Harnessing Data Parallel Hardware for Server Workloads
Alvin R. Lebeck, Duke University
Emmett Witchel, University of Texas
NSF XPS Collaborative (CCF-1335443)

Growth in Cloud Services
- Everything is in the Cloud (It’s raining data!)
  - Static content: files, images, audio and video streaming
  - Dynamic content using PHP, Perl, Python or AJAX
  - Search, Analytics
  - Compute (SaaS)
  - Non-traditional workloads
- Example: Facebook
  - 900 million users
  - ~1 trillion page views a month or ~350,000 per second
  - 1.2 million photos per second
  - >100,000 servers
- Challenge: How to best satisfy this demand?
  - Add more machines – Higher costs
  - Improve existing machines – Harness accelerator efficiency to achieve higher throughput/Watt

Application: SpecWeb Banking
- **Rhythm**, a software architecture for high throughput SIMT-based servers.
- Asynchronous, Event driven, Lock free and wait free, pipelined.
- Supports deep and wide pipelines for max throughput

Evaluation: SpecWeb Banking
- Titan A – Backend emulated as host thread
- Titan B – Backend emulated as device function call
- Titan C – Omits/offsets response transpose
- No network effects (but results similar w/ network)
- 99% latency: Titan B ~ 34 ms, Titan C ~ 14 ms

Application: K Nearest Neighbors
- **Given**: Sparsely populated high dimension feature space and sustained query arrival rate
- **Find** the k nearest neighbors for each of the query points (k << N) within a deadline T
- **Want to achieve** lowest Total Cost of Ownership (TCO)
- **Experiment**: Wikipedia, TF-IDF model for index
  - Return top 32 items across all of corpus
  - Sparse Matrix Matrix Multiplication
  - Custom algorithm: MKL, CUSPARSE too slow

Evaluation: TCO Analysis
- Accelerator based configurations lower TCO
- 25K Q/s, T = 50ms
- Similar results for 5K Q/s, T = 20ms

Evaluation: NIC-GPU
- GPU-based Networking Applications need
  - Pipelining & buffer management
  - Request Batching (Cohort formation)
  - NIC-GPU interacation
- **Problem**: CPU is the only boss?
- **GPU-IO (File System and Network APIs for GPU)**

GPU-IO (GPUUnet)
- **Simplicity**
- Reliable in-order streaming
- Non-RDMA Transports
- RDMA Transports
- **Performance**

GPUUnet Evaluation: Face Verification
- **GPU**: 4.5x throughput
- Similar latency
- **2xGPUUnet+ 10xCPU**

Summary and Future Directions
- **Progress to Date**
  - Demonstrated potential benefits of GPUs for non-traditional workloads
  - Rhythm
  - K nearest neighbors
  - GPUUnet
- **Future Directions**
  - Streaming Analytics
  - Multitenancy on GPUs
  - Integrating Rhythm w/ GPUUnet

Acknowledgements & Publications
- **Funding from NSF, NVIDIA and Duke University**
- Thanks to Sandeep Agrawal, Sangman Kim and Mark Silberstein (now at Technion), Valentin Pistol, Jun Pang, John Tran, David Tarjan