Application Architecture Adequacy through an FFT case study

Emilien Kofman12 Jean-Vivien Millo1 Robert de Simone1

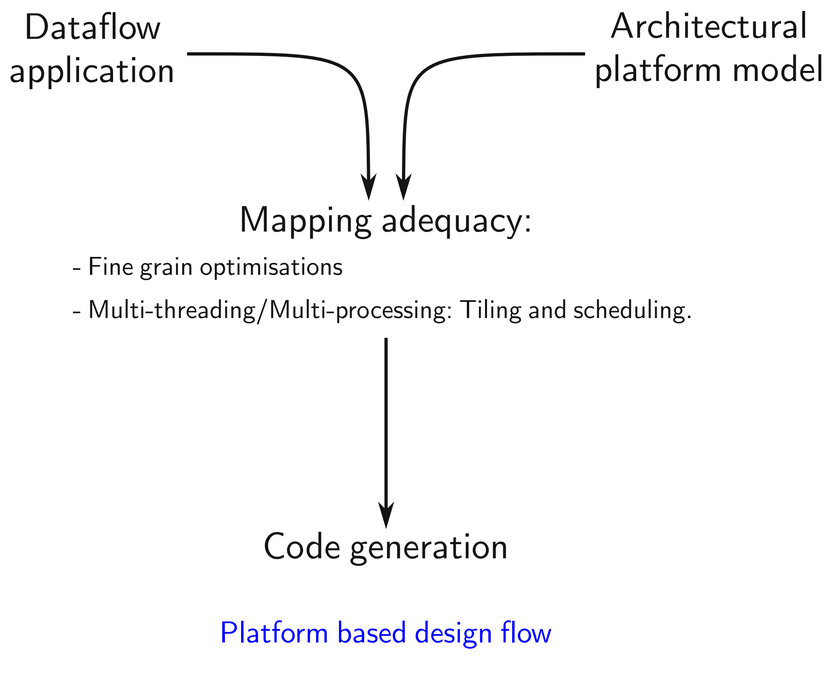
1 INRIA Sophia-Antipolis, Aoste team (INRIA/I3S/CNRS/UNS), 06560, Sophia-Antipolis, France
2 Univ. Nice Sophia Antipolis, CNRS, LEAT, UMR 7248, 06900 Sophia-Antipolis, France
{emilien.kofman, jean-vivien.millo, robert.de_simone}@inria.fr

Motivations

Application Architecture Adequacy (AAA) aims at tuning an application to a given hardware architecture. However it is still a difficult and error prone activity. As like as in Hardware/Software co-design, it requires a model of both the application and the architecture. With the new highly-parallel architectures, AAA should also allow a fast exploration of different software mapping granularity in order to leverage better the hardware resources without sacrificing too much productivity. This work extracts from a case study a methodology based on dataflow modeling to make the software both faster to develop and suited to the target.

Approach

This is a very broad topic which can be split in different sub problems. For each of them we give the solutions we have explored or considered. Problems 3 and 4 are related to non-functionnal specifications (Performance, Temperature, Power) and are studied within the HOPE project (Hierarchically Organized Power/Energy management).



Platform based design flow

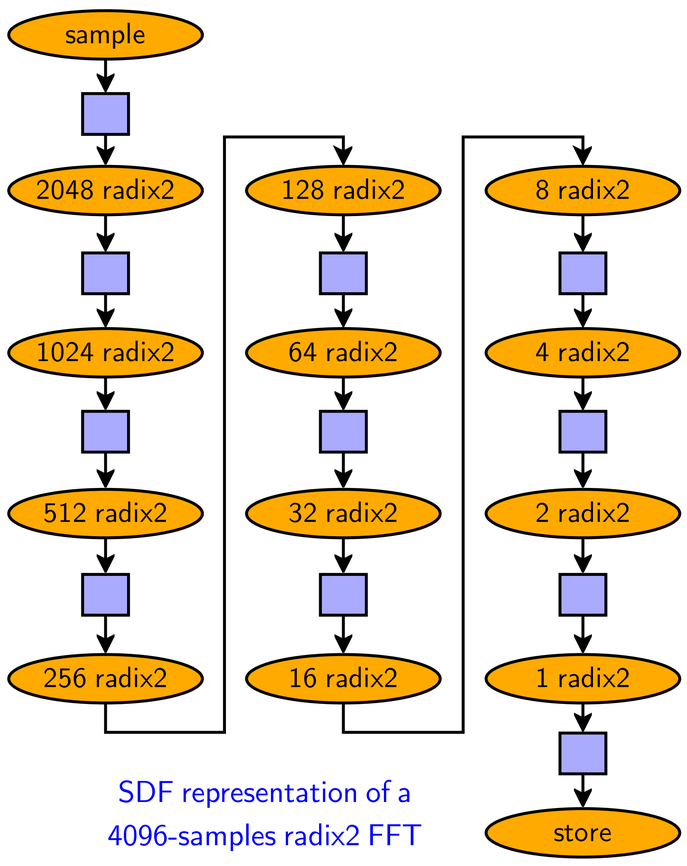
Dataflow graph representation

An SDF is a graph structure in which every vertex has a type: it is either a Place or an Agent. The places allow to model data communication between the agents which run the actual algorithm subfunctions. The places does not describe a physical memory: this is what the implementation should decide. In this example multiple implementations are possible:

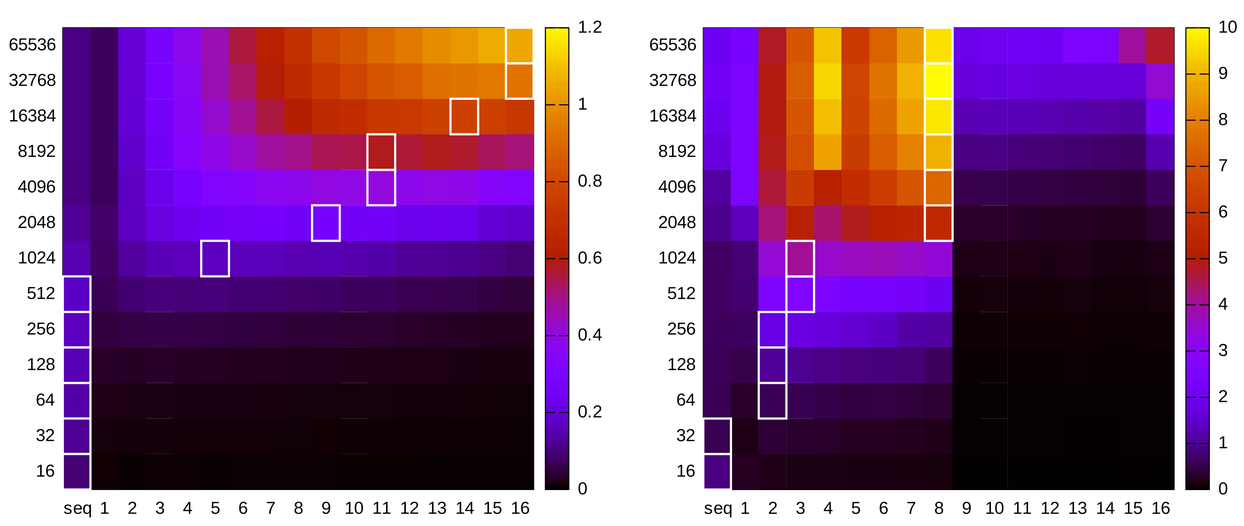
- Merge the places in one physical memory and assume they are never used at the same time.
- Split the places in different physical memories (this allows pipelining).

Tiling and scheduling

This requires either an estimation of agent processing time, or benchmarking these agents in order to map agents efficiently on devices. The dataflow process network description of SW exposes clearly the communications and the data sizes. Tiling is efficient (performance and power) if parallelism is achieved with the maximum data locality. Thus the knowledge of the memory hierarchy (number of cores: CPUs/GPUs/accelerators, caches, number of DMAs) is required.



SDF representation of a 4096-samples radix2 FFT



Benchmarking the fft for different tilings within two different plateforms: Kalray MPPA-

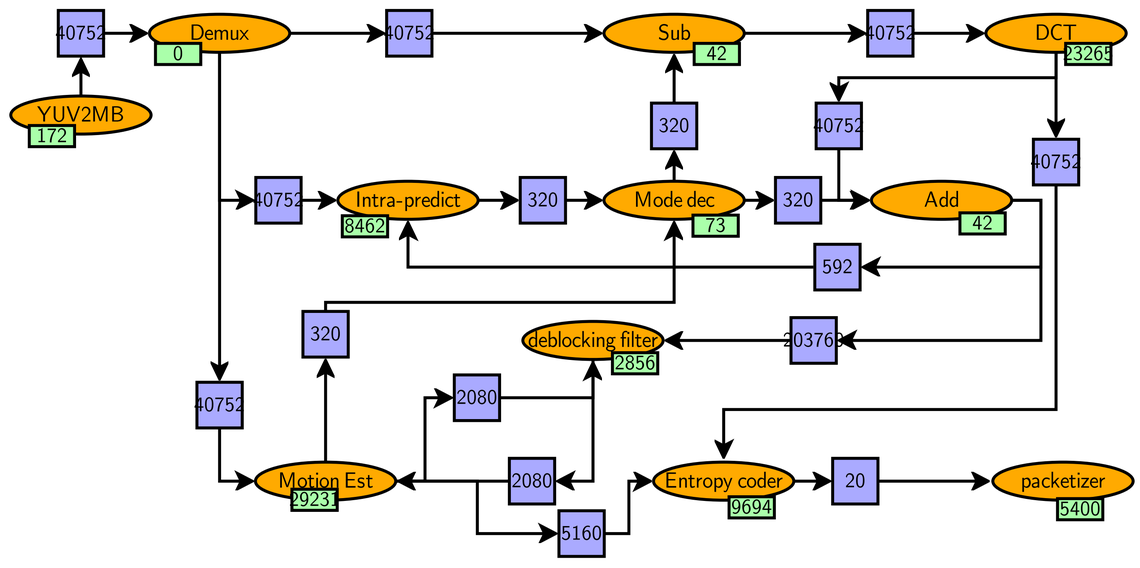
Fine grain optimisations

Most of these fine grain optimisations (Vector processing, VLIW, accelerators) are cumbersome when it comes to implementation and require deep knowledge and understanding of the HW.

Some are not in the scope of this study (VLIW optimisations) but most of them could benefit from an automated description of the HW (register sizes, DRAM burst sizes, vector sizes...).

Modeling HW

This work was conducted in the context of the HOPE project which investigates a relevant solution for designing power efficient system on chip devices early in the design flow. HOPE will develop a modeling approach on top of classical design flows based on existing standards (SystemC OSCI TLM-2.0 , UPF, IP-XACT). Although it is primarily focused on power modeling, the same workflow allows to explore other non-fonctionnal aspects.



H.264 graph annotated with benchmarking results.

Source: "A YAPI system level optimized parallel model of a H.264/AVC video encoder" H.K. Zrida et al., 2009 IEEE/ACS International Conference on Computer Systems and Applications.

Future works

http://anr-hope.unice.fr

SDF allows to model more sophisticated algorithms (ex: h.264 encoder). The multimedia applications are especially interesting because they have embarassingly parallel computataions which need to be optimised.