**CS 415/516 Paper Review**
**Paper Title: Debunking the 100x GPU vs CPU Myth**
**Reviewer Name: Matthew Lee**

## Paper Summary

**Overview:** This paper compares the computing power between an Intel CPU (i7 960) and Nvidia GPU (GTX 280) over a variety of applications. Specifically, these authors benchmark these processors over applications which contain a large amount of data level parallelism and non - blocking computations (computations which can be performed independently and in any order). The authors define this set of applications as ”throughput computing applications”, which are increasingly relevant as data stream size exceeds current computational limits. The main focus of this paper is to re-investigate prior papers which claim that GPUs outperform CPUs on the order of 100-1000x for many throughout computing applications. The paper also classifies common throughput computing applications by their attributes (bandwidth requirements, access patterns, etc.) and outlines the application and architecture specific optimizations required for performant computation. Furthermore, the paper outlines ”why” certain applications benefit from CPUs and GPUs from the hardware architecture perspective.

**Results Overview:** On various applications such as Monte Carlo simulations, FFTs, SpMVs and more, the authors benchmarked the performance of the CPU/GPU with relevant metrics. Contrary to prior research, the authors found that GPUs perform only 2.5X on average better than CPUs on throughput applications. Applications which show significantly increased performance on the GPU were GJK (10x i7 performance, due to it’s use of a GPU specific texture sampler feature), while radix sort and a constraint solve both performed better on CPUs.
The authors identified numerous architectural differences which explain the performance differences for different applications. The biggest feature categories were external memory bandwidth and available computational flops. Bandwidth bound applications included the SAXPY (Scalar Alpha X PLus Y)and LBM (Lattice Boltzmann method), where the GPU exhibited roughly 5x the performance of the CPU, which is in line with the ratio for peak memory bandwidth between the GPU and CPU (4.7X). Compute bound applications (applications which exhibit dependence on single thread performance and TLP) such as SGEMM, Conv, and FFT yielded performance ratios in the range from 2.8-4x, in line with the single precision flop ratio of 3-6x from GTX 280 to i7 960. Other architectural features which contributed to performance differences on other applications were cache size, gather/scatter support, efficient synchronization and fixed functional units.
Finally, the authors noted that the results of this paper differed from prior papers due to the optimizations made for both GPU and CPU code, as well as the comparison of a GPU/CPU in the same tier. To optimize their CPU code, they utilized multithreading, cache blocking, and the reogranization of memory accesses. For the GPU, the authors’ key software optimizations were the minimization of global synchronization and utilization of local shared buffers.

## Paper Strengths

1. This paper disproves the previous idea that GPUs exhibited performance improvements from 100-1000x for various computing applications, by providing a more realistic view at performance benchmarks between a high end gpu and cpu.

2. This paper provides a great overview of relevant throughput computing applications, as well as detailed explanation for the performance bottlenecks of each application.

3. Rather than just comparing CPUs and GPUs based on performance, this paper explains the architectural differences which contribute to those performance differences.

## Paper Weaknesses

1. The authors fail to mention other throughput computing applications which were not benchmarked. The inclusion or exclusion of certain applications could shift their amazing result of an avg 2.5x performance difference between GPU and CPU to something less impressive.

2. Although one of this paper’s strengths is the wide range of applications which are benchmarked, the paper suffers from a lack of depth for benchmarking and metric explanations. The paper doesn’t really discover any ”new” optimizations for CPUs and GPUs, but rather correctly applies preexisting optimizations for the sake of transparent benchmarking.

3. As a follow up to point 1 and 2, it is hard to be completely objective when analyzing performance benchmarks. Prior researchers had results which falsely painted GPUs as vastly superior to CPUs for certain applications. The authors here fail to mention any possible bias in their results.

## Detailed Comments

**Pros Justification:** I enjoyed the systematic methodology the authors employed for describing the universe of throughput computing applications and categorizing each application by it's characteristics. The authors‘ results were great, simply because they resulted from comparisons of similiarly powered CPUs and GPUs, and equal optimizations for both processors. The paper was very detailed in describing the application specific requirements, and provided intuitive and justified architecutural explanations for the performance gaps.

**Cons Justification:** Although I thought this paper was mostly good, I did notice the authors seemed to completely ignore that their benchmarks and results could be biased, even as they admonished the bias of prior research. I'm skeptical of the idea that these categories of applications capture ALL relevant throughput computing applications, yet the authors don't mention any possible gaps in their universe or caveats in their process.

## Other Comments

I found it interesting how the company of the authors who created this research (Intel) stands to directly benefit from the results of this paper.