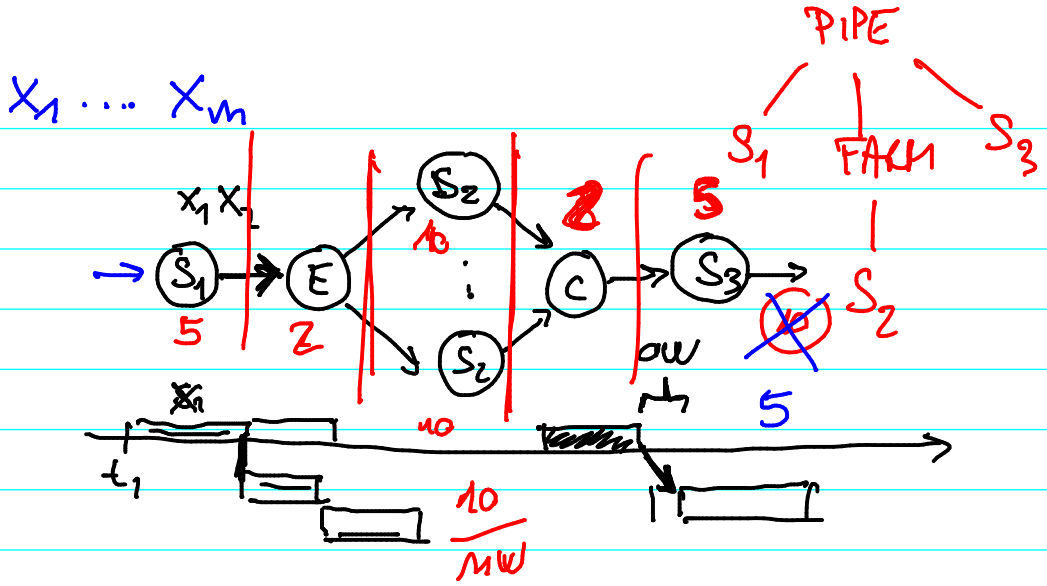
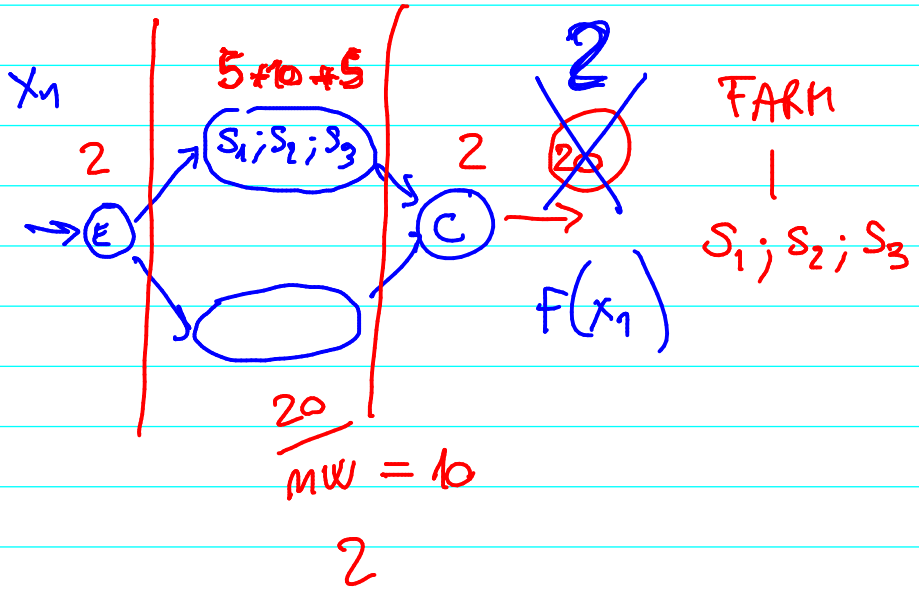
