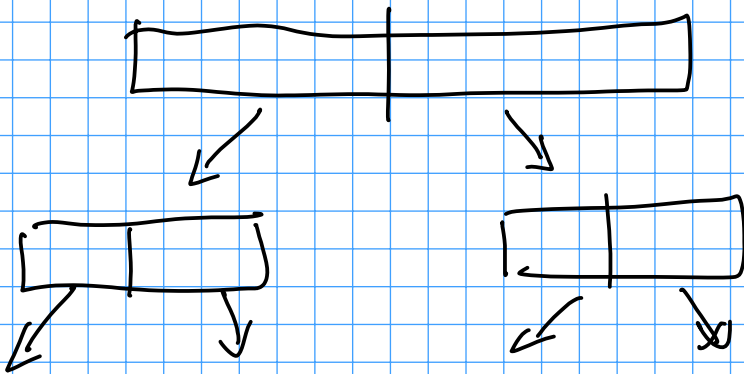


chk_for ()



Pattern

"recipe" modelling some useful/reusable parallel "form" (of computation)

DATA PARALLEL

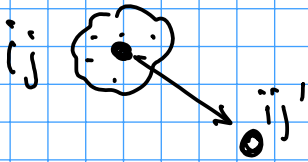
STREAM PARALLEL

CONTROL (TASK) PARALLEL

map (per for)

