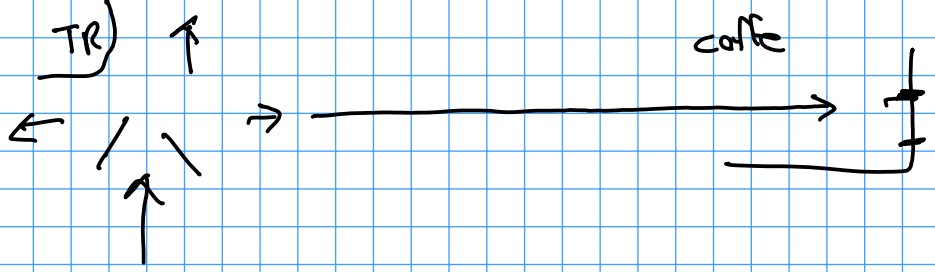
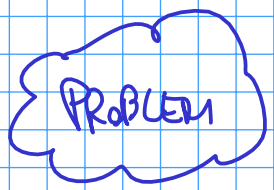


FINDING CONCURRENCY



Sol?

Introduce parallelism

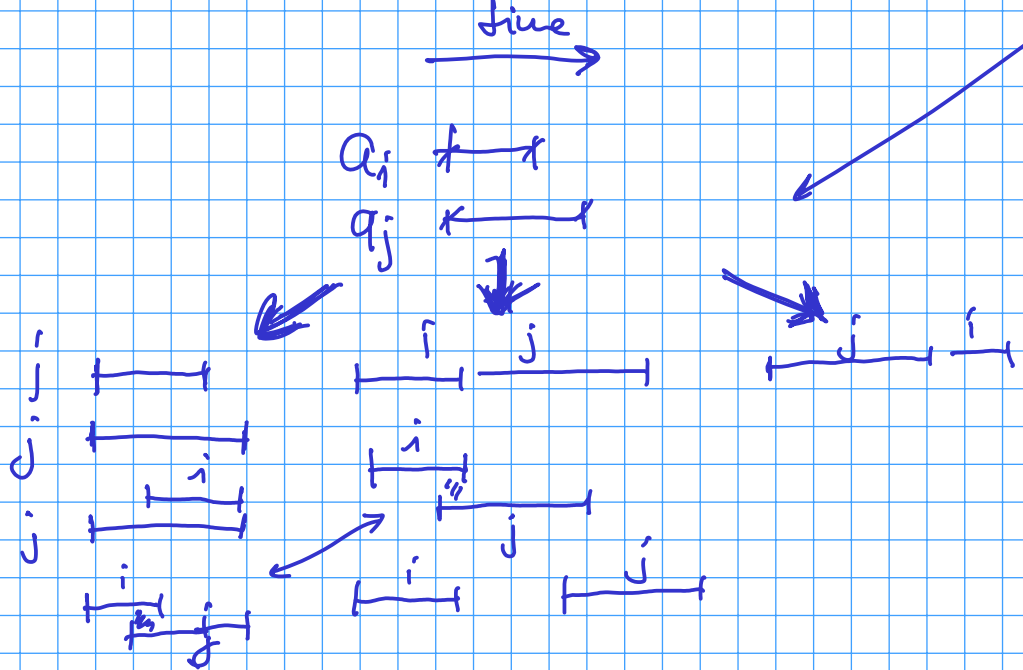
understand what I can compute in parallel

a set of concurrent activities

$\{a_1 \dots a_n\}$

a_i may run at the same time a_j $(i \neq j)$

parallel / distributed executors



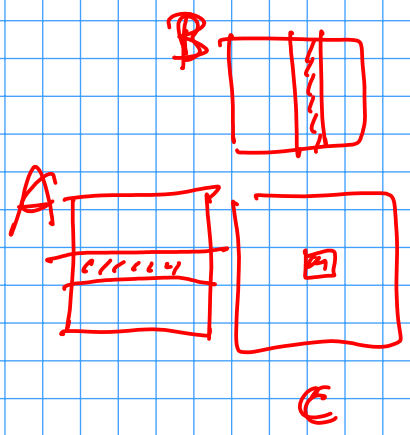
activity ≡ minimal unit of work

↓

which are the dependencies with other activities

Ex: $\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{m1} & \dots & & a_{mm} \end{bmatrix} \begin{bmatrix} b_{11} & & \vdots \\ \dots & & b_{mm} \end{bmatrix} = \begin{bmatrix} c_{ij} = \sum_k a_{ik} \cdot b_{kj} \end{bmatrix}$

