Abstract

Integrated Heterogeneous System (IHS) processors pack throughput-oriented GPGPUs alongside latency-oriented CPUs on the same die sharing certain resources, e.g., shared last level cache, network-on-chip (NoC), and the main memory. They also share virtual and physical address spaces and unify the memory hierarchy. The IHS architecture allows for easier programmability, data management and efficiency. However, the significant disparity in the demands for memory and other shared resources between the GPU cores and CPU cores poses significant problems in exploiting the full potential of this architecture.

In this work, we propose adding a large capacity stacked DRAM, used as a shared last level cache, for the IHS processors. The reduced latency of access and large bandwidth provided by the DRAM cache can help improve performance respectively of CPU and GPGPU while the large capacity can help contain the working set of the IHS workloads. However, adding the DRAM cache naively leaves significant performance on the table due to the disparate demands from CPU and GPU cores for DRAM cache and memory accesses. In particular, the imbalance can significantly reduce the performance benefits that the CPU cores would have otherwise enjoyed with the introduction of the DRAM cache. This necessitates a heterogeneity-aware management of this shared resource for improved performance. To address this, in this thesis, we propose three simple techniques to enhance the performance of CPU application while ensuring very little or no performance impact to the GPU. Specifically, we propose (i) *PrIS*, a prioritization scheme for scheduling CPU requests at the DRAM cache controller, (ii) *ByE*, a selective and temporal bypassing scheme for CPU requests at the DRAM cache and (iii) *Chaining*, an occupancy controlling mechanism for GPU lines in the DRAM cache through pseudo-associativity. The resulting cache,

HAShCache, is heterogeneity-aware and can adapt dynamically to address the inherent disparity of demands in an IHS architecture with simple light weight schemes.

We enhance the gem5-gpu simulator to model an IHS architecture with stacked DRAM as a cache, coherent GPU L2 cache and CPU caches and a shared unified physical memory. Using this setup we perform detailed experimental evaluation of the proposed HAShCache and demonstrate an average system performance (combined performance of CPU and GPU cores) improvement of 41% over a naive DRAM cache and over 100% improvement over a baseline system with no stacked DRAM cache.