Efficient Task Fusion Exploration for Data Intensive Computing SoC

Rosilde Corvino, Abdoulaye Gamatié and Pierre Boulet

r.corvino@tue.nl, abdoulaye.gamatie@lifl.fr, pierre.boulet@lifl.fr

Contribution

Data transfer and storage mechanisms are the major challenge in application specific parallel architecture design and their optimization is needed to achieve high performance and efficiency. We propose a method to take them into account in the early phase of architecture design.

Abstract

Synchronous Data Flow (SDF) are often used to describe the data-intensive application and to directly infer a corresponding Hardware Description. Problems occur when application tasks consume and produce data in different orders and appropriate methods are needed to model the data multidimensionality in the application, as well as, transfer and store mechanisms in the architecture. We propose to use an Array Oriented Language to adequately describe the application and an architecture abstract model including data-oriented optimizations and being configurable with respect to data transfer and storage mechanisms.

Related Works

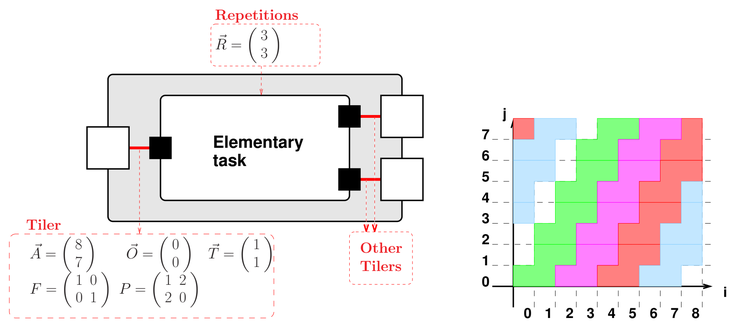
Application model comparison:

	SDF/(G)MDSDF	Polyhedral m.	Array-OL
A	- / ok	ok	ok
B	- / -	ok	ok
C	ok / ok	-	ok

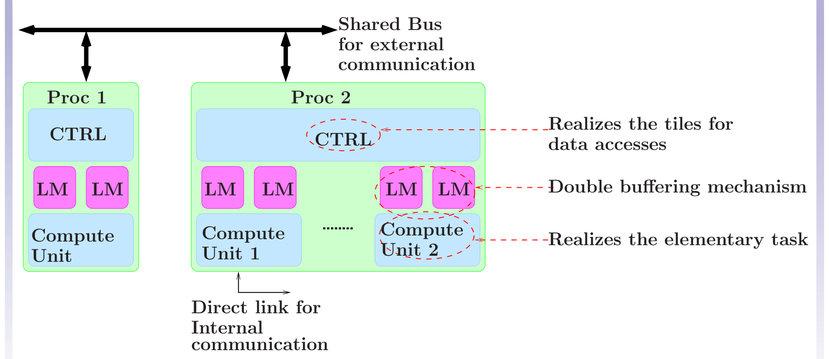
A = data multidimensionality; B = (loop) transformations; C = task level parallelism.

Architecture optimizations: SIMD architectural paradigm; power saving scratch pad memories; double buffering mechanism.

