HYDRA:
A Hybrid CPU/GPU Speech Recognition Engine for Real-Time LVCSR

Jungsuk Kim Jike Chong Ian Lane

Electrical and Computer Engineering

Carnegie Mellon University

Silicon Valley

March 21, 2013 @GTC2013

Overview

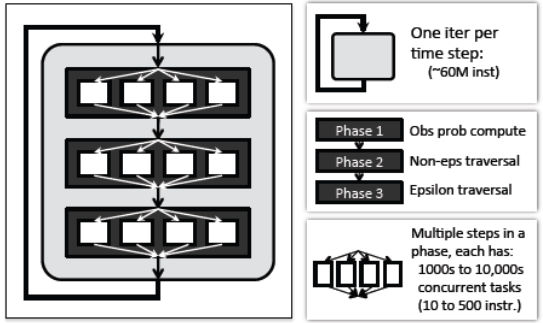
- Introduction
- Speech Recognition with Weighted Finite State Transducers (WFST)
- GPU-Accelerated Speech Recognition
- On-The-Fly Hypothesis Rescoring
- Evaluation Results
- Demonstration Video

Introduction

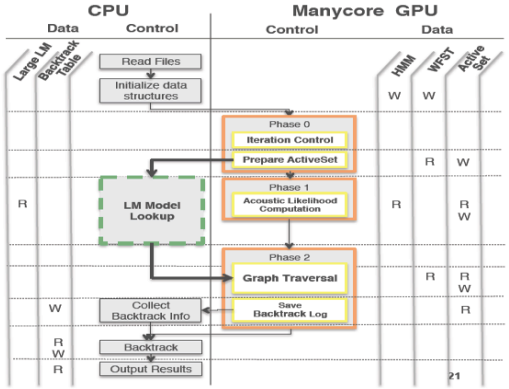
- Voice interfaces a core technology for User Interaction
  - Mobile devices, Smart TVs, In-Vehicle Systems, …
- For a captivating User Experience, Voice UI must be:
  - **Robust**
    - Acoustic robustness → **Large Acoustic Models**
    - Linguistics robustness → **Large Vocabulary Recognition**
  - **Responsive**
    - Low latency → **Faster than real-time search**
  - **Adaptive**
    - User and Task adaptation

A New Speech Recognition Architecture Required

Introduction



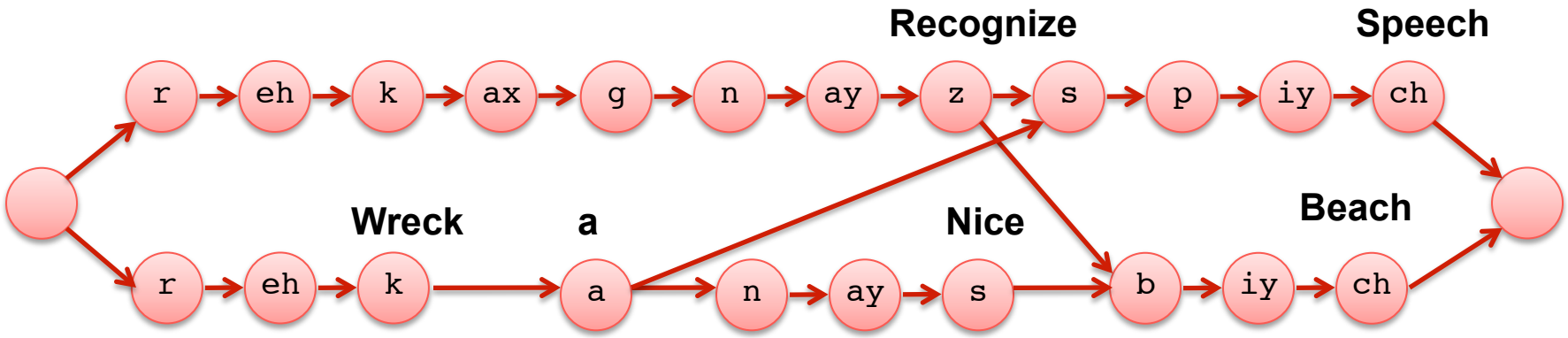
&



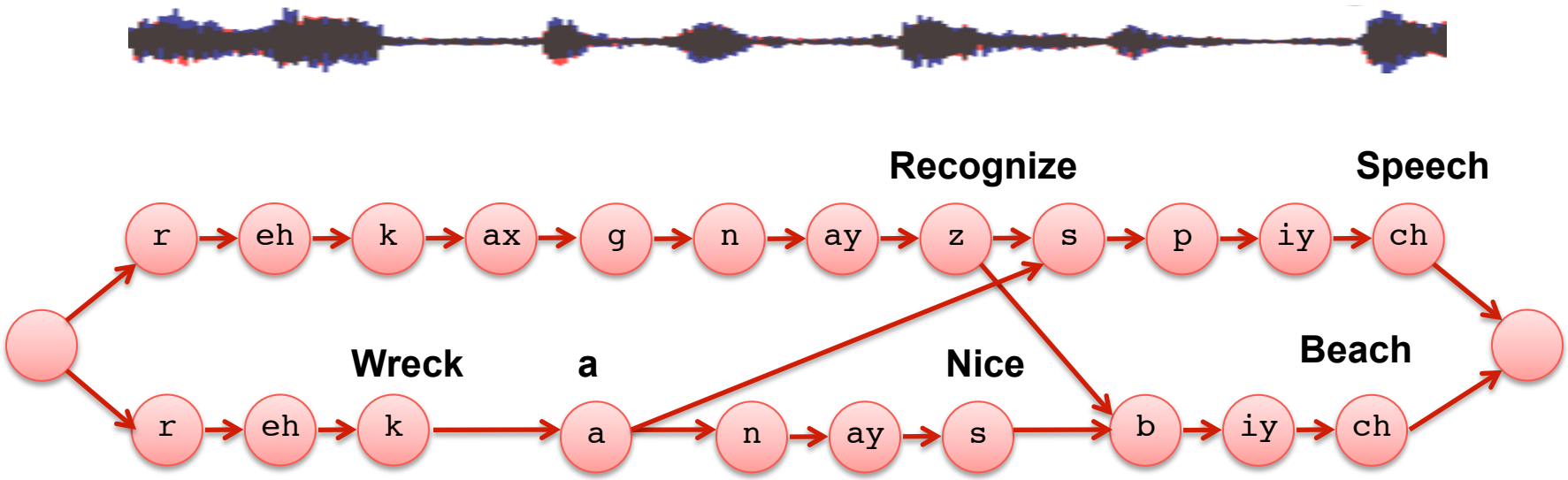
Speech recognition contains many highly parallel tasks + **GPU processors** optimized for parallel computing = **HYDRA** an ASR engine designed specifically for GPUs

Speech Recognition with Weighted Finite State Transducers (WFSTs)

A search space in speech recognition is represented as a WFST graph that includes acoustic, phonetic and linguistic constraints

