Scalable HMM based Inference Engine in Large Vocabulary Continuous Speech Recognition

Jike Chong, Kisun You, Youngmin Yi, Ekaterina Gonina, Christopher Hughes, Wonyong Sung, Kurt Keutzer

Accelerating Speech Recognition on Manycore Platforms

Target Parallel Platforms
- Platforms targeted – Multicore and Manycore with SIMD units
- Architecture trend:
  - Increasing vector unit width
  - Increasing numbers of cores per die
- Application implications:
  - Must optimize synchronization cost
  - Must increase SIMD efficiency

Speech Recognition Inference Engine
- 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
- Efficient graph traversal techniques will:
  - Reduce the task management overhead
  - Enable gaining additional speedup in scaling to more cores
- Efficient transition evaluation granularity is also important:
  - SIMD efficiency is indicative how well the algorithm would respond to HW increases in SIMD width

Patterns in the Graph Traversal Phase
- Efficient graph traversal techniques:
  - Reduce the task management overhead
  - Enable gaining additional speedup in scaling to more cores
  - Efficient transition evaluation granularity is also important:
    - SIMD efficiency is indicative how well the algorithm would respond to HW increases in SIMD width

Graph Traversal by Propagation
- A task parallel application on manycore platform with atomic operation support

Graph Traversal by Aggregation
- A task parallel application on manycore platform without atomic operation support

Conclusions
- We have defined and implemented a parallel software architecture:
  - Less than 2.5% sequential overhead
  - Significant potential for further speedup in future platforms
- We have explored the algorithmic design space on two HW platforms:
  - The fastest algorithm style differed between platforms
  - Propagate techniques were able to use efficient HW on both platforms
  - SIMD optimization will become more important going beyond four lanes