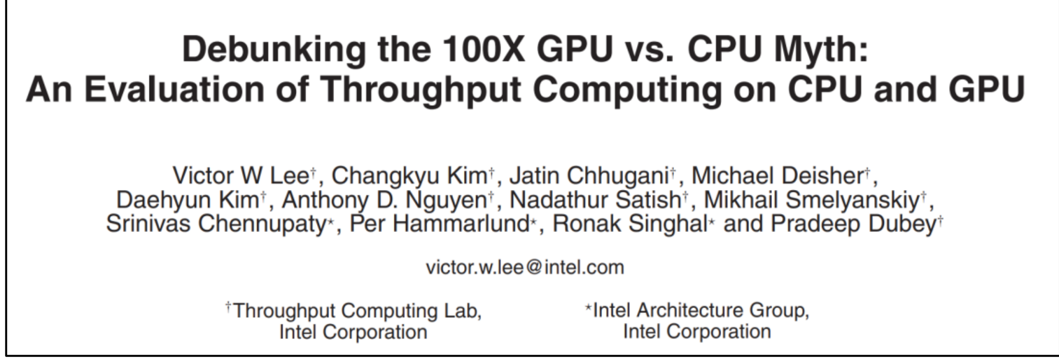
Debunking the 100X GPU vs. CPU Myth: An Evaluation of Throughput Computing on CPU and GPU

Benjie Miao Feb. 25^{th}, 2021

Paper Information



Debunking the 100X GPU vs. CPU Myth: An Evaluation of Throughput Computing on CPU and GPU

Victor W Lee[†], Changkyu Kim[†], Jatin Chhugani[†], Michael Deisher[†], Daehyun Kim[†], Anthony D. Nguyen[†], Nadathur Satish[†], Mikhail Smelyanskiy[†], Srinivas Chennupaty[*], Per Hammarlund[*], Ronak Singhal[*] and Pradeep Dubey[†]