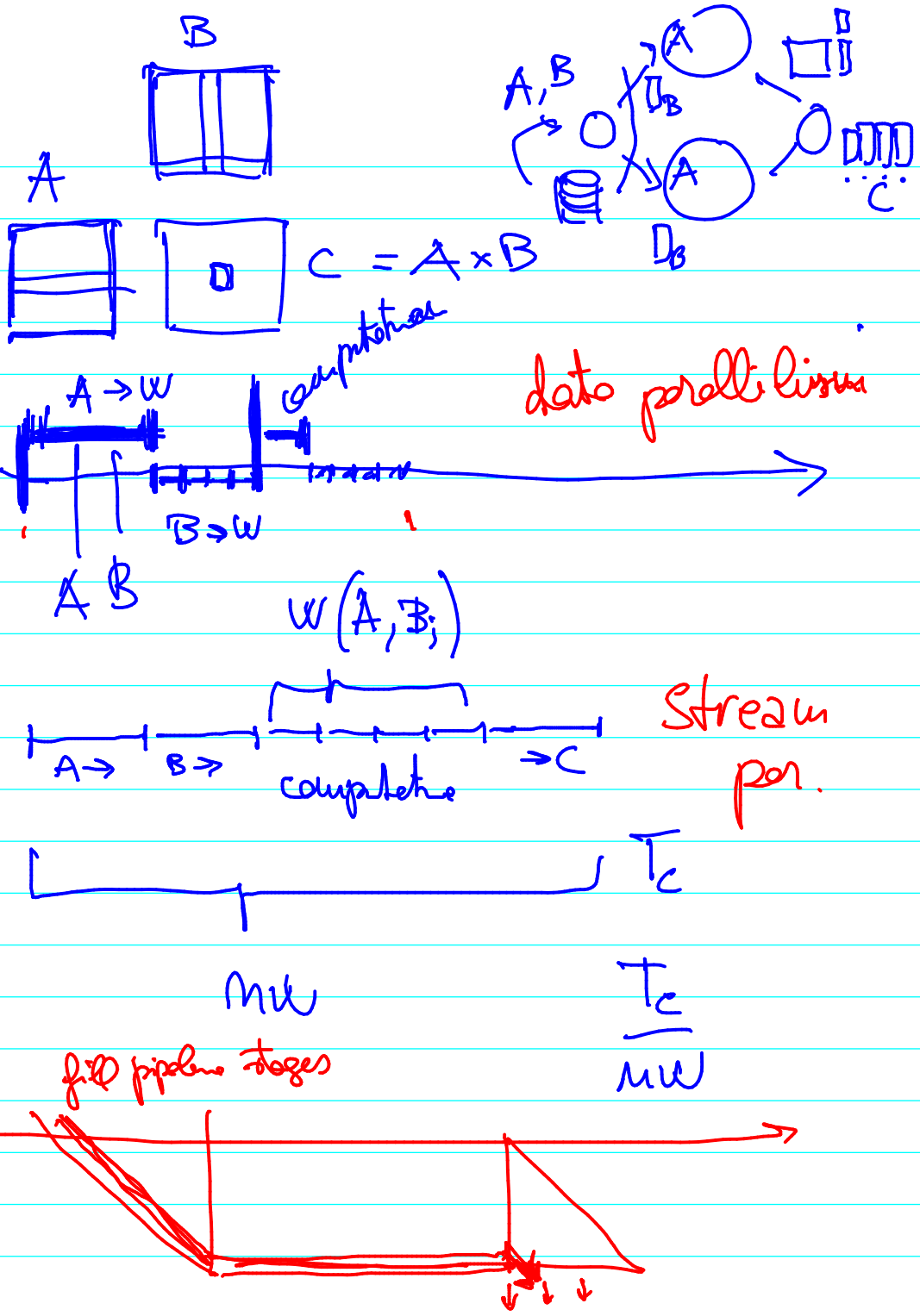

