Comparing DPDK with traditional Linux based networking

With the ever-increasing network speeds of physical links, the processing of packets on network nodes is becoming more and more of a bottleneck. Packet processing on a standard Linux-based network node traditionally involves the operating system (OS). Since an OS is usually optimized for a range of tasks rather than a specific task, using conventional Linux kernel functionalities for packet processing can degrade performance. For this reason, approaches to bypass the kernel have been proposed to perform network processing in user space.

One approach of bypassing the kernel that has attracted growing interest in recent years is Data Plane Development Kit (DPDK). By processing packets entirely in user space, DPDK avoids time-consuming context switches between user space and kernel space. This comes at the cost of one CPU core actively polling for new packets, instead of the network interface card (NIC) triggering interrupts for incoming packets. In addition, DPDK itself mainly provides the poll mode drivers for selected NICs, but the processing of the packets is the duty of the application using DPDK. Thus, while DPDK is suitable for certain application scenarios, there are also numerous use cases that are better suited to be implemented using the Linux networking stack. For example, to establish a Transmission Control Protocol (TCP) connection, an additional user space TCP/IP stack must be implemented or taken from open-source projects. These are generally not as feature-rich as the conventional Linux networking stack and do not necessarily improve performance.

This work aims to find a method to compare applications using DPDK with applications using the Linux network stack. Envisioned is a client-server application that uses iperf3 to generate data traffic.

Prerequisites

To successfully complete this work, you should have:

- very good programming skills in Python and C/C++,
- basic knowledge about Git,
- first experience with the Linux environment.

The student is expected to be highly motivated and independent.

Contact

Email: lars.nolte@tum.de

Advisors

Lars Nolte