OPPORTUNITIES AND CHALLENGES OF PARALLELIZING SPEECH RECOGNITION

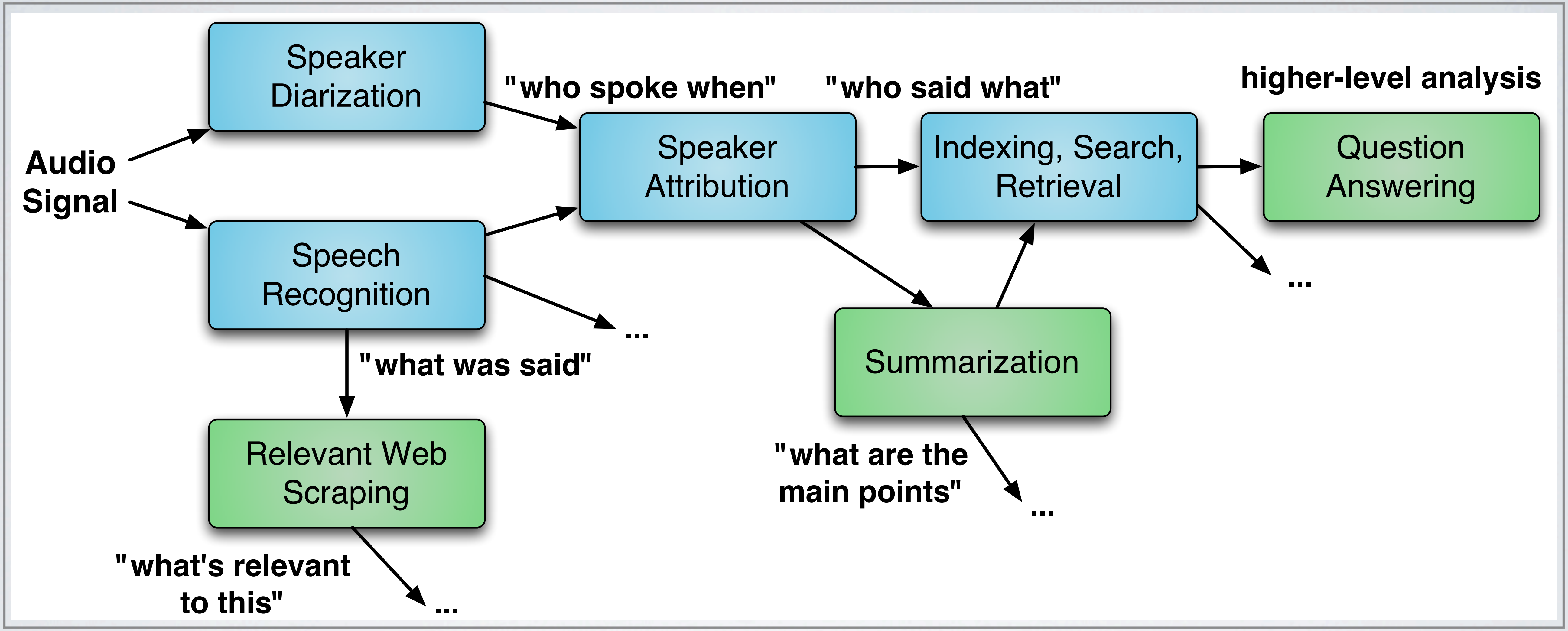
Jike Chong, Gerald Friedland, **Adam Janin**, Nelson Morgan, Chris Oei

OUTLINE

- Motivation
- Improving Accuracy
- Improving Throughput
- Improving Latency

Meeting Diarist Application
"Parlab All"

MEETING DIARIST



MOTIVATION

- Speech technology has a long history of using up all available compute resources.
- Many previous attempts with specialized hardware with mixed results.

