

Automatic Parallelization of Go programs

30. November 2016

candidate	Morten Fliflet Johannessen
supervisors	Martin Steffen and Violet Ka I Pun
group	PMA
type	60 ECTS
recommended background	programming languages, program analysis and compilers, concurrency
study program	computer science
planned date of completion	May 2018

Short description

The task is to design and implement a program analysis and transformation for Go, with the purpose of parallelization and performance increase.

Background and motivation

Go [2] is a relatively new general-purpose language, which gained traction recently, being backed by Google and being used to implement applications such as Docker and dropbox. Influenced by C in its syntax and its relatively compact definition, one core feature of Go is *concurrency*. It is supported in the form channel communication in combination with “goroutines”, a form of light-weight threads. The latter can be understood simply as a function body being executed concurrently.

Concurrency and parallelism are crucial programming mechanisms, though concurrent programs are notoriously hard to get right. One approach is to not to burden the programmer with coming up with a good parallel design initially, instead allowing to concentrate on a typically more straightforward) sequential design. Afterwards, the program is transformed automatically in one making use is asynchronous goroutines (and additional channels) instead of the original, sequential solution. The transformation, obviously, needs to preserve the semantics.

Problem setting

A prospective student would try design and implement and analysis and transformation. The task involves

- get an overview over relevant approaches in the field of automatic parallelization
- identify a useful set of patterns resp. problems that promise being parallelizable in a fruitful manner.
- design & implement the analysis and transformation
- run experiments on hand-made examples, resp. existing open-source software projects, gaining empirical results
- contribute to publications/conference submissions.

The work is done in connection with the *GoRETech*-project.

Keywords: program analysis, concurrency, Go language,

References

- [1] A. A. A. Donovan and B. W. Kernighan. *The Go Programming Language*. Addison-Wesley, 2015.
- [2] The Go programming language specification. <https://golang.org/ref/spec>, Nov. 2016.
- [3] M. Summerfield. *Programming in Go*. Addison-Wesley, 2012.