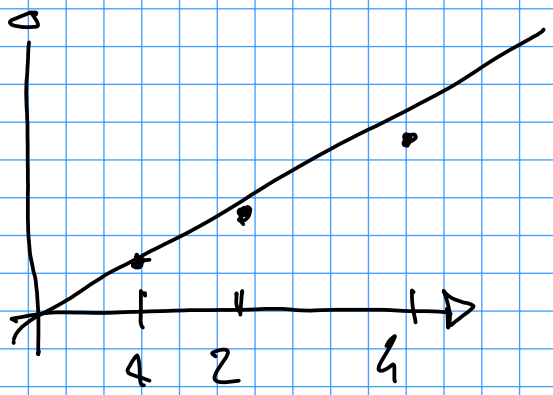
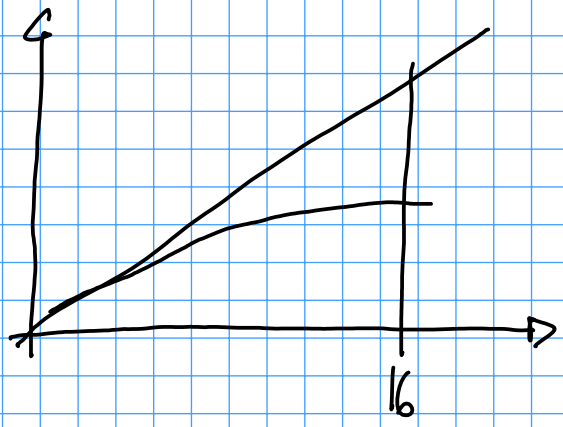
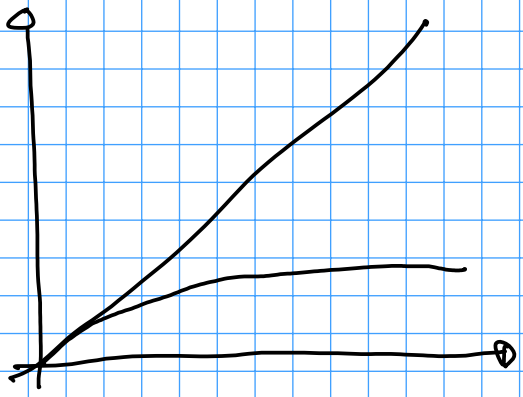


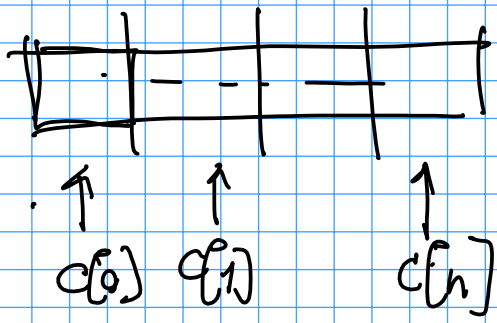




OpenMP

String text;

chunks[]




```
for(i=0; i<n; i++)
    spawn(cap(c[i], text));
sync;
```

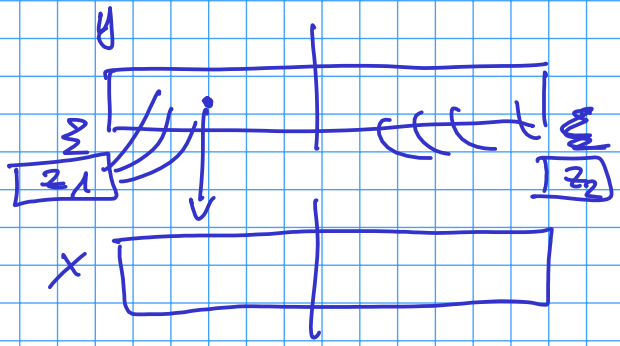
```
#pragma omp parallel for
for(i=0; i<n; i++)
    cap(c[i], text);
```