PIPE
S_1
FARM
S_3
$x_1 \dots x_m$
$x_1 x_2$
S_1
E
S_2
S_2
C
S_3
S_2
5
2
10
2
5
ow
5
x_1
t_1
10
10
MW


x_1
2
E
$S_1; S_2; S_3$
2
C
2
2
FARM
$S_1; S_2; S_3$
$F(x_1)$
5+10+5
20
MW = 10
2


B
A
C = A × B
A, B
A
A
C
A → W
B → W
completamento
dato parallelismo
A B
W(Ai, Bj)
A →
B →
completamento
→ C
Stream par.
Tc
n/w
Tc
n/w
fill pipeline stages

$f(x_1)$ $f(x_2)$
$f(x_3)$

1) top down

PIPE

S_1 FARM FARM

templ$_1$ templ$_1$

S_2 S_3

$\#PE = K = 10$

template library

FARM Template

nw = (..)

2) bottom up

P

S F F

5s

nw = 5

template$_1$ nw = f(...

nw = 10

S S

10s 20s

count resources

1 stage 1

7 stage 2

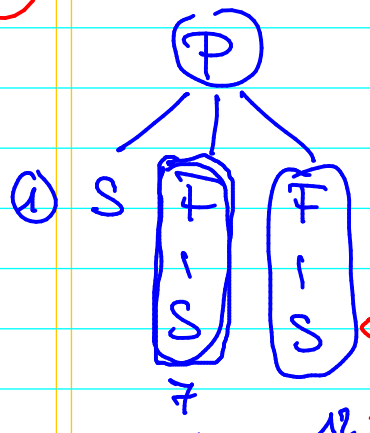
42 stage 3

} 20 PE me

③ reduction (PEs) process

top down

3.1 PIPE



↓

lower PEs in ∀ stages

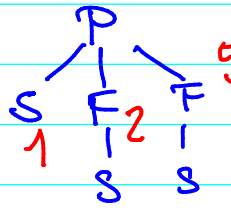
6 → Perf model

3.2 compute PEs

17

3.3 loop

eventually (after iterating the step 3)



≤ K = 10 PEs

Compile

PGM → PARAMETRIC OBJ CODE

Deploy

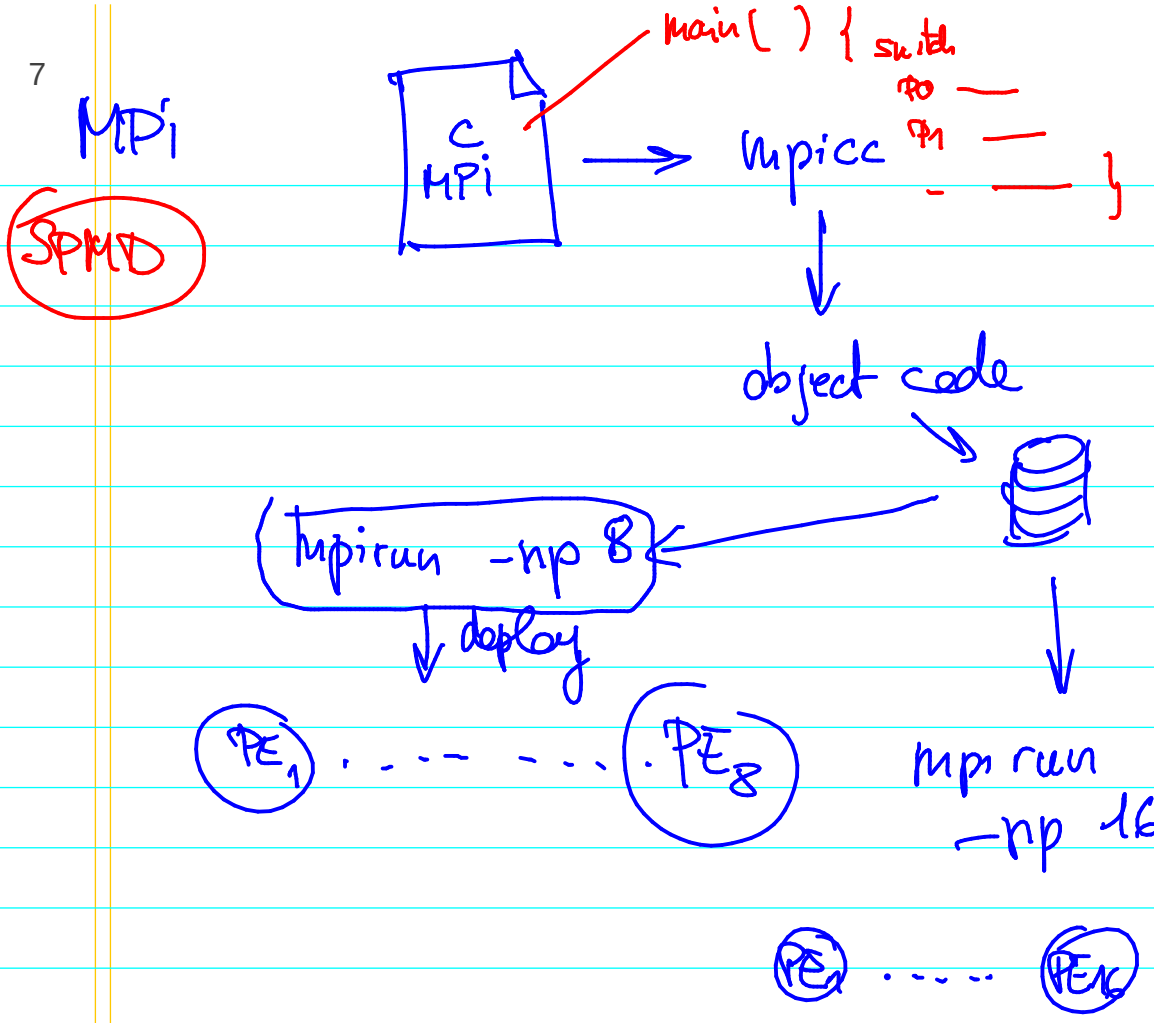
REDUCTION → GIVES PARMS

OBJ CODE (SENT TO REMOTE PES)

