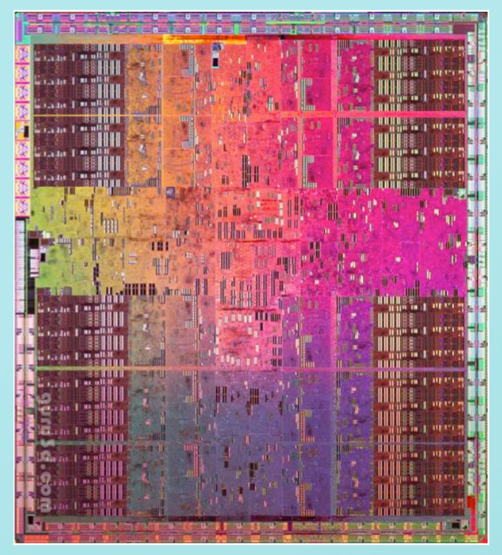
Application-level Trade-offs for WFST-based Large Vocabulary Continuous Speech Recognition on a Graphics Processing Unit

Jike Chong, Ekaterina Gonina, Youngmin Yi, Kurt Keutzer, Department of Electrical Engineering and Computer Science, University of California, Berkeley

Emerging Manycore Platforms

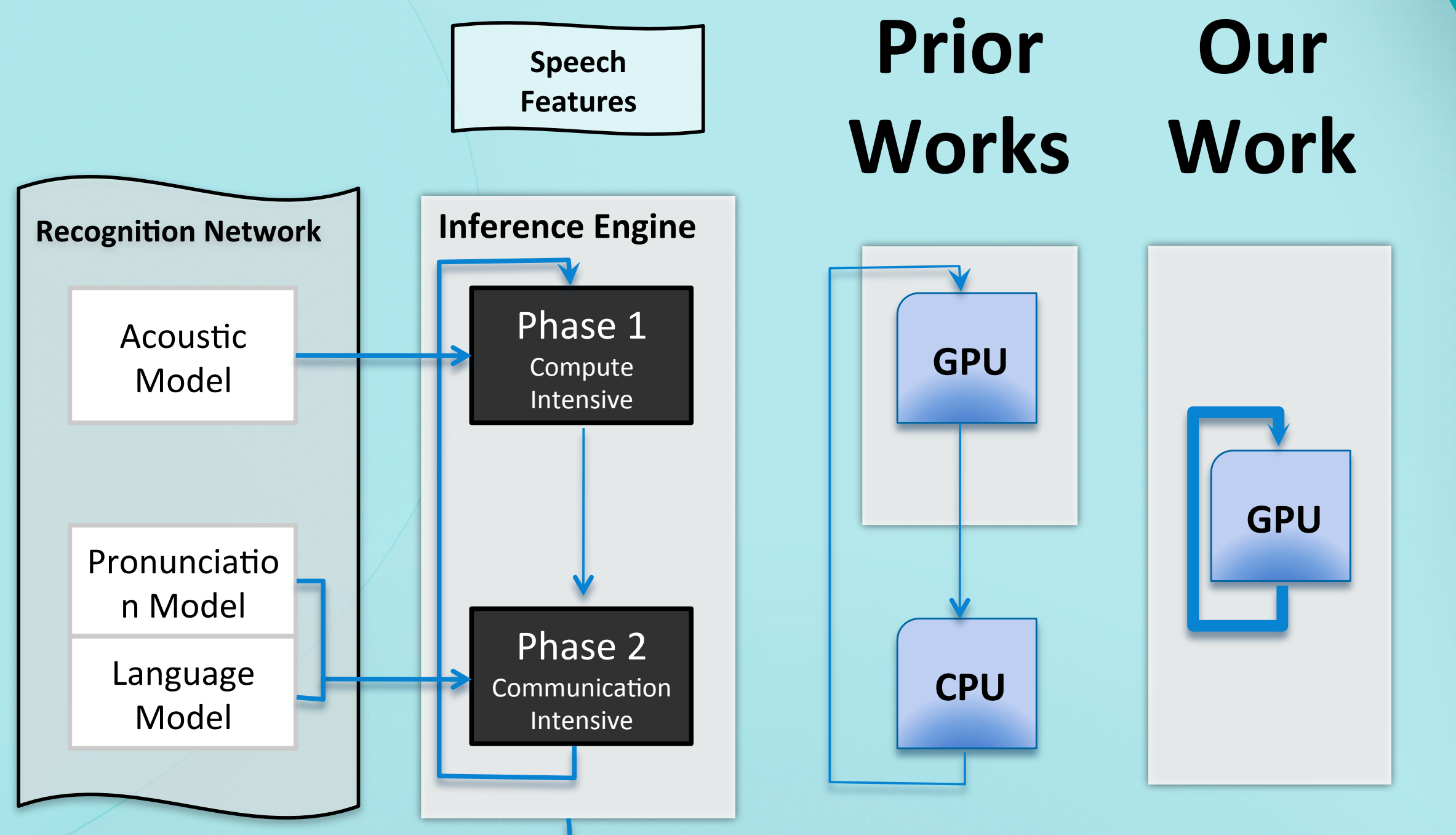


NVIDIA GTX280
30 cores

- Architecture trend:
  - Increasing vector unit width
  - Increasing numbers of cores per die
- Application implications:
  - Must optimize synchronization cost
  - Must increase SIMD efficiency

Ongoing work investigates algorithm design space that optimizes for data parallel manycore programming.

