Accelerator Design Optimization Using A Functional Data-Parallel Language

Naums Mogers

Supported by
arm Research
EPSRC Centre for Doctoral Training in Pervasive Parallelism
THE UNIVERSITY of EDINBURGH informatics

MOTIVATION

1. Compute-intensive **apps** need to be HW-accelerated

DNNs · SLAM · ISP · Graphs

2. It is hard to design accelerators that extract parallelism efficiently across problem domains and **dimensions**

VGG · LSTM · ResNet

3. An automatic approach to designing versatile **HW accelerators** is needed

ASIC · FPGA · CGRA

METHODOLOGY

1. Express applications with **high-level functional patterns**

Lift language

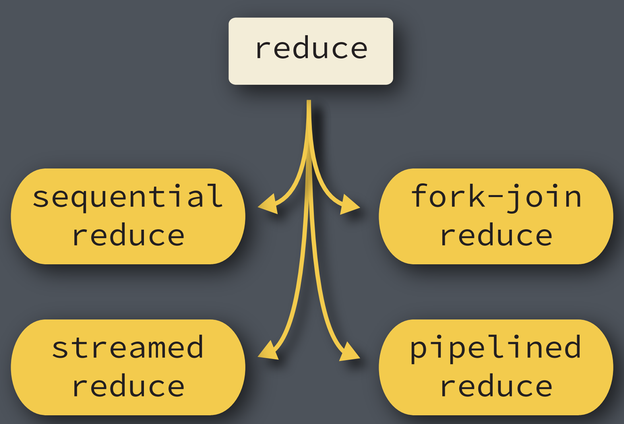
2. Leverage rich AST within the compiler to generate a search space of designs **through:**

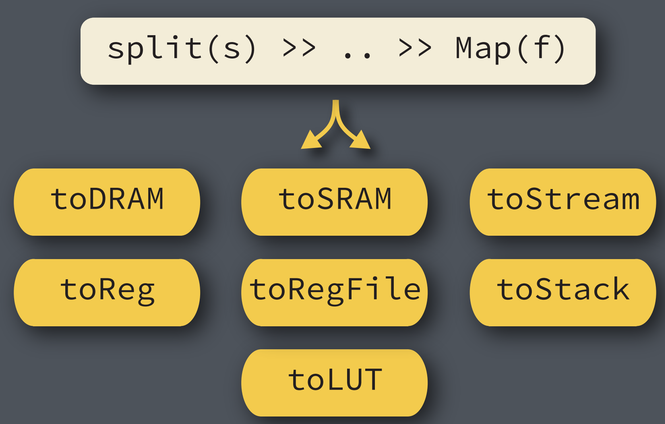
Rewrite Rules · Tuning parameter constraint inference

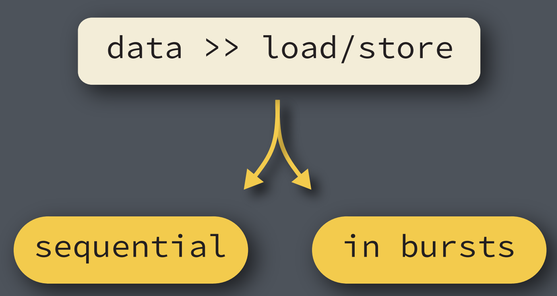
3. Compile to HDL that is **generic high-level expressive** and supports **ASIC FPGA CGRA**

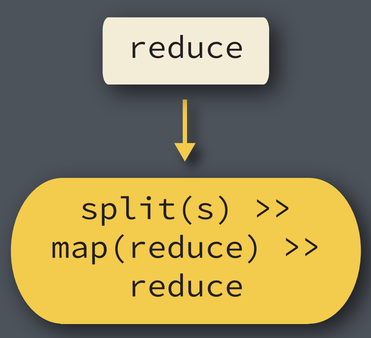
Spatial lang

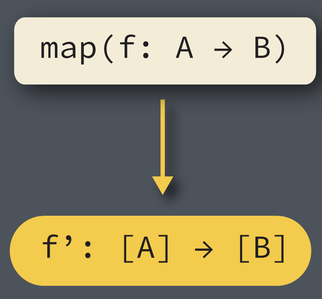
IR TRANSFORMATIONS

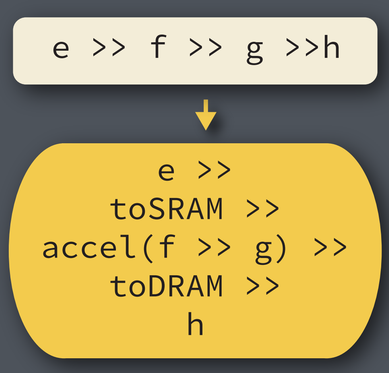








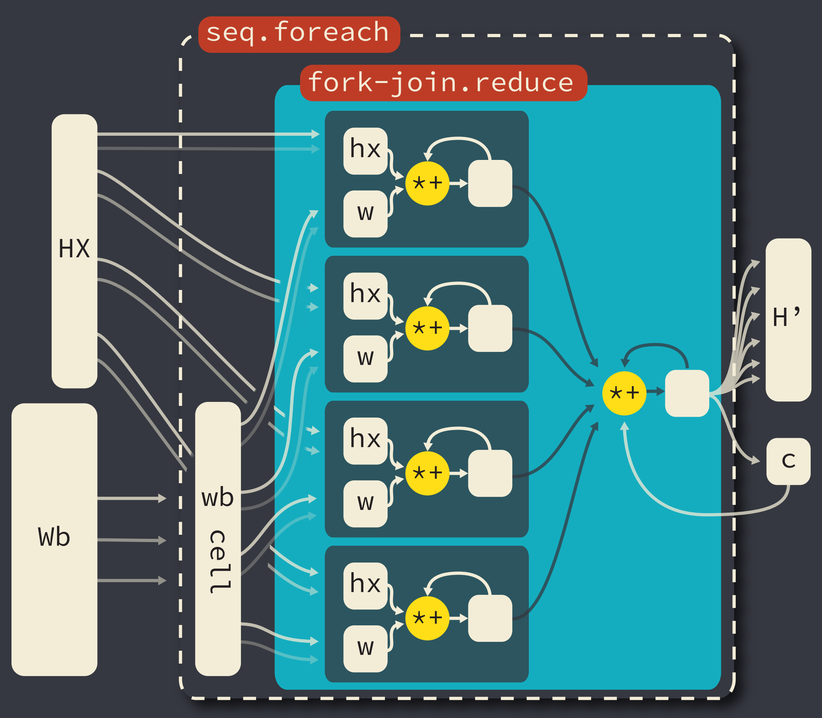




CASE STUDY: LSTM OPTIMISATION

```
lstm: ((x, h, c, w, b) >>
        mapAccum(reduce))
```

```
(x, h, c, w, b) >>
  mapAccum(forkJoin.reduce)
```




```
(x, h, c, w, b) >>
  mapAccum(pipe.reduce)
```