RUN TIME

EXECUTION

C`


MPI
SPMD
C
MPI
main( ) { switch
mpicc
object code
mpirun -np 8
deploy
PE1
PE8
mpirun
-np 16
PE9
PE16

$mw = f(\ \ldots\ )$

T_{seq_1} T_{seq_2} T_{seq_3}

sample inputs $\langle x_1 x_2 x_3 \rangle$

sample runs

$T_{seq_1} = t_{x_1} t_{x_2} t_{x_3}$

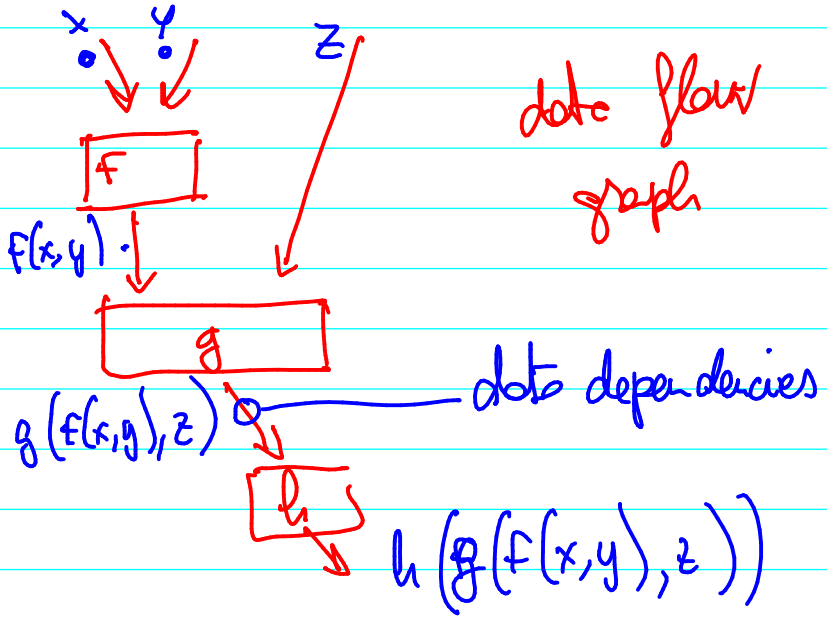
T_{seq_2}

Ask the user

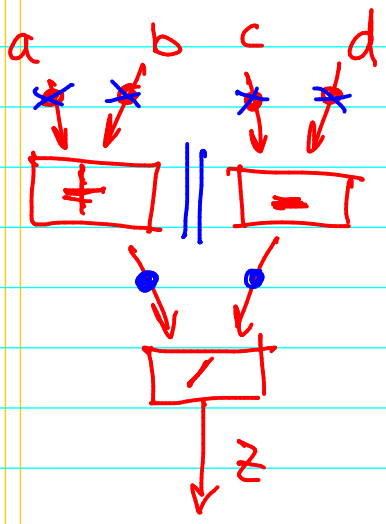
↳ approximate times

DATA FLOW

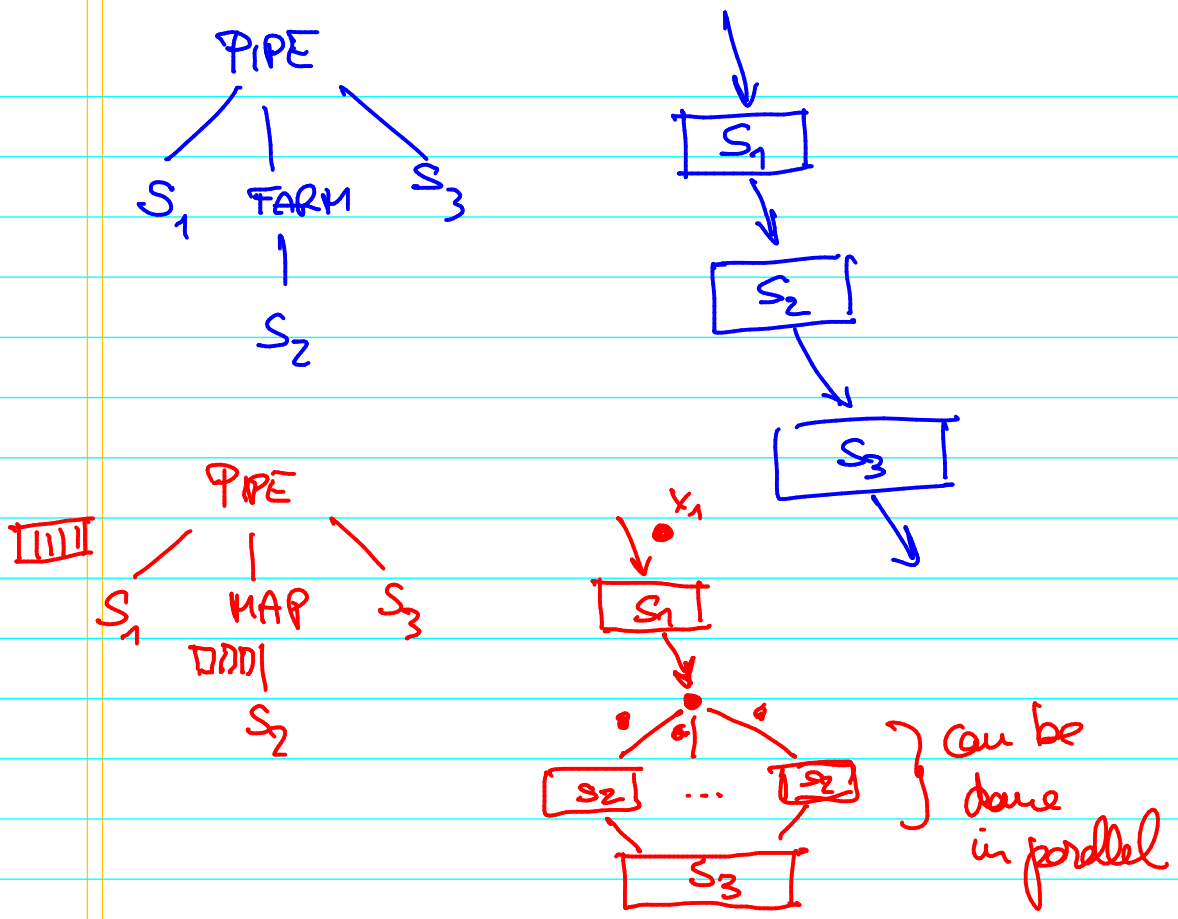
FIREABLE INSTRUCTION (NODE) = node with all inputs available