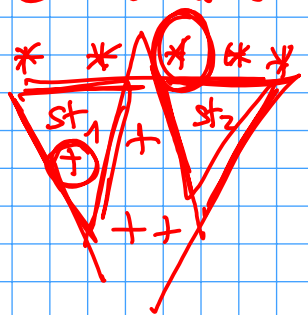
$A_{N \times N} \times B_{N \times N} = C_{N \times N}$



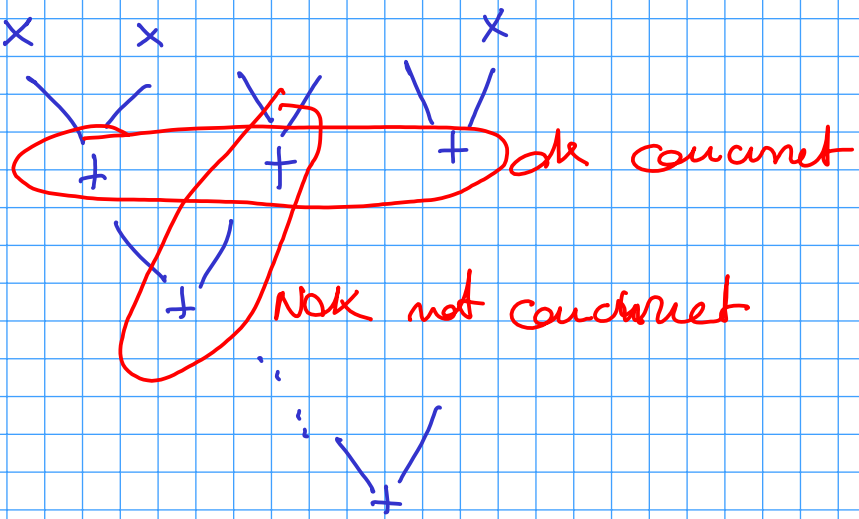
c_{ij} concurrent $c_{i'j'}$ $(i \neq i' \;\&\&\; j \neq j')$

$a_{ik} \cdot b_{kj}$ concurrent $a_{i'k'} \cdot b_{k'j'}$

"+"s & "×"s are concurrent?



$a_1 b_1 + a_2 b_2 + \dots + a_n b_n$



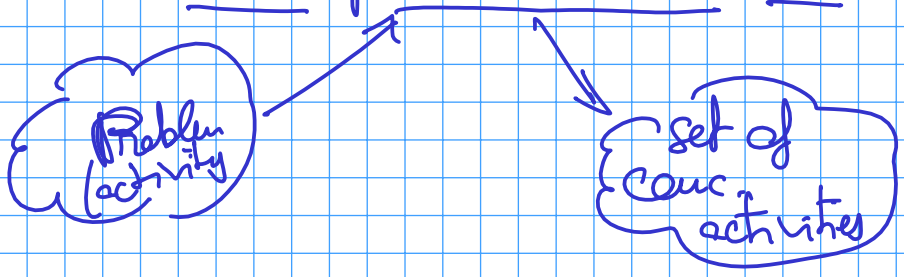
coarser grain than

any c_{ij} may be computed concurrently with the computation of a different c_{ij}

any mult. $a_{ik} \cdot b_{kj}$ may be comp. concurr. with the computation of $a_{ik'} \cdot b_{k'j}$