1: IMPROVING ACCURACY

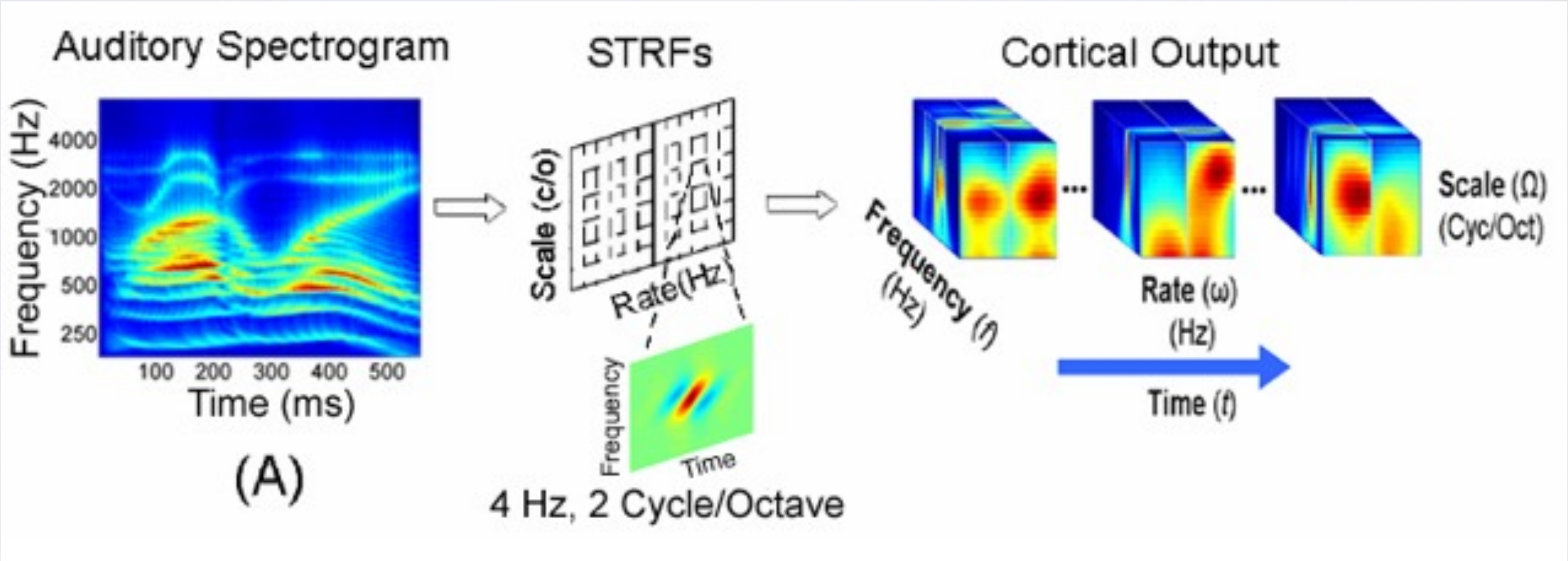
- Speech Technology works well when:
  - Large amounts of training data match application data
  - Small vocabulary; simple grammar
  - Quiet environment
  - Head-worn microphones
  - “Prepared” speech
- Each change adds 10% error!

FEATURES

- Most state-of-the-art features are loosely based on perceptual models of the cochlea with a few dozen features.
- Combining multiple representations almost always improves accuracy, especially in noise.
- Typical systems combine 2-4 representations.

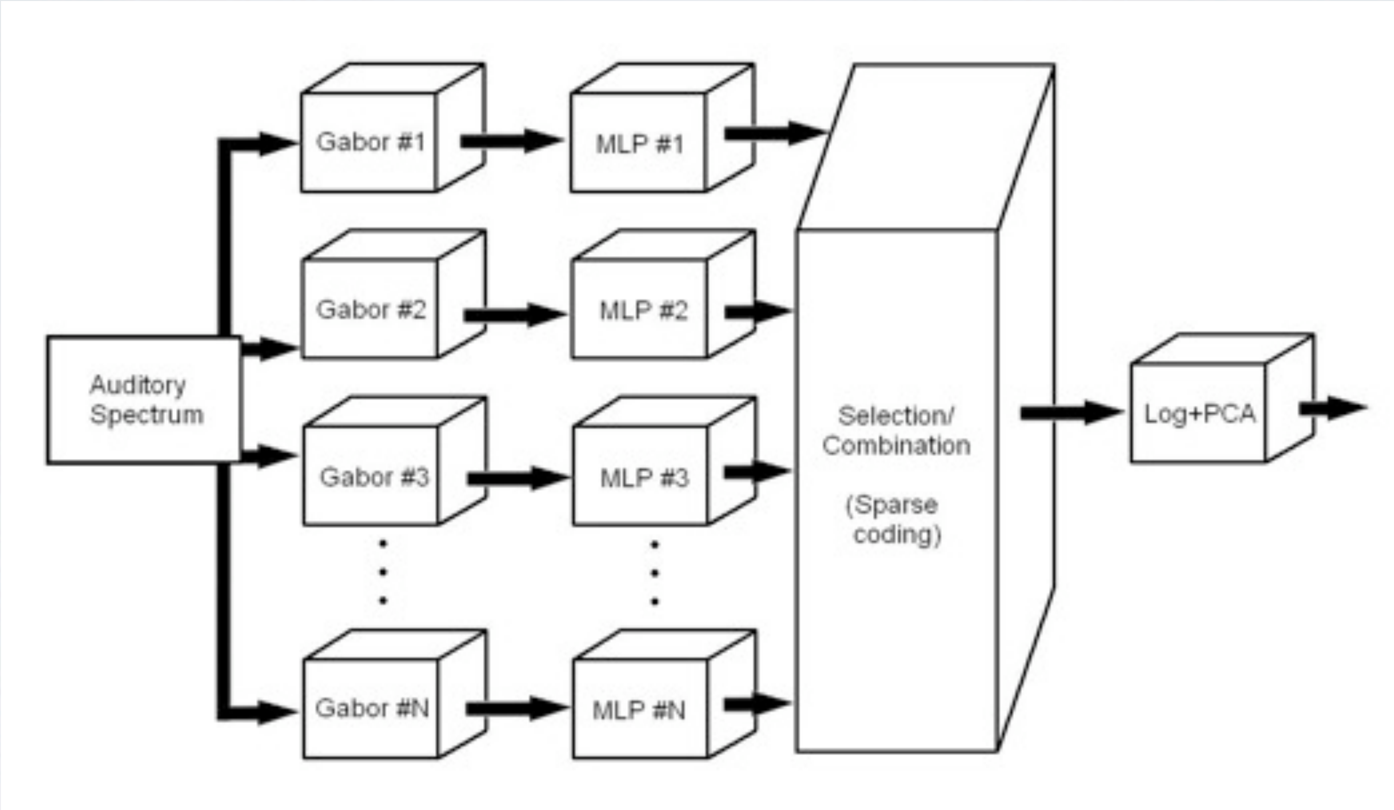
What if we used a LOT more?

