{ set of skeletons }

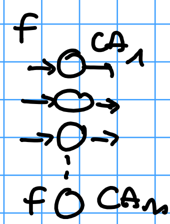
"RISC"

- a few general purpose skeletons
- very efficient
- easily combined into more complex patterns

"CISC"

- much more skeletons including "domain specific" skeletons ("closer" to the application domain programmer viewpoint/perspective)

skeletons {farm, map, stencil}
templates (DF)
mechanism
os + hw



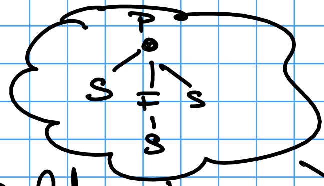
$$a_1 x_1 + a_2 x_2 \cdots + a_n x_n = a_b$$

$$b_1 x_1 + b_2 x_2 + \quad = b_b$$

$$\left(k \frac{a_1}{b_1}\right) x_1$$

$$f(a_1, b_1)\, x_1 + f(a_2 b_2)\, x_2 \quad \cdots = f(a_b, b_b)$$

Templete based implementation



skeleton tree

template library

compiler

executable combination of templates

Templates
- parametric
- process networks (C.A.)
- implementing SK_i
- on an architecture
- with perf. models

match

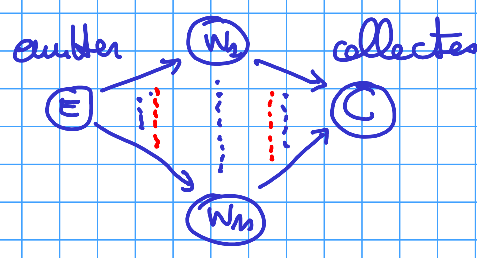
fixed

matched

optimize

< FARM , COW >

sk , hw



computing "f"

$$T_s = \max\left\{T_e, \frac{T_w}{n}, T_c\right\}$$

$T_s(n)$

time to emit (E)

time to collect (C)

time to compute "f"

par degree

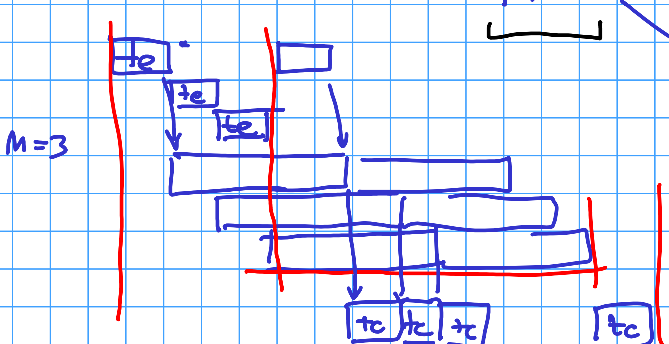
par degree

$$T_c(n, m) \cong T_s(n) \cdot m$$

tasks

parameters
- n: par degree
- P_{sched}: scheduling policy for E
- P_{coll}: gathering policy for C

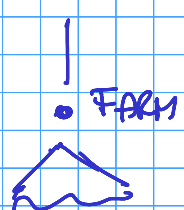
$$T_c(n, m) = nT_e + \frac{m}{n}T_w + T_c$$



$m \gg n$

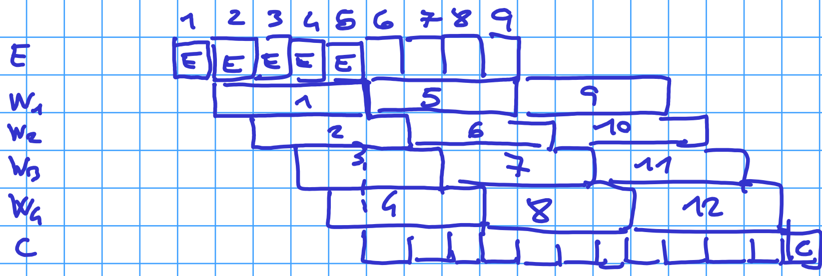
$nT_e + T_c$ negligible

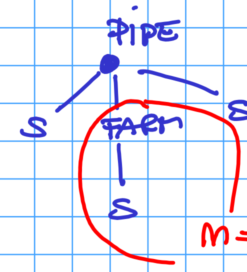
$$m\left(\frac{T_w}{n}\right) = m\,T_s(n)$$



