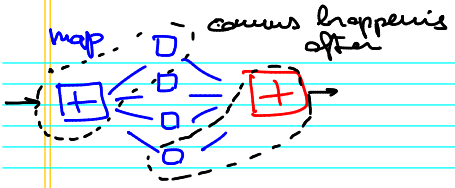
	DISTRIBUTED HW	MULTI CORE (SMP)
TEMPLATES or MDF INTERFR.	PROCESSES	THREADS (PROCESSES)
COMMS & SYNCHRO	INTER PE (TCP/IP SOCKETS, PROPRIETARY)	INTER THREAD (SH MEM + LOCKS SEMAPHORES SYS V)
LOCALITY	NICE/CLEVER PROCESS MAPPING	↘ NICE/CLEVER THREAD PINNING MEMORY ACCESS/ALLOC.
OPTIMIZATIONS	PROCESS MAPPING COMMUNICATION (GROUP COMMUNICATIONS)	SHARED MEM ACC (MINIMIZE SYNCHRO) MEMORY LAYOUT THREAD PINNING

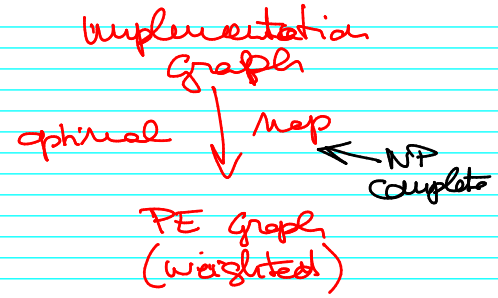
$t_{comm} = t_0 + d t_1$ (d: dim)

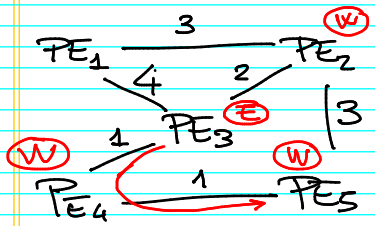
$comm_1$ d_1 $t_0 + d_1 t_1$

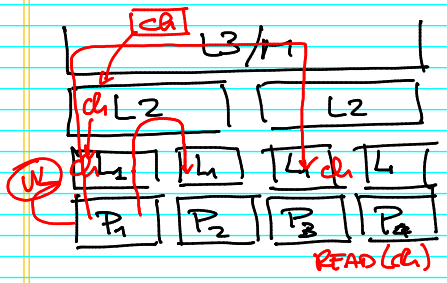
$comm_2$ d_2 $t_0 + d_2 t_1$

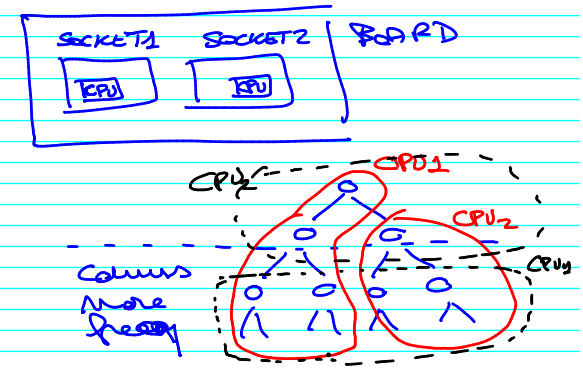
$t_0 + (d_1 + d_2) t_1$


map
comms happens after


implementation graph
optimal
map
NP complete
PE graph (weighted)


3
4
2
1
1
3
PE1
PE2
PE3
PE4
PE5

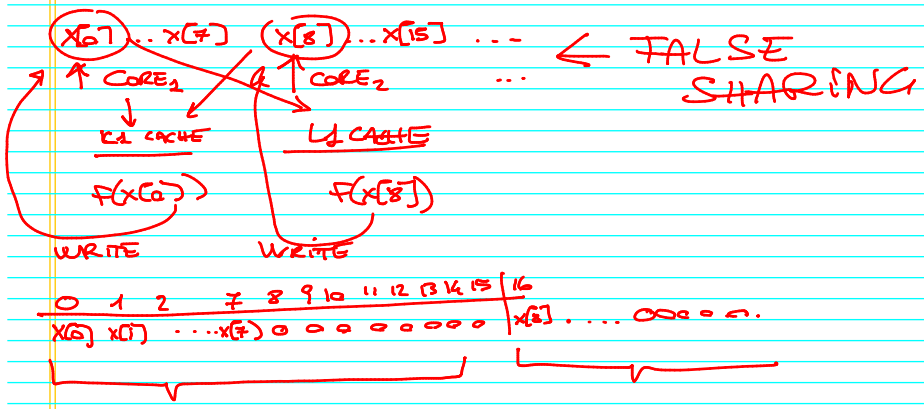

d1
L3
L2
L2
L1
L1
L1
L1
P1
P2
P3
P4
READ(d1)


2 QUAD CORE
SOCKET1
SOCKET2
BOARD
CPU
CPU
CPU1
CPU2
CPU3
comms more freely

P: LOAD ADDR(5)