MANYSTREAM



- Based on cortical models
- Large number of filters

MANYSTREAM



- Each filter feeds an MLP.
- Current combination method uses entropy-weighted MLP, but many other possibilities.

MANYSTREAM

It helps!

- 47% relative improvement over baseline for noisy "numbers" using 28-stream system.
- 13.3% relative improvement over baseline for Mandarin Broadcast News using preliminary 4-stream system.

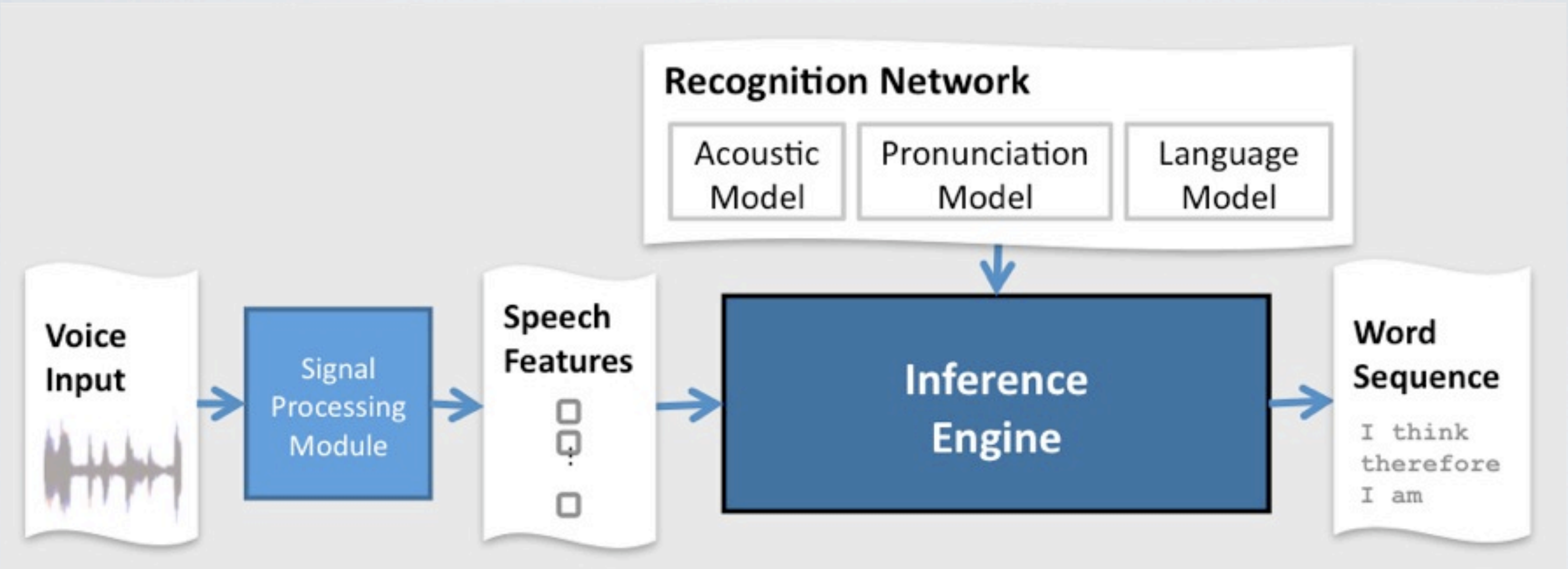
MANYSTREAM

- Next steps:
  - Fully parallel implementation
  - Many more streams
  - Other combination methods

