

# Treatment of state

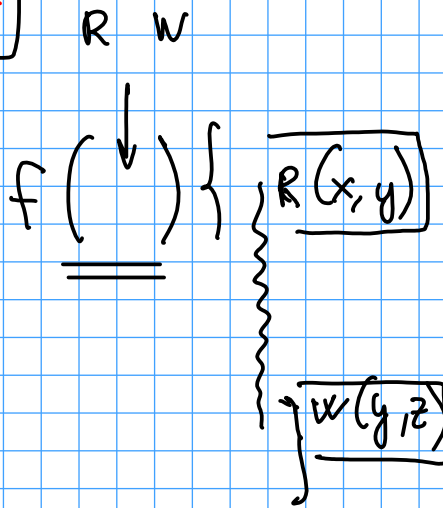
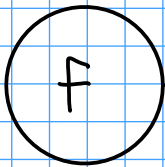
⇒ variabili condizionali

su cui si opera con + di una distinta componente alla volta

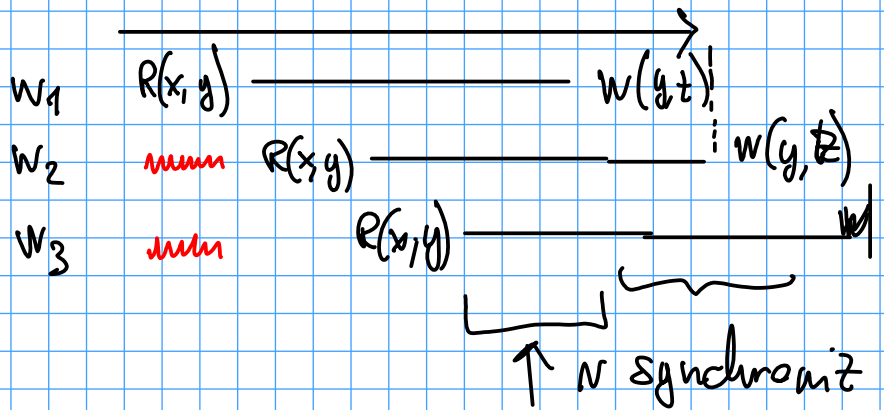
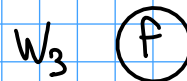
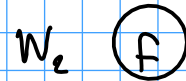
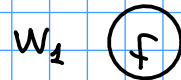
⇒ quali operazioni si usano

tipi degli accessi  
R W R/W

operazioni "logiche"



pattern parallelo



# OPERAZIONI "LOGICHE"

READONLY

$\forall AC_i$   
 può leggere

"Owner Computes"

$\forall AC_i$  può leggere  
 $\exists! k AC_k$  può scrivere

"ACCUMULATORE"

$\forall AC_i$  può operare sul dato

"RESOURCE"

Serializzare gli accessi

$\oplus$  associative binario commutativa

oppure

analisi degli accessi  
 $(R, W, RW)$

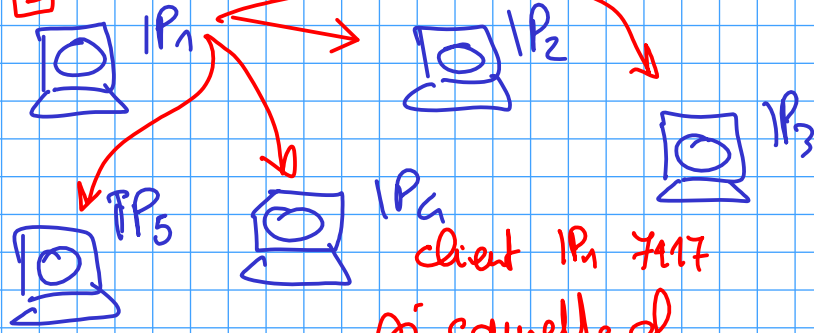
$x += y$

$R(x)$   
 $\overline{x+y}$   
 $W(x+y)$

$R(x)$   
 $x+y$   
 $W(x+y)$

SERIALIZZAZIONE degli ACCESSI

$\exists!$   $\forall AC_i$



client  $IP_1$   $\forall AC_i$   
 si connette al Master

si crea un socket  
 comunico al master il SS (Port)