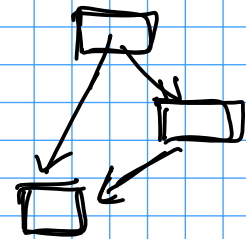
Reduce

stencil



pipeline form

task with dependencies



HIGH LEVEL PATTERNS

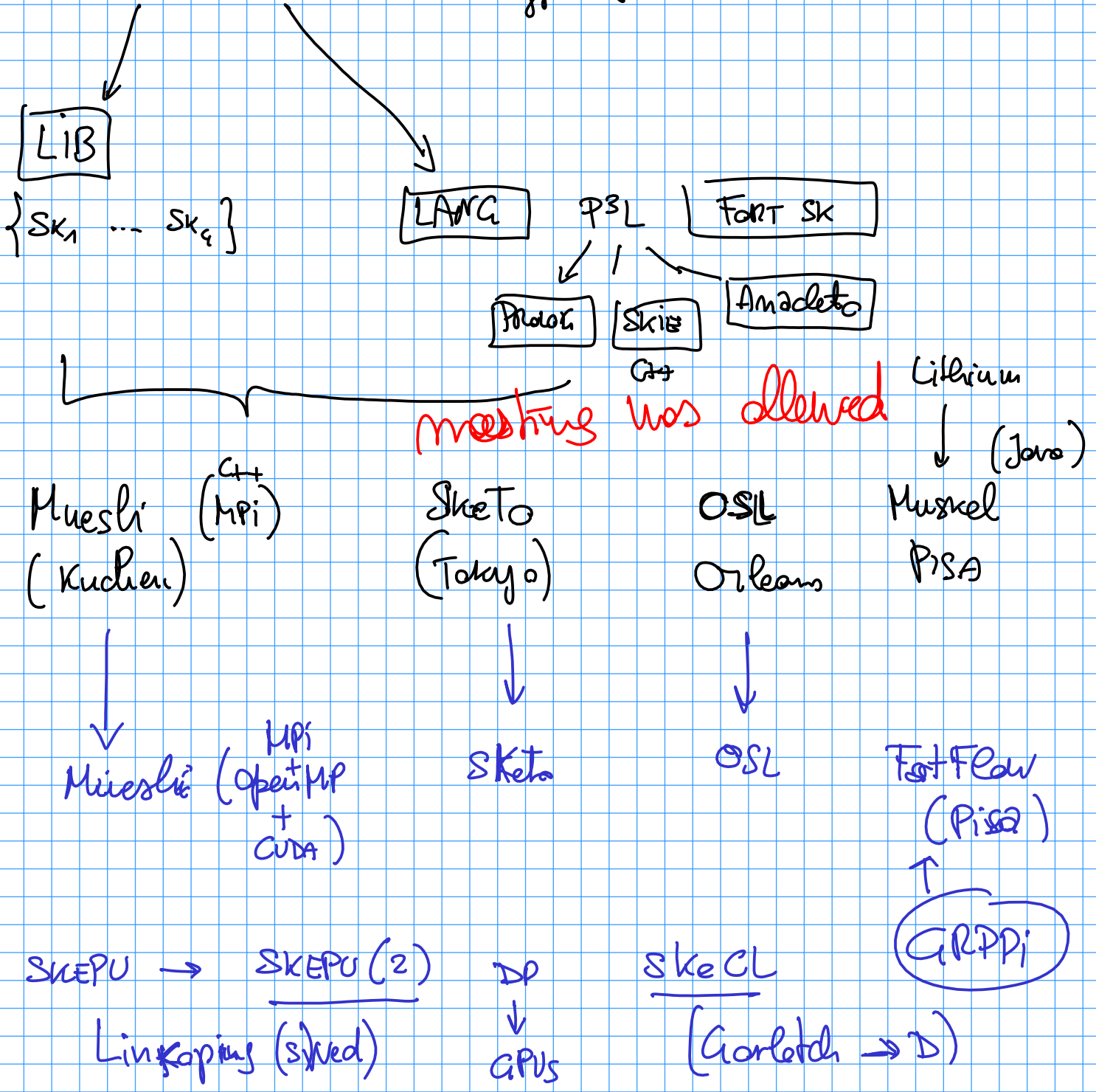
D&C

Pool Iteration

Algorithmic Skeletons (HPC)

→ tool (190)
 implementing a parallel function
 (f) (mw) ...

- parametric
- reusable
- efficient



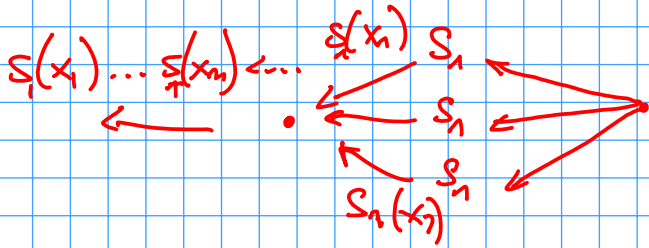
minimal disruptions principle

STREAM PARALLEL SKELETONS

pipeline $(S_1, S_2) : x_1 \dots x_n \dots \leftarrow$

$S_2(S_1(x_i)) \dots S_2(S_1(x_n)) \dots \leftarrow$

form $(S_1, mw) : x_1 \dots x_n \dots \leftarrow$

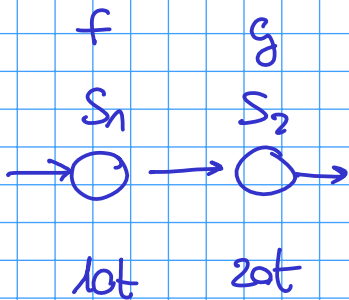


mw instances

$$T_{S_{pipe}} = \max \{ T_{S_i} \}$$

$$T_{S_{form}} = \max \left\{ \begin{array}{l} sched, \\ gather, \\ \frac{T_S}{mw} \end{array} \right\}$$