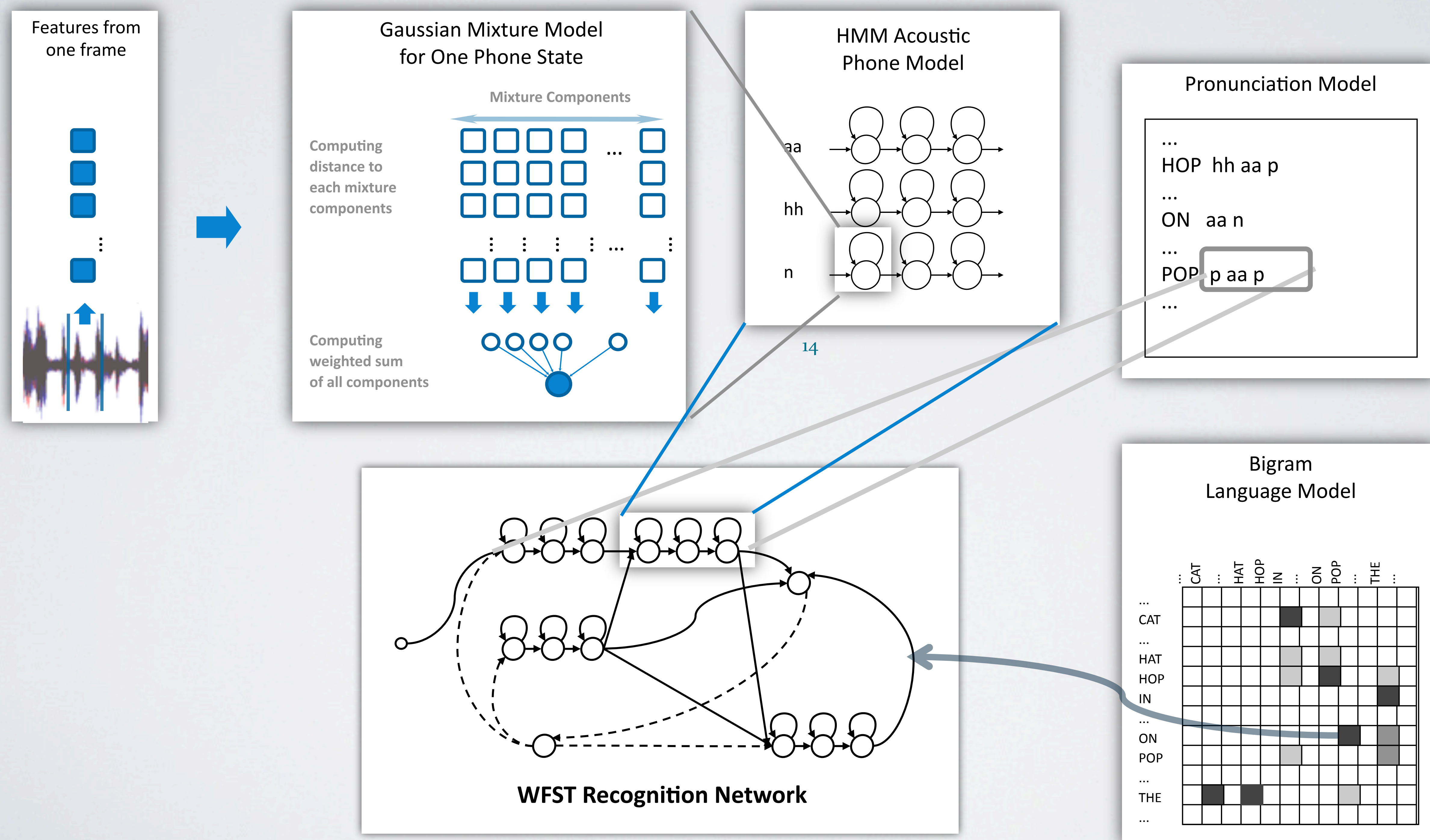
2: IMPROVING THROUGHPUT

- Serial state-of-the-art systems can take 100 hours to process one hour of a meeting.
- Analysis over all available audio is generally more accurate than on-line systems.
- Batch processing per utterance is “embarrassingly” parallel.

