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, eliminates transfer overhead, enables more optimizations

**Recognition Network**
- Acoustic Model
- Pronunciation Model
- Language Model

**Inference Engine**
- Phase 2 (Communication intensive)
- Phase 2 (Computation intensive)

**Speech Transducer**

Algorithm Design Space Exploration

- 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

Emerging Manycore Platforms

- **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.

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 Evaluation

- **Speed up varies between 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% speedup 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 phrases 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!