System Architecture

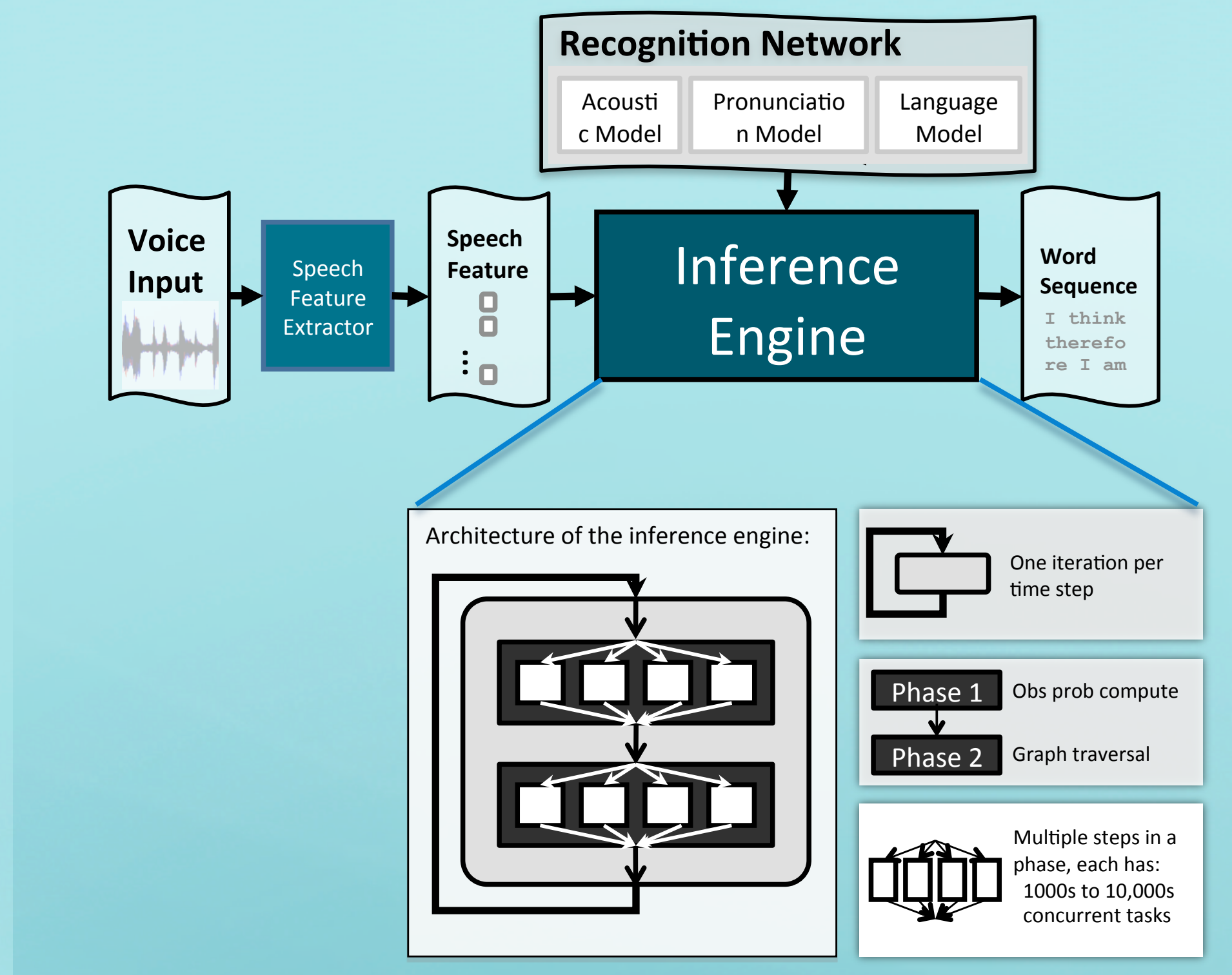


Prior Works: Data is passed between CPU and GPU in the inner loop, high transfer overhead

