National Technical University of Athens
School of Electrical and Computer Engineering
Department of Computer Science

I/O and Scheduling Techniques for the Efficient Utilization of Shared Architectural Resources on Clusters of SMPs

Ph.D. DISSERTATION

Evangelos L. Koukis

Athens, January 2010

Abstract

Clusters have become prevalent as a cost-effective solution for building scalable parallel platforms to power diverse workloads. Symmetric multiprocessors of multicore chips are commonly used as building blocks for clustered systems, when combined with high-performance interconnection networks, such as Myrinet. SMPs are characterized by resource sharing at multiple levels; Resources being shared include CPU time on cores, levels of the cache hierarchy, bandwidth to main memory, and peripheral bus bandwidth.

The increasing use of clusters for data-intensive workloads, in combination with the trend for ever-increasing cores per processor die, poses significant load on the I/O subsystem. Thus, its performance becomes decisive in determining overall system throughput. To meet the challenge, we need low-overhead mechanisms for transporting large datasets efficiently between compute cores and storage devices. In the case of SMP systems, this means reduced CPU, memory bus and peripheral bus contention.

This work explores the implications of resource contention in SMP nodes used as commodity storage servers. We study data movement in a block-level storage sharing system over Myrinet and find its performance suffers due to memory and peripheral bus saturation. To alleviate the problem, we propose techniques for building efficient data paths between the storage and the network on the server side, and the network and processing cores on the client side. We present gmblock, a system for shared block storage over Myrinet which supports a direct disk-to-NIC server-side data path, bypassing the host CPU and memory bus. To improve handling of large requests and support intra-request overlapping of network- and disk-I/O with minimal host CPU involvement, we introduce synchronized send operations as extensions to standard Myrinet/GM sends; their semantics support synchronization with an agent external to the NIC, e.g., a storage controller utilizing the direct-to-NIC data path.

On the client side, the proposed system exploits NIC programmability to support protected direct placement of incoming fragments into buffers dispersed in physical memory. This enables end-to-end zero-copy block transfers directly from remote storage to client memory over the peripheral bus and cluster interconnect.

Experimental evaluation of the proposed techniques demonstrates significant increases in remote I/O rate and reduced interference with server-side local computation. A prototype deployment of the OCFS2 shared-disk filesystem over gmblock shows gains for various application benchmarks, provided I/O scheduling can eliminate the disk bottleneck due to concurrent access.