Abstract

In both the CPU and GPU computing domains, manufacturers have recently announced NUMA systems designed around heterogeneous memory. By combining memories that are individually optimized for bandwidth, capacity, or cost, these systems can achieve better performance and cost efficiency than possible when employing a single memory technology. In these designs the operating system (OS) will be responsible for performing page migration between one or more NUMA nodes to optimize application throughput and system energy efficiency. In this work we show that current OS page migration mechanisms significantly underutilize the available interconnect bandwidth when migrating pages between NUMA nodes, resulting in poor page migration throughput. To remedy this we propose three page migration optimizations that: First, provide parallel page copies via a multi-threaded CPU routine or DMA based copy engine. Second, perform concurrent multi-page migration to improve memory copy granularity. Third, remove memory management overheads when symmetrically exchanging pages between two NUMA nodes. Implemented on Linux and tested on Intel Xeon and IBM Power systems, our work improves base page migration throughput up to 60%. For transparent huge pages (THPs), migration throughput is improved as much as 440%, while increasing NUMA interconnect bandwidth utilization from under 10% to over 90%.