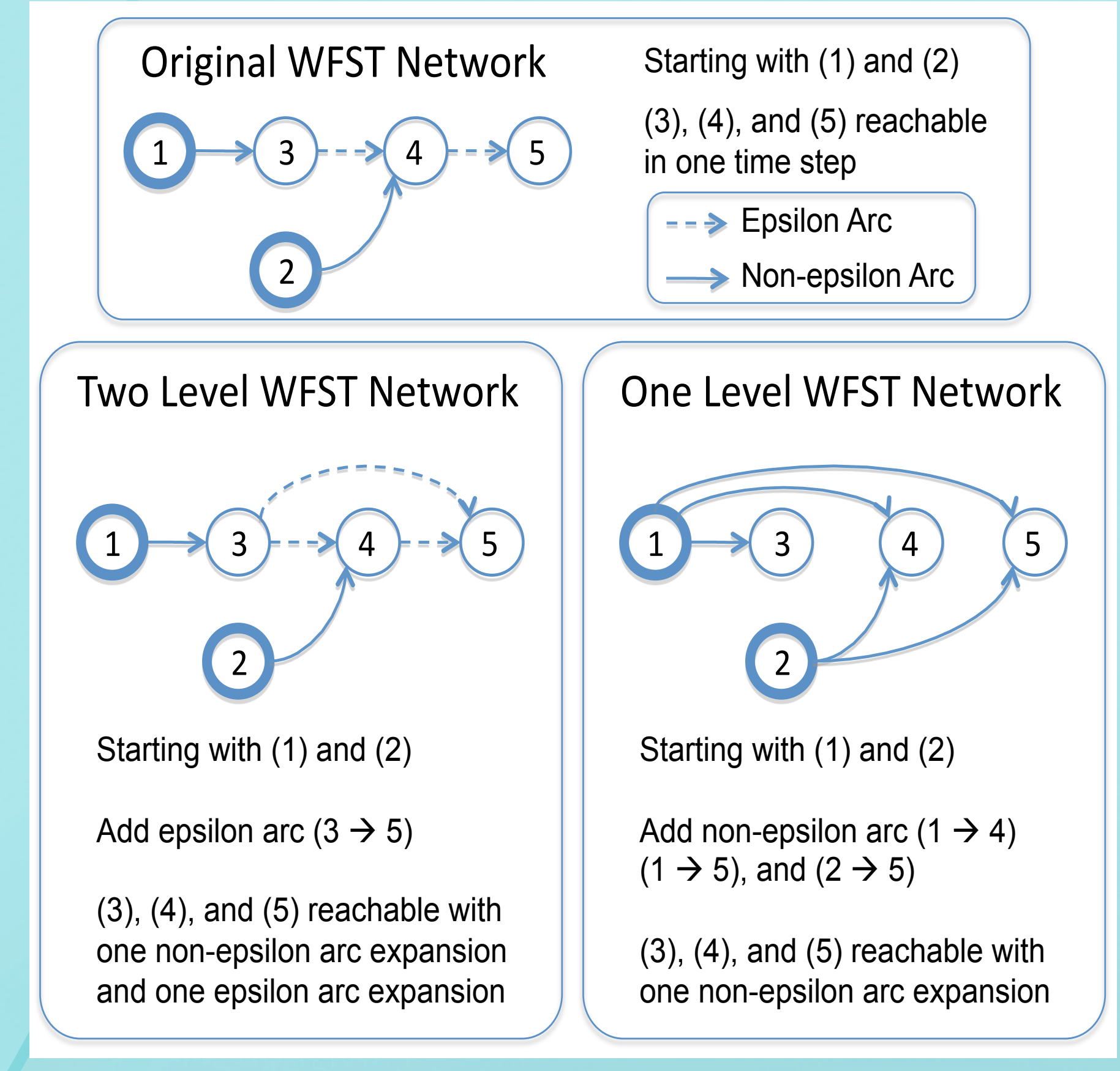
Our Work: Data stays on the GPU in the inner loop, eliminate transfer overhead, enable more optimizations