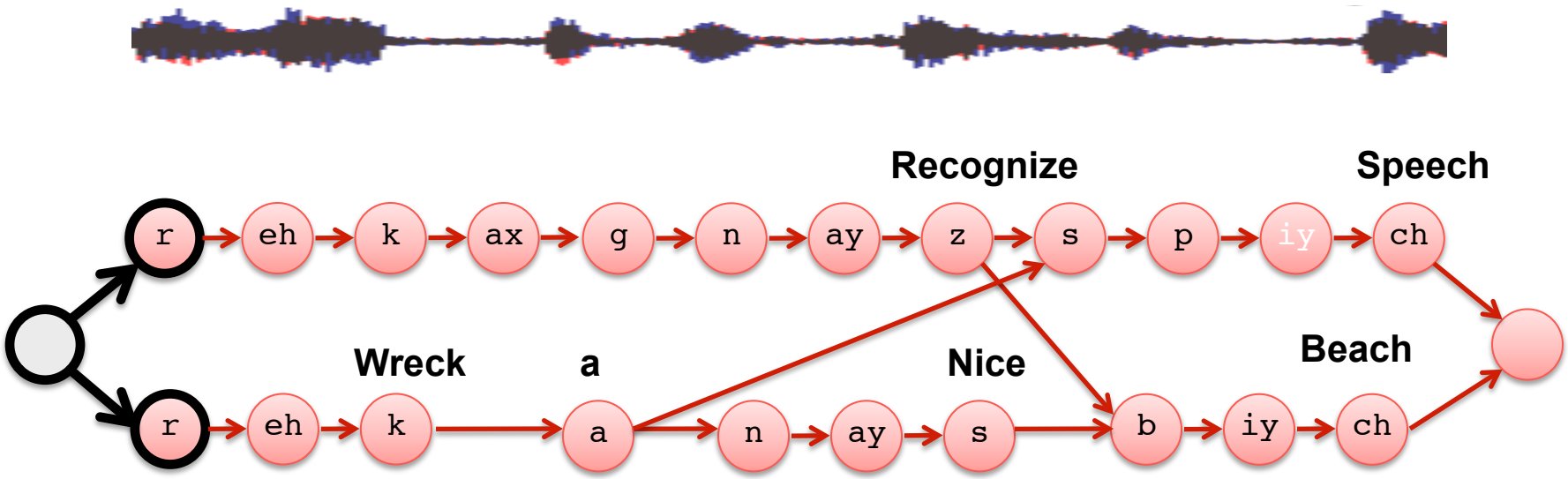


Speech Recognition with Weighted Finite State Transducers (WFSTs)



Search is performed in 3 phases

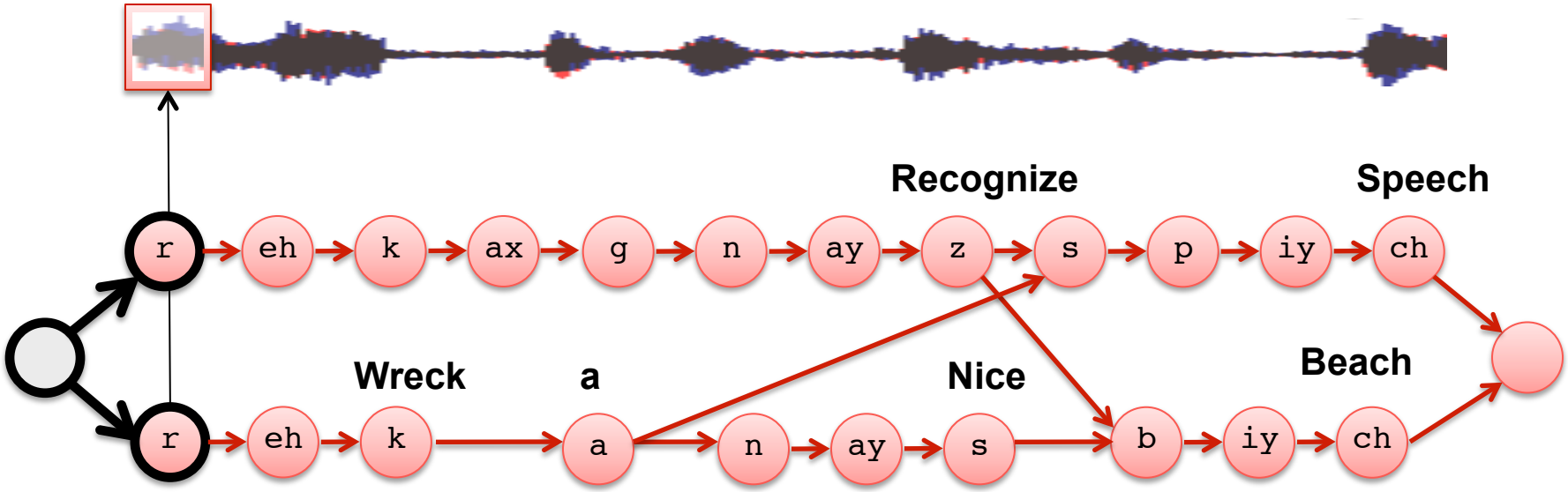
Speech Recognition with Weighted Finite State Transducers (WFSTs)



Phase 0: Prepare Active Set

Gather active states (speech recognition hypotheses) from previous frame.

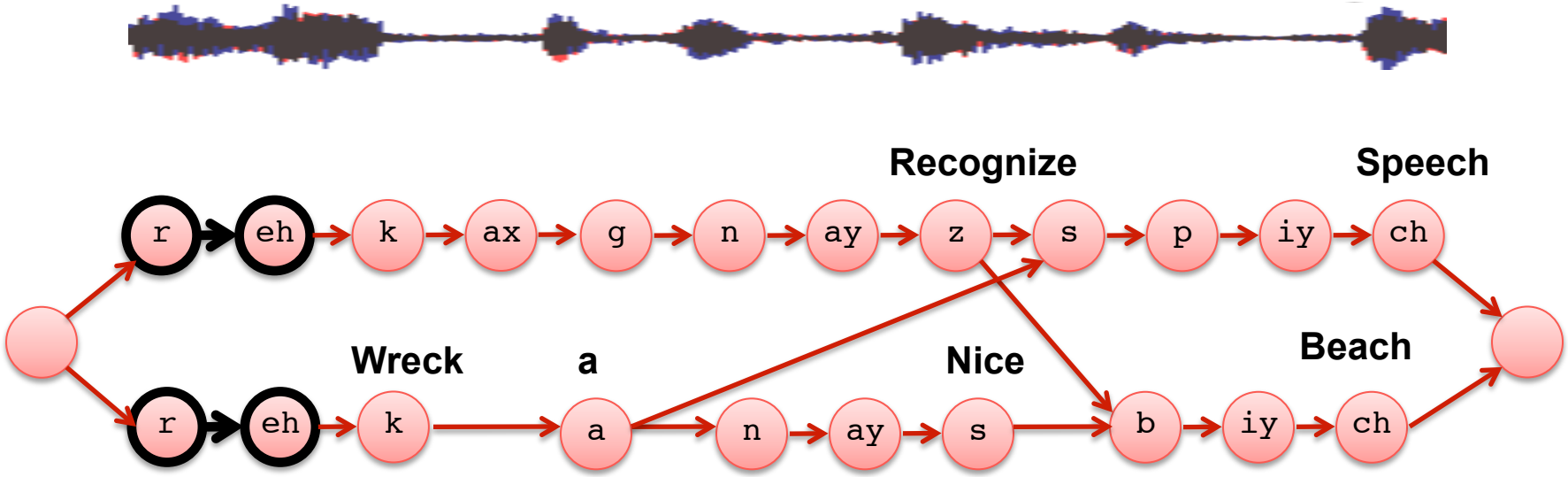
Speech Recognition with Weighted Finite State Transducers (WFSTs)



