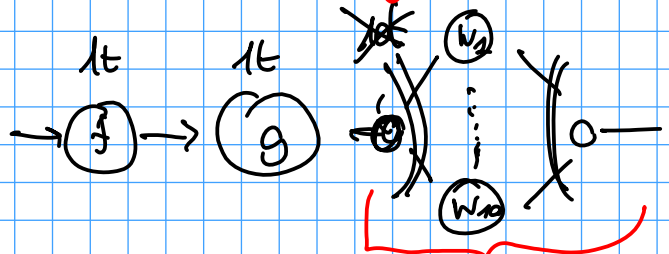
$\varepsilon(n) = \frac{\text{Sp}()}{n}$



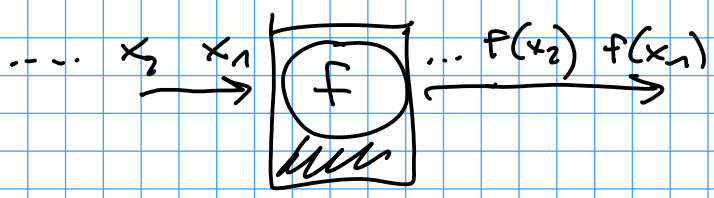
$T_{seq} = 12t$

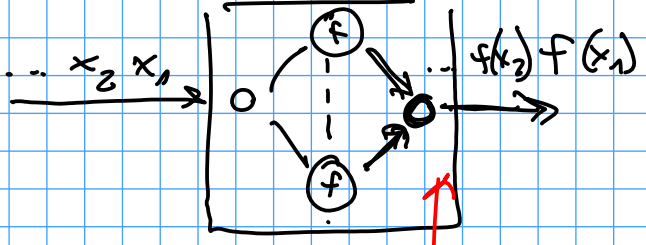


$T_s = 10t$

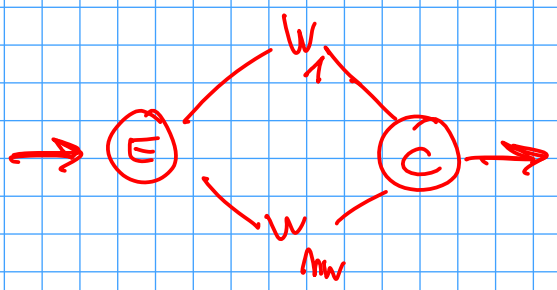


$T_e \simeq T_c \ll 1t$

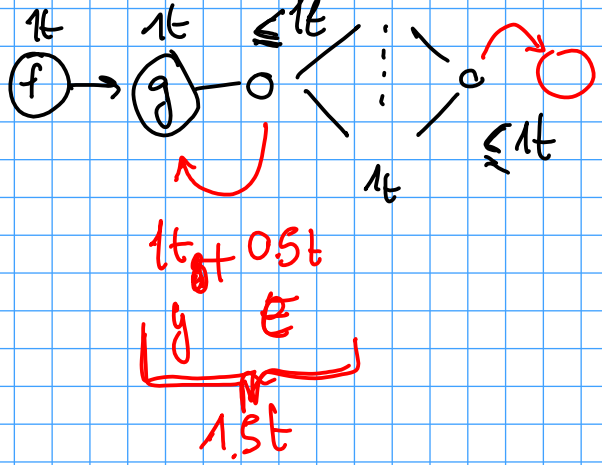




need to ensure input/output ordering (tasks/results)