- Гарантир

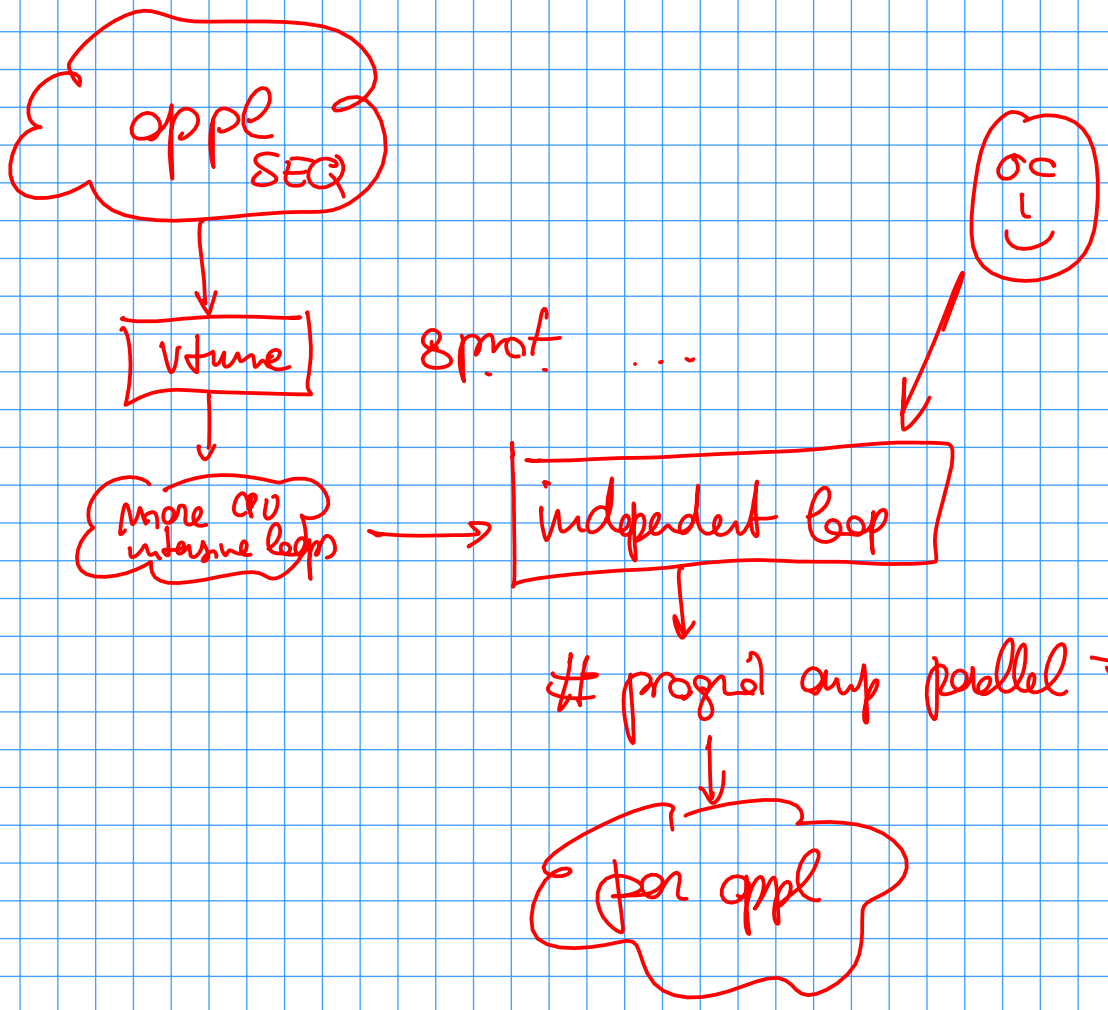
```
#pragma omp parallel for
for(i=0; i<length(text); i++) { if !is_small(text[i])
    cap(text[i]);
```

```
z = 0;
for(          ){
    x[i] = f(y[i]);
    z += x(i);
}
```

∀ i:



$z = z_1 + z_2$



DIFFICULTIES

(related to the correct usage of the mechanisms (thread, synch, sharing) available

PARALLEL PATTERNS (small set)

(map) forall reduce

DATA PARALLEL

PARALLEL STREAM PATTERNS

pipeline farm

"translation of the book" (small → capital letter)

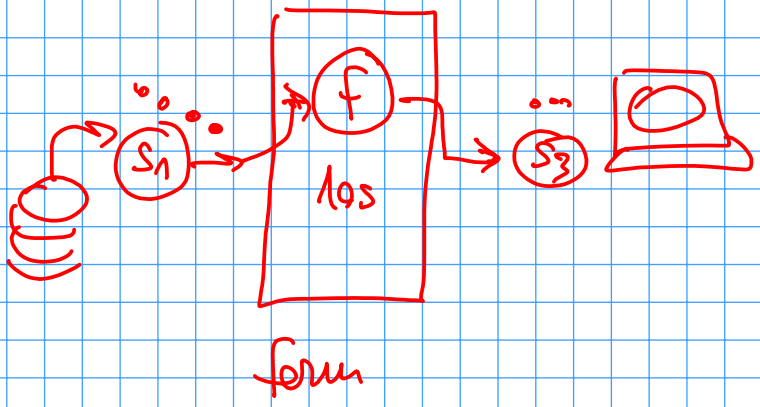
computing the payroll of unipi employs

forall map (f)