Phase 1: Compute Observation Probability

Compute likelihood of phonetic models (Gaussian Mixture Model) for current frame

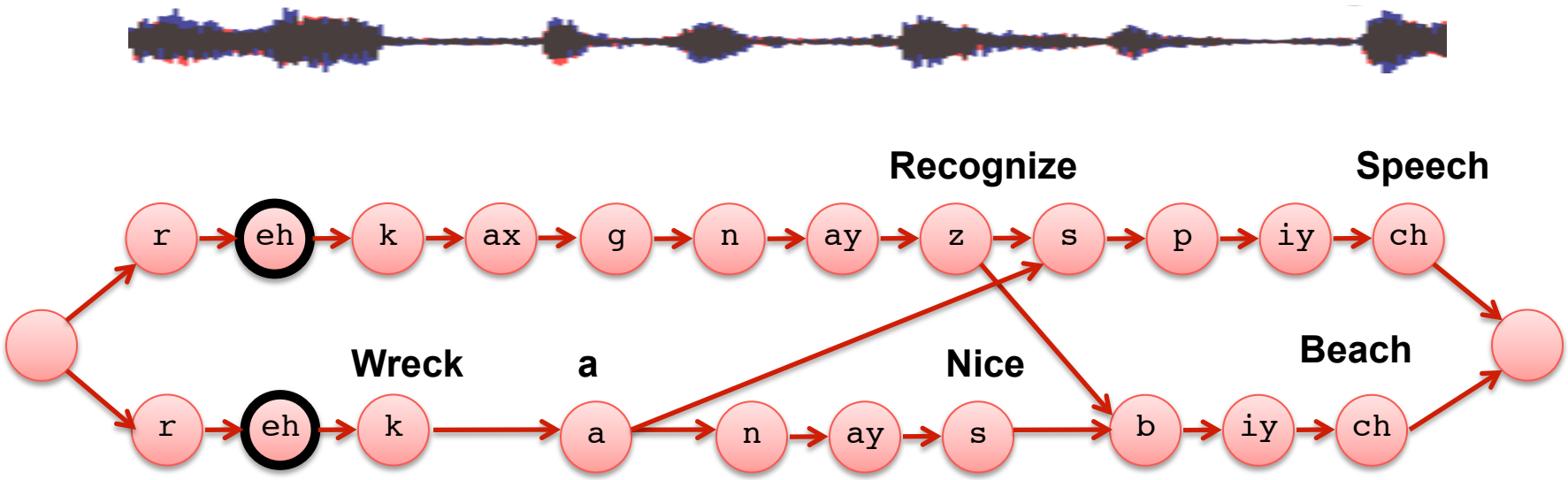
Speech Recognition with Weighted Finite State Transducers (WFSTs)



Phase 2: WFST Search

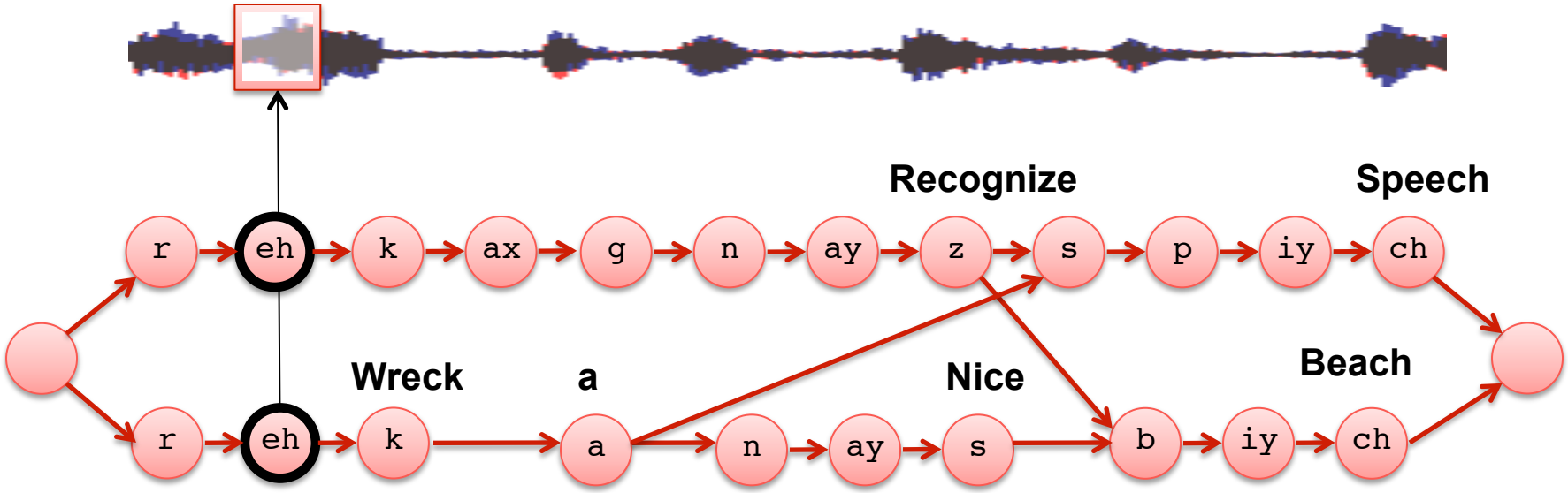
Frame synchronous Viterbi search is performed on WFST network

Speech Recognition with Weighted Finite State Transducers (WFSTs)



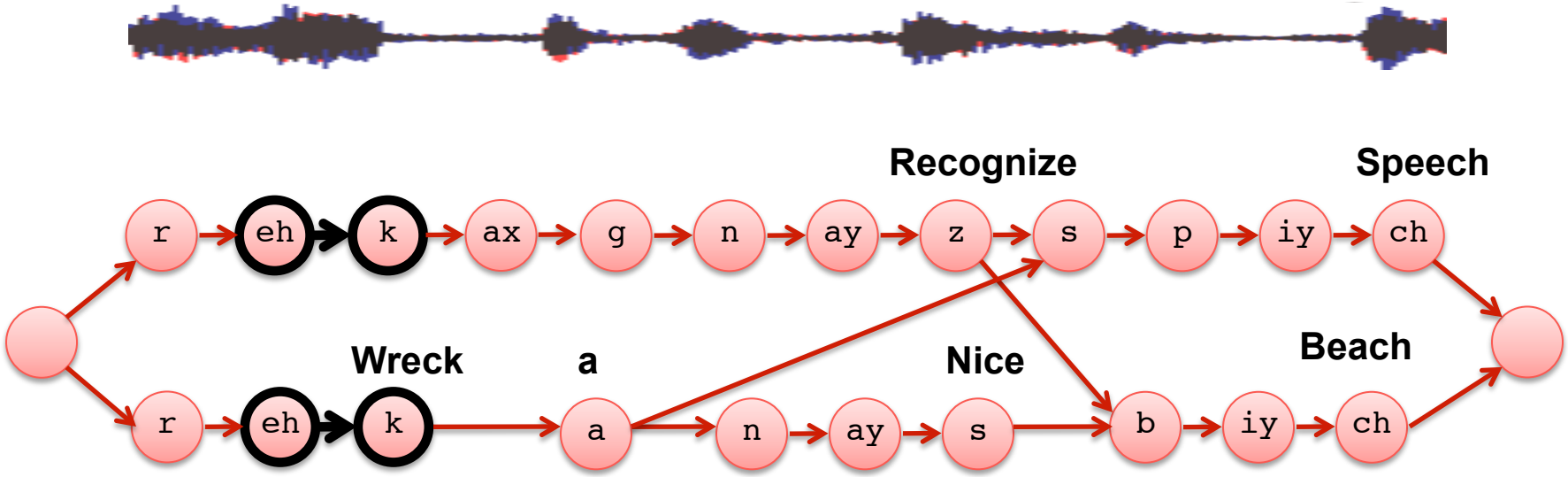
Phase 0: Prepare Active Set

Speech Recognition with Weighted Finite State Transducers (WFSTs)