Master

Master

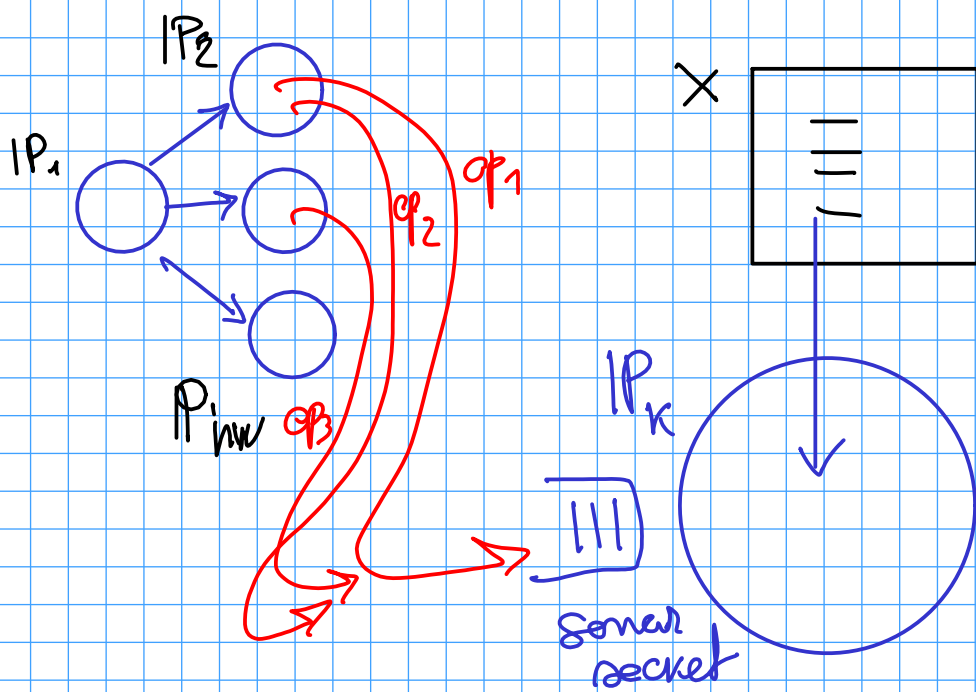


Phase 1

comunico al master "parte"

Phase 2

chiedo al master la parte di "variabile"