Performance models



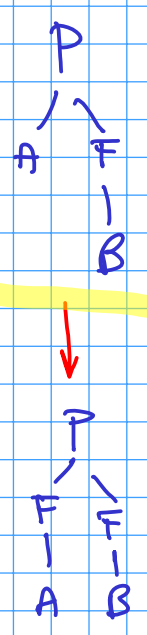
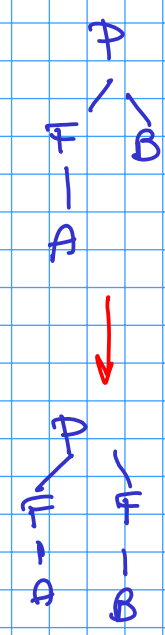
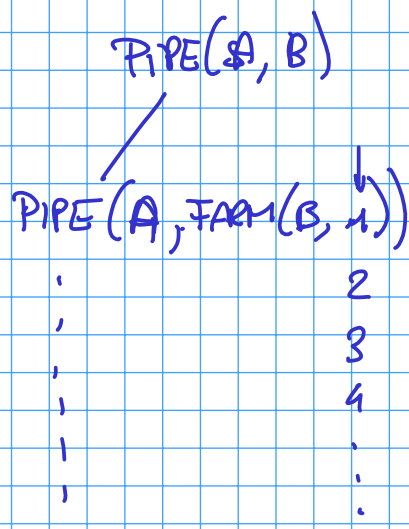
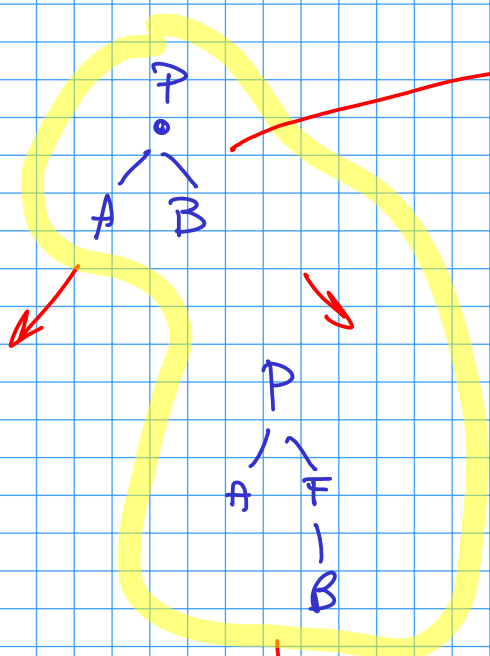
PIPE (seq(f), seq(g))

$$T_S = 2ot$$

PIPE (seq(f), form(seq(g), ?))
 $T_S = 1ot$
 $T_S = \frac{2ot}{2}$

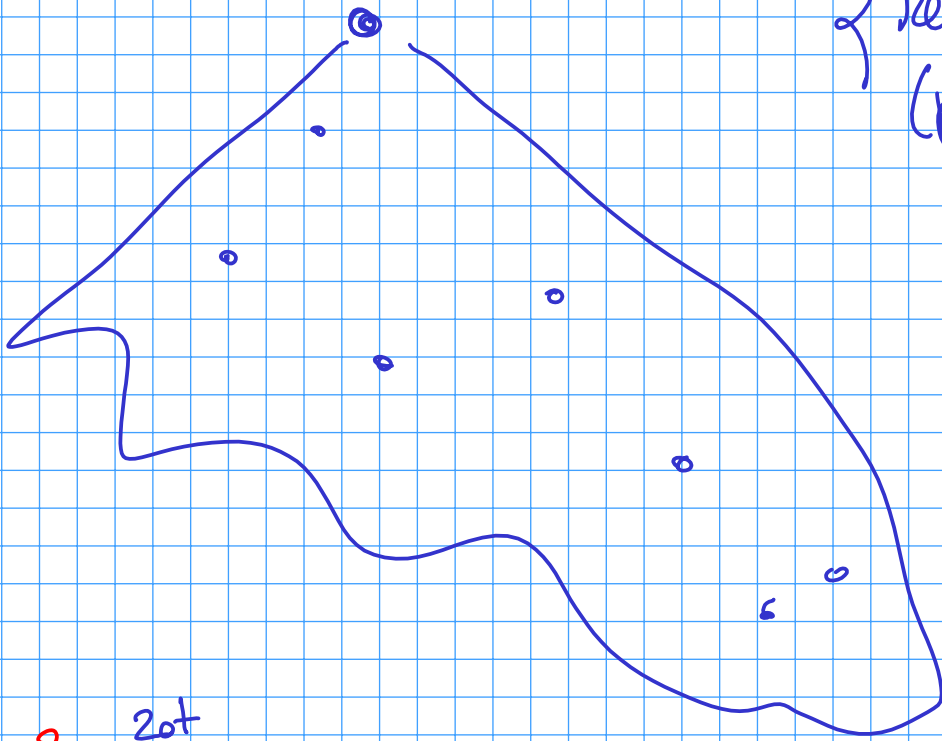
$S_1 \equiv form(S_1)$
 $pipe(S_1, S_2) \equiv comp(S_1, S_2)$
 ↑
 seq comp