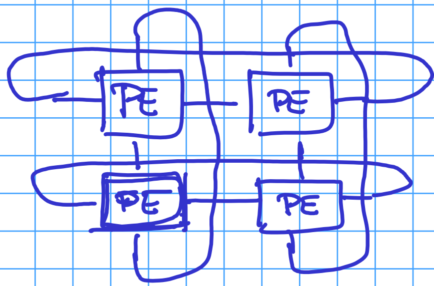
n such that $\max\left\{T_e, \frac{T_w}{n}, T_c\right\}$ is not T_w/n

$$\rightarrow \quad n = \left\lceil \frac{T_w}{\max\{T_e, T_c\}} \right\rceil$$



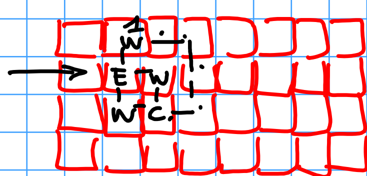


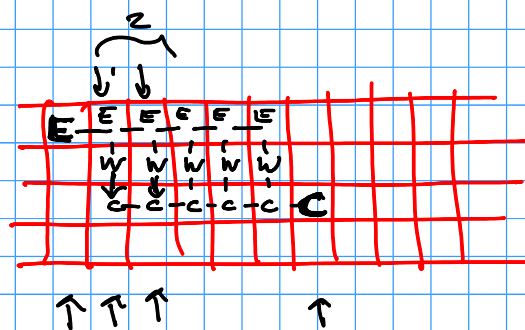
$m = \left\lceil \frac{T_w}{\max\{T_e, T_c\}} \right\rceil$

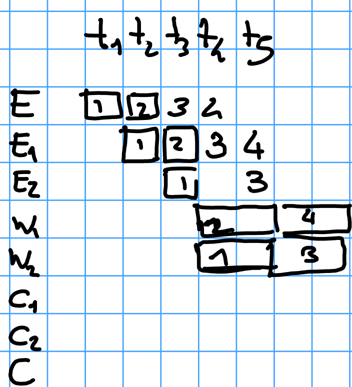


80 cores

8 x 10 mesh







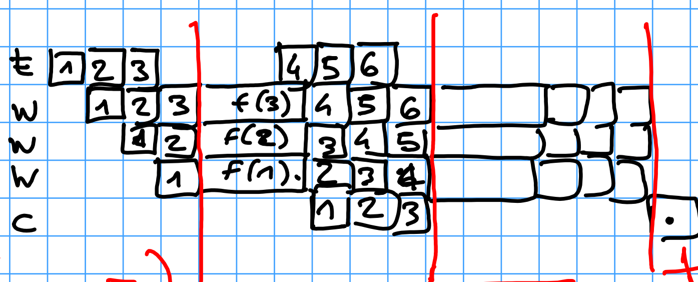
$T_e \cdot m + T_w + T_c \cdot m$

initial — compute a task ($\frac{m}{n}$ times) — at the end

$n >> m$

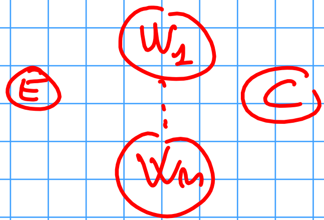
$T_c = \underline{(T_e + T_c)\,n} + \frac{m}{n} T_w$

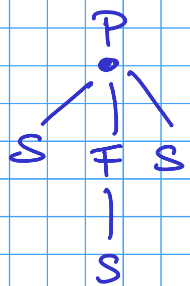
$\frac{m}{n}T_w = m\left(\frac{T_w}{n}\right)$



$(T_e + nT_e)$ $\qquad \frac{m}{n}(T_w + nT_e)$ $\qquad t_c$

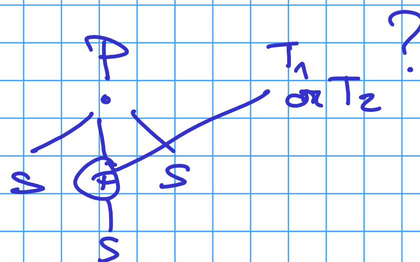
$T_c = (t_e + t_c + nt_e) + \frac{m}{n}(T_w + nt_e)$





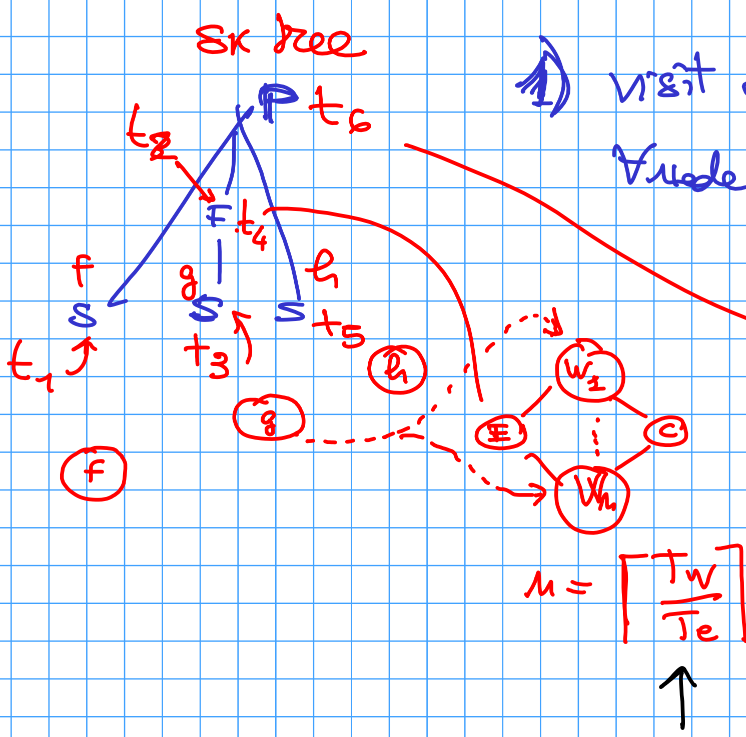
FARM	cow	T_1	o ⋮ o	$T_c = \dots$
FARM	cow	T_2	o–o–…–o	$T_c = \dots$

