TCP (Transmission Control Protocol) is one of the most important network protocols and is extremely widely used. Improving TCP performance can reduce server cluster scale and power consumption, brings both commercial and environmental benefits. Nowadays, the Ethernet technology is developing much faster than storage and CPU technologies, memory access and CPU processing network stack have become the bottleneck of TCP performance on end systems. The constantly increasing network bandwidth has caused a severe burden for CPU, optimizing TCP processing mechanism can relieve CPU from this and improve end system TCP performance.

Traditional TCP processing optimization techniques are focused on checksum calculation, high probability events handling, reducing data copy, increasing cache hit ratio and multithreading, protocol processing is done by host CPU. As the NIC (Network Interface Card)’s computing power gets stronger, the idea of TOE (TCP Offload Engine) comes along. TOE offloads the entire TCP protocol processing from host CPU to NIC, and remarkably improves the end system TCP performance, but its implementation is extremely complex and it can cause security and compatibility issues. TSO (TCP Segmentation Offload) offloads data segmentation and checksum calculation functions to NIC, it optimizes TCP processing from the data sending angle and is very maturely developed. LRO (Large Receive Offload) aggregates consecutive TCP packets into a single one and decreases CPU workload by reducing the number of packets processed by the network stack, but LRO works on the NIC driver level and packets aggregation is still done by host CPU, so it cannot decrease CPU workload greatly.

Considering TOE and LRO techniques’ defects and multicore NPU’s high packet processing capability, we used a multicore NPU as NIC and offloaded TCP packets reordering, checksum calculation functions to it, the multicore NPU aggregates small TCP packets into larger but much fewer packets, thus reducing the number of packets processed by network stack and the number of interrupts generated by NIC, eventually improving the TCP performance on an end system. By adding another storage level, we can also achieve the space-time tradeoff effect. Experiment results show that 4.9Gbps TCP receive data throughput is achieved in a 10Gbps network environment.