

## Parallel Prefix

1 2 | 3 4 → 1 3 6 10

PE<sub>1</sub>

PE<sub>2</sub>

vector <int> x(m) (m-1) " + "

1 3

3 7

$(\frac{m}{nw} - 1)t_0$

3 7

3 10

$(nw - 1)t_0$

1 3 | 10 6

$(m - \frac{m}{nw})$  is the amount of items we have in total

We have to parallelize this

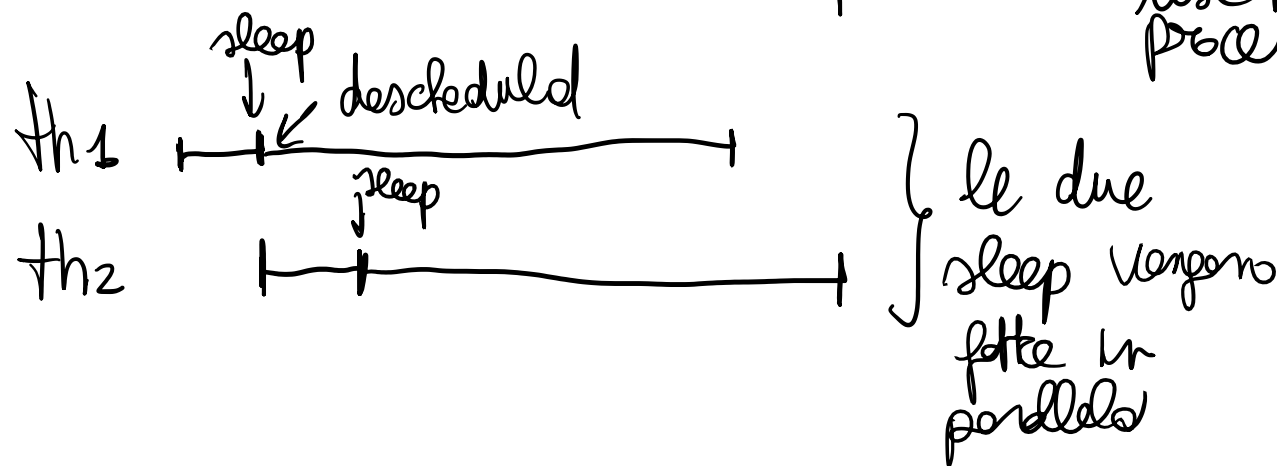
Active delay:

std::this\_thread::sleep\_for(15 μs)

↳ this kind of sleep suspend the thread and another thread works.

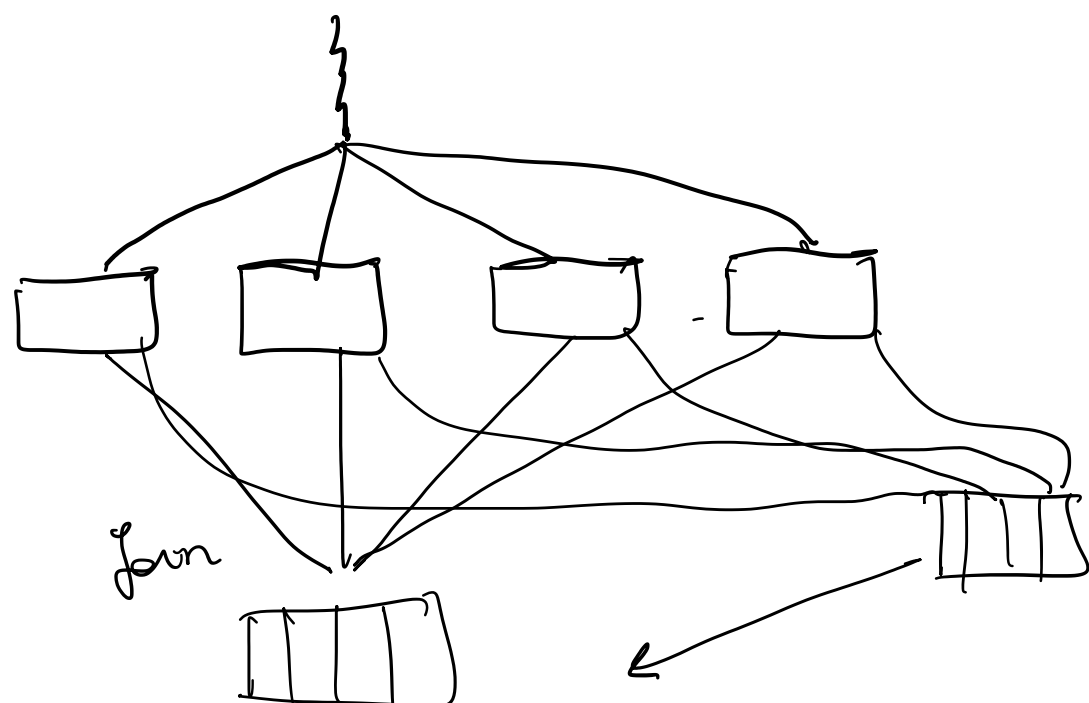
By the time to suspend,

the thread doesn't run on the processor. → don't use the processor



Se faccio active delay invece il thread continua a lavorare perché continua a controllare il tempo che passa.