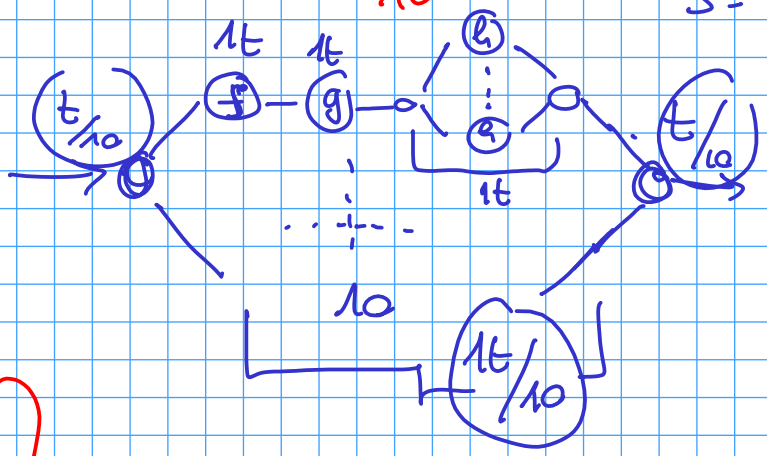


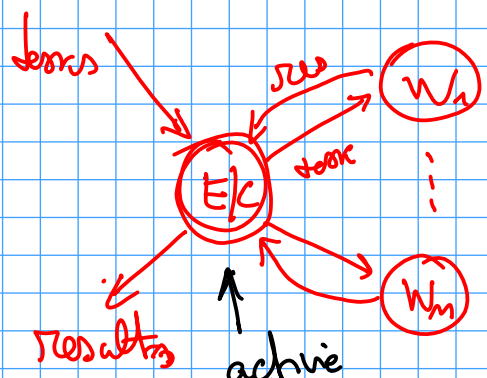
$\varepsilon(n) =$ $nw + 2$



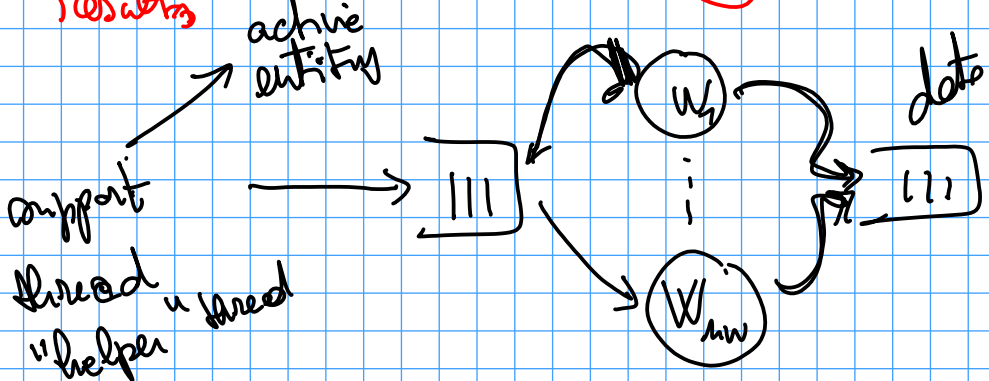
$t_e \approx t_c \approx \frac{t}{10}$

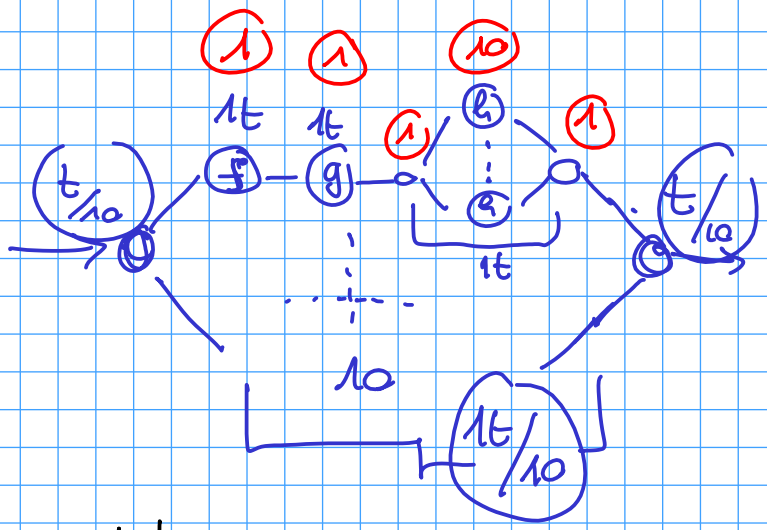
$t_s = t/10$





master/worker pattern





1W ⇒ 14

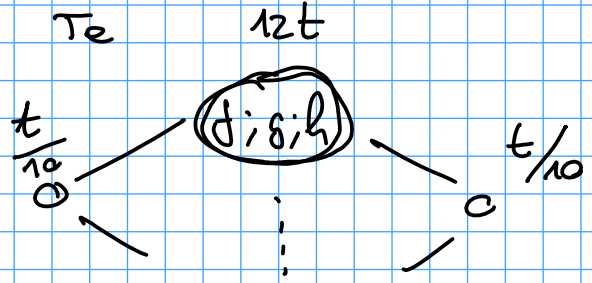
mwt = 10

w E C

140 + 1 + 1

142

$mw = \frac{T_w}{T_e}$



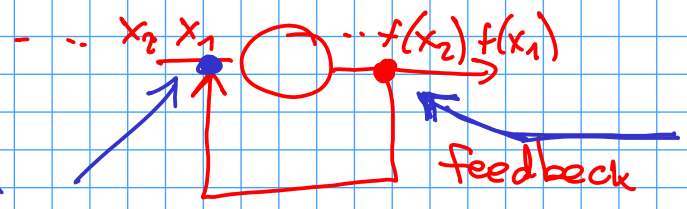