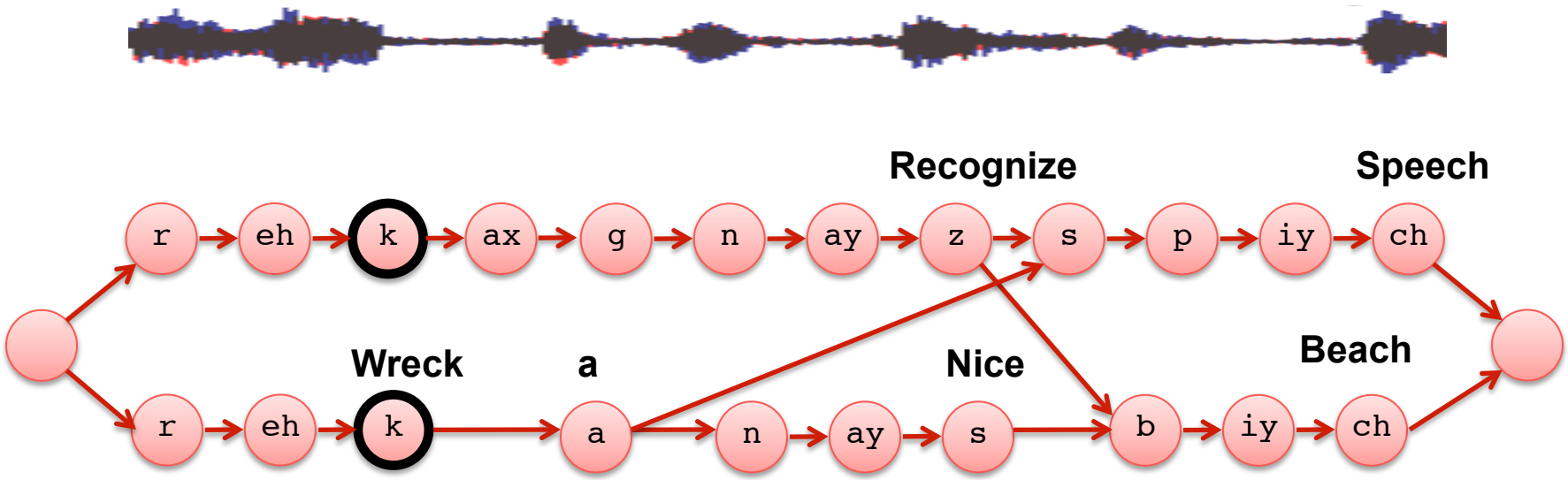
Phase 1: Compute Observation Probability

Speech Recognition with Weighted Finite State Transducers (WFSTs)



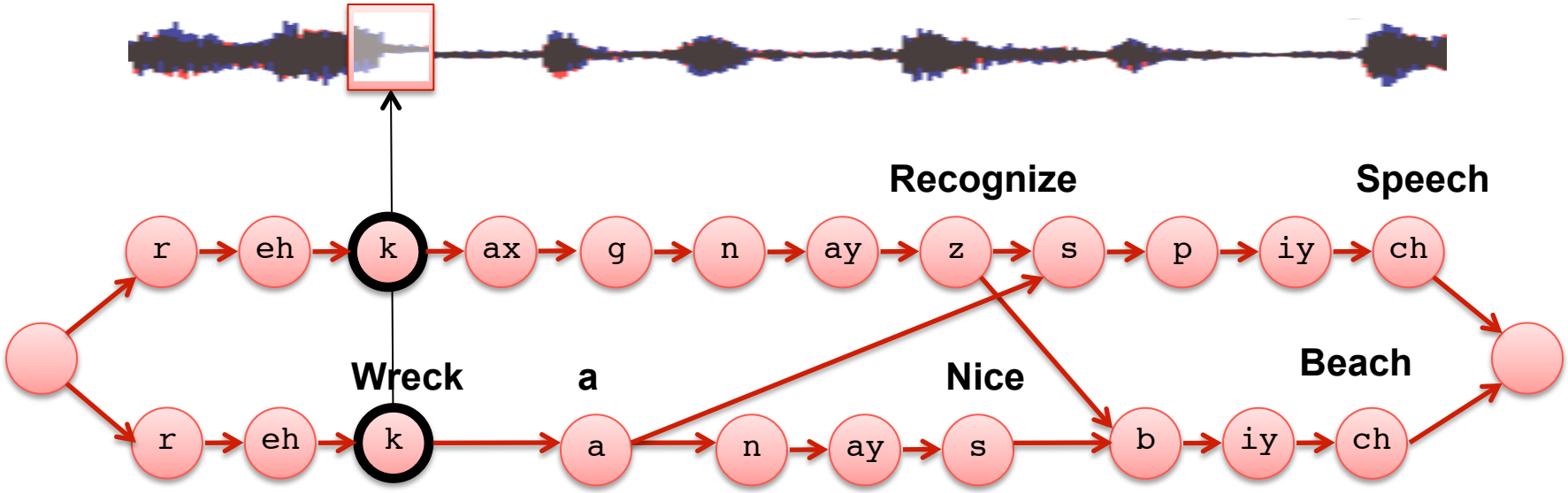
Phase 2: WFST Search

Speech Recognition with Weighted Finite State Transducers (WFSTs)



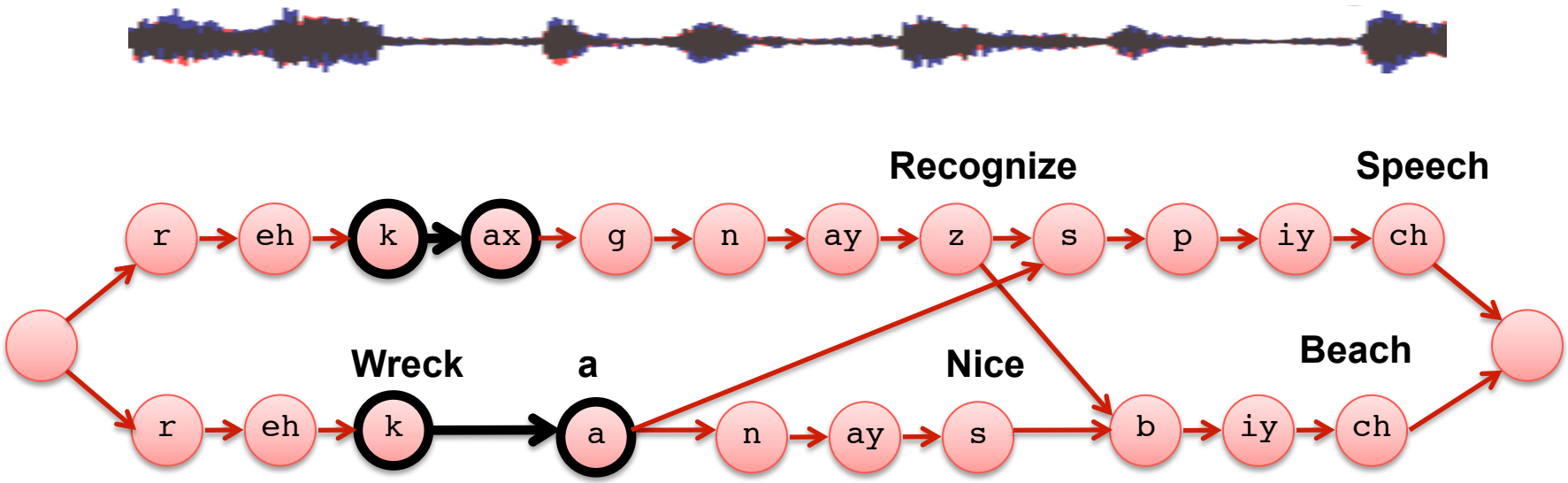
Phase 0: Prepare Active Set

Speech Recognition with Weighted Finite State Transducers (WFSTs)