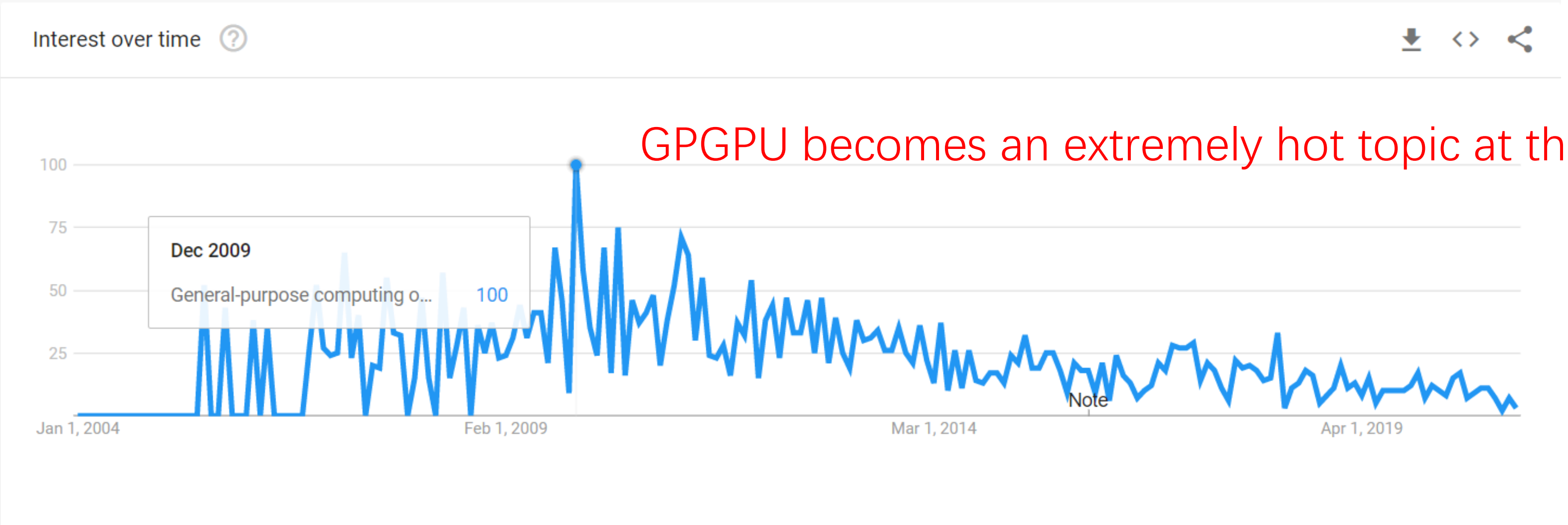
victor.w.lee@intel.com

[†]Throughput Computing Lab, Intel Corporation

[*]Intel Architecture Group, Intel Corporation

- Conference: ISCA'10 (Top-tier conference in Computer Architecture)
- Author: Victor W Lee, et. al. *Intel Corporation* (12 authors in all)
- Main idea: to argue that CPU is **not that bad** in scientific computing compared to GPGPU
  - *the performance gap between an Nvidia GTX280 processor and the Intel Core i7 960 processor narrows to* ***only 2.5x*** *on average*

Google Trends for GPGPU (General Propose Graphic Processing Unit)

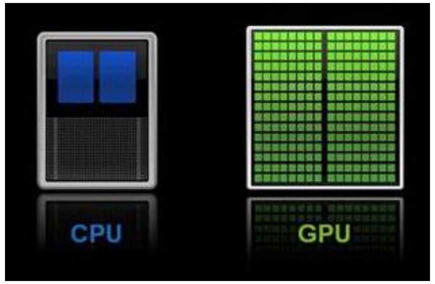


Paper Outline

- Introduction
- Methodology
  - The workload: Throughput Computing Kernel
  - Performance Benchmark Platform
- Result & Analysis
  - Performance comparison (GPU v.s. CPU)
  - Performance gap analysis
- Conclusion

Introduction

https://www.adslzone.net/2018/01/12/cpu-vs-gpu-diferencias/



- CPU v.s. GPU architecture: very different philosophy
  - CPU: fast response-time to **a single task**
  - GPU: a large degree of data parallelism, latency tolerant
- GPU is (claimed to be) suitable for **throughput computing**
  - throughput computing: complete **a large task** in a short time period
    - All scientific computing programs fall in this category
  - a number of papers claim that ***GPUs*** *perform 10X to 1000X better than CPUs on a number of throughput kernels/applications*

Introduction

- reexamine claims that GPUs perform much better than CPUs; after tuning the code for **BOTH** CPU and GPU, found that the GPU only **performs 2.5X better** than CPU
- analyze the difference between CPU and GPU and **identify the key architecture features** that benefit throughput computing workloads
- provide a **systematic characterization** of throughput computing kernels regarding 1) the types of parallelism available 2) the compute and bandwidth requirements 3) the access pattern and 4) the synchronization needs
- identify the important **software optimization techniques** for efficient utilization of CPU and GPU platforms

Workload: Throughput Computing Kernel

Kernel	Application	SIMD	TLP	Characteristics
SGEMM (**SGEMM**) [48]	Linear algebra	Regular	Across 2D Tiles	Compute bound after tiling
Monte Carlo (**MC**) [34, 9]	Computational Finance	Regular	Across paths	Compute bound
Convolution (**Conv**) [16, 19]	Image Analysis	Regular	Across pixels	Compute bound; BW bound for small filters
FFT (**FFT**) [17, 21]	Signal Processing	Regular	Across smaller FFTs	Compute/BW bound depending on size
SAXPY (**SAXPY**) [46]	Dot Product	Regular	Across vector	BW bound for large vectors
LBM (**LBM**) [32, 45]	Time Migration	Regular	Across cells	BW bound
Constraint Solver (**Solv**) [14]	Rigid body physics	Gather/Scatter	Across constraints	Synchronization bound
SpMV (**SpMV**) [50, 8, 47]	Sparse Solver	Gather	Across non-zero	BW bound for typical large matrices
GJK (**GJK**) [38]	Collision Detection	Gather/Scatter	Across objects	Compute Bound
Sort (**Sort**) [15, 39, 40]	Database	Gather/Scatter	Across elements	Compute bound
Ray Casting (**RC**) [43]	Volume Rendering	Gather	Across rays	4-8MB first level working set, over 500MB last level working set
Search (**Search**) [27]	Database	Gather/Scatter	Across queries	Compute bound for small tree, BW bound at bottom of tree for large tree
Histogram (**Hist**) [53]	Image Analysis	Requires conflict detection	Across pixels	Reduction/synchronization bound
Bilateral (**Bilat**) [52]	Image Analysis	Regular	Across pixels	Compute Bound

Table 1: Throughput computing kernels characteristics. The referred papers contains the best previous reported performance numbers on CPU/GPU platforms. Our optimized performance numbers are at least on par or better than those numbers.

