Abstract

In-memory key-value systems offer low latency access to non-persistent data. These systems are typically used as a caching layer and thus generally process a high proportion of read requests. The highly parallel architecture of GPUs is an ideal fit to accelerate high throughput, read-heavy workloads. In this thesis, a hybrid CPU/GPU in-memory key-value system is developed that can process up to 252.4 billion read requests per second. The system utilizes the CUDA platform and is tested using an NVIDIA GTX 1080. The system design takes advantage of optimizations such as latency hiding, CUDA kernel overlapping, pinned memory transfers, and minimizing memory access instructions. All system optimizations are implemented to improve read performance; however, some optimizations are found to benefit both the read and write performance.

The optimizations are implemented incrementally upon an initial design. The impact on performance is evaluated for each incremental change. The effects on performance from different batch sizes, key sizes, word (value) sizes, load factors, client threads, and block sizes is also evaluated.