

Abstract

This project aims to implement a fast packet processing framework in Java for use within network middleboxes, which are required to inspect, transform and forward packets at line rate of speeds in excess of 10Gbps. Such applications are normally custom designed for the specific task and struggle with scaling and evolving technologies. Programmable middleboxes using medium to high level languages offers a better solution, considering the emerging cloud computing support which utilise dynamic virtual machine allocation.

In this report, a number of techniques for data sharing between native memory and Java are explored, performance tested and evaluated to determine the most suitable. This technique is then utilised to develop a fast packet processing framework written in Java which uses the existing Data Plane Development Kit (DPDK).

Middleware applications are then implemented using the same algorithm in Java and C and tested to compare performance in packet and data throughput of the application. Through this, further improvements are suggested and implemented to fully maximise memory and CPU usage. The initial performance comparison looked promising as the Java implementation reached speeds of 90% of the native application. However, as the middleboxes become more complicated, this speed dramatically drops between the range of 20% and 75% depending on the packet sizes.

Reasons for these problems are discussed and a number of improvements to the project are suggested which could potentially allow performance to become matched.