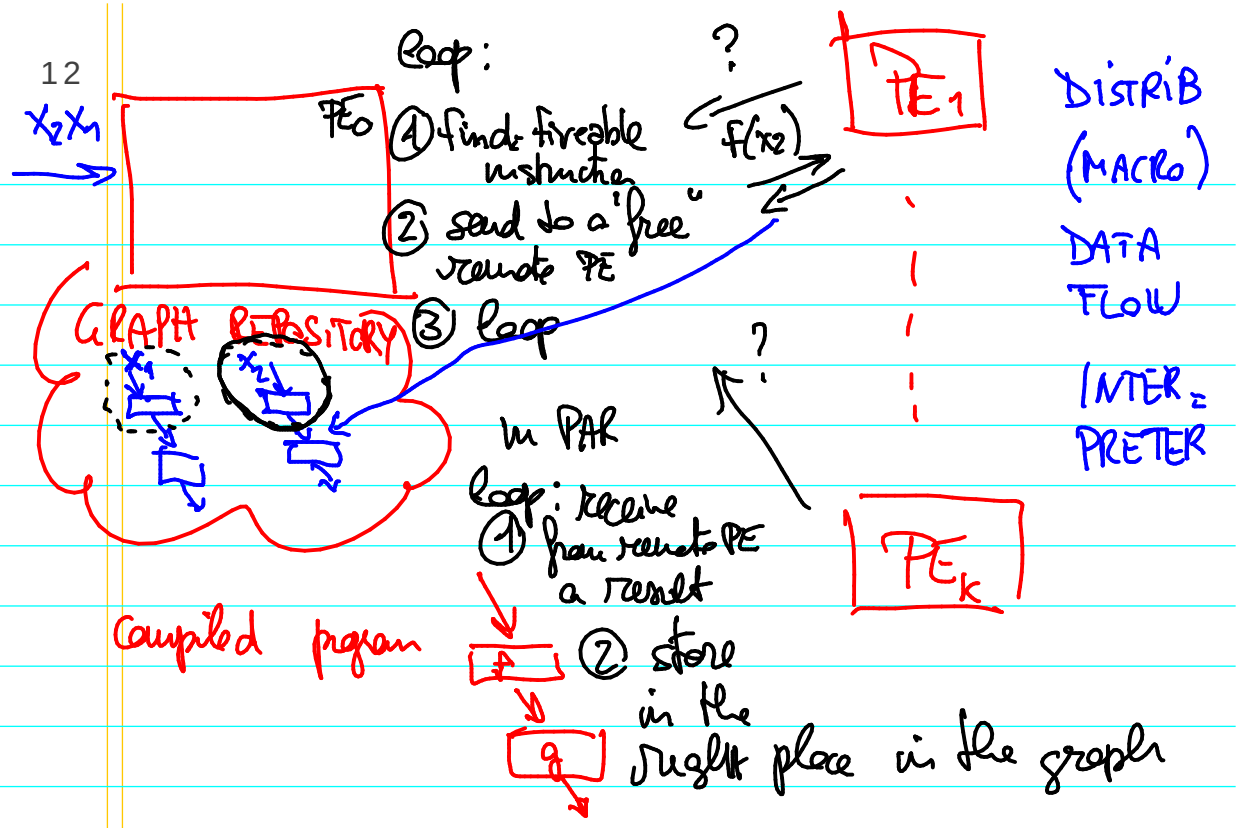


$z = (a+b)/(c-d)$

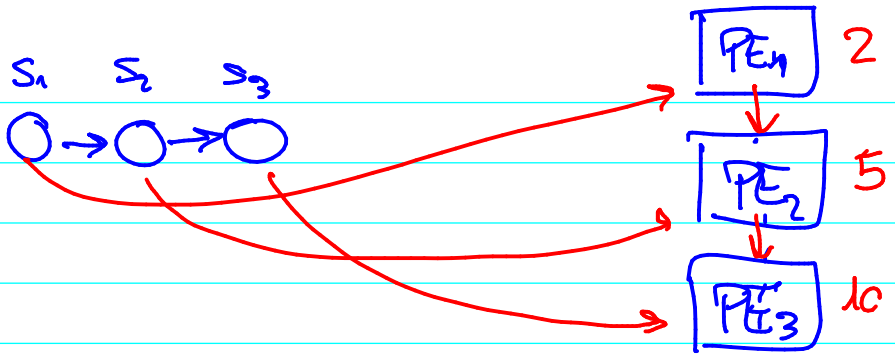



```
main( ) {
int a,b,c,d;
a= read( )
b=
c=
d=
z=(a+b)/(c-d)
printf( z);
}
```

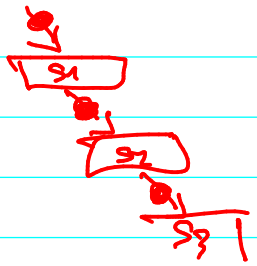

PIPE
S_1
FARM
S_2
S_3
S_1
S_2
S_3
PIPE
S_1
MAP
S_2
S_3
x_1
S_1
S_2
S_2
S_3
can be done in parallel


Loop:
PE0
① find a fireable instruction
② send to a "free" remote PE
③ loop
GRAPH REPOSITORY
x_1
x_2
x_2x_1
?
f(x2)
PE_1
PE_k
DISTRIB (MACRO) DATA FLOW INTER = PRETER
In PAR
loop: ① receive from remote PE a result
② store in the right place in the graph
Compiled program