$\frac{12t}{\frac{t}{10}} = 120$

+ E + C = 122 workers

$T_s = t/10$

- Laptop ~ ok
- c/c++ ok
- Linux ~ ok

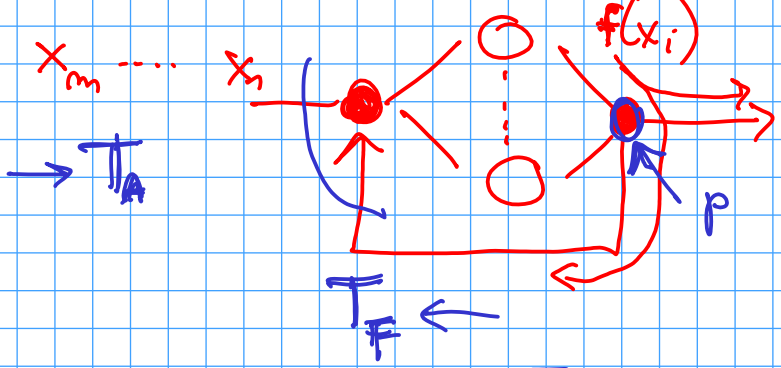
Stream parallel patterns
- pipeline
- farm (master/worker)
- feedback

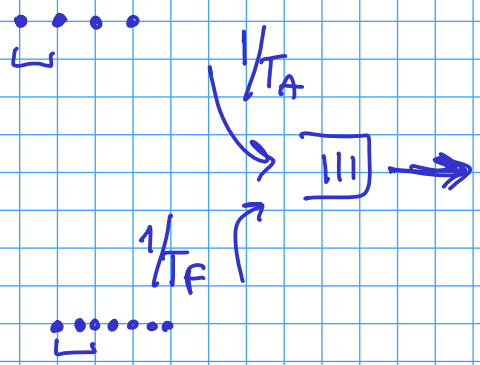


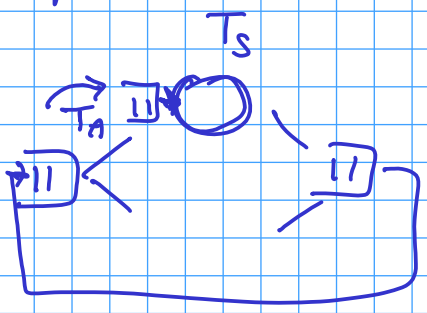
decide whether or not route back the result

⇒ additional activity that takes the decision

implement a "merge" operator







$$p = \frac{T_S}{T_A} < 1$$