Quantifying memory bandwidth limitations of current and future microprocessors

Science, Computer Science

Introduction

We propose a new metric, called traffic inefficiency, which quantifies the opportunity for reduction in the traffic ratio. We define traffic inefficiency as the ratio of traffic generated by a cache and some optimally-managed memory. This quantity gives an upper bound on the achievable effective bandwidth for a given memory size, package, and program. Technological trends have produced a large and growing gap between CPU speeds and DRAM speeds. The number of instructions that the processor can issue during an access to main memory is already large.

Methodology

We first surveyed a wide range of these techniques and qualitatively showed that each one exacerbates bandwidth limitations, either directly or indirectly. We also qualitatively analyzed technology trends, showing that future technology is likely to aggravate the bottleneck of the chip boundary. To permit quantification of future bandwidth limitations, we decomposed execution time into processing cycles, raw memory latency stall cycles, and limited bandwidth stall cycles. Using this decomposition, we measured how bandwidth stalls increase, as processors tolerate memory latencies more aggressively.

For our applications running on our most aggressive processor, we saw that the stall cycles due to bandwidth exceeded latency stall cycles in all cases but two. Excluding those benchmarks that fit comfortably in the cache, the stall cycles due to bandwidth limitations ranged from 11% to 31% of the programs’ total execution time. These measurements have significant implications for the designs of future processors.

Result

We have the potential exists to use on-chip memory much more effectively, greatly reducing the number of requests that must be made off-chip. Not surprisingly, no single technique emerged for making better use of the on-chip memory. This fact suggests that future designers should consider on-chip memory systems that are more flexible, allowing the programmer or compiler to tune the on-chip memory system parameters.

We used traffic ratios to compute effective pin bandwidth, and measured these ratios for a range of programs and cache sizes. We found that comparatively large caches eliminated about half of the processor-generated traffic for our small benchmarks.

Conclusion

We concluded that the memory bandwidth about old processors and currently processors had lot of difference . A large percentage of today’s typical processor chip is already devoted to on-chip memory. When enough transistors are available, a greater capacity on-chip will be more important than having all of the on-chip memory be fast memory.