Speech Recognition Inference Engine Characteristics

- Parallel graph traversal through irregular network
  - Guided by a sequence of input audio vectors
  - Computing on continuously changing data working set
- Implementation challenges
  - Define a scalable software architecture to expose fine-grained application concurrency
  - Efficiently synchronize between an increasing number of concurrent tasks
  - Effectively utilize the SIMD-level parallelism



Algorithm Design Space Exploration



Original WFST Network: Starting with (1) and (2). (3), (4), and (5) reachable in one time step

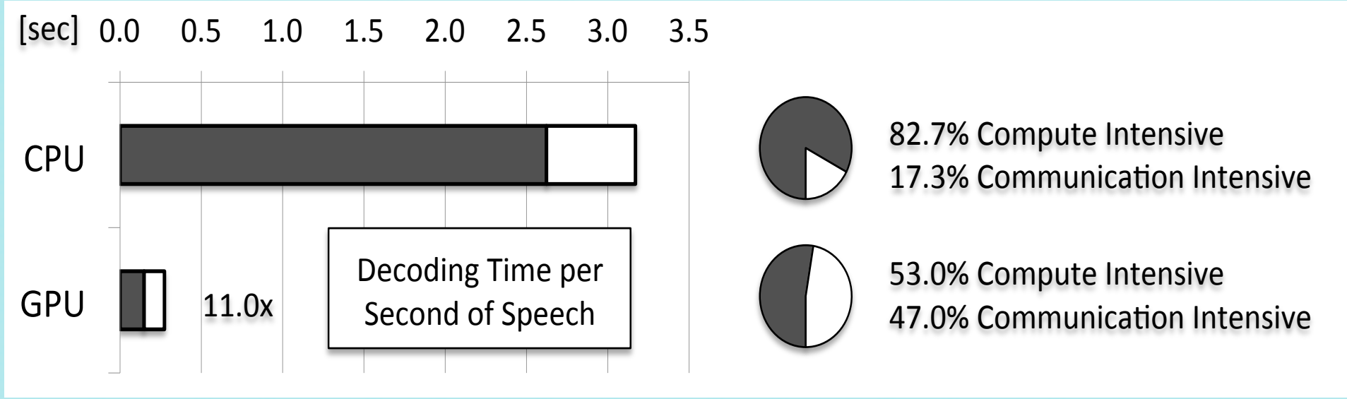
Two Level WFST Network: Starting with (1) and (2). Add epsilon arc (3 → 5). (3), (4), and (5) reachable with one non-epsilon arc expansion and one epsilon arc expansion

One Level WFST Network: Starting with (1) and (2). Add non-epsilon arc (1 → 4) (1 → 5), and (2 → 5). (3), (4), and (5) reachable with one non-epsilon arc expansion

