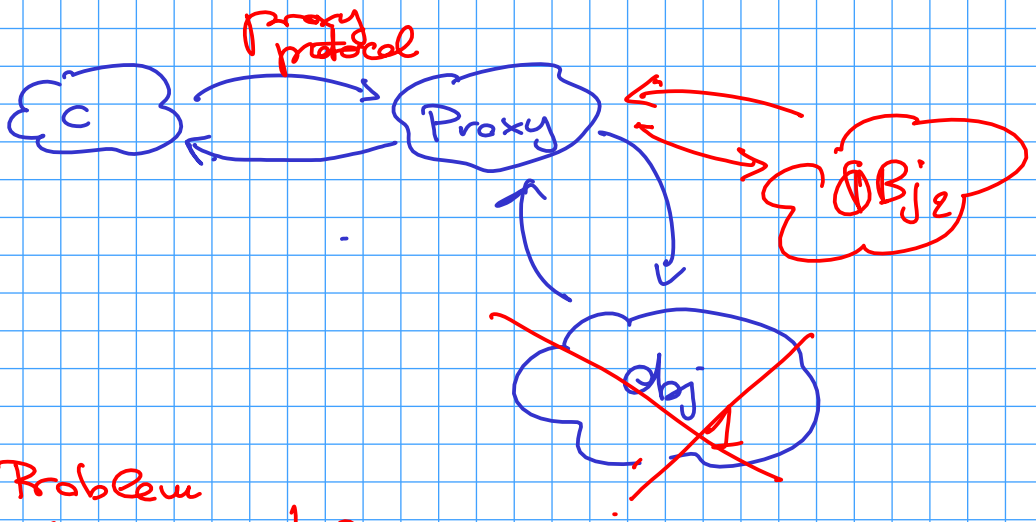
Design patterns SW engineering

Gamma et Al. "Design patterns"

Proxy



Façade

Recipes

Problem
Strategies & Policies
Typical implementation
Usage

Parallel design patterns

↳ Mattson Sanders Massingill

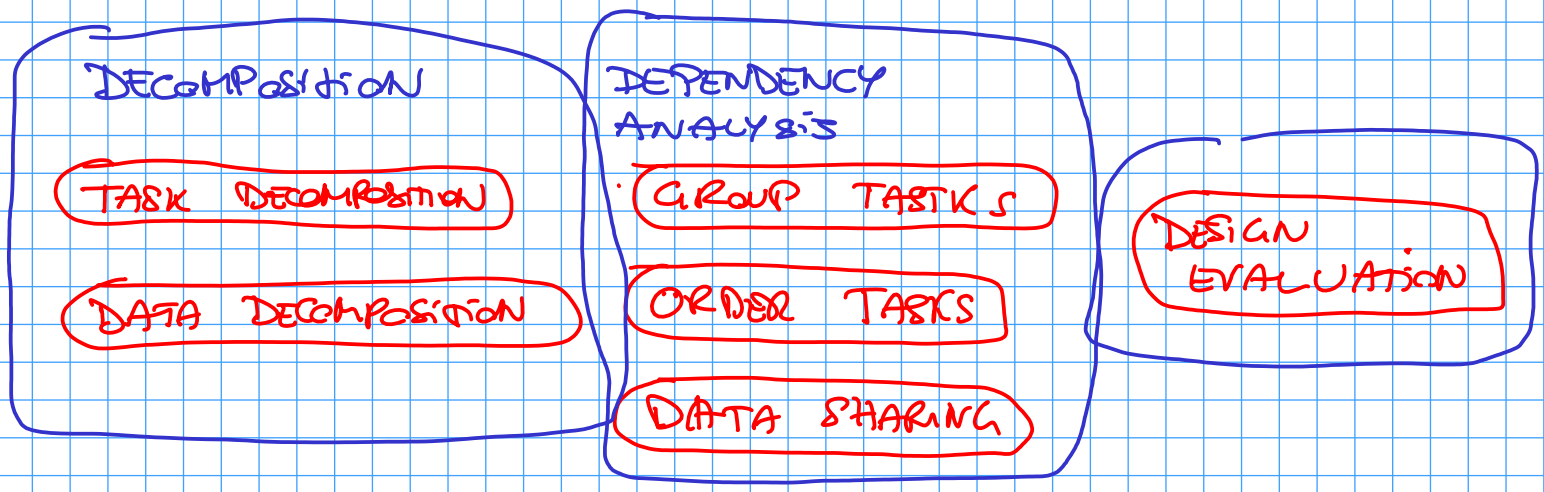
Set of features
- Problem — what kind of problem may be solved
- Context — scenarios
- Forces
- Solution

Structured design space

Finding concurrency
Algorithmic structure
} close to application domain

Supporting structures
Implementation mechanisms
} close to the hardware (Hw + Sw)

FINDING CONCURRENCY (DESIGN SPACE)



TASK DECOMPOSITION pattern

PROBLEM : How decompose problem into tasks that can run concurrently?