RIASCOLTARE parte verso le 14:45 e anche prima.

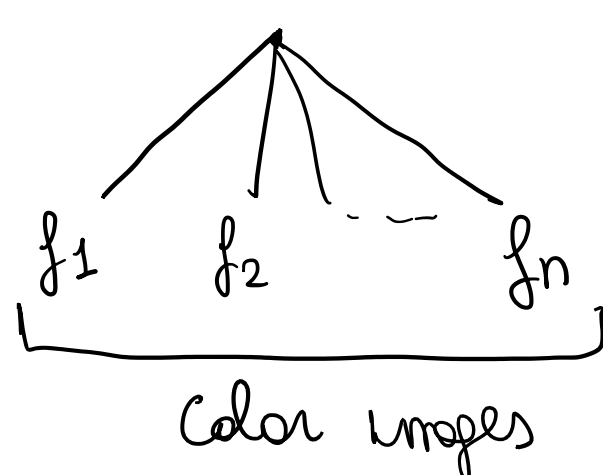


Se il tempo che ci mette a fare l'esecuzione sequenziale è minore del parallel anche se aumento i thread vuol dire che l'overhead è maggiore rispetto al costo effettivo delle varie operazioni.

Sequential code → parameters for performance model

parallel algorithm

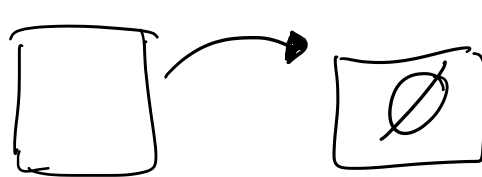
performance model



"CImg.h"

Stamp image b&w images

b&w 0



Map pattern(f): ∀ images

① load

② bracket map(g):

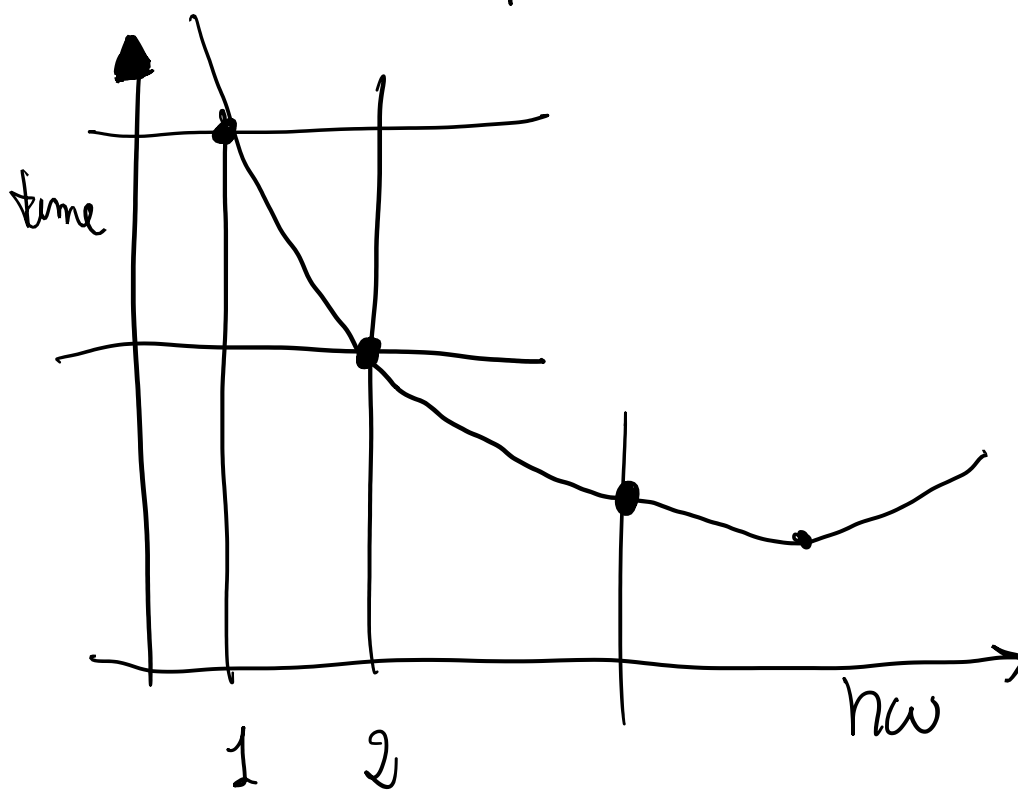
③ store

∀ pixel of the image

Init:

→ read stamp image

→ set par...



nw = m number of images