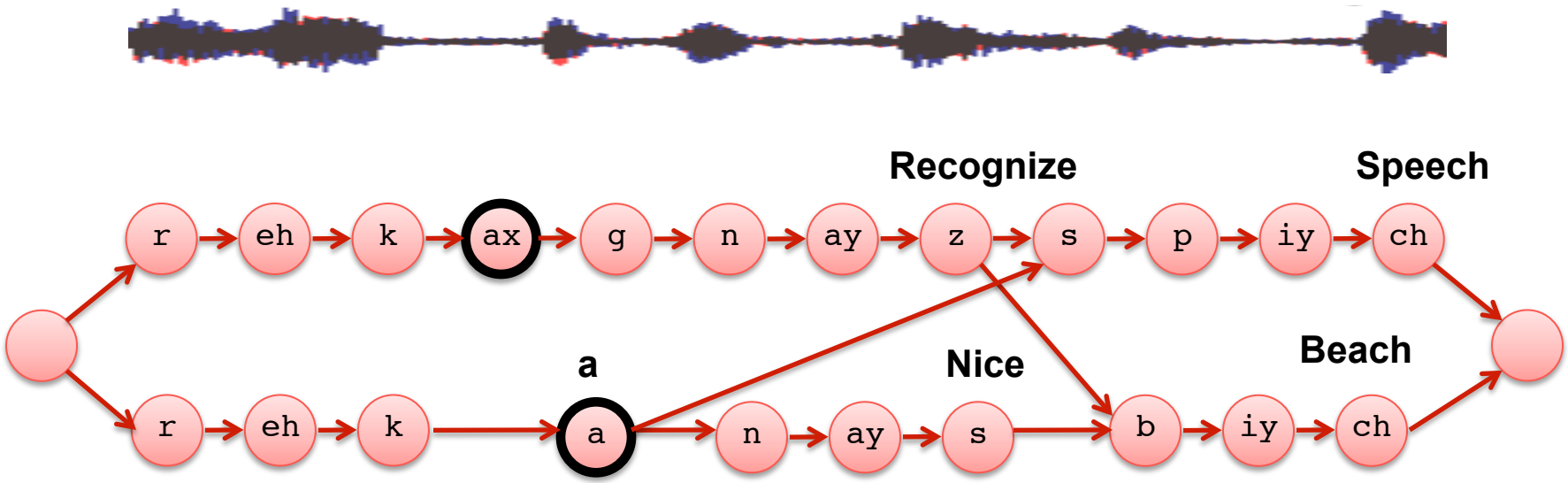
Phase 1: Compute Observation Probability

Speech Recognition with Weighted Finite State Transducers (WFSTs)



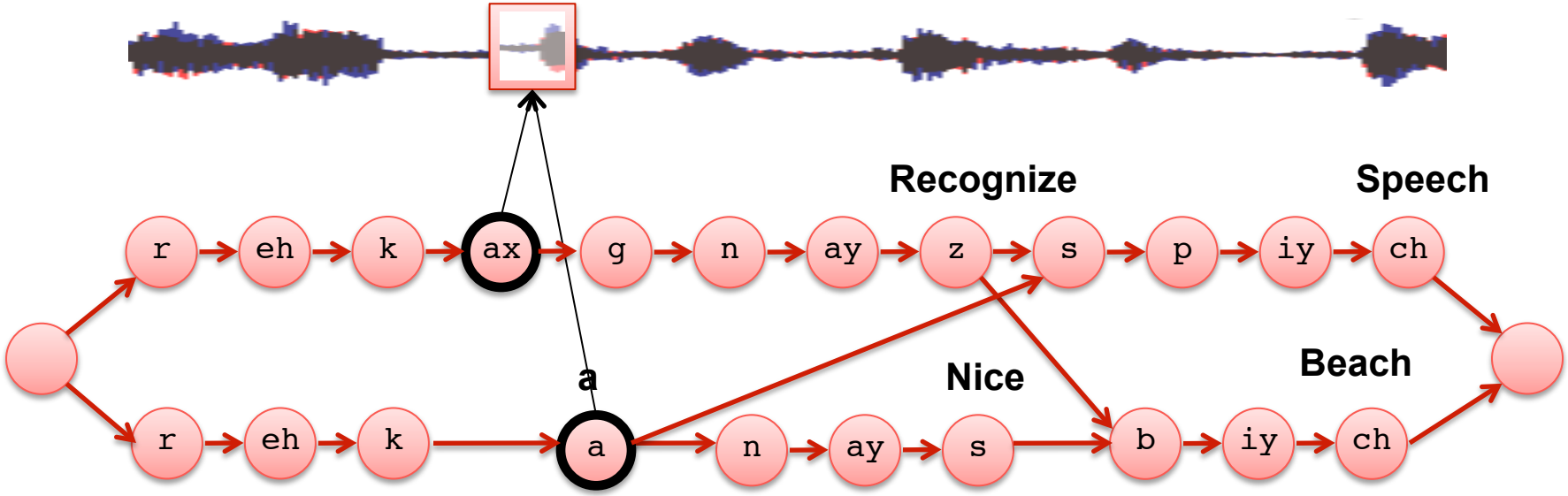
Phase 2: WFST Search

Speech Recognition with Weighted Finite State Transducers (WFSTs)



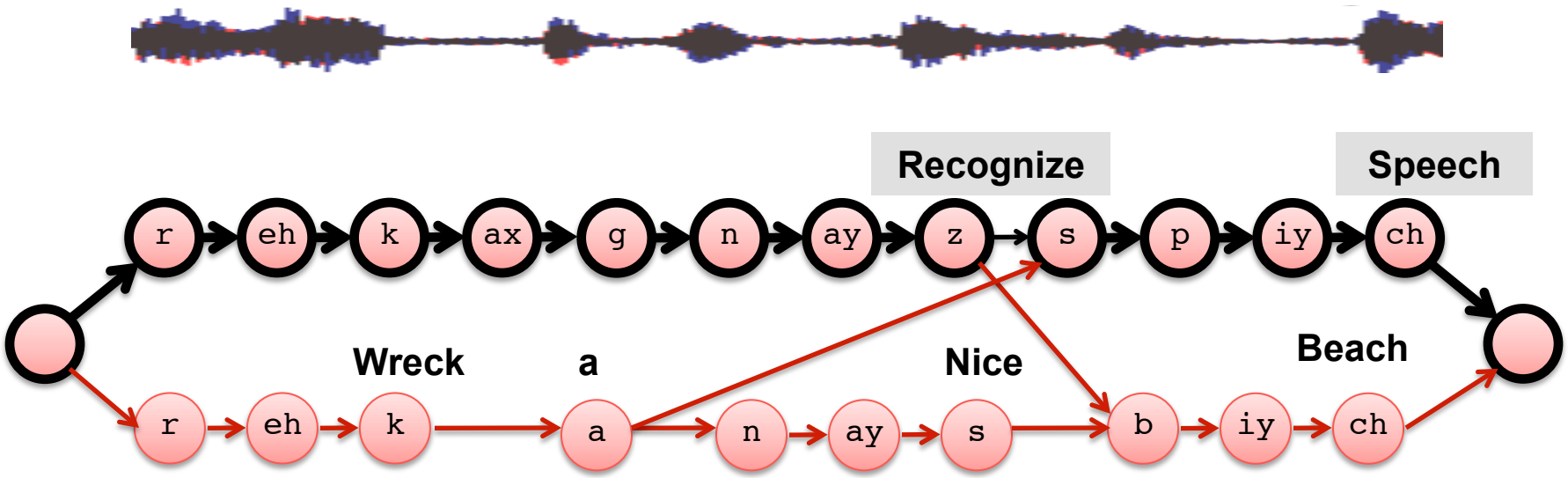
Phase 0: Prepare Active Set

Speech Recognition with Weighted Finite State Transducers (WFSTs)