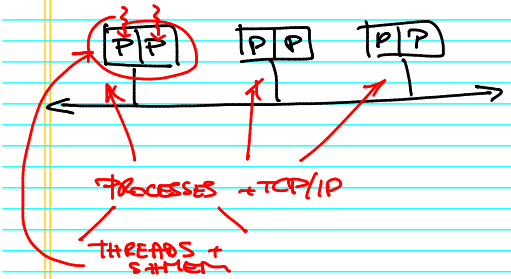
CACHE → 16 w lines

M[0,1,2.. ...14,15] → CACHE LINE



← FALSE SHARING

MULTICORES NETWORKED



3 PE with 2 CORES EACH

"SIMPLE" SOLUTION

ASSUME I HAVE A COW/NOW OF 3×2 PE

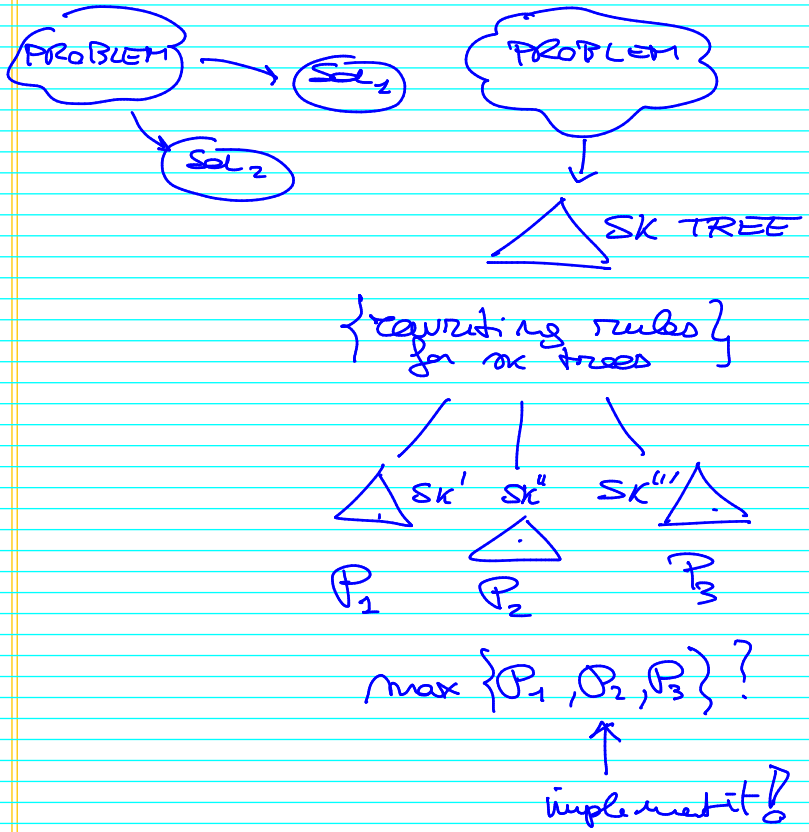
↓

use PROCESSES & TCP/IP

Also to forget MULTI CORES

This does not (in general) provide "optimal" performance

PROGRAM REWRITING (RULES)



REWRITING RULE SET

Δ : skeleton tree
{a single skeleton or a composition of skeletons}

FARM ELIMINATION RULE

farm(Δ) → Δ

$x_1 x_2 x_3 \ldots$ → $\Delta(x_1)\,\Delta(x_2)\ldots$

FARM INTRODUCTION RULE

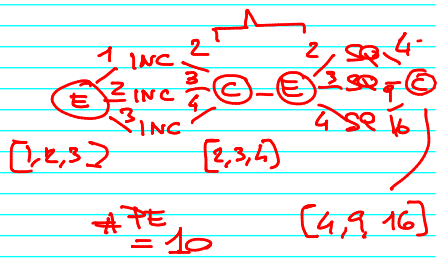
Δ → farm(Δ)

MAP/PIPE PROMOTION / REDUCTION

map(pipe(Δ_1 Δ_2)) ⇄ pipe(map(Δ_1), map(Δ_2))

$x_1 x_2 x_3$ → $\Delta_2(\Delta(x_1))$

pipe(Δ_1, Δ_2)



Δ_1 seq(INC)

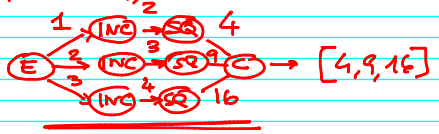
Δ_2 seq(SQ)

MAP(PIPE(SEQ(INC), SEQ(SQ)))

x_1 is a vector

$x_1 = [1,2,3]$
$x_2 = [3,2,1]$

#PE = 8



PIPE ASSOCIATIVITY

$PIPE(\Delta_1, PIPE(\Delta_2, \Delta_3)) \equiv PIPE(PIPE(\Delta_1, \Delta_2), \Delta_3)$