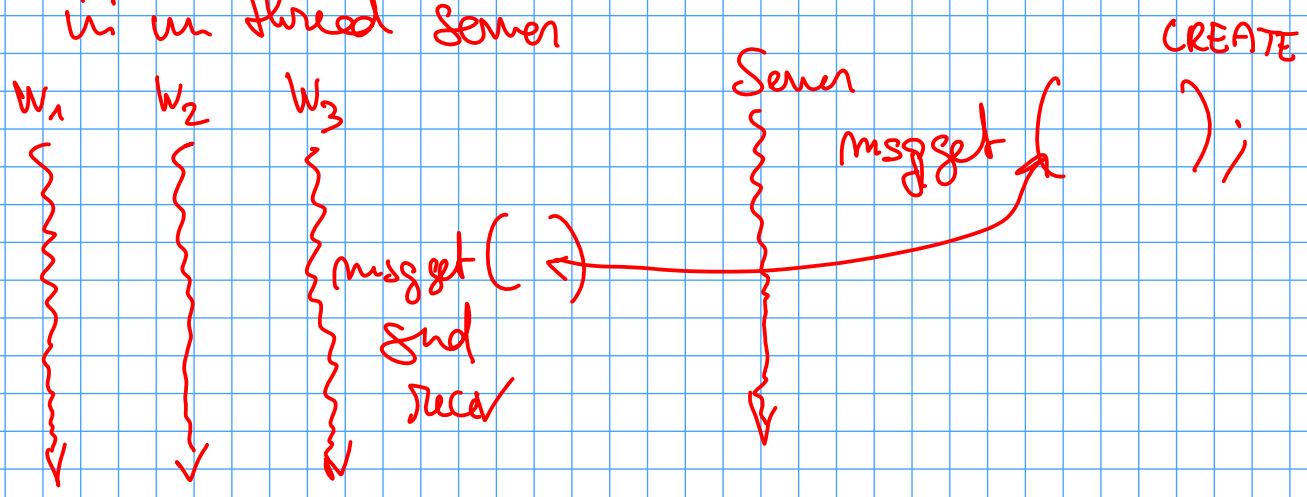


sequenza degli eventi  
 ↓  
 incapsulato  
 in un server  
 "sequenziale"

precorso

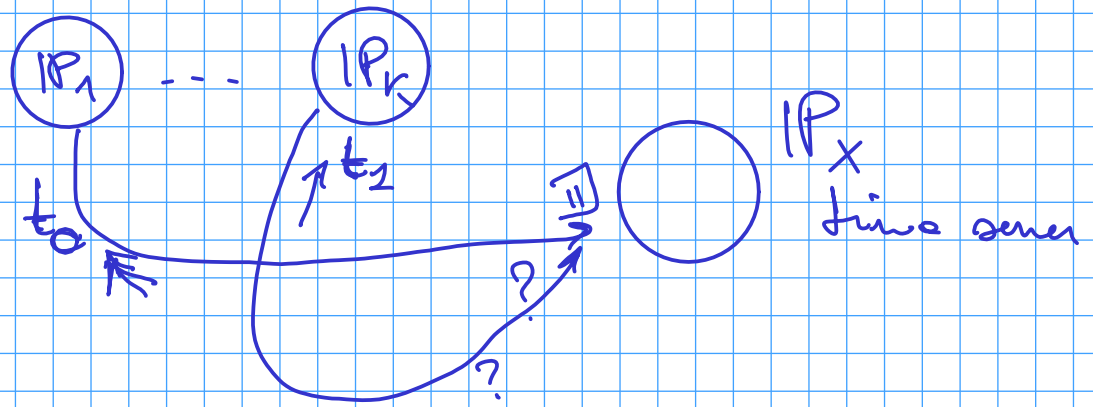
↳ accetta messaggi per qualunque  
 tipo di richiesta di operazione  
 sullo stato X