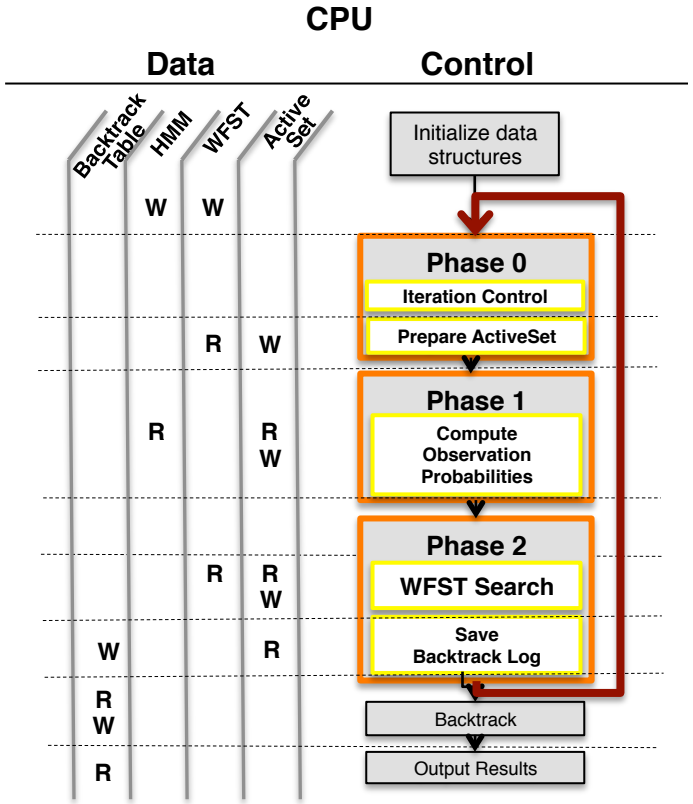
Phase 1: Compute Observation Probability

Speech Recognition with Weighted Finite State Transducers (WFSTs)



Backtrack from Viterbi Search is Speech Recognition Hypothesis

Speech Recognition with Weighted Finite State Transducers (WFSTs)

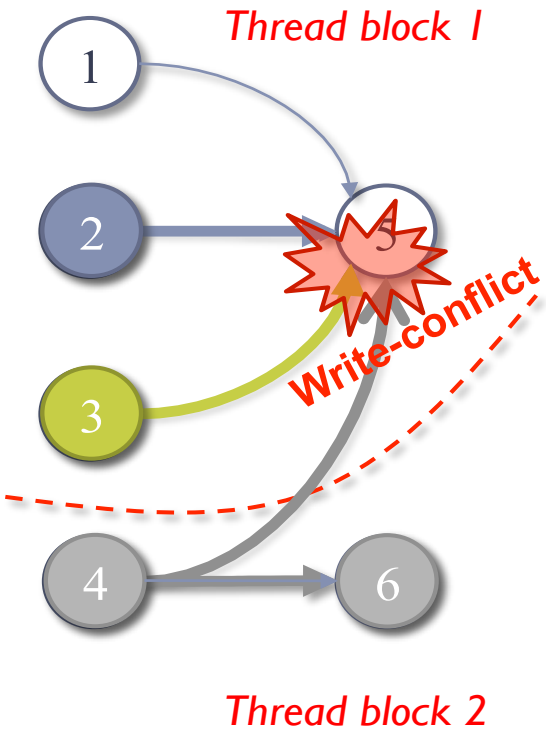


Decoding Process

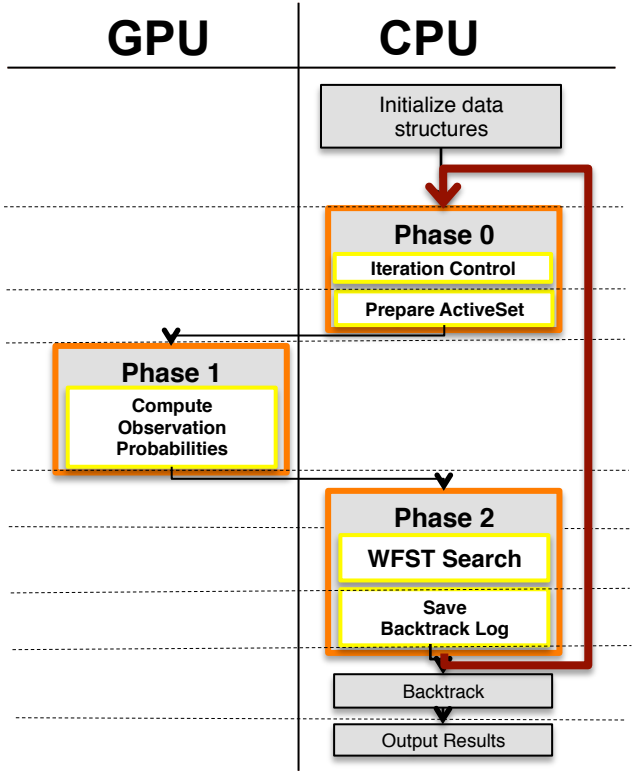
- **Phase 0: Prepare Active Set**
  - Gather active speech recognition ***hypotheses*** from previous frame.
- **Phase 1: Compute Observation Probability**
  - ***Compute likelihood*** of phonetic models (Gaussian Mixture Model) for current input features.
- **Phase 2: WFST Search**
  - Frame synchronous ***Viterbi search*** is performed on WFST network.

Concurrency Opportunities

- **Phase 0: Prepare Active Set**
  - ***Communication-intensive phase***
- **Phase 1: Compute Observation Probability**
  - 1000~2000 GMM clusters are activated per step.
  - ***Computation-intensive phase***
- **Phase 2: WFST Search**
  - 10,000s partial hypotheses tracked per step.
  - Handling ***irregular graph structure*** with data parallel operation.
  - ***Conflict-free reduction*** in graph traversal to resolve write-conflict.
  - ***Communication-intensive phase***



GPU-Accelerated Speech Recognition



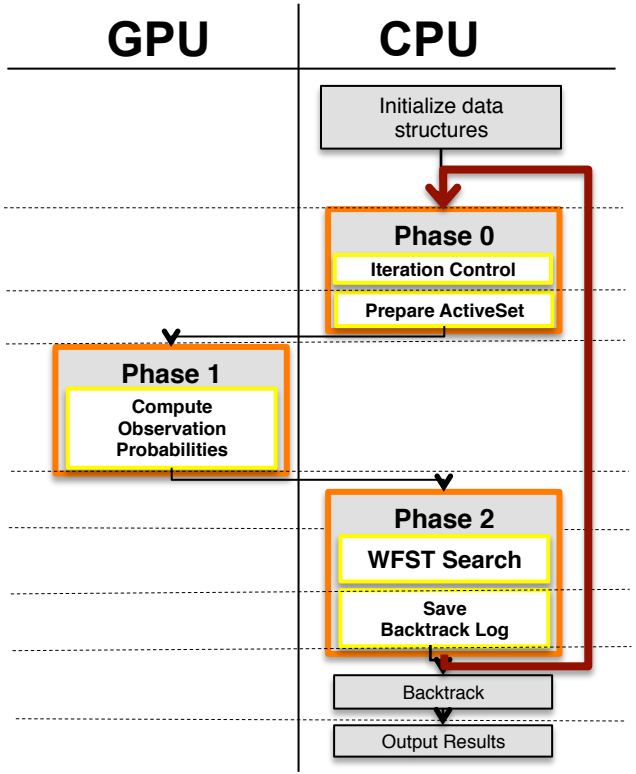
Paul R. Dixon [ICASSP 2009]