COMP $\not\sim$ PIPE

↳ BUT USING the SAME PES

a sort of SEQ composition

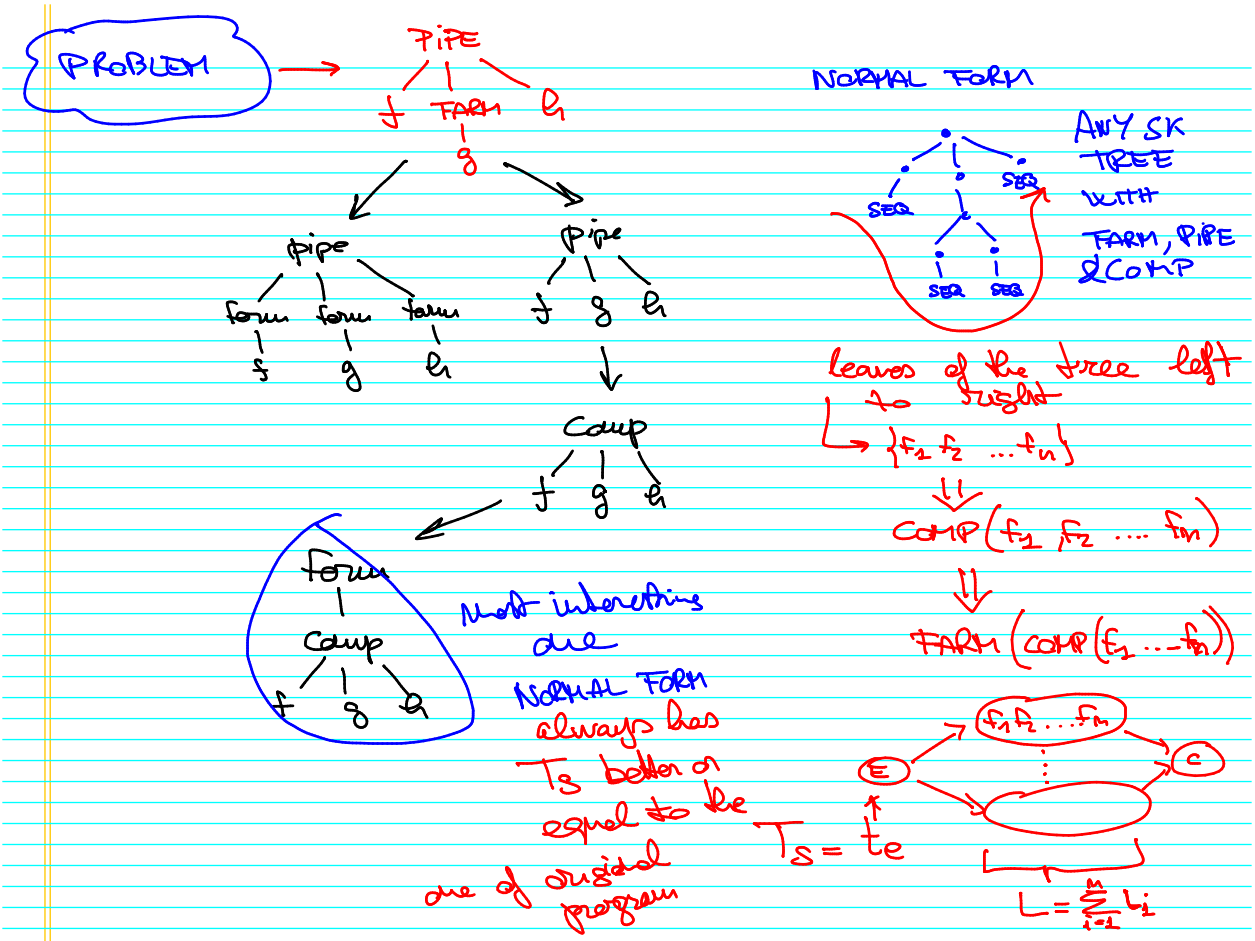
COMP(SEQ(f), SEQ(g))

$pipe(\Delta_1, \Delta_2) \rightarrow comp(\Delta_1, \Delta_2)$

$comp(\Delta_1, \Delta_2) \rightarrow pipe(\Delta_1, \Delta_2)$

$map(comp(\Delta_1, \Delta_2)) \rightarrow comp(map(\Delta_1), map(\Delta_2))$

$=$ $\leftarrow$ $=$


PROBLEM
PIPE
f
FARM
g
h
pipe
farm
farm
farm
f
g
h
pipe
f
g
h
comp
f
g
h
farm
comp
f
g
h
Most interesting one
NORMAL FORM always has T_s better or equal to the one of original program
NORMAL FORM
SEQ
SEQ
SEQ
SEQ
ANY SK TREE WITH FARM, PIPE & COMP
leaves of the tree left to right
$\{f_1 f_2 \dots f_n\}$
$COMP(f_1, f_2, \dots, f_n)$
$FARM(COMP(f_1 \dots f_n))$
E
$f_1 f_2 \dots f_n$
C
$T_s = t_e$
$L = \sum_{i=1}^{n} L_i$