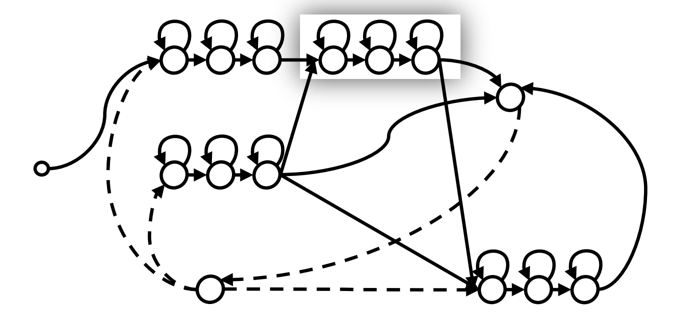
SPEECH RECOGNITION PIPELINE



INFERENCE ENGINE



INFERENCE ENGINE



WFST Recognition Network

- At each time step, compute likelihood for each outgoing arc using the acoustic model.
- For each incoming arc, track all hypotheses.
- Regularlize data structures to allow efficient implementation.
- The entire inference step runs on the GPU.

INFERENCE ENGINE

- 11x speed-up over serial implementation.
  - 18x speed-up for compute intensive phase.
  - 4x speed-up for communication intensive phase.
- Flexible architecture
  - Audio/visual plugin added by domain expert.

