

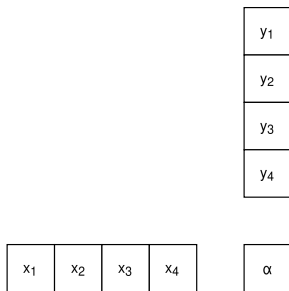
Concurrent Map Reduce (CMR)

Guillaume Dindart

Gilles MARAIT (Encadrant), Emmanuel AGULLO (Encadrant)
Équipe Concace INRIA

2024

Dot product

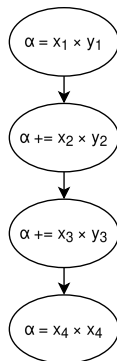


$$\alpha = x_1 \times y_1$$

$$\alpha += x_2 \times y_2$$

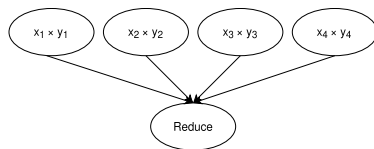
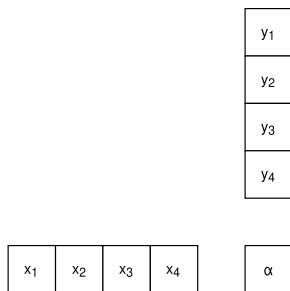
$$\alpha += x_3 \times y_3$$

$$\alpha += x_4 \times y_4$$



Using reduce to optimize?

Sequential Task Flow (STF)



$$\alpha_1 = x_1 \times y_1$$

$$\alpha_2 = x_2 \times y_2$$

$$\alpha_3 = x_3 \times y_3$$

$$\alpha_4 = x_4 \times y_4$$

$$\text{Reduce}(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$$

Advantage:

- Natural to pipeline

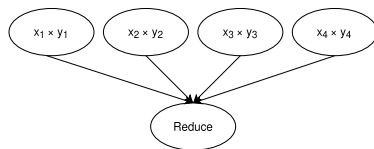
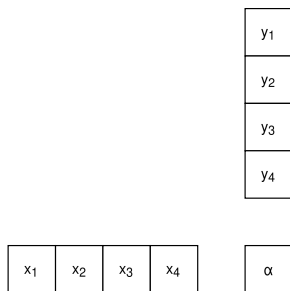
Disadvantage:

- Dynamic control flow

Question:

- Similar stencil ?

Map Reduce (MR)



Map = \times and Reduce = $+$

$$\alpha_1 = \text{Map}(x_1, y_1)$$

$$\alpha_2 = \text{Map}(x_2, y_2)$$

$$\alpha_3 = \text{Map}(x_3, y_3)$$

$$\alpha_4 = \text{Map}(x_4, y_4)$$

$$\alpha = \text{Reduce}(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$$

Advantage:

- Static control flow

Question :

- How to make a matrix product with it?

Concurrent Map Reduce (CMR): Matrix product (GEMM)

A

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

B

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

C

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

Datamap: Who has what block

Taskmap: Who works on which block

Each coordinate in C correspond in one MapReduce operation.

How to pipeline multiple CMR?

Concurrent Map Reduce = multiple simultaneous Map Reduce

CMR: Pipeline ($D = (A \times B) \times C$)

A

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

B

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

tmp

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

C

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

D

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

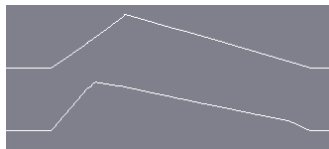
Tasks submission

With datamap and taskmap, we generate all the tasks. When a block is calculated, we send it to the other nodes if they need it.

Without pipeline

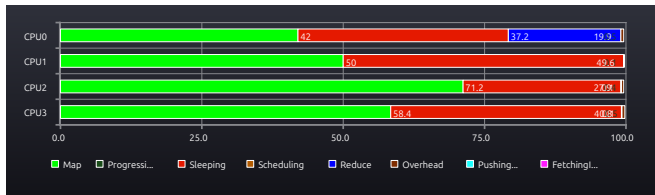


With pipeline

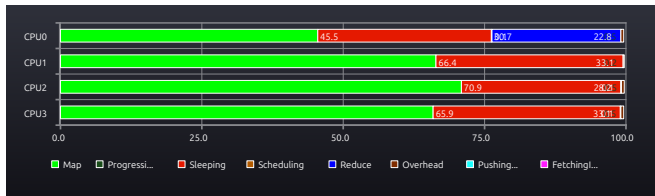


CMR: Pipeline CPU sleeping

Without pipeline



With pipeline



On going

- ▶ More general CMR:
 - ▶ OpenMP backend instead of StarPU
 - ▶ TRSM algorithm (Hugo pre-thesis)
- ▶ Vectorization of Map and Reduce (presentation by Aurélien)