Approach 1:

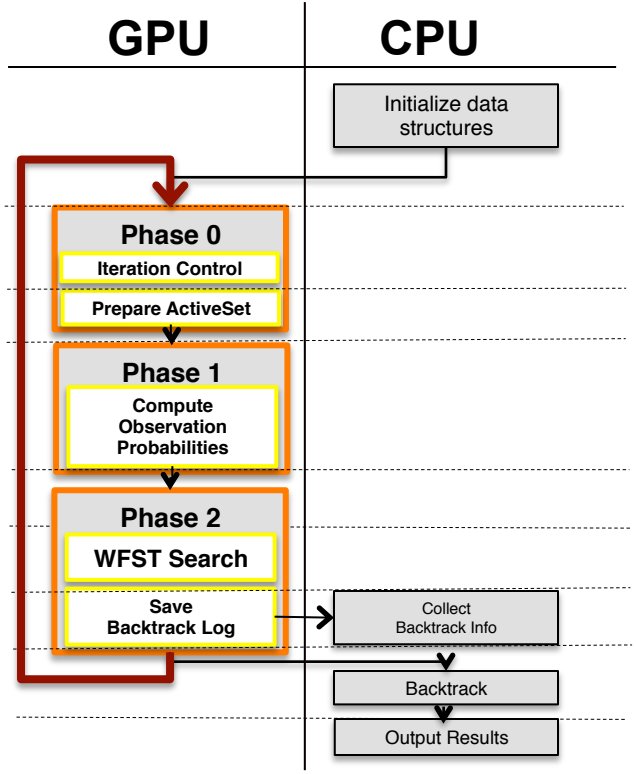
- Conduct compute intensive phase only ***on GPU***
- Incurs ***high data-copying overheads*** between the CPU and the GPU.
- ***Less scalable*** as transfer become a sequential bottleneck in algorithm.

Faster than CPU but less scalable

GPU-Accelerated Speech Recognition

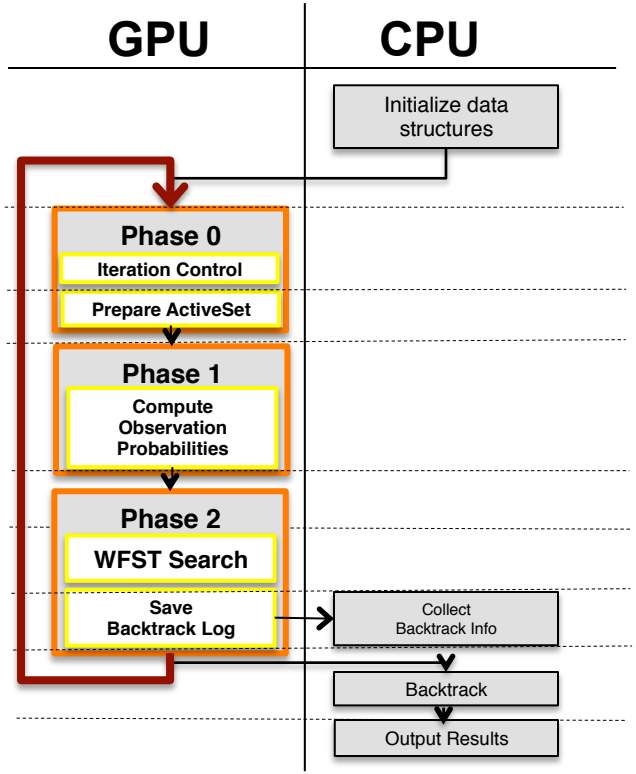


Paul R. Dixon [ICASSP 2009]



Jike Chong [INTERSPEECH 2009]

GPU-Accelerated Speech Recognition



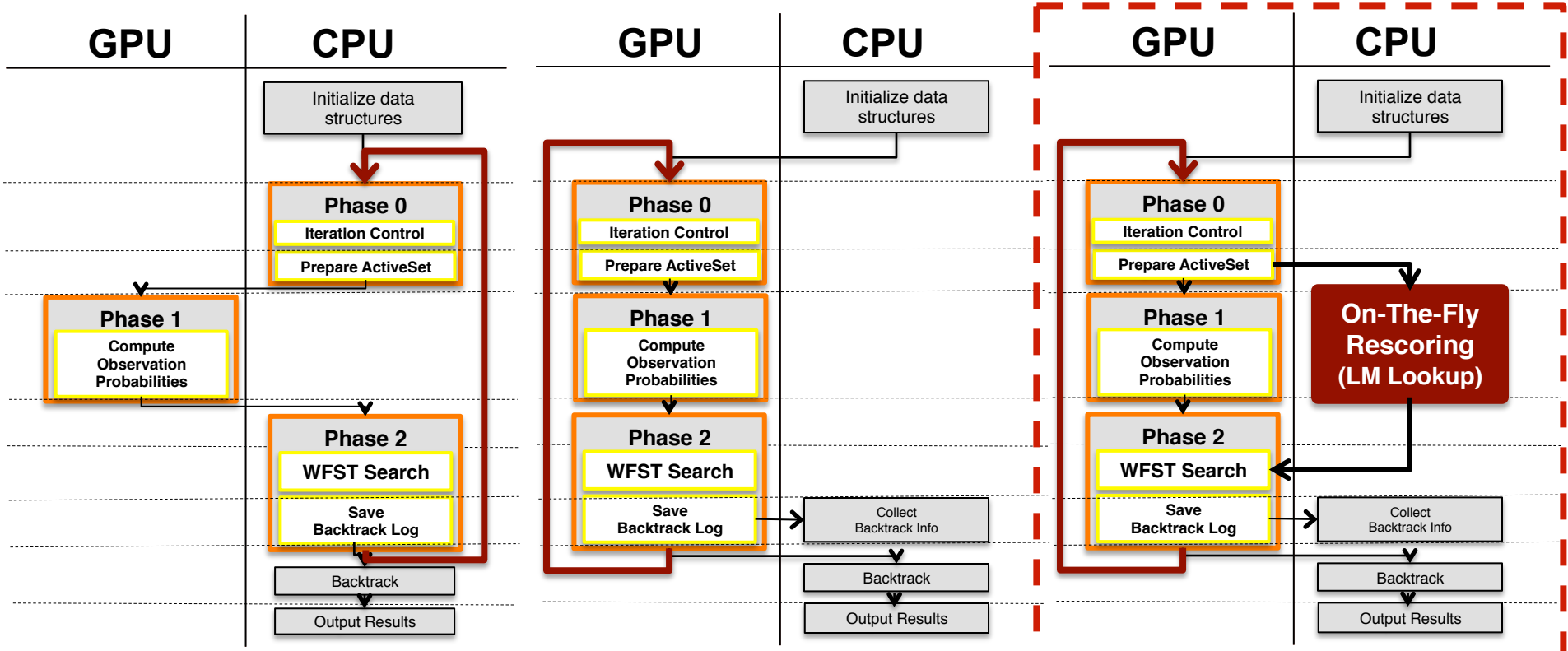
Jike Chong [INTERSPEECH 2009]

Approach 2:

- Both phases are ***on GPU.***
- ***Scalable*** algorithm
- Only suitable for small models (vocab / linguistic context) due to limited GPU memory (1~6GB)

