

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

1	Introduction	1
2	Background	1
3	Design	1
3.1	Structure of the threshold parameter	1

May 2019

Abstract

Stuff

1 Introduction

Stuff

2 Background

3 Design

3.1 Structure of the threshold parameter

The aforementioned threshold parameters, can be dependent on each other. Imagine a nested loop where it is determined, for the outer loop, that it should not be executed in parallel, therefore the inner loop should also not be executed in parallel, thereby making the inner dependent on the outer. These dependencies builds a tree, which an example of can be seen in figure 1.

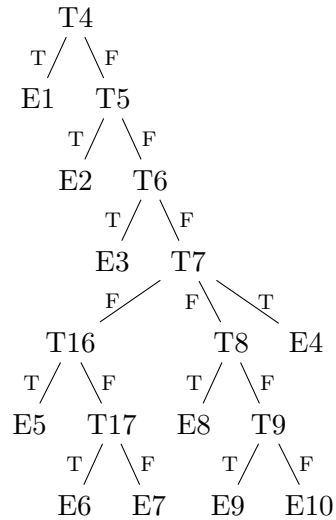


Figure 1: The dependencies between thresholds, of the test program `LocVolCalib.fut`. The dependency is based on a comparison of the threshold (T), the edge to the next node is then taken based on that comparison. Each path through the tree, to one, or more, end nodes, is then an execution of the program.

To tune a program we need to examine each execution path. It is important not to get an end node confused with an execution path. Two example of paths through the tree in figure 1, that show this, could be;

- $\{(T4, \text{False}), (T5, \text{False}), (T6, \text{False}), (T7, \text{True})\}$
- $\{(T4, \text{False}), (T5, \text{False}), (T6, \text{False}), (T7, \text{False}), (T8, \text{False}), (T9, \text{True}), (T16, \text{False}), (T17, \text{True})\}$

The first path is simple, the code represented by T4, T5, T6 is executed in parallel, where everything after it, is executed sequentially. The second path is more interesting, T7 has two child nodes, that are reached with a false comparison. Here it is clear that two end nodes is reached, namely (E6, E9). This is also important to note, because we could have a forest, instead of a single tree, and this would leave multiple end nodes.