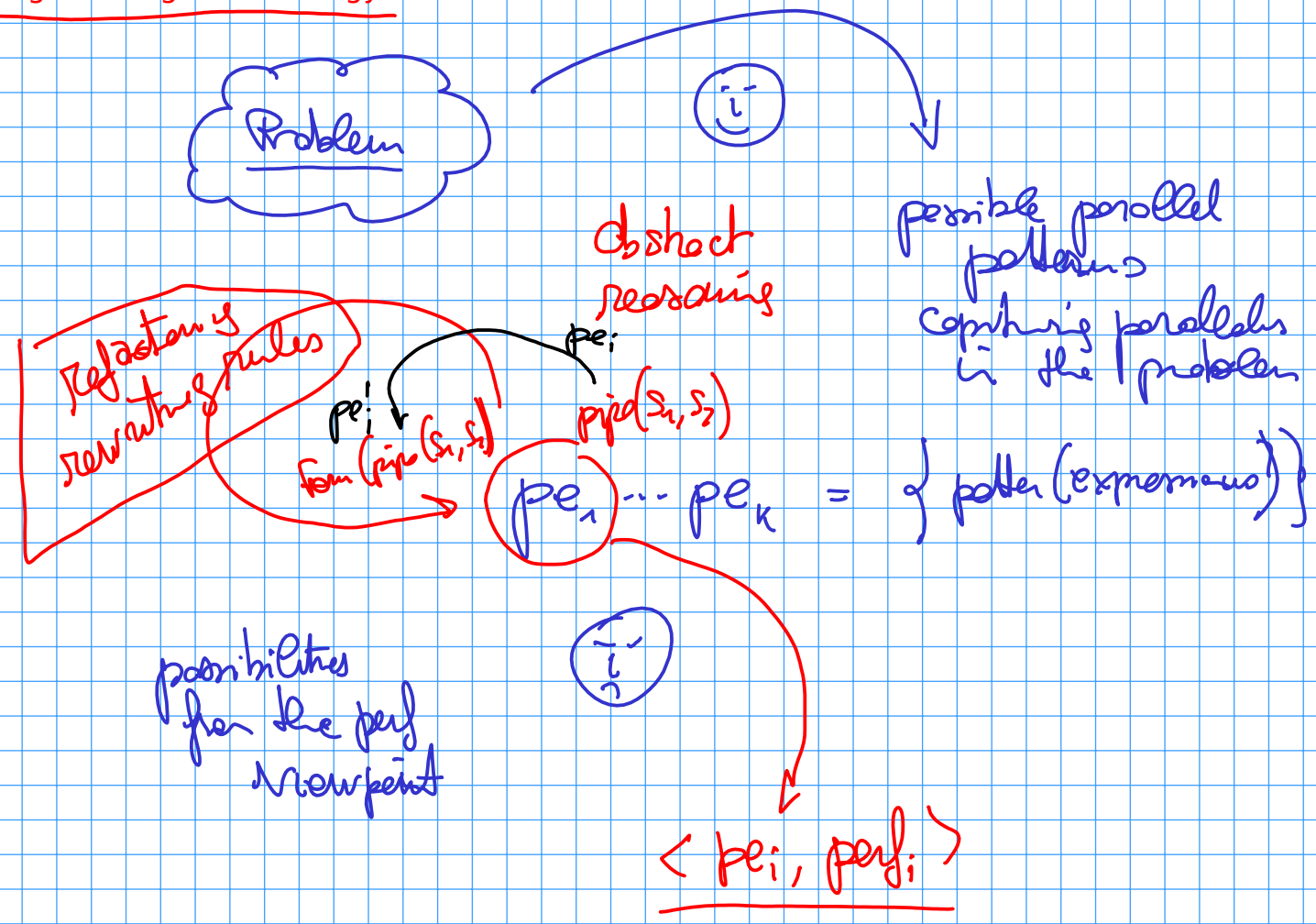

Problem
i
possible parallel patterns capturing parallelism in the problem
Abstract reasoning
pe'
pe
pipe(s1,s2)
Refactoring rewriting rules
pe1 ... pek = { pattern(expression) }
possibilities for the perf viewpoint
i
< pei, perfi >
Problem
i
possible pattern expressions
Perf
Best
perf
perf
n
n'
implement the parallel code

embarrassingly parallel

t_0

for(i=0; i<N; i++) ← size

x[i] = f(x[i]);

t_1

$(t_1 - t_0)/N$

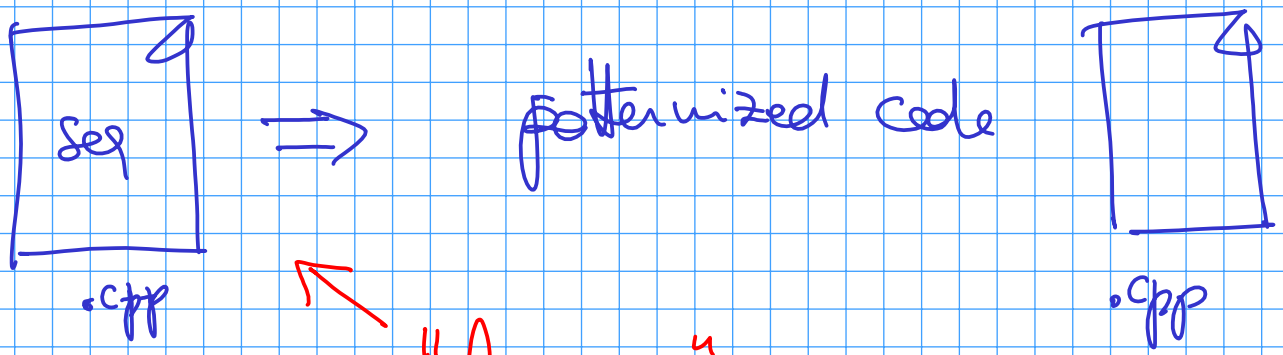
weight?

