Golang Bindings for VPP TCP Stack

Internship Projects/Mentors

Title	Golang Bindings for VPP TCP Stack
Status	APPROVED BY TSC
Difficulty	MEDIUM

Description

VPP provides a host stack that supports TCP, UDP, and QUIC. The VPP transport stack is both much more scalable and much more performant than the kernel transport stack. Golang is a very commonly used Cloud Native Development langauge. Because Golang utilizes its own net.Listener and net.Conn instead of the C style BSD socket API, it should be much easier to produce a drop in replacement binding to the VPP TCP stack rather than the kernel stack. This project is to write an opportunistic (meaning it falls back to normal kernel networking if VPP Host Stack isn't available) Go binding to the VPP Hoststack.

Additional Information

1. VPP Hoststack Wiki page

Learning Objectives

1. Learn to use Github for SCM and Github Actions for CI
2. How to use Go to pass file descriptors over unix file sockets
3. How to write high performance memory protocols (tcp buffers in this case) over shared memory in Go
4. Learn how the native thread model and the Go goroutine model interact around concurrency.
5. How to write functional and performance tests in Go, as well as build robust CI utilizing those tests
6. How to benchmark TCP performance
7. How to present a technical topic in a Webinar

Expected Outcome

Deliverables:

1. Write and test a Golang binding for the VPP TCP stack
2. Write functional tests for Golang bindings for the VPP TCP stack
3. Write performance tests for Golang bindings for VPP
4. Write CI utilizing the funtional and performance testing
5. Benchmark Golang+VPP vs Golang+Kernel
6. Prepare and present an FD.io Webinar on the experience
7. Optionally, if things go smoothly, there may be an opportunity to explore a Golang binding for the VPP QUIC stack.

Relation to LF Networking

FD.io

Education Level

Undergraduate should be fine as long as they have programming experience. This projects **presumes** the ability to program generally.

Skills

Skills in Go are highly preferred. Ability to read C code is extremely helpful.

Future plans

Work is underway to allow the opportunistic use of VPP's Host Stack in Envoy. That work plus this work could lead to Cloud Native applications that could utilize an Envoy based Service Mesh with their 'TCP' traffic moving purely over a memory buffer to Envoy and then out via kernel bypass via VPP. This approach would significantly improve performance and scaling when deployed.

Preferred Hours and Length of Internship

Either 40 hours a week for 12 weeks or 20 hours a week for 24 weeks are acceptable.

Mentor(s) Names and Contact Info

Ed Warnicke - hagbard@gmail.com - edwarnicke on the FD.io slack

Florin Coras - florin.coras@gmail.com - Florin Coras on the FD.io slack