Data Parallel Patterns

$$Collection\langle T\rangle \times f: T \to T' \longrightarrow Collection\langle T'\rangle$$

any "splittable" collection (easy)

map

$(\alpha f)(x)$ ≡ $map(f)(x)$

```
float x[N]; y[N];
y = map(f, x)
```

float → float

TURING AWARD LECTURE 1978

John Backus

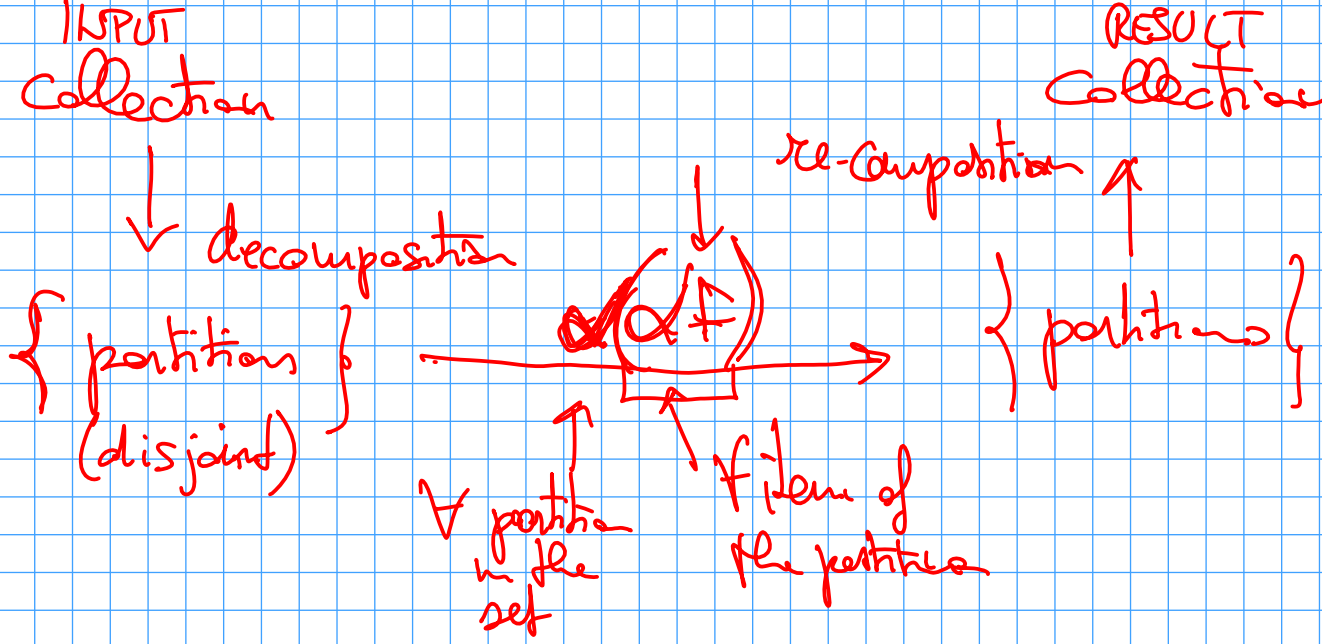
Can programming be liberated from the Von Neuman style? . . . .

FP: apply to all (map) $\alpha(f) : \langle x_1 \dots x_n \rangle \to \langle f(x_1) \dots f(x_n) \rangle$

insert / $\oplus$: $\langle x_1 \dots x_n \rangle \to \langle x_1 \oplus x_2 \oplus \dots \oplus x_n \rangle$