$\text{PIPE}(\text{Seq}^A(A), \text{Seq}^B(B))$

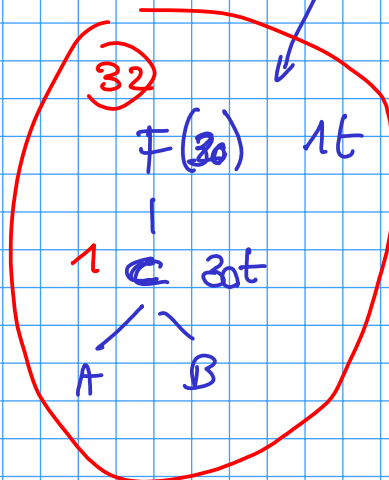
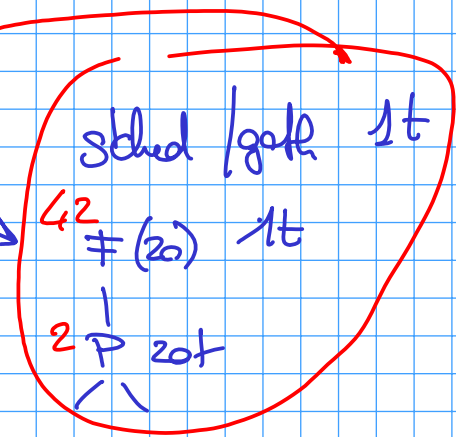
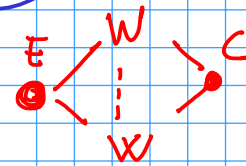
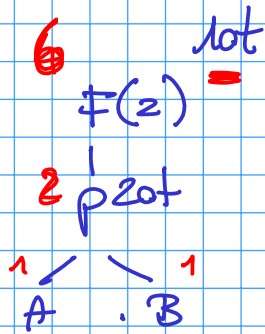
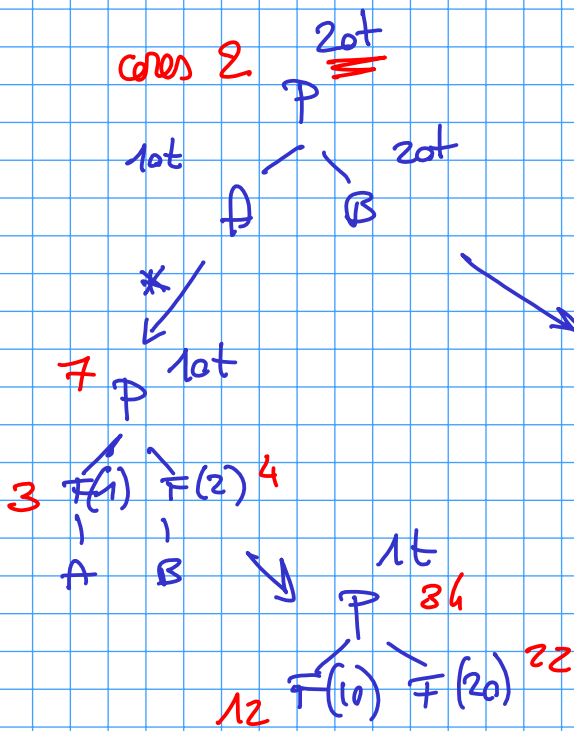


