Data-Centric Parallelisation

Magnus Morton
magnus.morton@ed.ac.uk
Post-doctoral Researcher at the University of Edinburgh

Björn Franke
bfranke@inf.ed.ac.uk
Reader at the University of Edinburgh

Traditional parallelising compilers e.g. ICC, Clang/Polly, GCC/Graphite parallelise loops by building parallel schedules while respecting all sequential data dependencies. This process is code-centric and applied on a loop-by-loop basis, ignoring the rest of the program, and is particular to the semantic intentions of any data structures used in the loops. With the Data-Centric Parallelisation (DCP) project, we aim to replicate the holistic approach of the human expert, taking into account the context of the whole program and knowledge of the most commonly known abstract data types and their implementations.

Our main aim is to develop a novel "data-first" paradigm for automatic parallelisation of sequential legacy code, outperforming existing parallelising compilers on irregular, pointer based or control flow dominated applications.

As a motivating example, consider a program that sequentially traverses and increments each element in a simple singly-linked list. Traditional parallelising compilers are unable to parallelise such a program, as they are unaware of the context of the complete program. We observe that the program is traversing over a linked list and **increase the level of abstraction** e.g. to a set if we observe that the element operations are independent of each other and traversal order, and now our loop is trivially and perfectly parallelisable.

This linked list example is naive, but important. Hand-implemented linked lists are commonly found in legacy and scientific code, and library implementations are often used needlessly. Similar pointer-chasing code is often used to implement graph structures. This is conceptually convenient, but again makes parallelisation and sequential optimisation difficult. Graph data structures are commonly used in data science and network analysis. Software and hardware techniques for accelerating graph programs is an open research problem, and there are several popular and high-performance graph libraries available.

In this talk, I will give an overview of the DCP project, discussing the background, motivation and our planned approach to the problem, along with some early results. I will also discuss my planned work applying the principles of DCP to the domain of graph data structures.