With the computing industry’s recent adoption of the Heterogeneous System Architecture (HSA) standard, we have seen a rapid change in heterogeneous CPU-GPU processor designs. State-of-the-art heterogeneous CPU-GPU processors tightly integrate multicore CPUs and multi-compute unit GPUs together on a single die. This brings the MIMD processing capabilities of the CPU and the SIMD processing capabilities of the GPU together into a single cohesive package with new HSA features comprising better programmability, CPU-GPU coherency, shared Last Level Cache (LLC), and shared virtual memory address spaces. These advancements can potentially bring marked gains in heterogeneous processor performance and have piqued the interest of researchers who wish to unlock these potential performance gains. Therefore, in this dissertation I explore the heterogeneous CPU-GPU processor and application design space with the goal of answering interesting research questions, such as, (1) what are the architectural design trade-offs in heterogeneous CPU-GPU processors and (2) how do we best maximize heterogeneous CPU-GPU application performance on a given system. To enable my exploration of the heterogeneous CPU-GPU design space, I introduce a novel computer architectural simulation tool called M2S-CGM. M2S-CGM includes all of the simulation elements necessary to simulate coherent execution between the CPU and GPU with shared LLC and shared virtual memory address spaces. I then utilize M2S-CGM to conduct three architectural studies. First, I study the architectural effects of shared LLC and CPU-GPU coherence on the overall performance of non-collaborative GPU-only applications. Second, I profile and analyze a set of collaborative CPU-GPU applications to determine how to best optimize them for maximum collaborative performance. Third, I study the impact of varying four key architectural parameters on collaborative CPU-GPU performance by varying GPU compute unit coalesce size, GPU to memory controller bandwidth, GPU frequency, and system wide switching fabric latency.