finer grain than

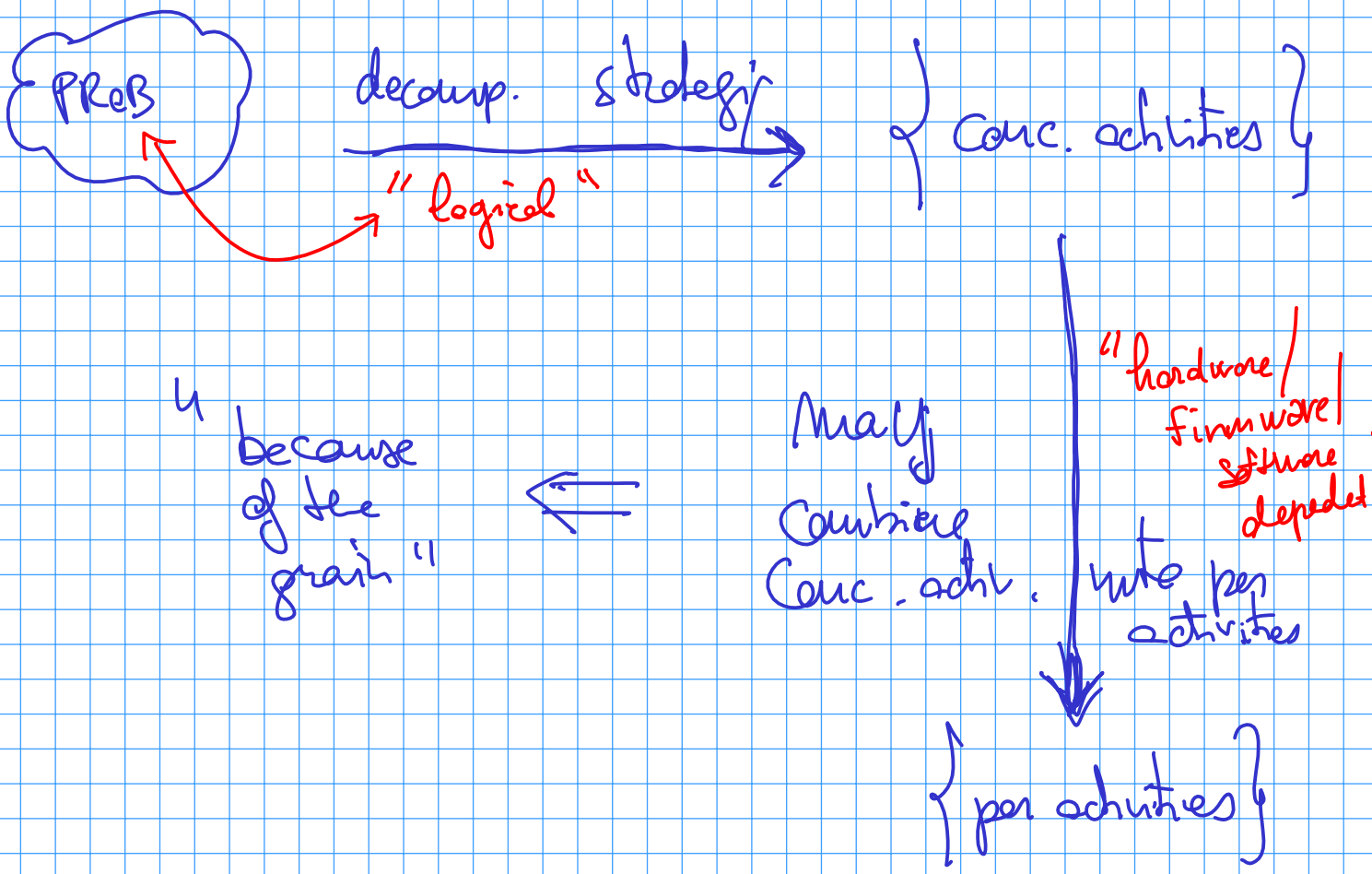
two decomposition strategies

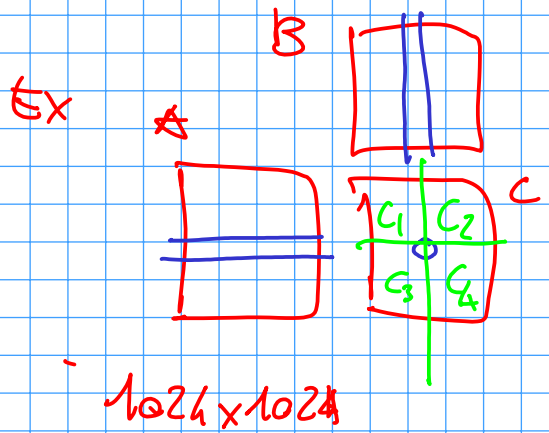


"grain" of the decomposition

↓

ratio in between the time I spend to compute a concurrent activity (in parallel) & the time needed to do this in parallel





