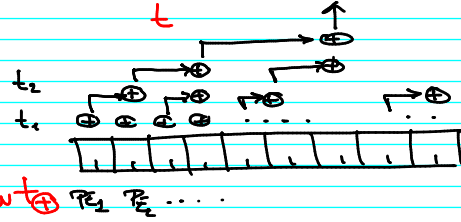
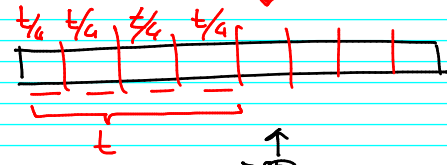
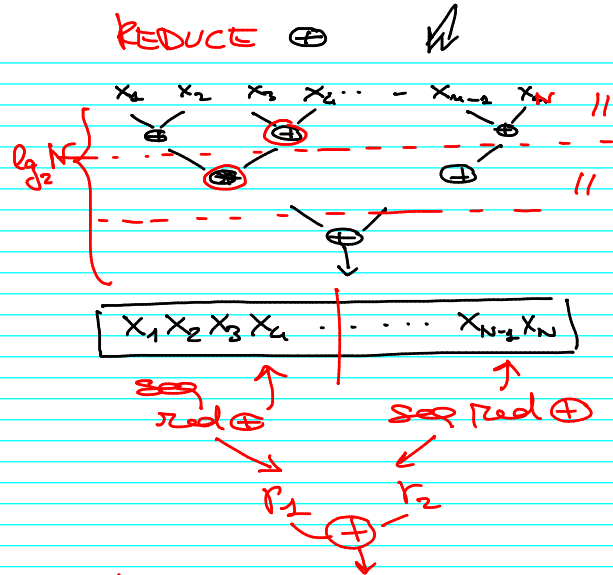


# TEMPLATE DESIGN

- 1) ALTERNATIVES
- 2) EVALUATION
- 3) (PERFORMANCE MODELS)
- 4) DESIGN INTERFACE
- 5) IMPLEMENTATION  
(TARGET MACHINE  
DETAILS NEEDED)



## Performance models

1) template (tree)  $\frac{N \cdot (N-1)}{2}$

$\log_2(N)$  layers  $t_{\oplus}$

$\log_2(N) \cdot \dot{o}(t_{\oplus})$

2) template (small tree)

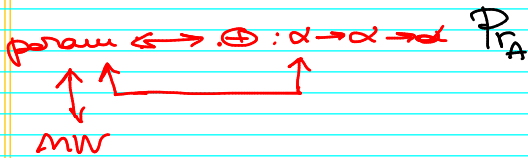
$\frac{m \cdot w}{m \cdot w} t_{\oplus} + m \cdot w t_{\oplus} p_1 p_2 \dots$

DESIGN INTERF (templ 2)

Posix / TCP



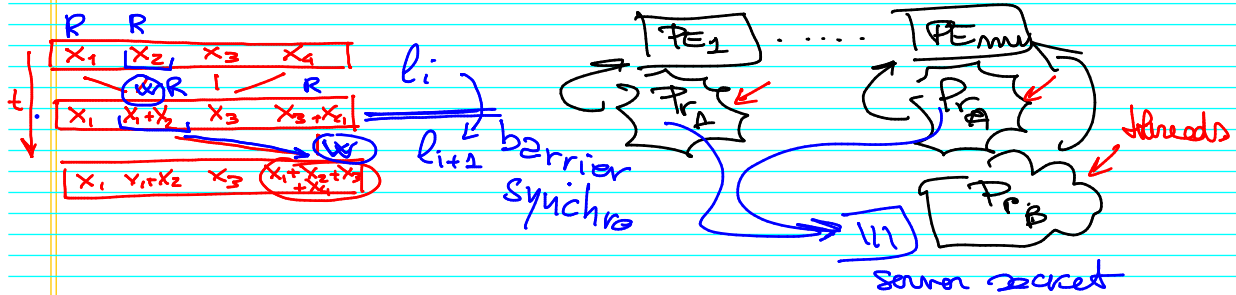
distri (row / row)



1 process x "PE" (mwr processes)  
body : { compute local partition  
then send result }

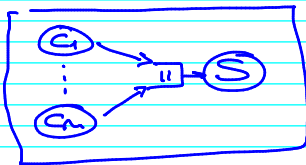
PrB + 1 process "final result" collector

body : { for (mwr)  
receive partial result  
& sum it up (⊕)



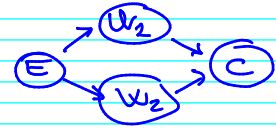
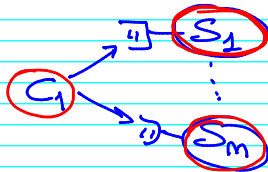
# BUILDING BLOCKS for TEMPLATES

CLIENT/SERVER



perf model

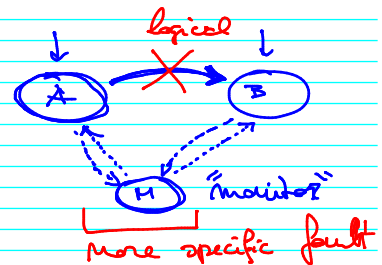
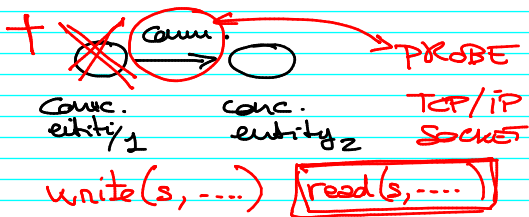
convenient imple  
decent interface



C can perceive  
faults of  
W1 W2  
W1 & W2  
" " "  
of E

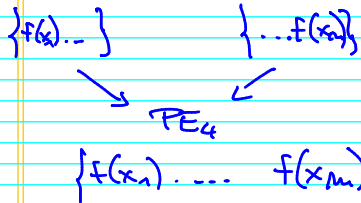
## "STRATEGY" for FAULT TOLERANCE

→ MECHANISMS TO DETECT FAULTS



MAP · F :

$$\{x_1 x_2 x_3 \dots x_m\} \rightarrow \{f(x_1) f(x_2) \dots f(x_m)\}$$



$\text{map}(f) \equiv$

pipeline ( $S_1, S_2, S_3$ )

$S_1 \equiv \text{split} : \text{collection} \rightarrow \text{stream}(\text{items})$

$S_2 \equiv \text{form}(f)$

$S_3 \equiv \text{gather} : \text{stream}(\text{items}) \rightarrow \text{collection}$