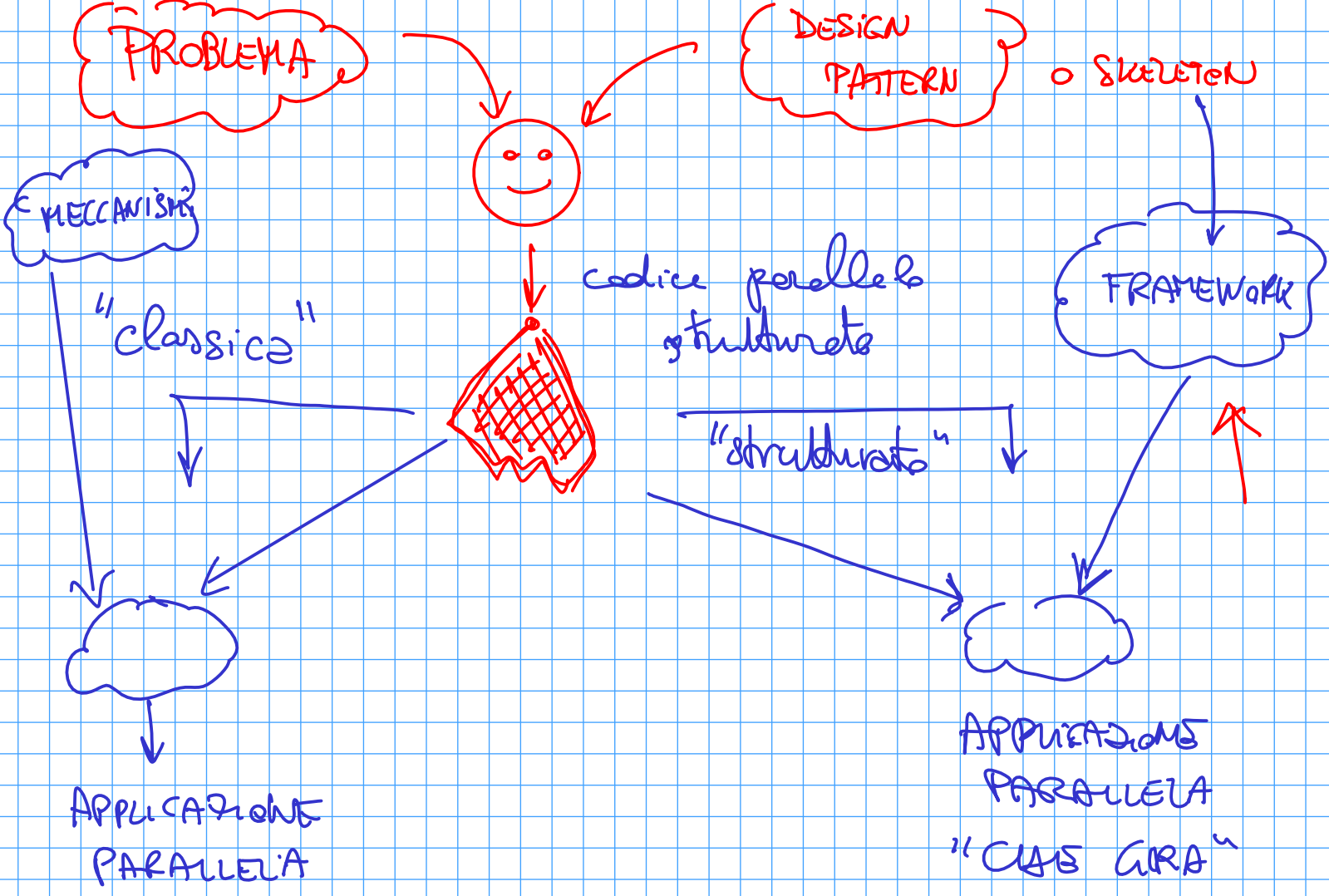
incapsulo X  
 in un thread server



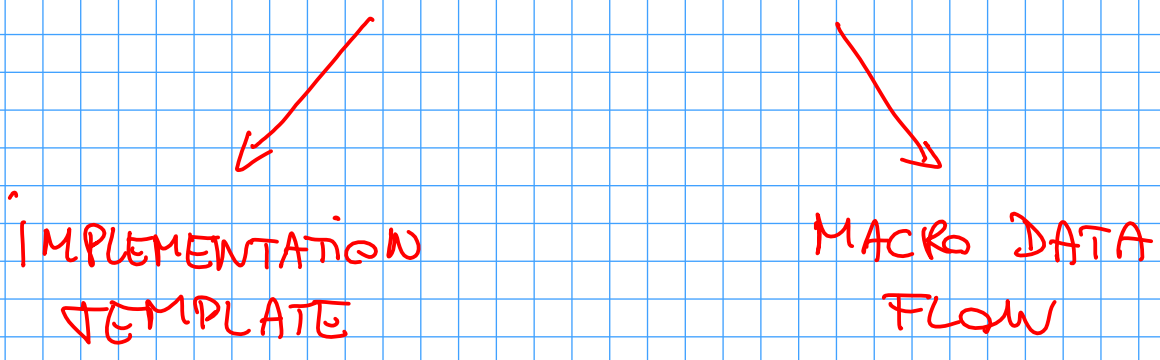
"Tempo" in ambiente distribuito

CoW





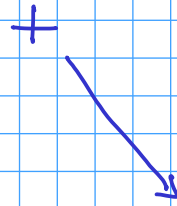
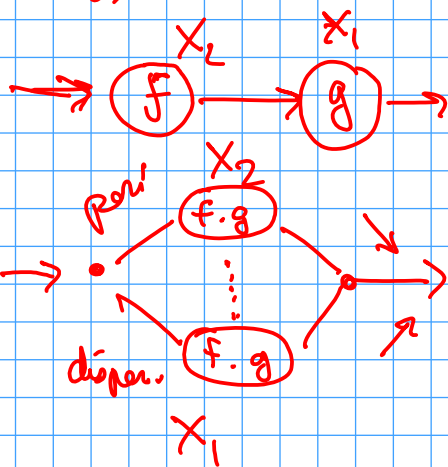
MODELLI di IMPLEMENTAZIONE SKELETON



# IMPLEMENTATION TEMPLATE

schema (parametrico) per implementare  
un pattern in una certa architettura

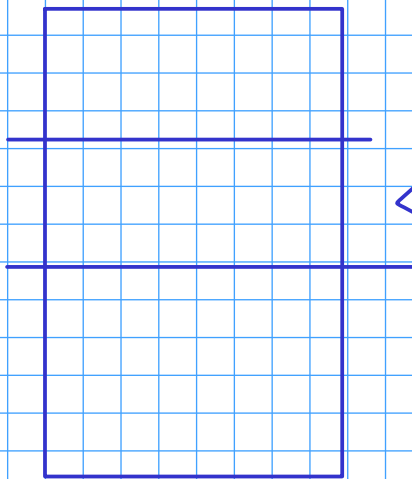
