

Abstract

**A novel automatic parallelization model for sequential code using
Python programming language.**

By:

Ahmad Bassam Marzouq

Supervisor

Prof. Afif Almgawish

Co- Supervisor

Dr. Ayman M Abdalla

In this thesis, we propose and design a model called PyParallelize for automatic parallelization of a given source code. This thesis evaluates different approaches done by other researchers and compares it to our model. The model implemented using Python programming language. The model uses different approaches involved around Abstract syntax tree (AST) manipulation helped by the usage of external libraries. This thesis is focused on task level parallelism. The model can be used on other problems besides automatic parallelization.

This thesis identified the problems around parallel conversion and proposed solutions around them by designing a model that implements methods manipulating ASTs, patterns generation, pattern rules and pattern matching to analyze the source code. The usage of ASTs helped the proposed model easily automate the process of parallel conversion by providing information about the source code without the need of any modification or additional knowledge from the programmer.

The proposed model was implemented and then tested in Al-Zaytoonah Private University JadHPC lab. The obtained results show that PyParallelize is able to achieve parallel speed up and performance similar to PIPS (Irigoin, Jouvelot, & Triolet, 1991) and Cetus (Lee, Johnson, & Eigenmann, 2004) . The model PyParallelize has met limitation of this thesis. PyParallelize fails to focus on data level parallelism but focuses on task level parallelism. In addition, PyParallelize fails to parallelize loops with more than two levels due to the complexity of the analysis of such loops.

It is concluded that PyParallelize is sufficiently efficient for automatic parallel conversion and provided a model that never been implemented before to our best knowledge while giving very accurate results. In addition, PyParallelize was able to implement a method for detecting recursive functions.