$[f_1; f_2; \dots f_n](x) \to \langle f_1(x), f_2(x) \dots f_n(x) \rangle$

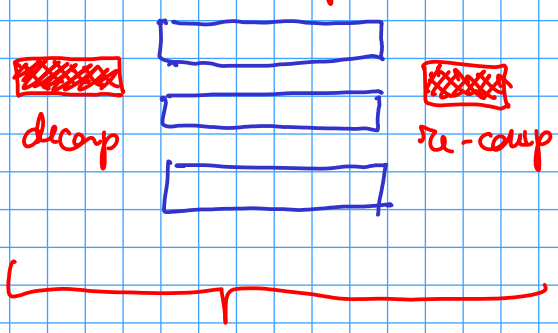
MISD



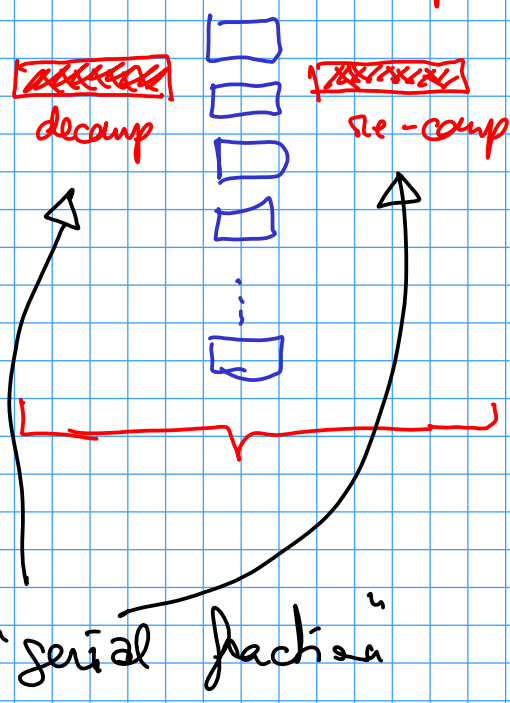
the more you decompose — the more work you have to do (decomp / re-comp)

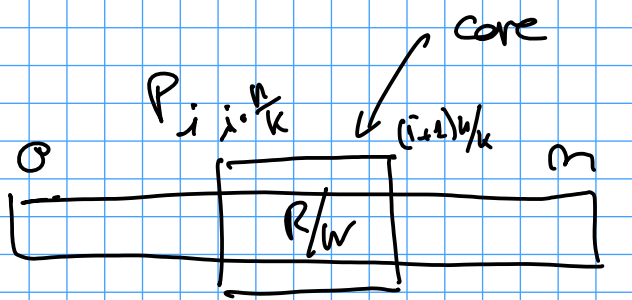
— the more parallelism you get

coarse grain decomp



fine grain decomp





K sub partitions

$\frac{n}{K}$ len of partition

$\left[i \cdot \frac{n}{K} , \frac{i+1}{} \frac{n}{K} - 1 \right]$

$$L = T_{decomp} + T_W + T_{recomp} = \underline{T_{decomp} + T_{recomp} + \frac{n}{n} t_f}$$

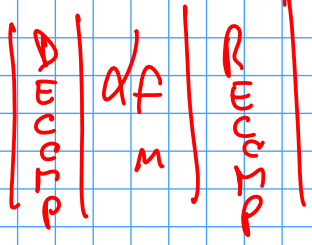
len(collection) = n — t_f time spent to compute a single instance of f (f(x_i))

#collection

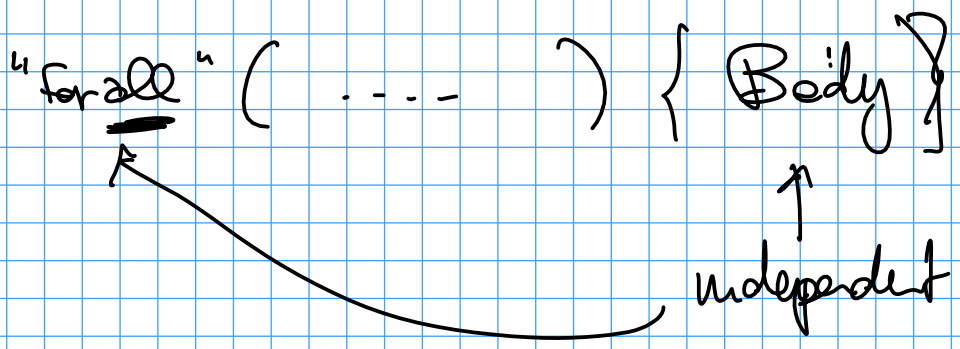
$T_{seq} = n \cdot t_f$

$T_W(n) = T_{seq}/n = \frac{n}{n} t_f$

Decomp ;
αf ;
Recomp ;
$\Rightarrow$ PIPELINE



$T_C = \#\text{loops} \cdot T_S$ $\quad T_S = \max\left\{ T_{decomp}, \frac{m}{n} t_f, T_{recomp} \right\}$



C / C++
OpenMP

Do INDEPENDENT $\approx$ map

Iterations
Do 10 I=1,10
⋮
10 CONTINUE
} body

# pragma omp parallel for
for( ) { $\equiv y(i) = f(x(i))$ }

Reduce pattern (parallel scan)