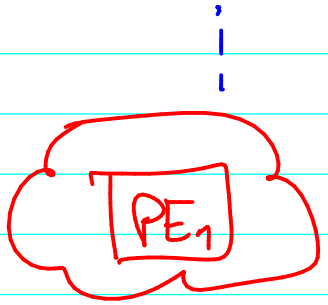
TEMPLATE



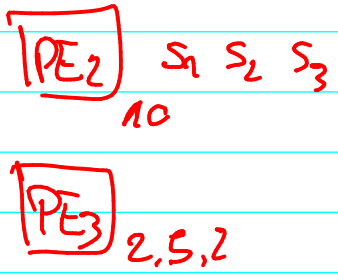
MACRO DATA FLOW

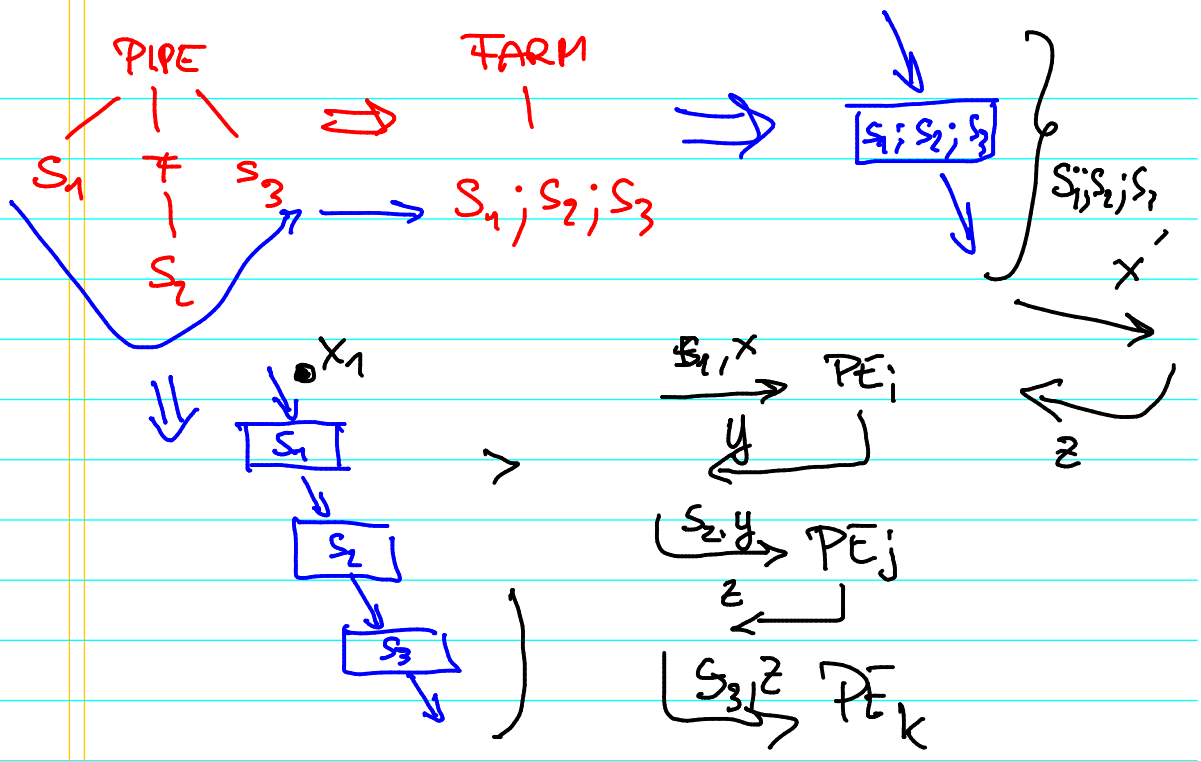


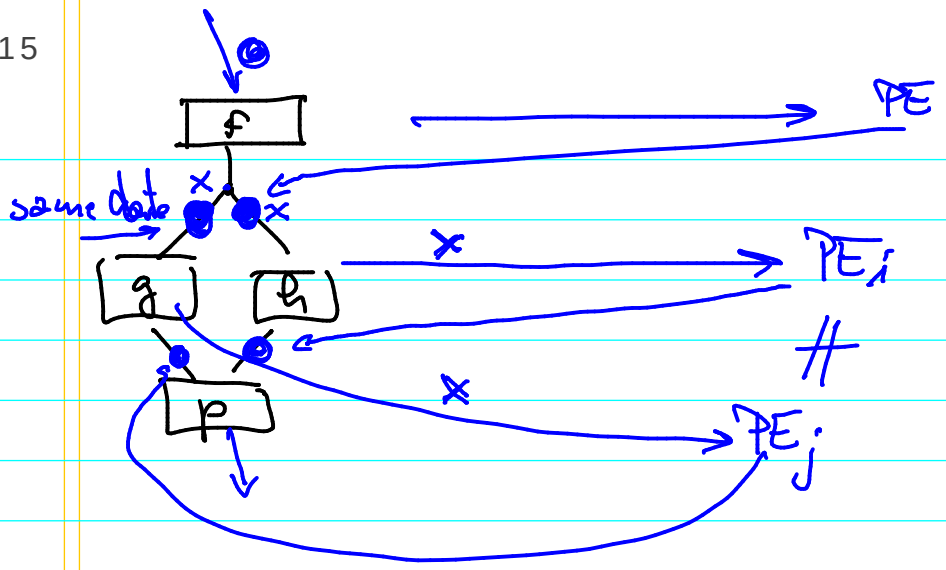
TASK GRAPH



REMOTE INTERPRETERS

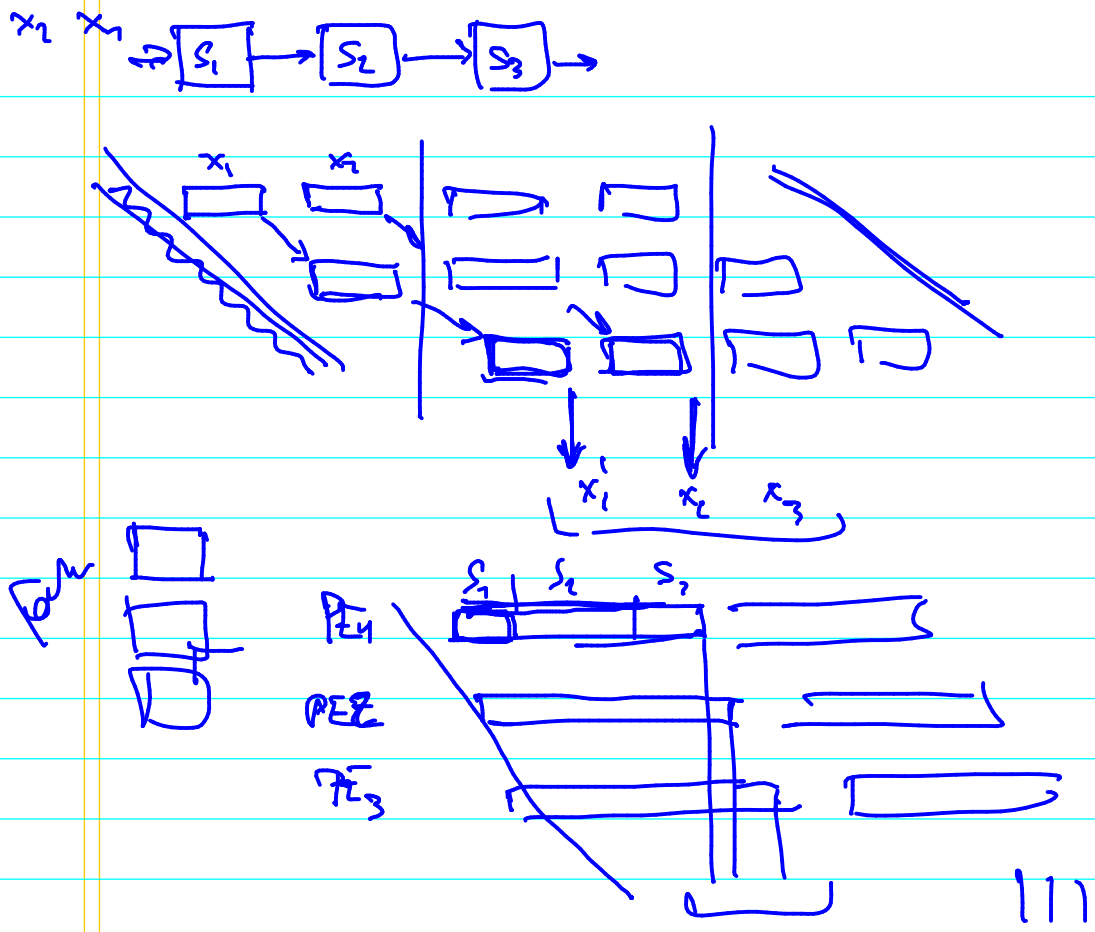



PIPE
FARM
S_1
S_2
S_3
$S_1;S_2;S_3$
$S_1;S_2;S_3$
$S_1;S_2;S_3$
x'
x_1
S_1
S_2
S_3
S_1, x
PE_i
y
z
S_2, y
PE_j
z
S_3, z
PE_k



1) send h, x to PE_i
x (cacheable)

2) send g, x to PE_i
↳ reference


x_2 x_1
S_1
S_2
S_3
x_1 x_1
x_1' x_2 x_3
S_1 S_2 S_3
P_{E1}
P_{E2}
P_{E3}