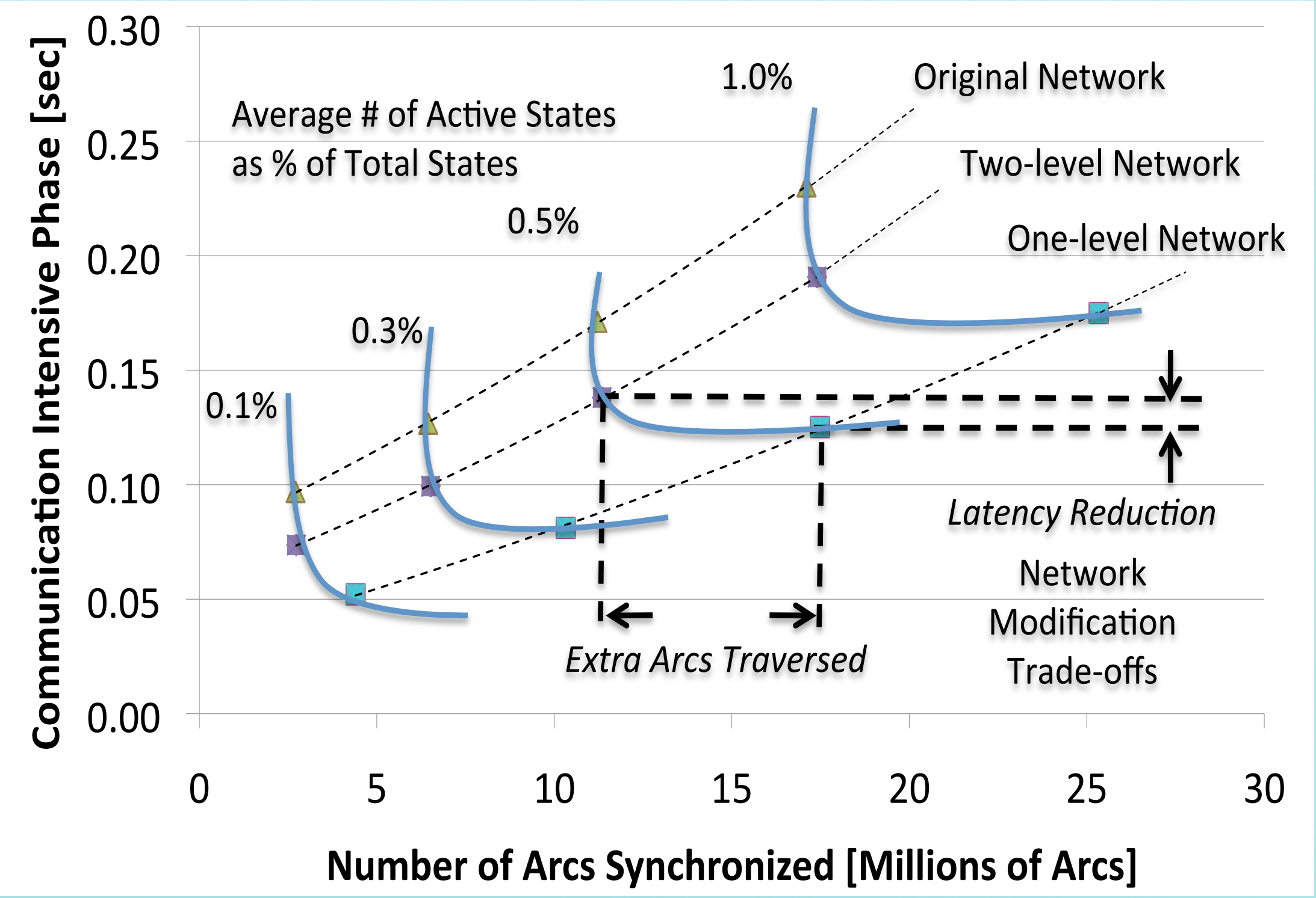
- Explore efficient graph traversal technique
  - Vary the amount of flattening of the WFST network to two levels or one level
  - The flattening increases the number of arcs to traverse in the algorithm

Evaluation of the Inference Engine

- Speed up varies btw phases:
  - 4-18x for compute intensive phases
  - 3-4x for communication intensive phases
  - Communication intensive phases becoming proportionally more important



- Speedup for phase 2:
  - Moving to 2-level WFST network provides 17-24% speed up with minimal increase in arcs traversed
  - Moving to one level WFST network provides an additional 8-29% speedup at the expense of traversing 48-62% more arcs
- Less than 8% sequential overhead



Conclusions

- Defined and implemented a parallel software architecture:
  - 5-8% sequential overhead
  - Significant potential for further speedup in future platforms
- Implemented the entire inference engine on the GPU
  - Both GMM computation and graph traversal phase are implemented in data-parallel routines
- Explored the algorithmic design space for WFST network optimization for data parallel operations
  - Network flattening critical for efficient data parallel operation

We expect that an efficient speech recognition engine will be a key component in many exciting new applications to come!

Thanks to Yen-Kuang Chen and Pradeep Dubey at Intel for initiating and supporting this research.
Thanks to Nelson Morgan, Andreas Stolcke, and Adam Janin at ICSI for insightful discussions and continued support in the infrastructure used in this research.
This research is supported in part by Intel Ph.D. Fellowship.
This research also supported in part by Microsoft (Award #024263) and Intel (Award #024894) funding and by matching funding by U.C. Discovery (Award #DIG07-10227).