Platform

- CPU (Intel Core i7-960) v.s. GPU (Nvidia GTX 280)

	Num. PE	Frequency (GHz)	Num. Transistors	BW (GB/sec)	SP SIMD width	DP SIMD width	Peak SP Scalar FLOPS (GFLOPS)	Peak SP SIMD Flops (GFLOPS)	Peak DP SIMD Flops (GFLOPS)
Core i7-960	4	3.2	0.7B	32	4	2	25.6	102.4	51.2
GTX280	30	1.3	1.4B	141	8	1	116.6	311.1/933.1	77.8

Table 2: Core i7 and GTX280 specifications. BW: local DRAM bandwidth, SP: Single-Precision Floating Point, DP: Double-Precision Floating Point.

- SIMD (Single Instruction Multiple Data)
  - CPU: Intrinsic instructions, Out-of-order, etc.
  - GPU: Warp (32 threads at the same time)

Result (after tuning both CPU/GPU codes)

- GPU 2.5x faster than CPU on average
- GJK, Bilat, SAXPY:
  - >5x (suitable for GPU)
- Solv, Sort:
  - CPU version faster

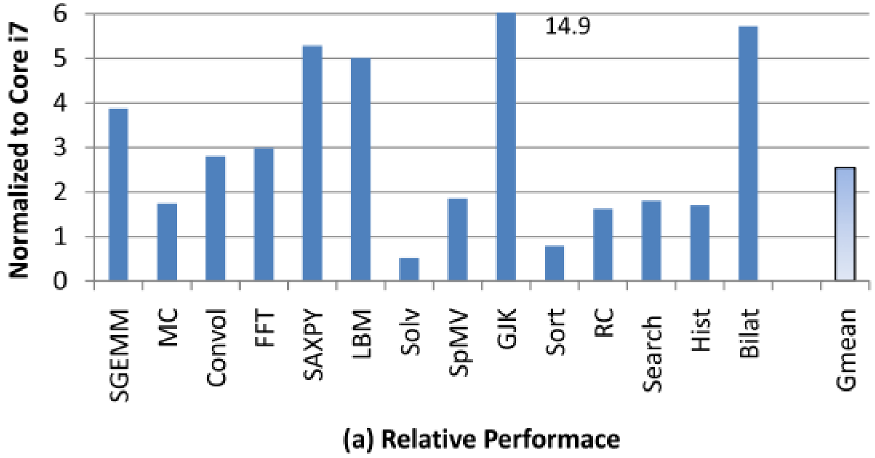


Figure 1: Comparison between Core i7 and GTX280 Performance.

Analysis

- Categorize the kernels by their **computing characteristics**
  - **Bandwidth-bound**: SAXPY, SpMV, LBM
  - **Compute-bound**: SGEMM, Conv, FFT, Bilat
  - **Cache-bound**: Sort, Search
  - **Gather/Scatter**: GJK, RC
  - **Reduction and Synchronization**: Hist, Solv
  - **Fixed Function:** Bilat, MC

Bandwidth-bound: SAXPY, SpMV, LBM

- SAXPY (Scalar Alpha X Plus Y), SpMV (Sparse Matrix * Vectors), LBM (Lattice Boltzmann method in CFD)
- SAXPY & LBM:
  - sets that require **much global memory accesses** without **much compute**
  - are *purely bandwidth bound*
- Platform peak memory bandwidth ratio: 4.7X
- Speedup: SAXPY - 5.1X, LBM - 5.0X
- SpMV: 1.9X
  - Reason: in CPU, column index fit in cache

Software Optimization Techniques

- For CPU:
  - multithreading
  - cache blocking
  - reorganization of memory accesses for SIMD-ification
- For GPU:
  - minimizing global synchronization
  - using local shared buffers

Conclusion

- CPUs and GPUs are **much closer in performance (2.5X)** than the previously reported orders of magnitude difference
- many factors affect the reported performance
- **Characterization of kernels**: compute/bandwidth, cache, gather/scatter, synchronization, fixed functional units
- **Guideline for performance optimization** on CPU and GPU programs

- Future: Power efficiency

Questions?