Zero Copy Strategies for Distributed CORBA Objects in Clusters of PCs

Christian A. Kurmann, Laboratory for Computer Systems
ETH - Swiss Institute of Technology, CH-8092 Zurich

Clusters of Personal Computers (CoPs) offer the best compute performance at the lowest price. Workstations with 'Gigabit networking to the Desktop' can enable a new game of multimedia applications that benefit from higher communication bandwidth and lower latency. In order to reach the full Gigabit's speed on normal PCs, with their typically weak memory subsystems, it requires either additional hardware for protocol processing or alternatively a highly efficient software system that circumvents data copies.

We state that Zero-Copy is an essential principle to achieve efficient communication system software. A clean and efficient Zero-Copy regime through all levels of a modern system is challenging and can indeed be achieved in a standard environment using commodity hardware.

We successfully introduced speculation techniques into system software design and managed to implement a clean zero-copy solution entirely in software that runs with commodity network interface cards (NICs). On the middleware layer the MICO CORBA ORB (www.mico.org) has been enhanced to support Zero-Copy by separating control and data transfers.

Optimized ORB with Control/Data Transfer Separation on ORB Level.

TCP/IP-Stack

Even if the socket interface is optimized for Zero-Copy, there still remains a driver copy for IP-fragmentation and header/payload enclosure/segmentation. Ethernet still has an MTU that is smaller than a memory page which requires a memory copy for deterministic packet (de)fragmentation. Getting rid of the remaining copy allows extreme high bandwidth.

DMA-Descriptor Initialization for Zero-Copy TCP/IP and the Socket API.

Speculative Defragmentation: Make the Common Case Fast!

Performance Analysis

By combining both technologies, Zero-Copy Techniques at the Socket Interface and Speculative Defragmentation in the driver, the last copy and with that the largest overhead can be eliminated. An 80% increase in bandwidth has been achieved! Application traces show that speculation often works very well and a control architecture can prevent the speculation rate from dropping. The fallback case when speculation misses, is not much more expensive than the standard operation. It is therefore not much slower.

That means that the common case is fast at the price of a not much slower fallback.