Rapid Training of Acoustic Models using GPUs

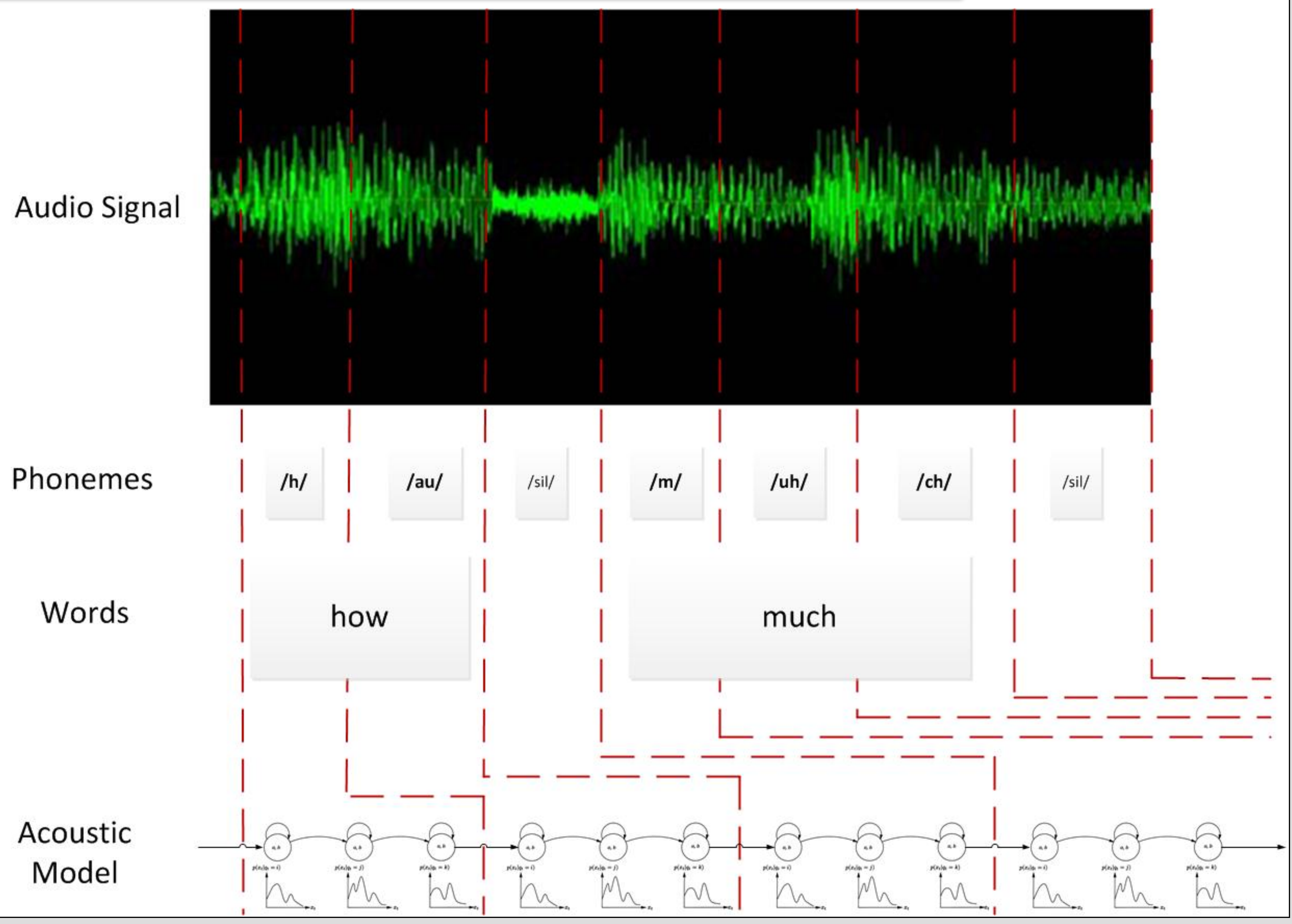
Senaka Buthpitiya, Jike Chong, Ian Lane

Carnegie Mellon University

Training of Acoustic Models for Speech Recognition

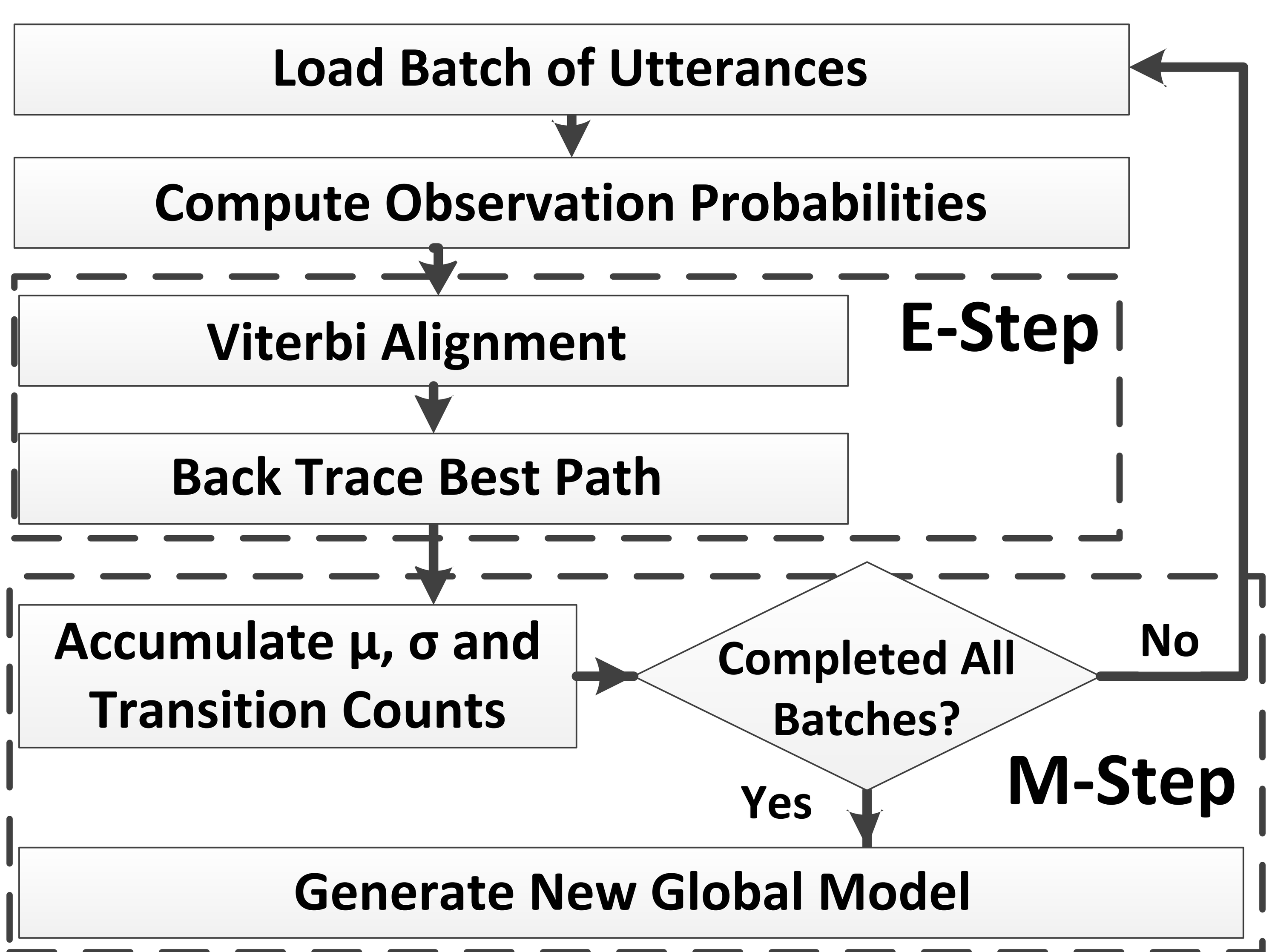
- State-of-the-art speech recognition systems are trained on thousands of hours of speech data
- Training requires:
  - Calculating observation probabilities
  - Aligning audio with transcripts
  - Estimating model parameters
  - Repeat process multiple times
- Training can take many weeks even on large clusters
- Evaluating new approaches challenging

Can acoustic model training be parallelize on GPUs using GPUs to parallelize the computation?



Parallelizing Acoustic Model Training on the GPU

Viterbi training used to estimate the parameters of an Hidden-Markov-Model (HMM)-based acoustic model



Training flow for one training iteration

Observation Probability Computation
- GMM-level parallelism - 10KB of model data - fits into scratch space on the GPU
- Threads parallelize over the observation samples
- Thread blocks parallelize over the GMMs
- Each thread in a thread block performs all computations for one time step