cow



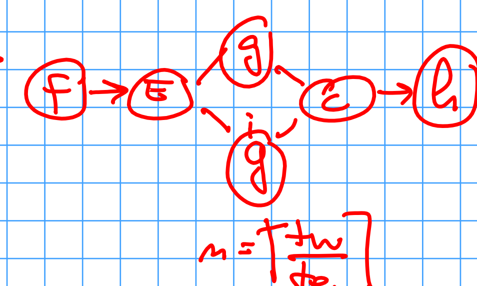
$n = X$
$P_{sched} = Y$
⋮

Hp: #templates × snel × target hw in templ library



1) visit the tree bottom up

∀node → fix optimal template parameters

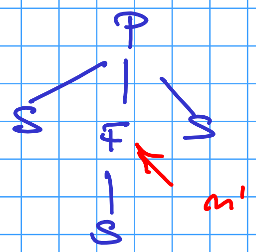


Pipe perf model to be considered

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

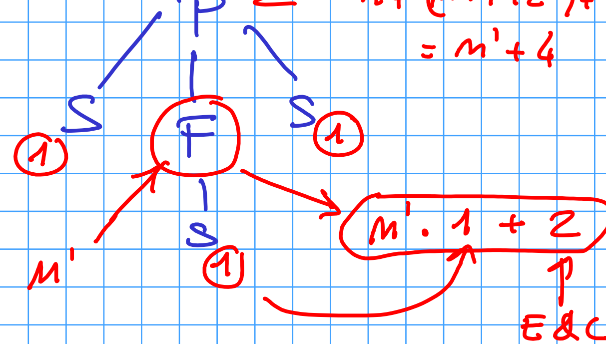
$T_S = \max\{t_f, T_e, t_h\}$

May require new n' for FARM matching



resources ?

$\Sigma = 1 + (m' \cdot 1 + 2) + 1$
$= m' + 4$



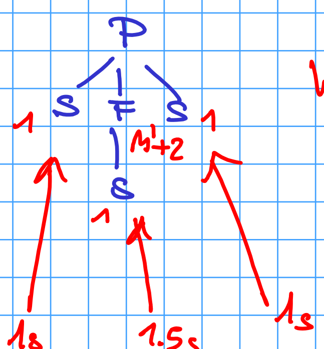
$\frac{T_w}{n} \gg \frac{T_e}{T_c}$

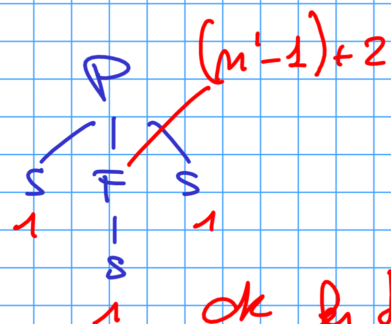
$T_s = \max\{T_e, \frac{T_w}{n}, T_c\}$

2) iteratively:

take away the smallest #PE such that the performance models of the templates still hold

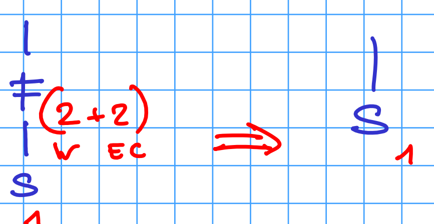
what if

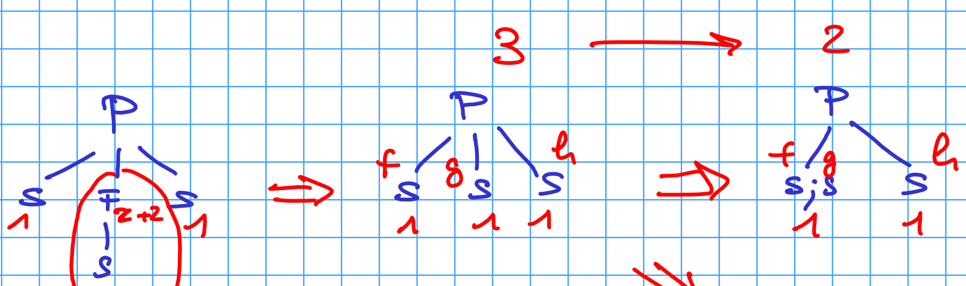




ok for the performance models

? $(m'-1)+2+2 <$ #available resources'





$t_f + t_g \neq t_h$