Collection<T> → f : T×T→T → T

reduce(⊕) ⟨x1 ··· xn⟩ = x1 ⊕ x2 ⊕ ··· ⊕ xn

collection
decomp ↓
{ partitions } —d(reduce(⊕))→ { values } —recomp ≡ reduce(⊕)→ value

parallel step ⇒ ⊕ associative

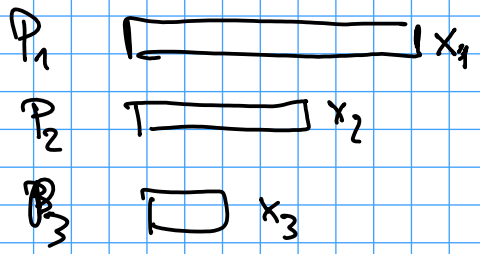
(x1 ⊕ x2) ⊕ (x3 ⊕ x4) ⊕ (x5 ⊕ x6) ⊕ (x7 ⊕ x8)
a1 a2 a3 a4

foldleft (⊕)
right (+)

((x1 ⊕ x2) ⊕ x3) .. ⊕ xn
··· ⊕(xn-2 ⊕(xn+1 ⊕ xn))

collect
↓
{ P1 P2 P3 }

d(reduce(⊕))



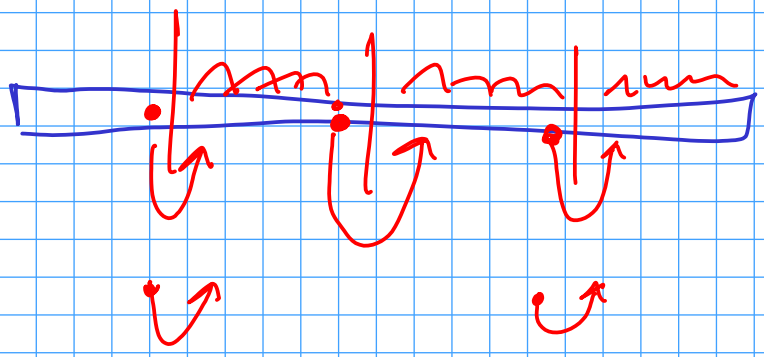
} (x3 ⊕ x2) ⊕ x1

scan (⊕) ⟨x1 ··· xn⟩ → ⟨ x1, x1⊕x2, x1⊕x2⊕x3, ···, x1⊕x2···⊕xn ⟩

$X_1\ X_2\ X_3\ X_4 \mid X_5\ X_6\ X_7\ X_8$

per $X_1\ X_{12}\ X_{123}\ X_{1234}\ X_5\ X_{56}\ X_{567}\ X_{5678}$

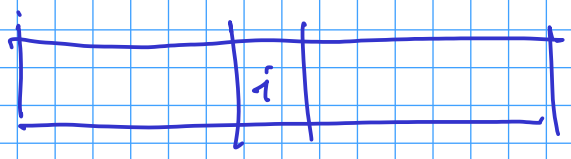
X_{12345}

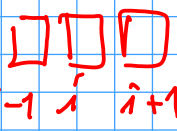


STENCIL

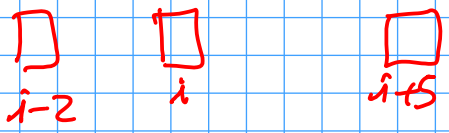
Collection<T> → neighborhood<T> → {T} → f: {T} → T