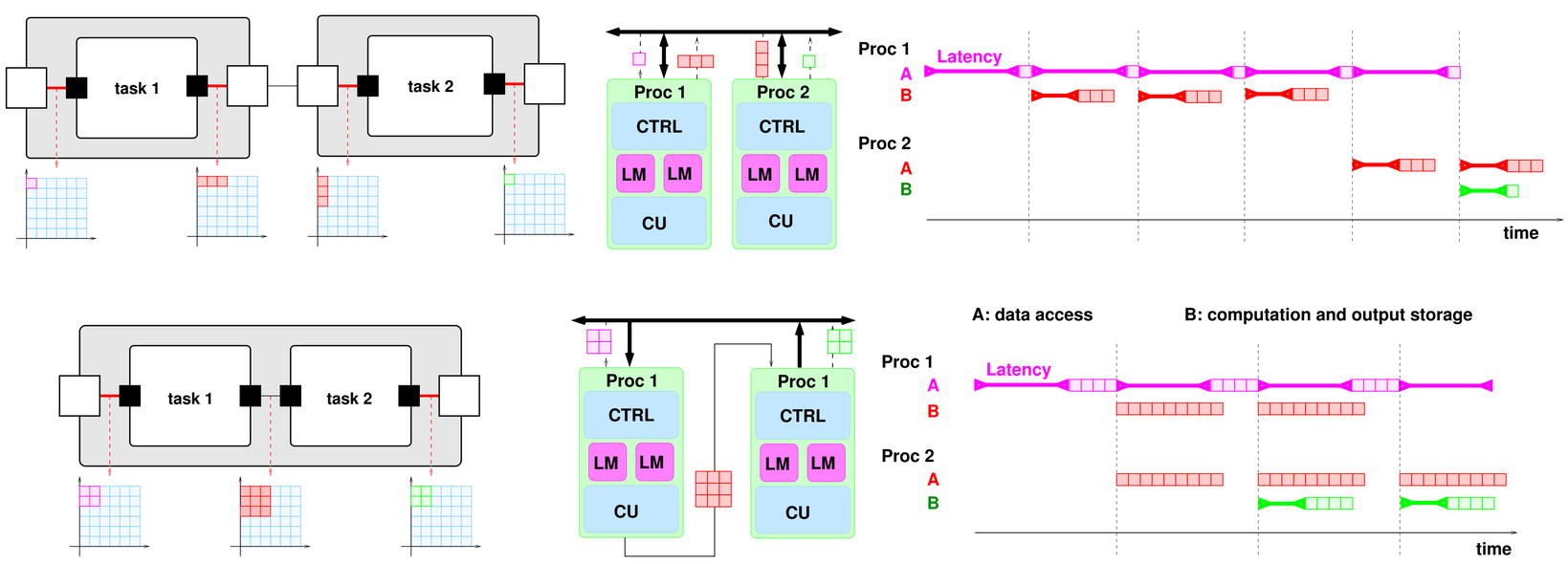
Array Oriented Language



Data Oriented Architecture



Fusion and Communication Mapping



Formal Problem Statement

X is a tuple of data blocks classified as X_{in}, X_{out}, X_{in}^{Mem}, X_{out}^{Mem}, X_{in}^{Com} and X_{out}^{Com}. The indexes Mem and Com indicate data blocks from/to an external memory or an internal communication respectively. A communication configuration is described by 4 matrices:

$$\begin{array}{lllll} \mathbf{I}_{\delta}^{\mathbf{Com}}: & X_{out}^{Com} & = & \mathbf{I}_{\delta}^{\mathbf{Com}} & X_{out}; \\ \mathbf{I}_{\delta}^{\mathbf{Mem}}: & X_{out}^{Mem} & = & \mathbf{I}_{\delta}^{\mathbf{Mem}} & X_{out}; \\ \mathbf{K}_{\delta}: & X_{in} & = & \mathbf{K}_{\delta} & X_{out}; \\ \mathbf{K}_{\delta}^{\mathbf{Mem}}: & X_{in}^{Mem} & = & \mathbf{K}_{\delta}^{\mathbf{Mem}} & X_{out}; \end{array} \quad (1)$$

Each processor P_j consumes input data blocks $\{x_i : x_i \in X_{in}\}$ and produces a single output $\{x_j : x_j \in X_{out}\}$. Due to the chosen execution model, for each processor:

$$t_{fetch}(x_i) \leq t_{compute}(x_j) \quad (2)$$

t_{fetch} = time to fetch data into the local memories and $t_{compute}$ = time to compute x_j.

Assuming pipelined compute units (i.e. modulo scheduling) and a data parallelism N_j:

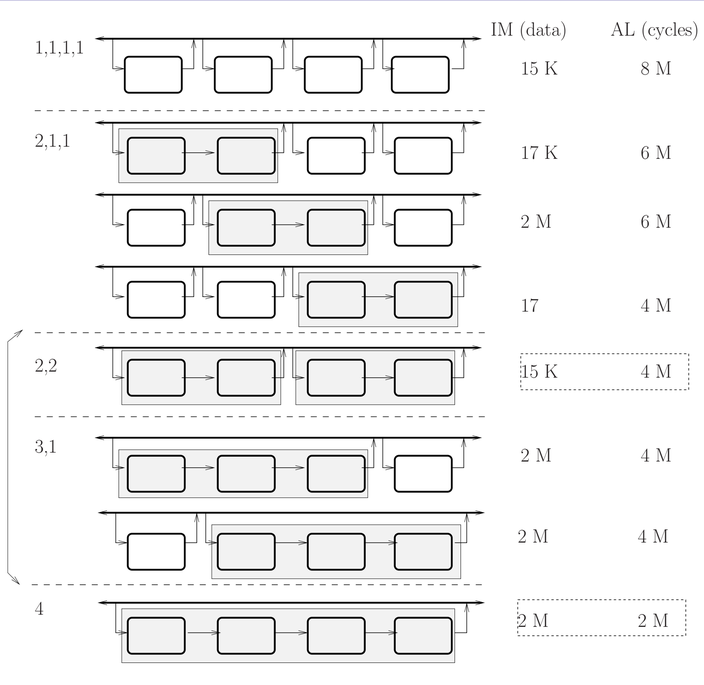
$$t_{compute} = \frac{C_j x_j}{N_j} \quad (3)$$