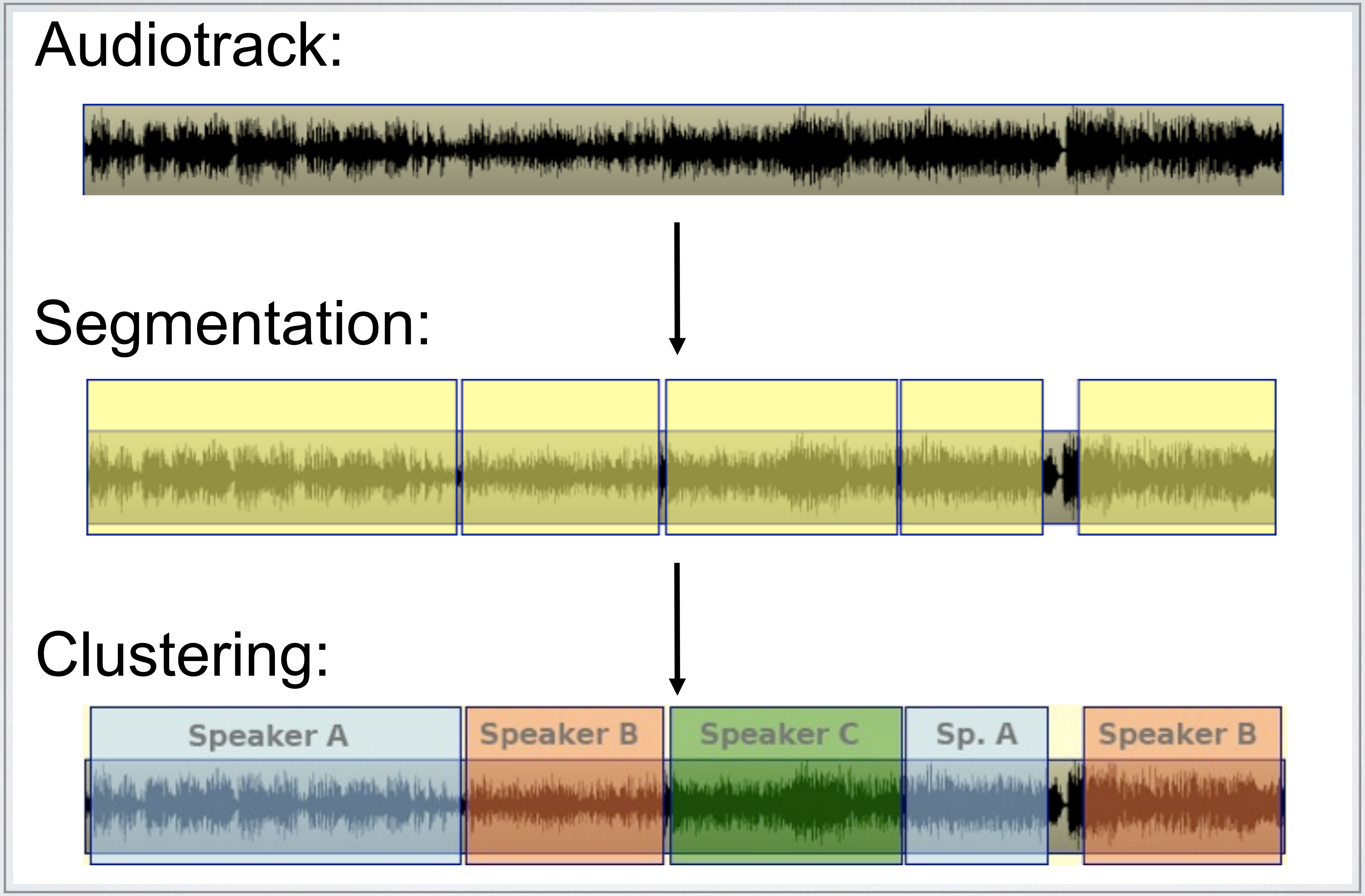
INFERENCE ENGINE

- Next steps:
  - Generate lattices and/or N-best lists.
  - Explore other parallel architectures.
  - Distribute to clusters.
  - Explore accuracy/speed trade-offs.

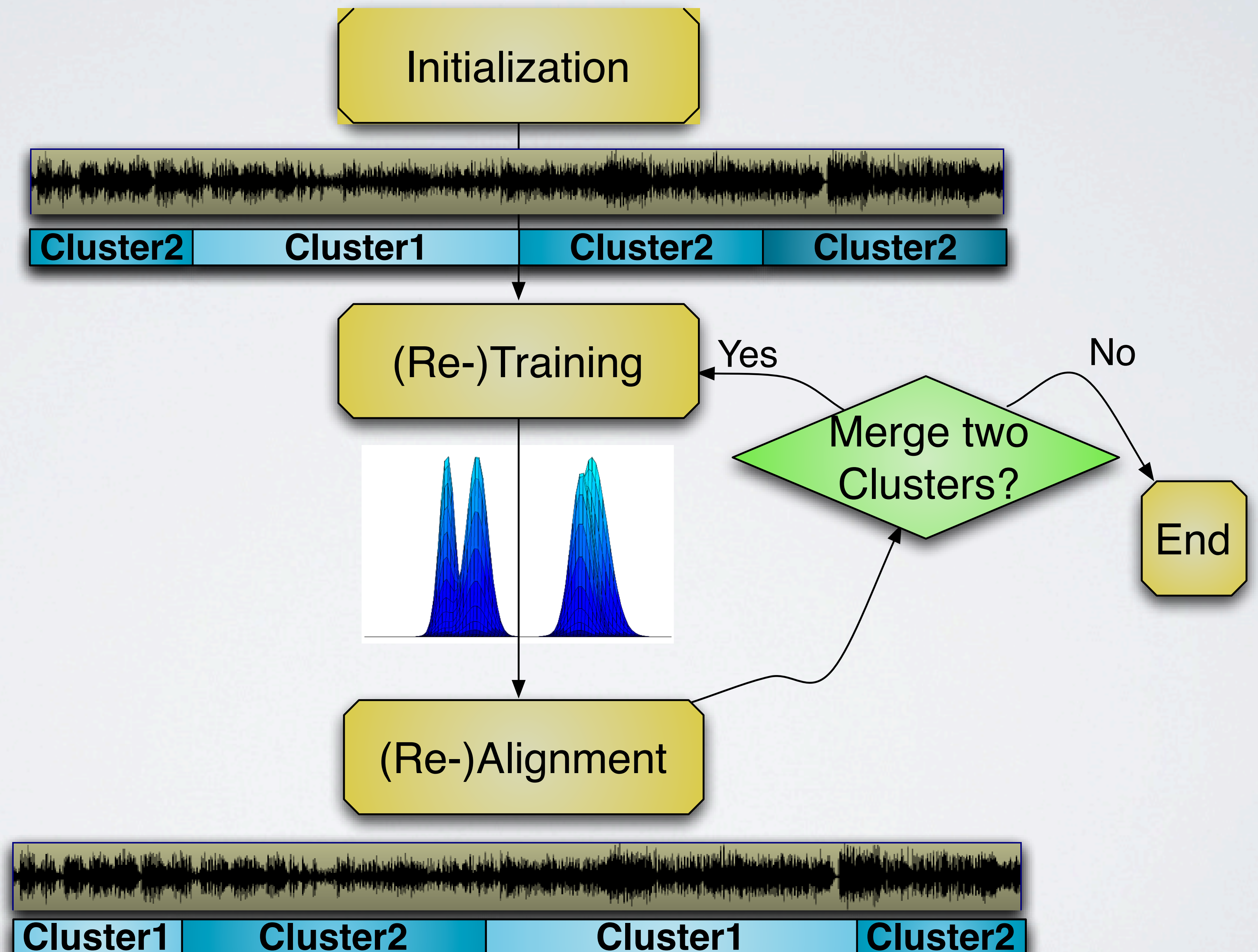
3: IMPROVING LATENCY

- For batch, latency = length of audio + time to process.
- On-line applications require control of latency.
- Parallelization allows lower latency **and** potentially better accuracy.

