Go Compiler

Release Plan

January 29th, 2018

Baiwen Huang (Product Owner), Kyle Remmert (Scrum Master),
Petar Zaninovich, Trevor Ching, Vincent Kim

High level goals

Using Oracle's Multilingual Engine (MLE) we will map the Abstract Syntax Tree nodes of Go onto Truffle, a language implementation framework. Truffle builds a self optimizing AST, with each node being specialized based on the content. Each node in the AST will have an execute method in which it executes its children and returns its result. When the execution count of a node reaches a predetermined threshold, Truffle will trigger a partial evaluation call to Graal. Graal is a dynamic compiler that will optimize or deoptimize the selected node. Our task is to parse the Go AST into Truffle, and create the appropriate execute methods.

User stories for release

Sprint 1

- As a user, I would like to parse the Go AST nodes into the Truffle interpreter.
- As a user, I would like my nodes to be optimized.
- As a developer, I would like learn Truffle/Graal and know the technologies behind it.

Sprint 2

- As a user, I would like my Go source code to have a good IR.
- As a user, I would like simple control flow to be implemented.
- As a user, I would like variables to be implemented.
- As a user, I would like to be able to import packages and libraries.
- As a customer, I want a toolchain to parse my Go source code.