$$t_{fetch} = \begin{cases} L_m + \frac{x_i}{m} & \text{if } x_i \text{ comes form } Mem \\ \frac{C_i x_i}{N_i} & \text{if } x_i \text{ comes form } P_i \end{cases} \quad (4)$$

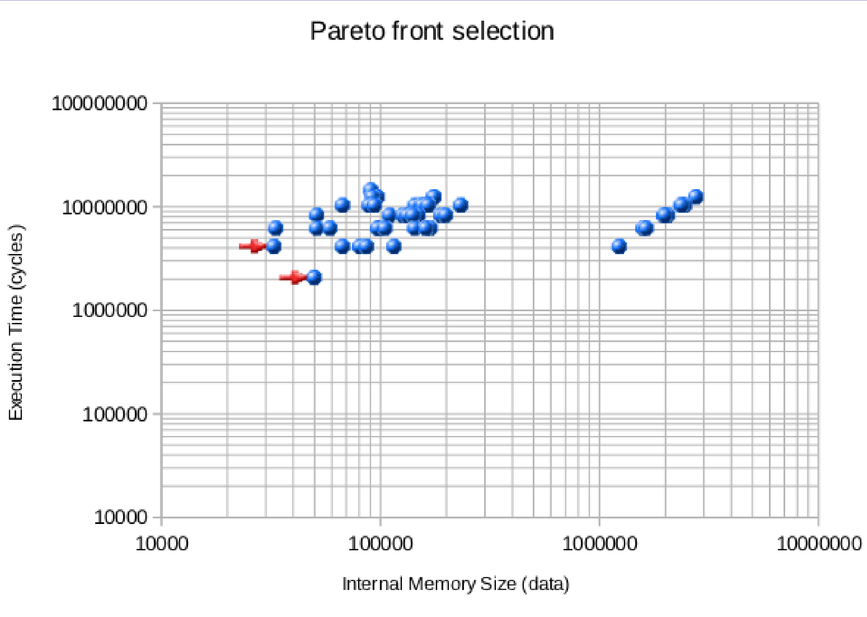
where m is the external memory bandwidth.

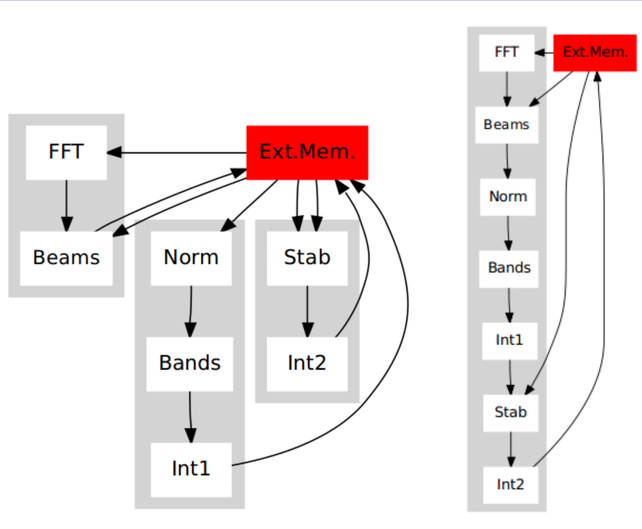
Using equations (2)(3)(4) in the matrix system (1), we obtain a system of inequalities in the unknown N and X_{out} describing a communication configuration. Solving the system means finding N and X_{out} that mask the time to access data (including latency) and respect the external memory bandwidth.

Design Space Exploration



Results: Hydrophone monitoring





The design space contains 54 fusions and 2 are selected. The exploration lasts 3 seconds.

References

[1] R. Corvino, A. Gamatié and P.Boulet. Architecture Exploration for Efficient Data Transfer and Storage in Data-Parallel Applications In *Euro-Par 2010 - Parallel Processing*

[2] R. Corvino, A. Gamatié and P.Boulet. Design Space Exploration for Efficient Data Intensive Computing on SoCs. In *Springer Handbook of Data-intensive computing*