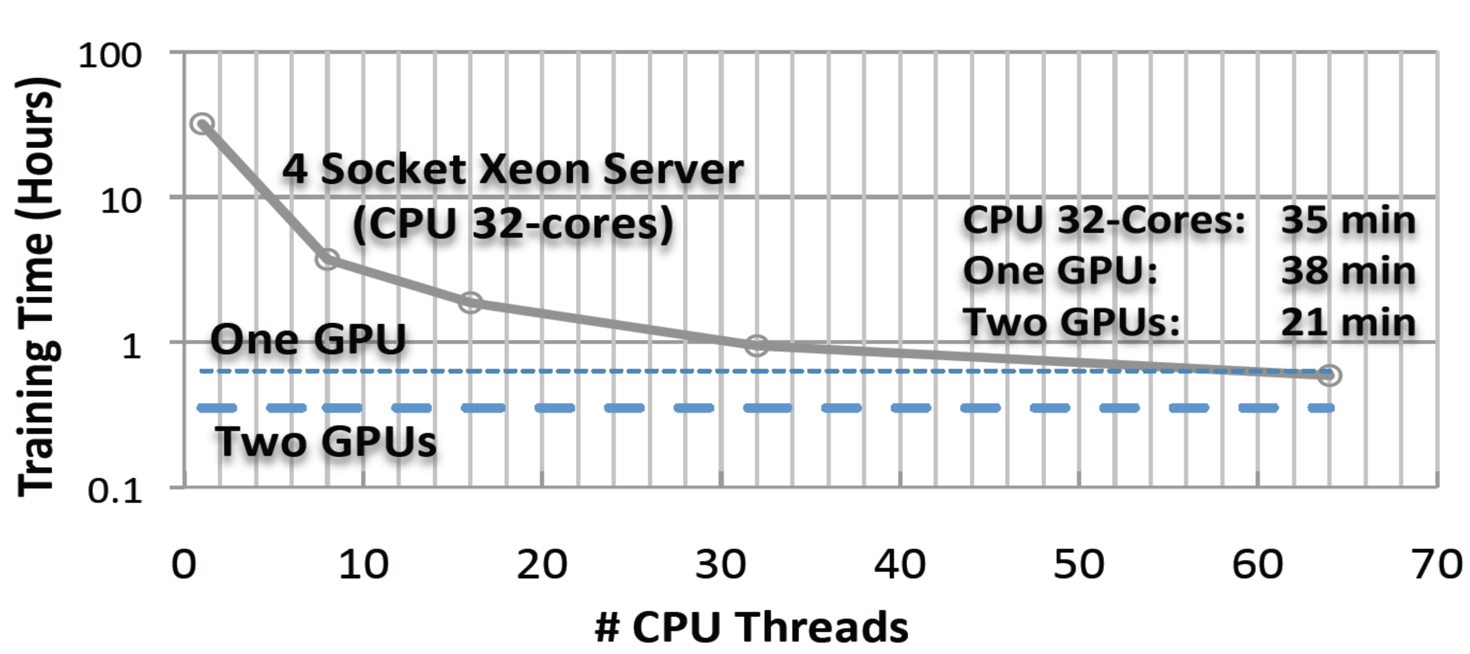
Alpha Computation
- Calculate optimal match between the transcript and the acoustic input
- Calculation is time-synchronous – present output depends of previous outputs
- Parallelize utterances per thread block – For optimal memory access speed

Backtracking Computation
- Trace one-best path best aligning GMM states to acoustic input observations
- Naïve implementation causes severe bottleneck with excess memory reads
- We implement using a prefetch optimization
  - Fully utilize load bandwidth
  - Minimize memory latency caused by the pointer chasing operations

Maximization Step
- Updates aggregated statistics using aligned and labeled input observations
- Extremely large number of values to update – suffers from over/underflows
- Parallelize by mapping each utterance to a thread block
- First aggregate the histogram information within an utterance locally
- Then merge local results from each thread block to the main model

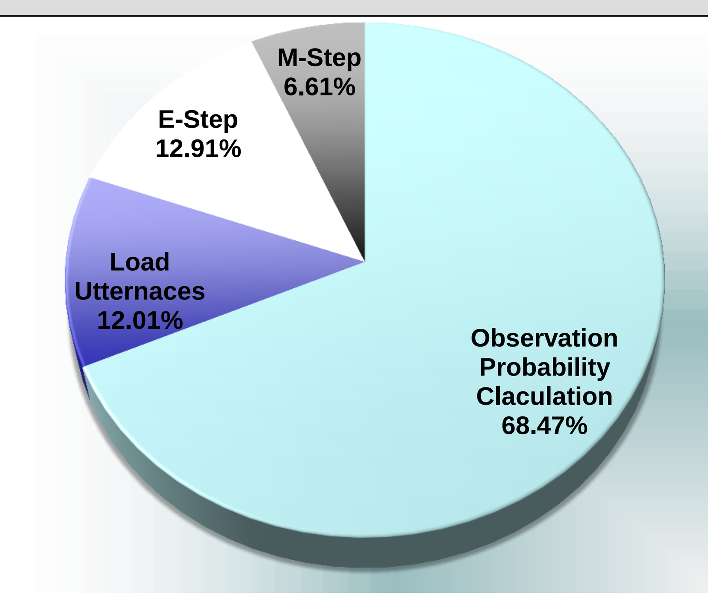
Experimental Evaluation

- GPU implementation on Intel Core i7-2600k CPU machine with two NVIDIA GTX580 GPU cards (approx. $2k)
- Traditional implementation on a 32-core Xeon server (approx. $30k)



Time required for single training iteration with on a 1000hr corpus

Component-wise timing breakdown (GPU)



- A 32-core Xeon server has *only* 7.5% performance advantage over a single GPU system
- With two GTX580 cards training **67%** faster than a 32-core Xeon server

Conclusions:

1. Proposed approach is 51x faster than a sequential CPU implementation
2. Trains an acoustic model with 8000 codebook of 32-component GMs on 1000 hours of data in 9 hours
3. Empowers researchers to rapidly evaluate new ideas to build accurate and robust acoustic models on very large training corpora

This work was partially supported by Nvidia.