Efficient Packet Processing in User-Level OSes: A Study of UML

Submitted by Ravi Prakash Giri (rgiri8)

The paper presents a thorough study of TCP/IP stack in User-mode Linux (UML). Their study attributes the resulting slow-downs of User-level Operating Systems to three main sources: the execution of privileged code, memory management across layers and additional instructions to execute. The authors further describes the User-level operating systems and the set of qualities that they provide to enable server consolidation. They also have described the packet processing in UML both in unoptimized and optimized manner. Finally, the paper evaluates the performance of packet processing in UML.

The paper identifies the three main sources to slow down the TCP/IP stack in User-mode Linux (UML). They are also successful in presenting five optimization techniques to mitigate these bottlenecks. One key strength of this paper is that they have performed a thorough analysis of the performance of their optimization techniques. Their experimental results shows a considerable improvement for network performance over EUL. Although the performance results shown are good enough on outdated processors (Pentium processor) with kernel version 2.4.26, I wonder if their results will still be the same on the latest core processors and recent linux kernel versions.

The results presented by the authors depicts that a ULOS can achieve high network throughput and become a serious alternative for network server consolidation. Similar to the work of this paper, `Operating System Support for Virtual Machines`, describes similar optimization techniques in type-II VMMs. They have also discussed few ways to eliminate the performance overload due to switching between guest kernel and guest user space.