CONTEXT : —

FORCES : FLEXIBILITY EFFICIENCY SIMPLICITY

SOLUTION : evidence the actions needed to decompose the problem in tasks
dependencies in between the actions

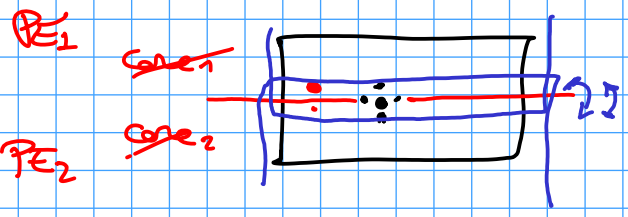
list of typical situations where I can look for tasks
- function or procedure calls

```
for(i= ...
  for(j= ...
    (fp)( a_i*, b_*j, c_ij )
```

- loop iterations
  independent iterations ⇒ tasks

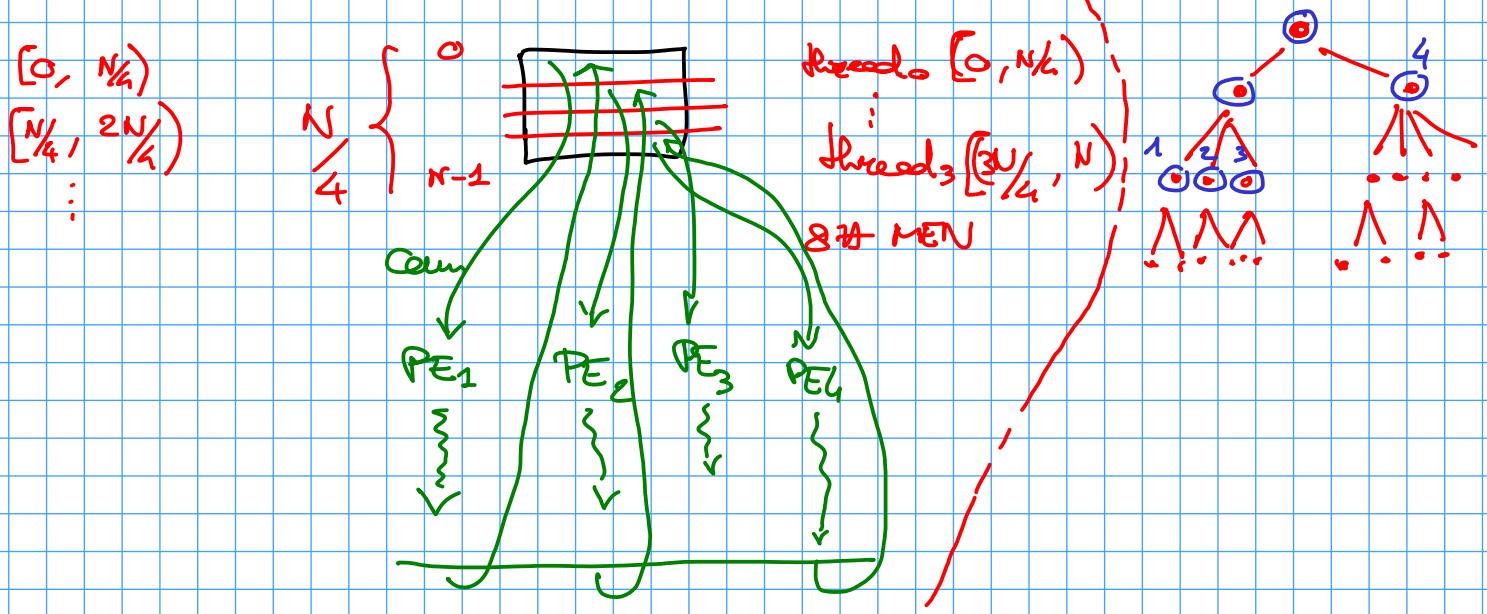
DATA DECOMPOSITION PATTERN

PROBLEM: data (input) → units that can be operated "relatively" independently
(partitions [possibly overlapping])



FORCES: flexibility, efficiency, simplicity

SOLUTIONS: array-based vs recursive data



GROUP TASKS Pattern

PROBLEM: { Tasks } how can i group them

TASK decomp — DATA decomp

⇒ dependency management is simplified?

