

Performance Modeling Reconfigurable Computing with Structured Parallelism

Making high-level RC more predictable

Pascal Jungblut
ISC PhD Forum 2019
17.06.2019



With **Structured Parallel Programming** patterns can be used to separate the implementation from the semantics.

FPGAs promise better efficiency than CPUs but programming models are only slowly evolving. Additionally performance modeling for high-level programming is difficult compared to cycle-accurate low-level simulations.

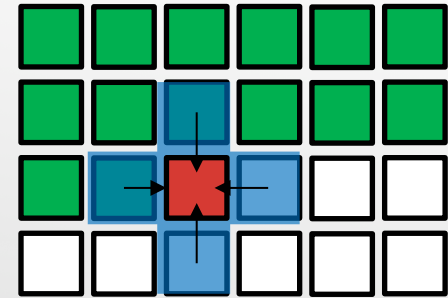
Higher-level abstractions need accurate performance models!



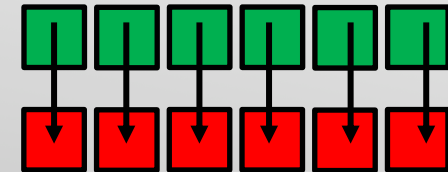
Patterns or **algorithmic skeletons** can be combined to form more complex algorithms.

Each pattern has unique properties, e.g.:

- parallelism, decomposition
- complexity
- memory access patterns
- memory and cache pressure



stencil pattern



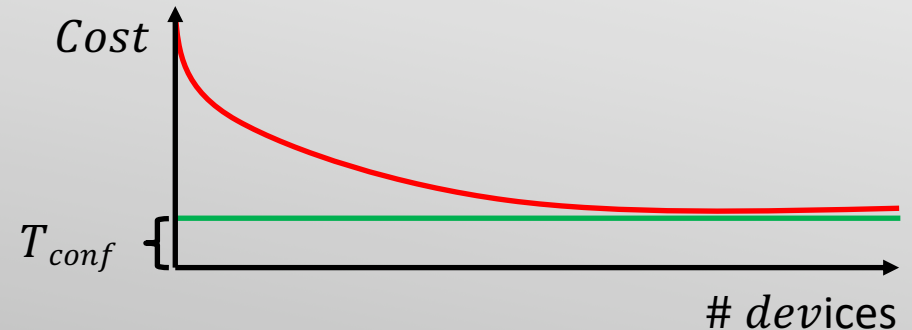
map pattern



Reconfigurable Computing adds several challenges

- *When* to reconfigure?
- *What* to reconfigure? → partial reconfiguration is possible
- NICs' bandwidth exceed host-device bandwidth
- How to split the work with other accelerators/CPU's?

→ How good is the performance model?



Early in PhD! Currently gathering data to build a reliable model:

Single Patterns

Model Building + Evaluation

Combined Patterns

Evaluation

Example: Stencil → shape, blocking, sizes, precision, ...

First results: reconfiguration time is almost linear with bitstream size and very predictable!

→ Dynamic reconfiguration at runtime seems possible/beneficial

Hardware:

- Zybo Z7 for student works
- Stratix 10 (and hopefully more soon)