decomp

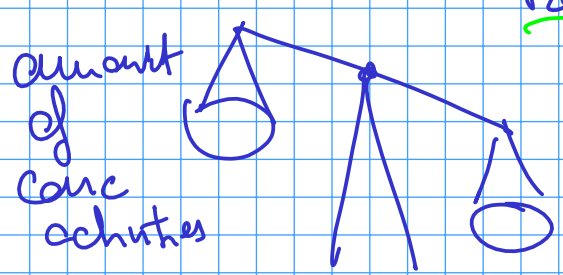
Couc. activities

$a_{i*} \cdot b_{*j}$

Recomp

Run multicore (threads)

2 cores with 4 contexts

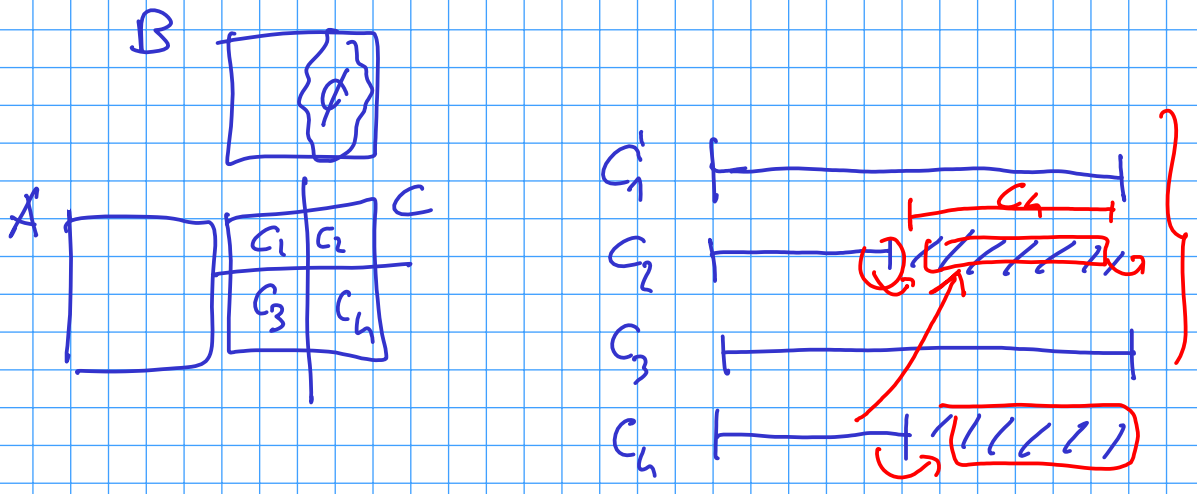


Scheduling

parallel overheads

C_1, C_2, C_3, C_4

Match the amount of parallel executors



$\#\{\text{conc act}\} >> \#\{\text{por executors}\}$

excess parallelism

rather than $\#\{\text{por act}\} = \#\{\text{por executors}\}$

I choose more por activities

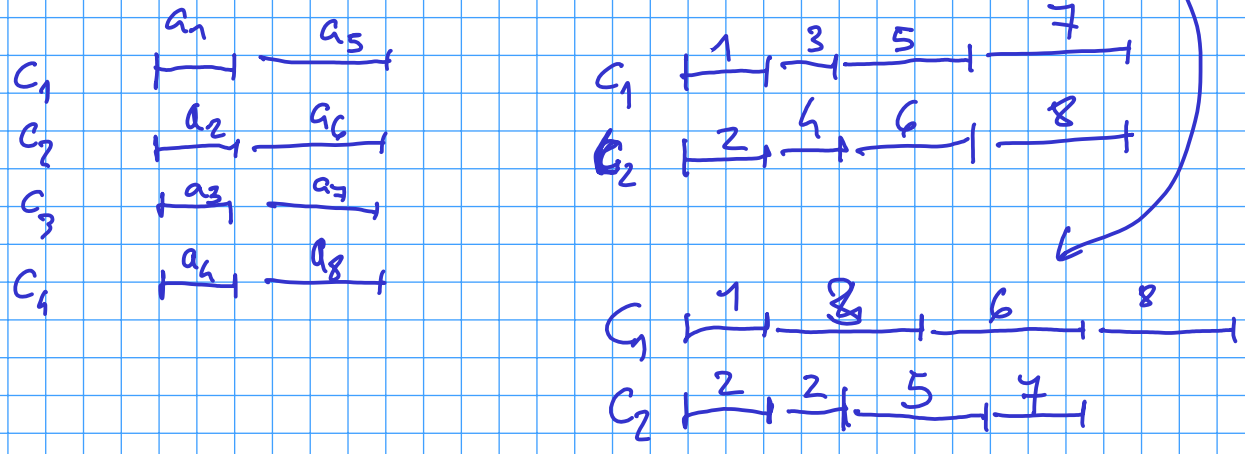
$\#\{\text{por act}\} > \#\{\text{por executers}\}$

$\{\text{por act}\}$ $a_1, a_2, a_3, a_4, \vdots, a_8$

a_1, a_2, a_3 } 2t to complete

$a_4, \vdots, a_8$ } 4t to complete

	1	2	3	4	5	6	7	8
	2	2	4	2	4	4	2	4



Orchestration of (conc) activities

1) execution coordination
   (start, stop, "wait" activities)

2) communications
   (shared memory, message passing)

3) synchronisations
   (point to point, "collective")

DATA DEPENDENCY between a_i & a_j (conc. activities)

in two "different places"

a_i produces some data needed by a_j

$R(a_i)$ = {data read by a_i}

$W(a_i)$ = {data written by a_i}

a_i before a_j

0) $R(a_i) \cap R(a_j) \neq \emptyset$
1) $R(a_i) \cap W(a_j) \neq \emptyset$ ANTIDEP @a_i; @a_j
2) $W(a_i) \cap R(a_j) \neq \emptyset$ TRUE DEP
3) $W(a_i) \cap W(a_j) \neq \emptyset$ OUTPUT DEP