SOLUTIONS:

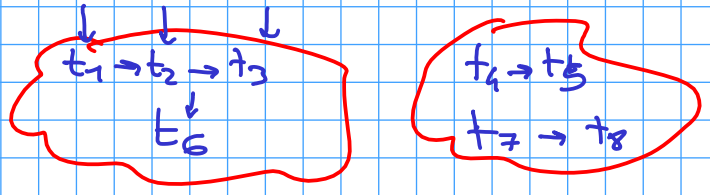
- temporal dependencies ⇒ groupings

$t_1 \to t_2 \to t_3$

$t_4 \to t_5$ t_6 $t_7 \to t_8$

∀ task t_i I need the same amount of time to compute

2 PE ?



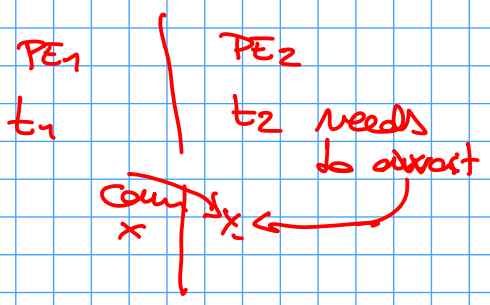
$t_4 \to t_5$

produces x — uses x

⇓

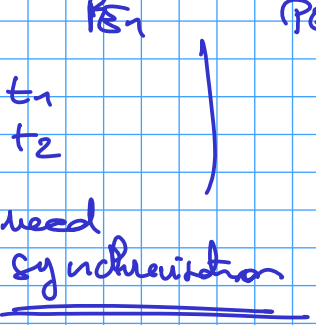
t_4 must end before t_5 starts

Ex 1:




```
for(i= ---
  execute (task[i])
```

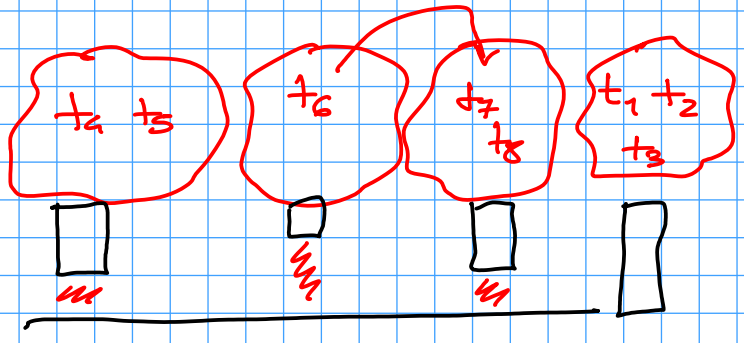
Comm x no more needed

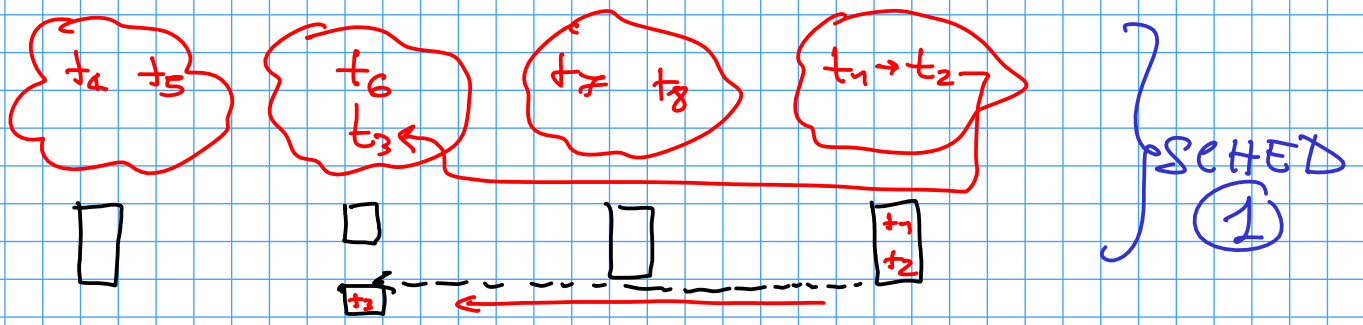


need synchronization

$t_1 \to t_2 \to t_3$

$t_4 \to t_5$ t_6 $t_7 \to t_8$





ORDER TASKS pattern

PROBLEM : { tasks } dependencies → { group }

task decomp

data decomp

which ordering can we take to ensure that all constrains are satisfied