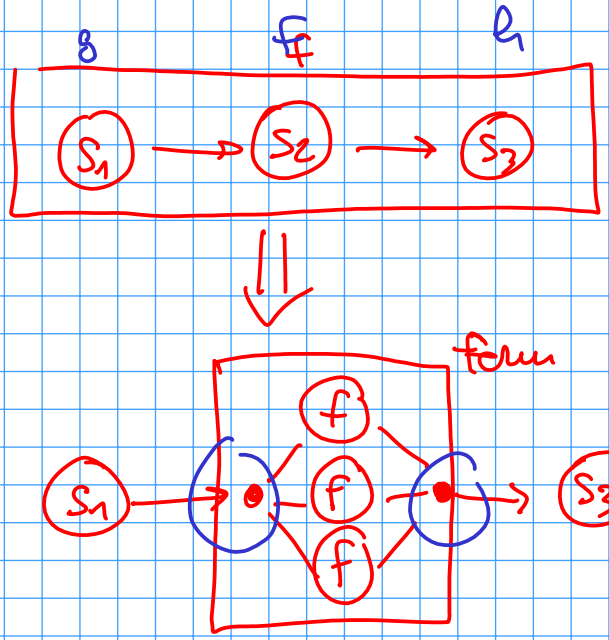
small → capital

map (g)

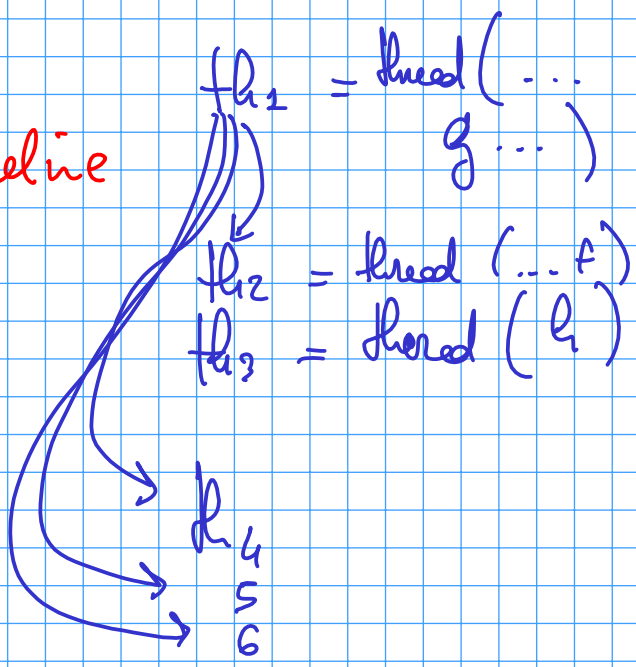
(my)data → (my)salary



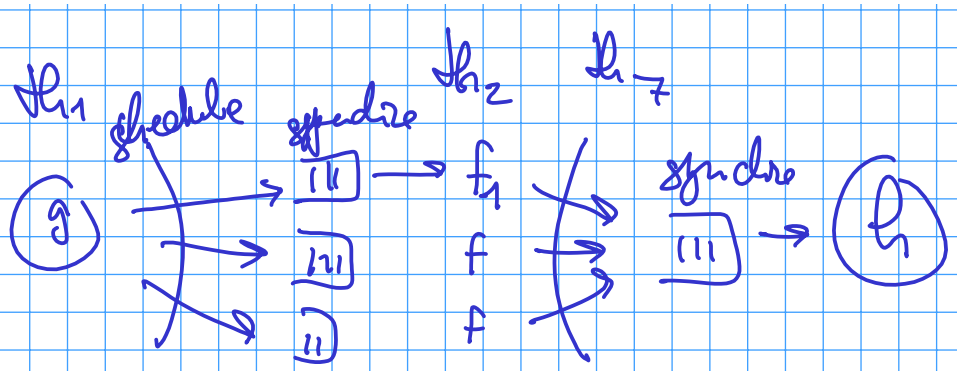
farm



pipeline



```
Pipeline( parallel_execution_omp,
   // Stage 0: read values from a file
   [&]() -> optional<vector<int>> {
      auto r = read_list(is);
      return ( r.size() == 0 ) ? {} : r;
   },
   // Stage 1: takes the maximum value of the vector
   Farm(parallel_execution_omp{6},
      []( vector<int> v ) {
         return ( v.size() > 0 ) ?
            max_element(v.begin(), v.end()) :
            numeric_limits<int>::min();
   }),
   //Stage 2: prints out the result
   [&os]( int x ) {
      os << x << endl;
   }
);
```



DATA PARALLEL

map $(x_1 \dots x_n)$ f

$(f(x_1) \quad f(x_n))$

reduce $x_1 \dots x_n$ $\oplus$

$x_1 \oplus \dots \oplus x_n$

map-reduce f $\oplus$ $x_1 \dots x_n$

$f(x_1) \quad f(x_n)$

$\langle y_1, k_1 \rangle \quad \langle y_n, k_n \rangle$

$\oplus$ $\forall$ items with same key

stencil $(x_1 \dots x_n)$ $f \equiv$

$f(x_{i-1}, x_i, x_{i+1})$

$\downarrow$

new x_i

$\equiv$

STREAM PARALLEL

farm $\forall$ items of stream (f)

f

$f \dots f(x_3) \; f(x_2) \to f(x_1)$

f

pipeline

$(f) \to (g) \to (h)$

$\to f(x_3) \to g(f(x_2)) \to h(g(f(x_1)))$

filters

$\to (c)$

$\text{if}(c(x)) \to x$

else discard

windowed