overhead
forking a thread

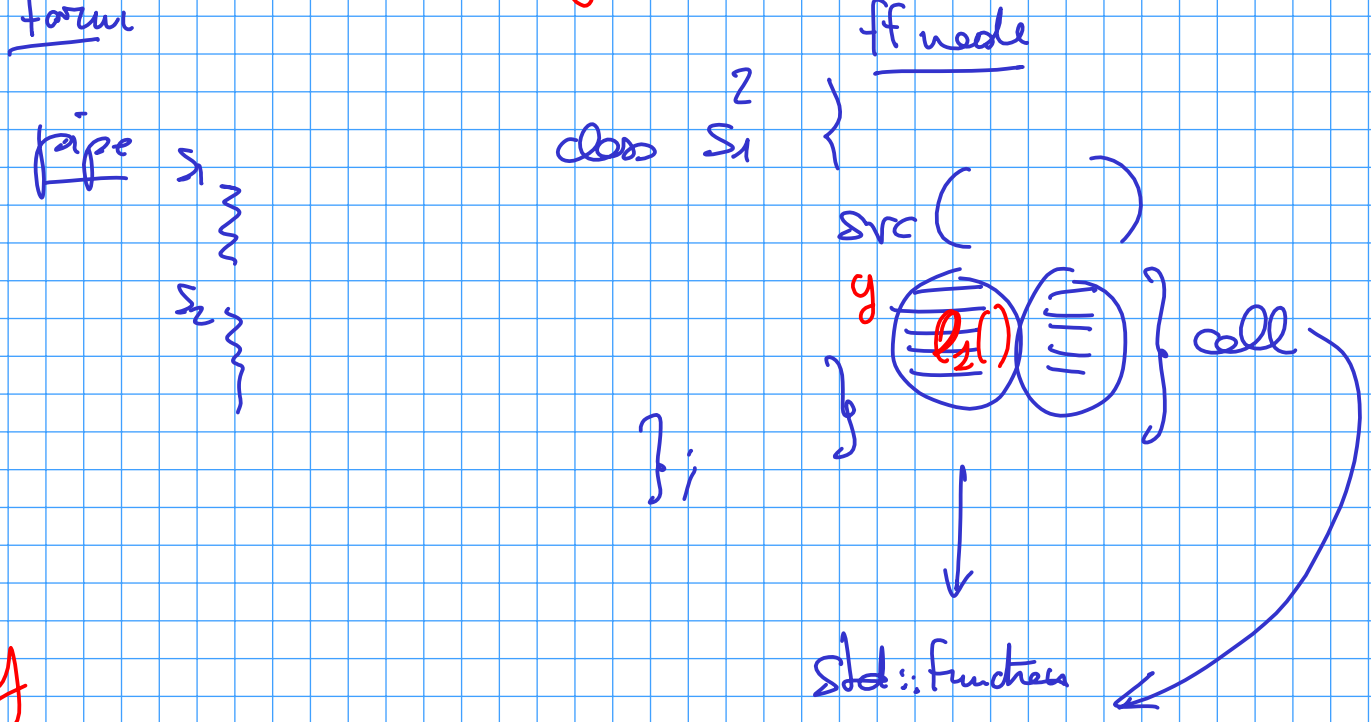
Parallel For

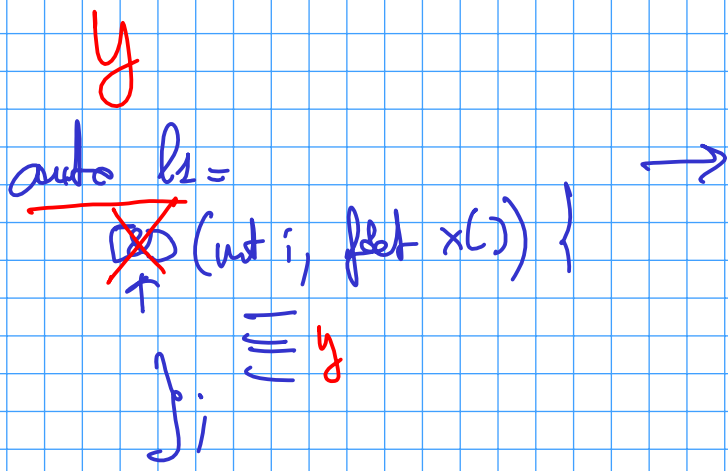
#pragma omp parallel_for

Parallel For


Seq
.cpp
pattern ized code
.cpp

"Mygenic"


farm
pipe
S1
S2
ff node
class S1 : 2 {
svc ( )
y
l1()
call
};
std::function


y
auto l1 =
[&] (int i, float x()) {
y
};
lambda
pass
fun

pe x

poly( )