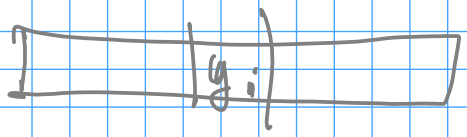
→ Collection<T>

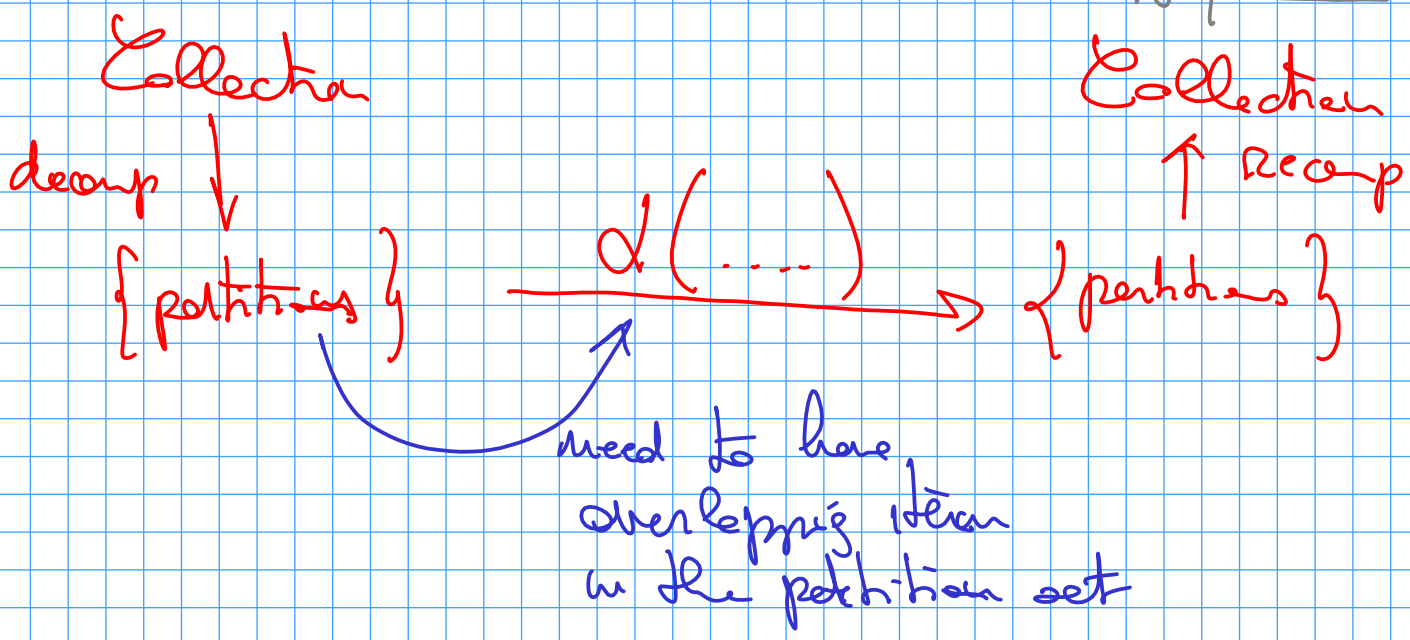




$y_i = f(x_i, x_{i-1}, x_{i+1})$



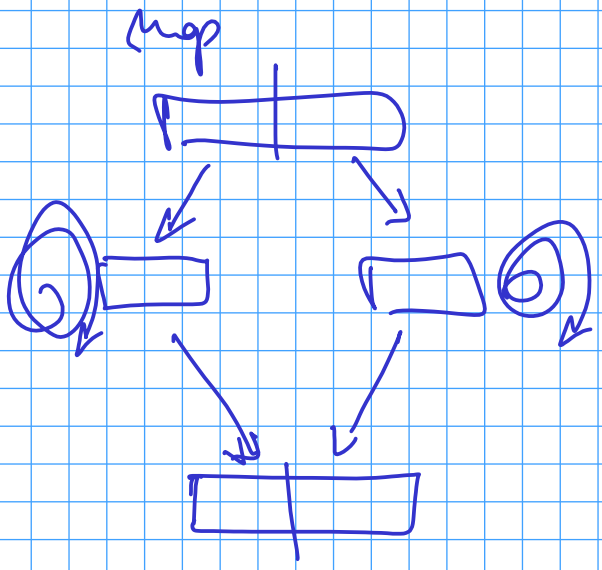


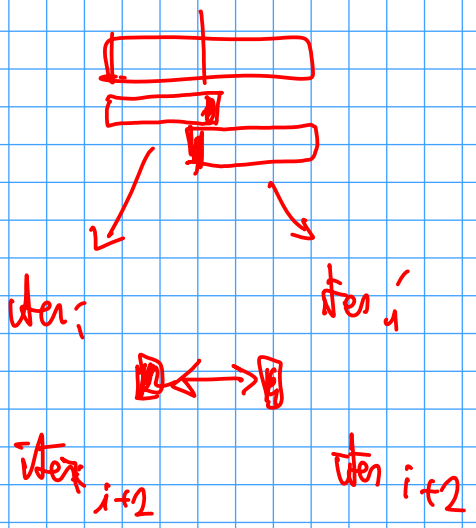


for ( ) {

map / stencil pattern

}





Google MapReduce

$$comp(f, \oplus)(x) \equiv reduce(\oplus) \circ (\alpha f(x))$$

scalar — vector — reduce — map

g. MapReduce

→ map (f)

→ α(reduce ($\oplus$))

$f: X \to \langle K, x \rangle$

{ { $\langle K, x \rangle$ with same key } }

$x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8$

αf

$\langle E, x_1 \rangle\ \langle E, x_2 \rangle\ \langle O, x_3 \rangle\ \langle E, x_4 \rangle\ \langle \oplus, x_5 \rangle\ \langle \oplus, x_6 \rangle\ \langle E, x_7 \rangle\ \langle E, x_8 \rangle$

{ { $\langle E, x_1\ x_2\ x_4\ x_7\ x_8 \rangle$ } { $O, x_3\ x_5\ x_6$ } }

$\alpha\oplus$ $\alpha\oplus$

{ $x_1 \oplus x_2 \oplus x_4 \oplus x_7 \oplus x_8$, $x_3 \oplus x_5 \oplus x_6$ }