DEPARTMENT OF COMPUTER SCIENCE AT THE UNIVERSITY OF COPENHAGEN

Bachelors project

Auto-tuning Futhark

Simon Rotendahl (mpx651) & Carl Mathias Graae Larsen (pwh334)
{*simon, cala*}@di.ku.dk

Contents

May 2019

Abstract

Stuff

1 Introduction

Stuff

2 Background

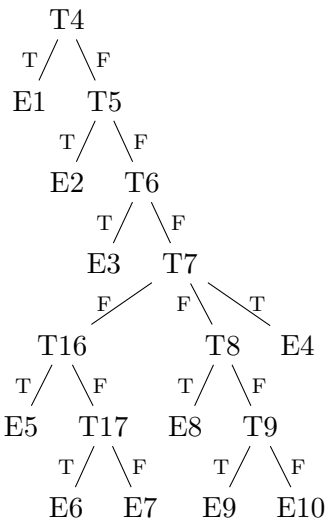
2.1 Futhark

A common way to increase computer performance, is to increase the capacity for parallelism. For practical usage, however, this is difficult to implement, due to low-level GPU-specific languages requiring domain specific knowledge to make full use of. A wast amount of work has gone into transforming high-level hardware-agnostic code into these low-level GPU-specific languages [3].

The programming language **Futhark** aims to solve this problem. The creator of Futhark writes this nicely on the home page for the language *"Because it's nicer than writing CUDA or OpenCL by hand!"* [1]. On the same page, Futhark is described, more specifically, as a *"a statically typed, data-parallel, and purely functional array language"*. Better than a description, is an example:

```
1 let dotprod [n] (xs: [n]f32) (ys: [n]f32)
    : f32 =
2     reduce (+) 0f32 (map2 (*) xs ys)
3
4 let main [n][m][p] (xss: [n][m]f32) (yss:
     [m][p]f32): [n][p]f32 =
5     map (\xs -> map (dotprod xs) (
    transpose yss)) xss
```

Figure 1: Matrix-matrix multiplication in Futhark [2]



2.2 Incremental flatting

3 Design

References

[1] Troels Henriksen. *Why Futhark?* May 2019. URL: https://futhark-lang.org/.

[2] Troels Henriksen et al. *Experimental infrastructure for the paper "Incremental Flattening for Nested Data Parallelism" at PPOPP'19.* May 2019. URL: https://github.com/diku-dk/futhark-ppopp19.

[3] Troels Henriksen et al. "Incremental Flattening for Nested Data Parallelism". In: *Proceedings of the 24th Symposium on Principles and Practice of Parallel Programming*. PPoPP '19. Washington, District of Columbia: ACM, 2019, pp. 53–67. ISBN: 978-1-4503-6225-2. DOI: 10.1145/3293883.3295707. URL: http://doi.acm.org/10.1145/3293883.3295707.