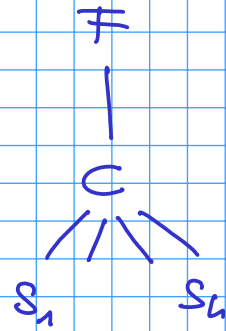
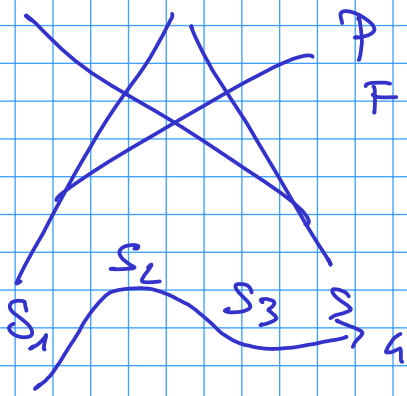
initial pattern expressions

rewrite rules
(preserve Funct Semantics)

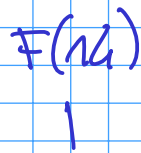
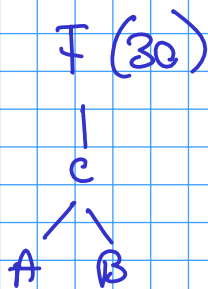
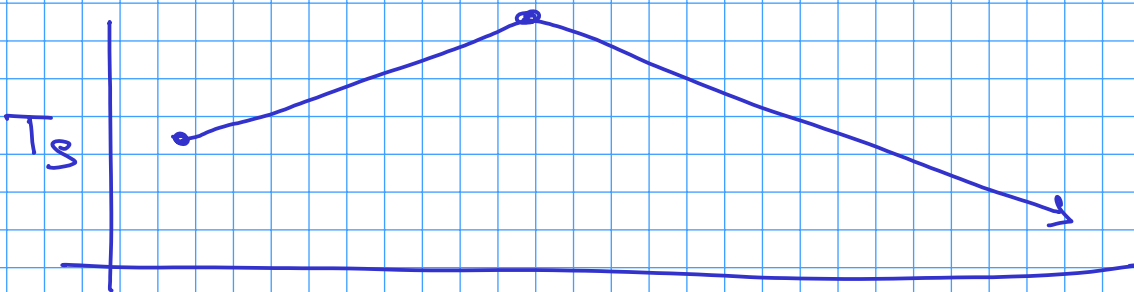


$mw \rightarrow mw+2$

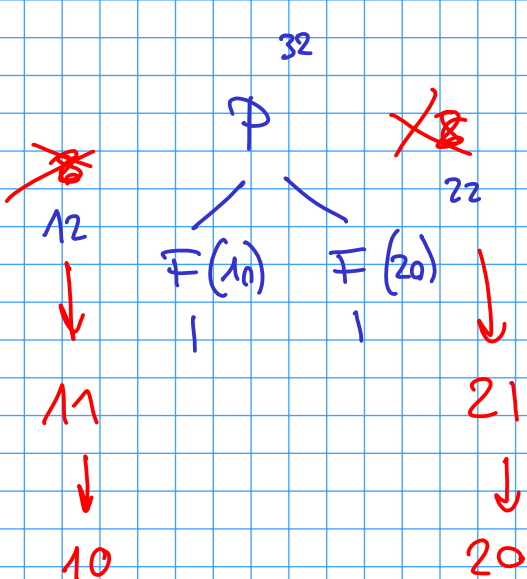




$$P(s_1, s_2) \rightarrow C(s_1, s_2) \rightarrow F(s_1, s_2)$$



fits the 16 core I have



$$T_{s_1} = \frac{T_{s_2}}{2}$$

$$\left\{ \begin{aligned} (mw_1 + 2) + (mw_2 + 2) &\leq 16 \\ mw_2 &= 2mw_1 \end{aligned} \right.$$