$y = f(x, z);$
$k = g(y)$

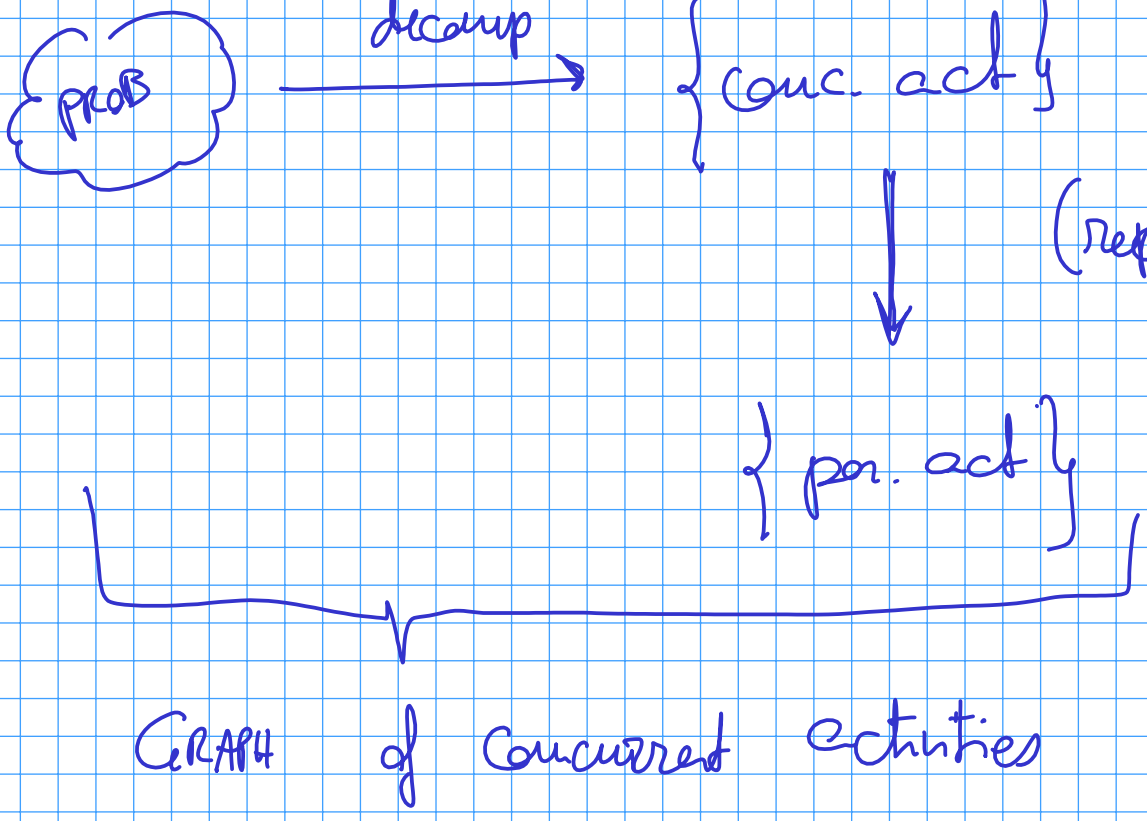
$W( ) = \{y\}$
$R( ) = \{y\}$

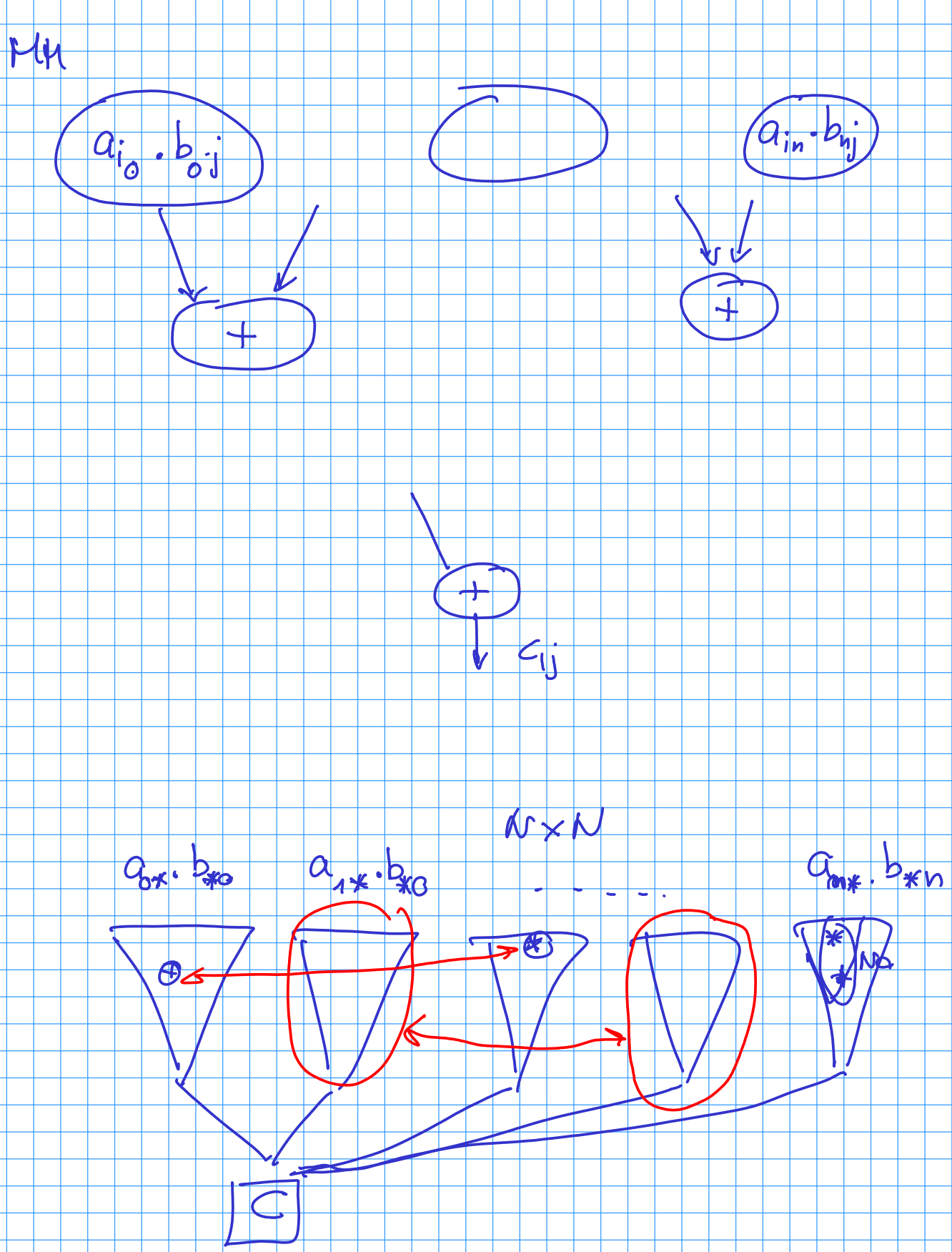
f & g are pure fnctes

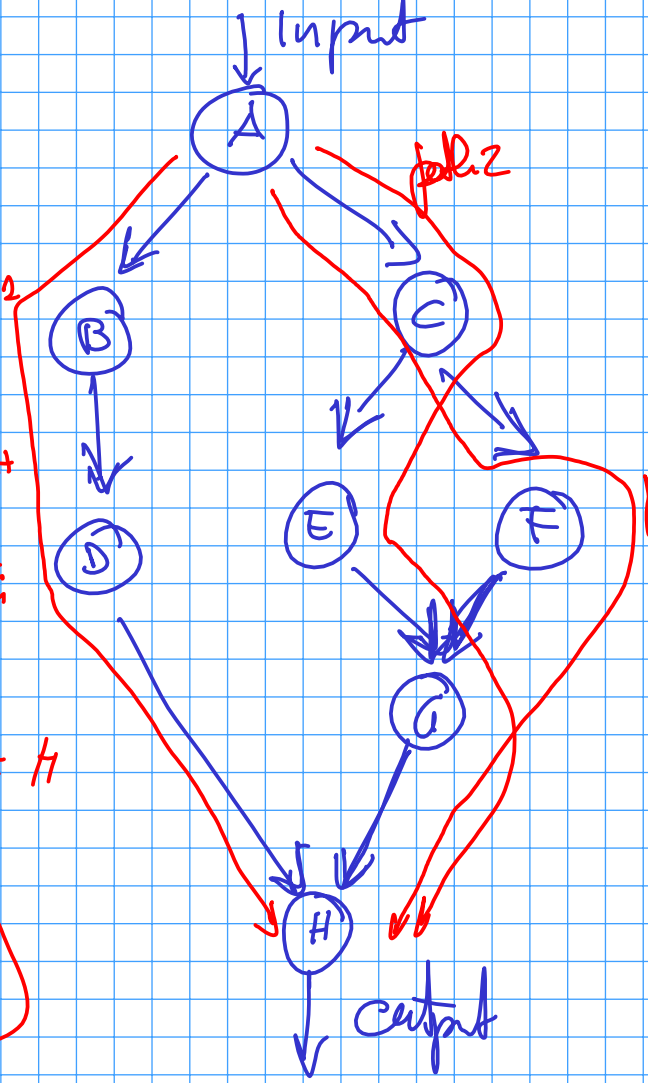
2) holds

a_i (f)
and
a_j (g)

must be computed serially
⇓
they are not conc activites




$a_{i_0} \cdot b_{0j}$
$a_{in} \cdot b_{nj}$
+
+
+
c_{ij}
$N \times N$
$a_{0*} \cdot b_{*0}$
$a_{1*} \cdot b_{*0}$
$a_{m*} \cdot b_{*n}$
C