PIPE (f, g)



HW?

un modello dei costi  
(delle prestazioni)

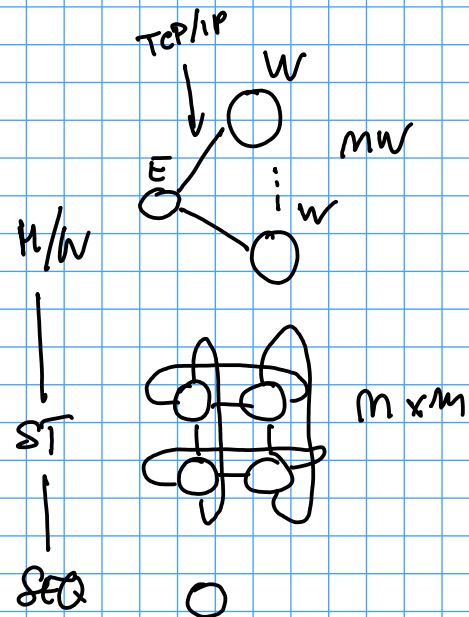
## LIBRERIA di IMPLEMENTATION TEMPLATE

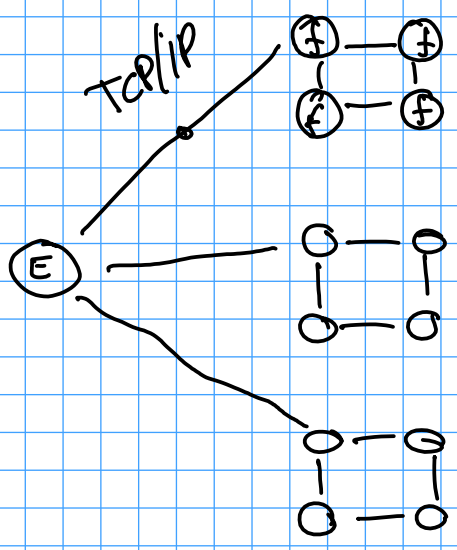


< name, ~~skeleton~~, ~~hw~~, grafo di attività, concatenati  
modello di costi >

APPL  
MASTER/WORKER  
•  
• STENCIL  
• SEQ(j)

HW  
annotato l'albero dei pattern con template presi dallo sito





$M=2$   
 $m=2$

$mw=3$

}  $Sp( )$   
 $Tc( )$