Cmpe491 Midterm Progress Report

Yiğit Özkavcı

October 2017

Contents

1 Introduction

This is the midterm report for project Ivy, a programming language for writing smart contracts on Ethereum Virtual Machine.

Ethereum is a decentralized plaform that runs smart contracts: pieces of codes that have the ability to run on any blockchain network. In order to write smart contracts, one should deploy a EVM-executable bytecode into blockchain network, which is not practical since bytecode is a sequence of hex characters; nothing more. To overcome this problem, EVM-compatible programming languages are being designed in order to abstract the problem of having to deploy plain bytecode.

There are already programming languages targeting EVM, including Solidity[1], Bamboo[2] and Viper[3].

2 Concerns on EVM

Designing a programming language while targeting EVM has several problems that general purpose programming languages don't. In this section, we will investigate problems that we have / may encounter.

2.1 Gas Cost

Gas is the pricing of computations that are run by smart contracts. Basically, each interaction via a smart contract must be paid in units of gas. This also means that the higher abstraction level we have for EVM computations, the more costly our computations will be, because of the abstraction layer switch of the compiler.

This is where compiler optimizations are critically important. In the context of smart contracts, compiler optimization is not just about how much of the RAM or CPU of the user you consume, but also the real money of the user you waste in the runtime. We will investigate optimisations in terms of stack operations (see 3.1) and memory allocations (see 3.2) in further sections.

2.2 Entrance Point

A smart contract has two main phases in its life time:

1. **Construction**

   When a smart contract is deployed on EVM, a code is being executed in the same fashion Java[4] constructs instances of objects described in their class.

2. **Code to execute upon receiving a message**

3 Optimisations

3.1 Stack

3.2 Memory

4 Syntax

References

[1] Solidity: Contract-Oriented Programming Language, https://github.com/ethereum/solidity

[2] Bamboo: a morphing smart contract language, https://github.com/pirapira/bamboo

[3] Viper: an experimental programming language targeting EVM, https://github.com/ethereum/viper

[4] https://en.wikipedia.org/wiki/Java_(programming_language)