what is the min time to complete? path 2

$t_1 = t_A + t_B + t_D + t_H$

$t_2 = t_A + t_C + t_E + t_G + t_H$

$t_3 = \dots + t_E + \dots$

t_F

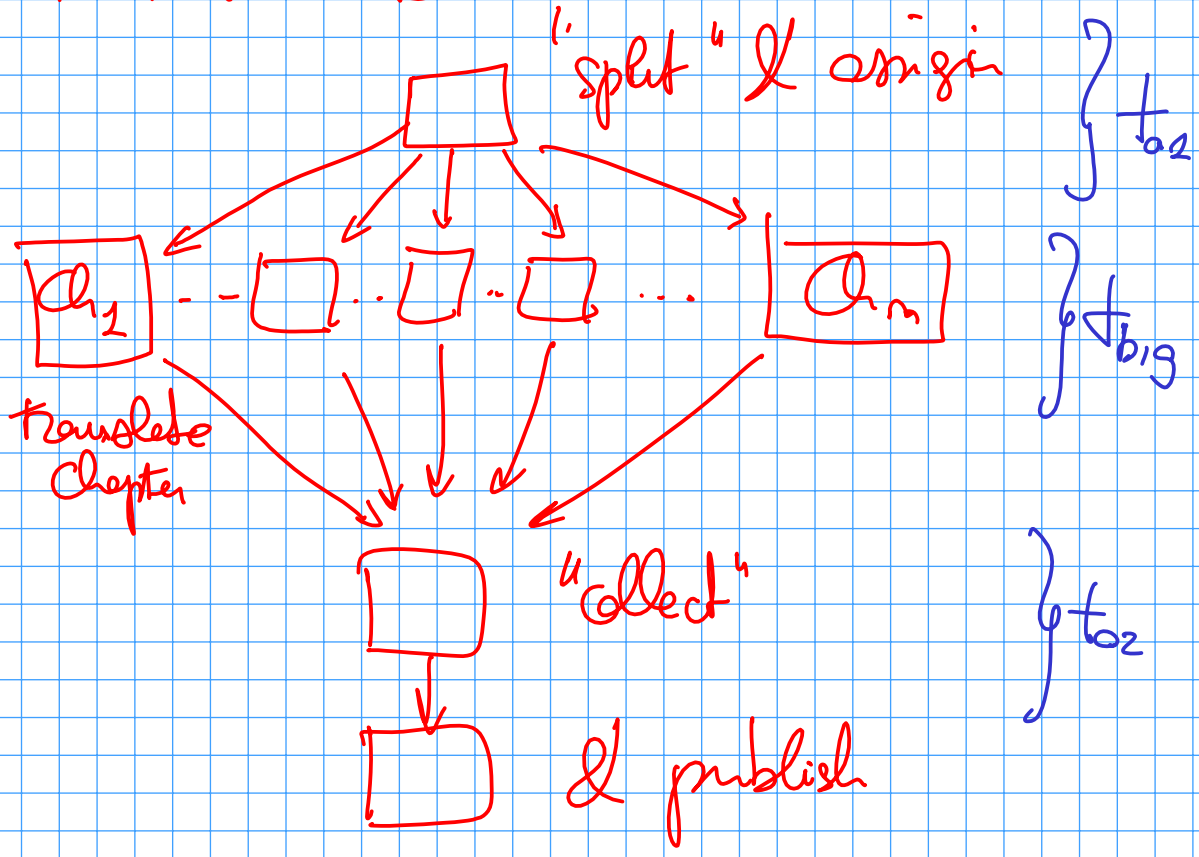
$\max\{t_i\}$

t_A t_B t_C

t_D .... t_H

What is the maximum parallelism degree?

Ex. TRANSLATING a Book



split(chap)

doall(chap)

collect'(chap)

VS

split(sentences)

doall(sentences)

collect(sentences)

Ex 2 : building object from pieces $p_1 \dots p_n$ added in order