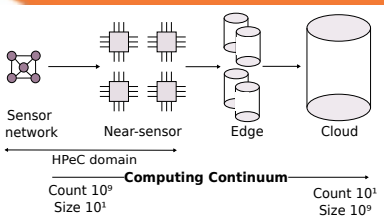
Chaire Internationale : Design Methodologies and Tools for Adaptive Machine Learning at the Network Edge

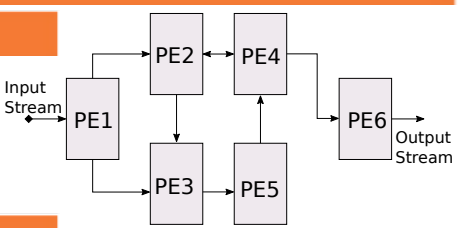
Shuvra S. Bhattacharyya, **J. Faye**, J-F Nezan,
K. Martin, F. Kermarrec, M. Pelcat

Anisotropic and heterogeneous architectures for streaming applications



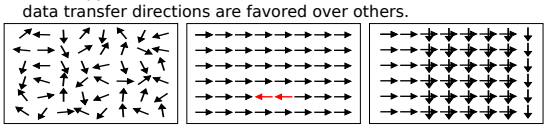
Introduction

In the design of a computing continuum, the trend is to compute data as close as possible to the sensors. Increasingly complex workloads need to be supported by HPeC systems yunder strong energy constrains which can only be satisfied by the use of heterogeneous SoCs. A dataflow MoC models coarse-grain datapath in which data is transfered locally. This poster presents the notion of anisotropy and its relationship with throughput on a canonical application example.



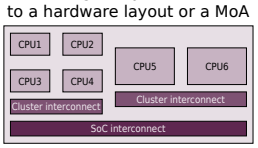
Anisotropy and Heterogeneity

1) Anisotropy refers to SoC architectures where some data transfer directions are favored over others.



a) Isotropy b) pipelined anisotropy c) systolic anisotropy

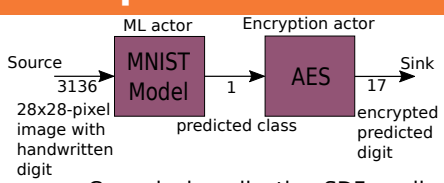
2) Heterogeneity either refers to a hardware layout or a MoA



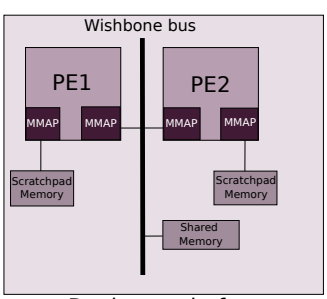
Anisotropy is the property of an object to assume different properties in different directions. When applied to SoC architectures, anisotropy implies that some data transfer directions are favored over others.

We reduce the definition domain to a design level of abstraction, considering services offered by PEs which can be regarded as computing black boxes.

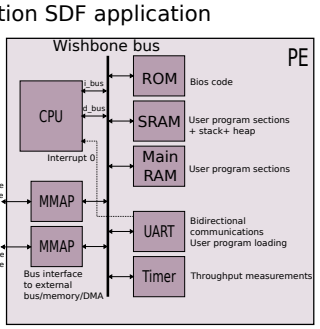
Experimental Setup



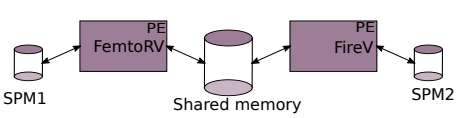
Canonical application SDF application



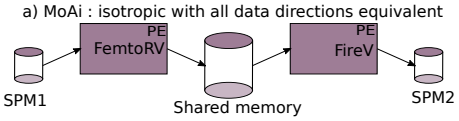
Dual core platform



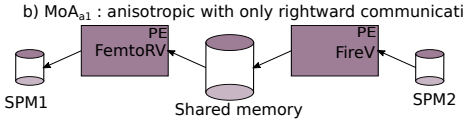
Internal structure of a PE.
The CPU depends on core type.



a) MoAi : isotropic with all data directions equivalent



b) MoA_{a1} : anisotropic with only rightward communications allowed



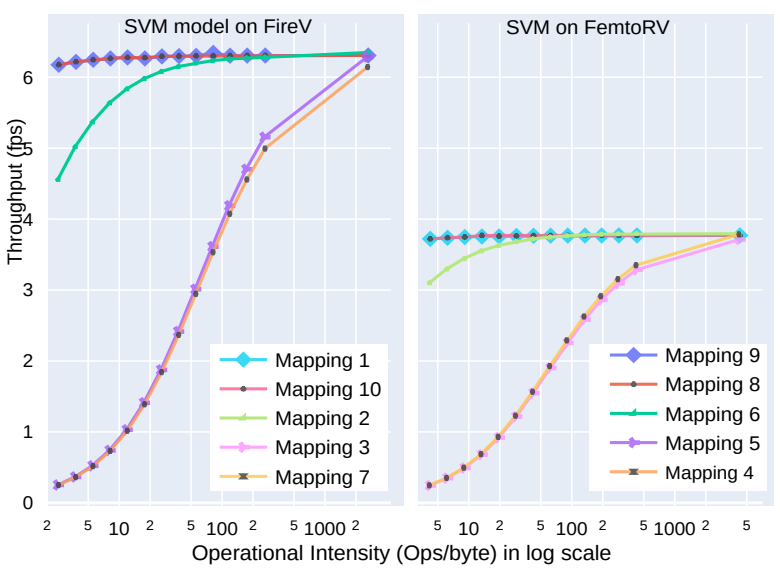
c) MoA_{a2} : aniisotropic with only leftward communications allowed

The 3 MoAs considered in our experiments out of 81 possible MoAs with the given PEs and memories.

Results

Mapping	MoAs	SPM1	FemtoRV	Sh. Memory	FireV	SPM2
Mapping 1	MoA_{a1}, MoA_i	Source	MNIST	Pred.	AES	Sink
Mapping 2	MoA_i	Source/Sink	MNIST	Pred.	AES	-
Mapping 3	MoA_i	Sink	MNIST	Pred.	AES	Source
Mapping 4	MoA_i	Source	AES	Pred.	MNIST	Sink
Mapping 5	MoA_i	Source/Sink	AES	Pred.	MNIST	-
Mapping 6	MoA_i	-	AES	Pred.	MNIST	Source/Sink
Mapping 7	MoA_i	-	MNIST	Pred.	AES	Source/Sink
Mapping 8	MoA_{a2}, MoA_i	Sink	AES	Pred.	MNIST	Source
Mapping 9	MoA_i	-	AES	Pred./Sink/Source	MNIST	-
Mapping 10	MoA_i	-	MNIST	Pred./Sink/ Source	AES	-

Algorithm-to-architecture mappings with supported MoAs. MoA_{a1} and MoA_{a2} are anisotropic while MoA_i is isotropic.



Conclusion

This poster introduce the concept of architecture anisotropy and demonstrated in a small example that using an anisotropic model of architecture (MoA) can force the system to exploit the natural flow of data, reaching higher throughput and isolating data sources from data sinks. Like heterogeneity, anisotropy can apply to a hardware layout or an MoA. Using anisotropic MoAs creates new opportunities for studying whether hardware should be made anisotropic and to constrain mapping solutions for reaching higher throughput.