SPEAKER DIARIZATION



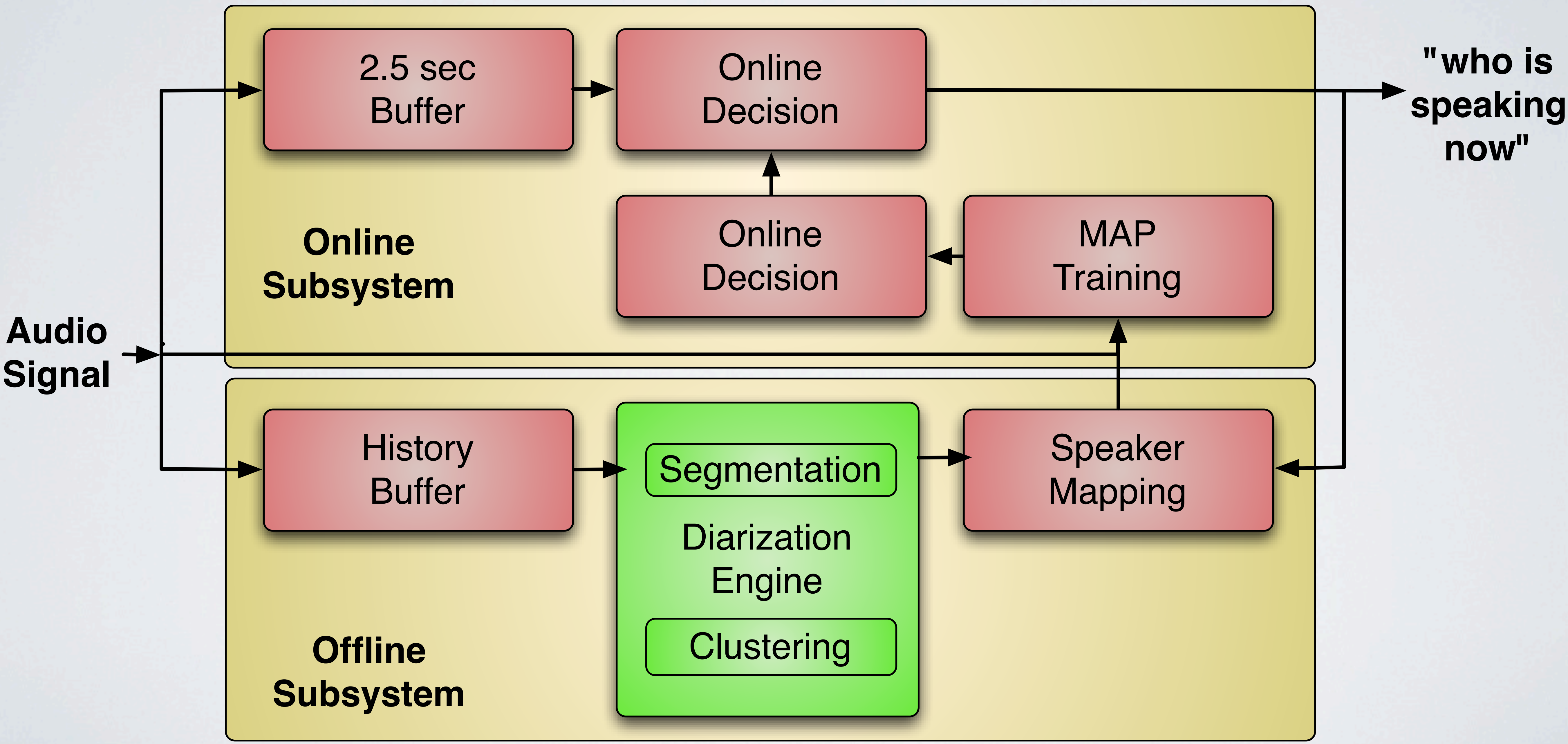
OFFLINE SPEAKER DIARIZATION



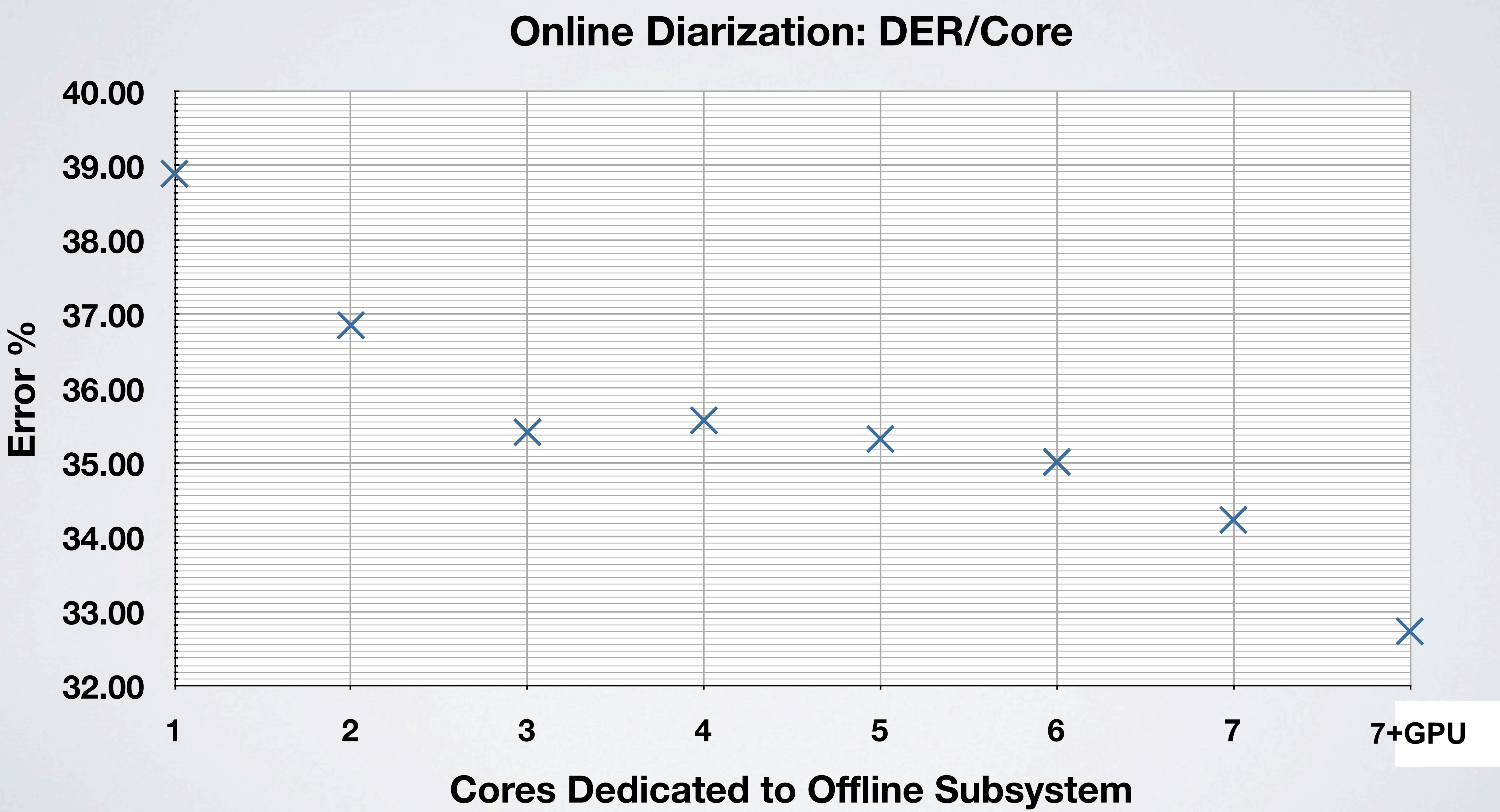
ONLINE SPEAKER DIARIZATION

- Precompute models for each speaker.
  - Run offline diarization on the start of a meeting.
  - Train models on first 60 seconds from each resulting speaker.
  - Another approach: stored models per speaker.
- Every 2.5 seconds, compute scores for each speaker model and output the highest.

HYBRID ONLINE/OFFLINE DIARIZATION



HYBRID ONLINE/OFFLINE DIARIZATION



DIARIZATION

- Next steps:
  - CPU/GPU hybrid system
  - Implement serial optimizations in parallel version
  - Integrate with manystream approach

CONCLUSION

- Speech technology can use all resources that are available.
- Parallelism enables improvements in several areas:
  - Accuracy
  - Throughput
  - Latency
- Programming parallel systems continues to be challenging.