Fast and Scalable but Limited

GPU-Accelerated Speech Recognition

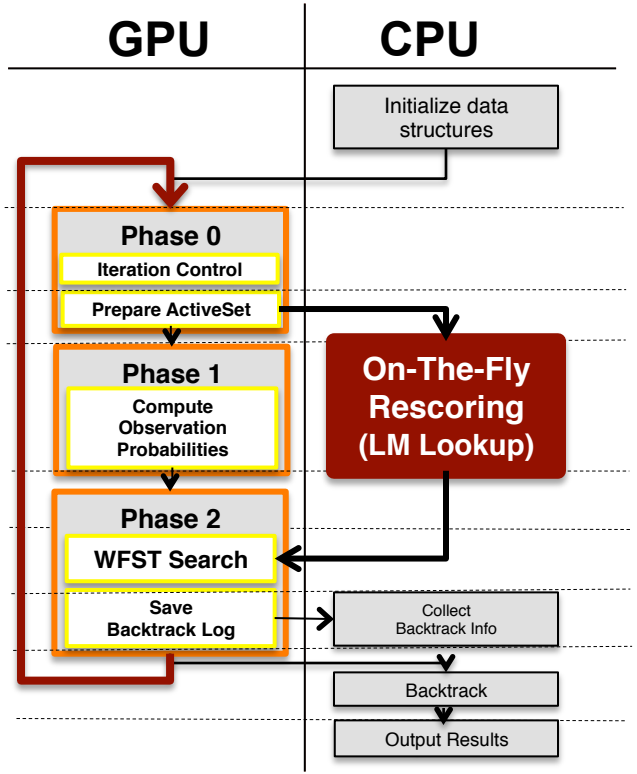


Paul R. Dixon [ICASSP 2009]

Jike Chong [INTERSPEECH 2009]

Proposed

GPU-Accelerated Speech Recognition



Proposed

Proposed Approach:

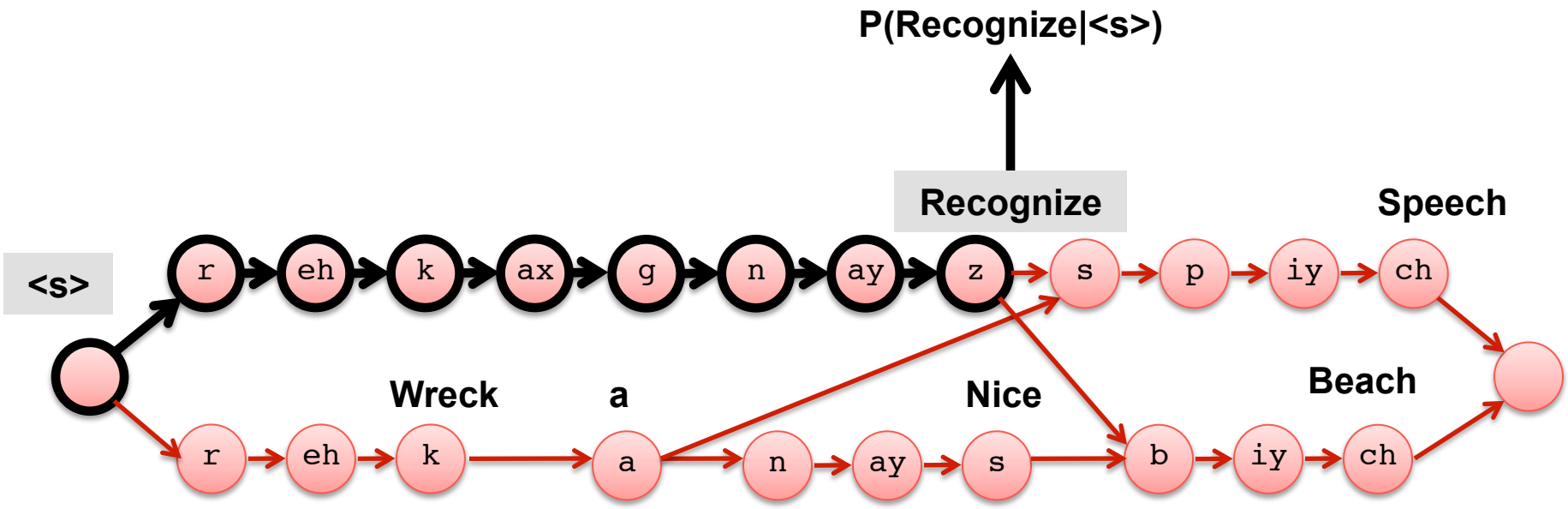
- Both phases are on GPU ***using unigram WFST***
- Rescore hypothesis ***"On-The-Fly"*** using ***larger language model on CPU***

Vocab. size	5k	64k	1M
unigram	2	12	92
bigram	114	1,880	N/A*
trigram	676	4,644*	N/A*

WFST network size (MB) ***(* Unable to decode)***

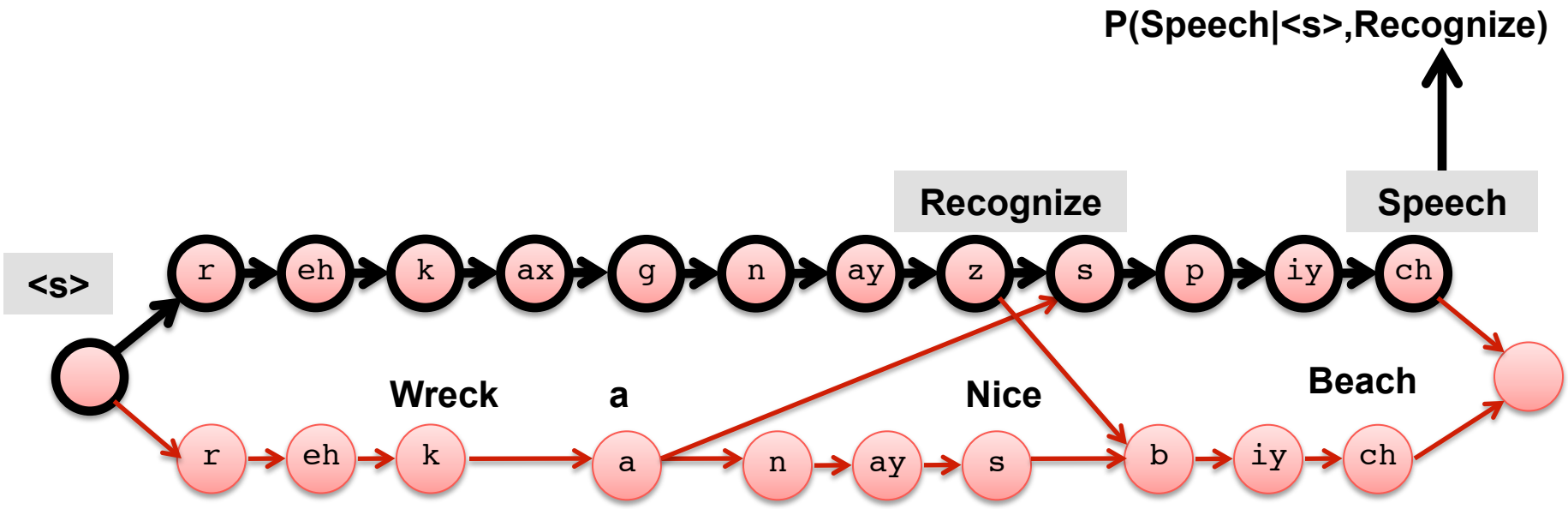
Fast, Scalable, Large Models

On-The-Fly Hypothesis Rescoring



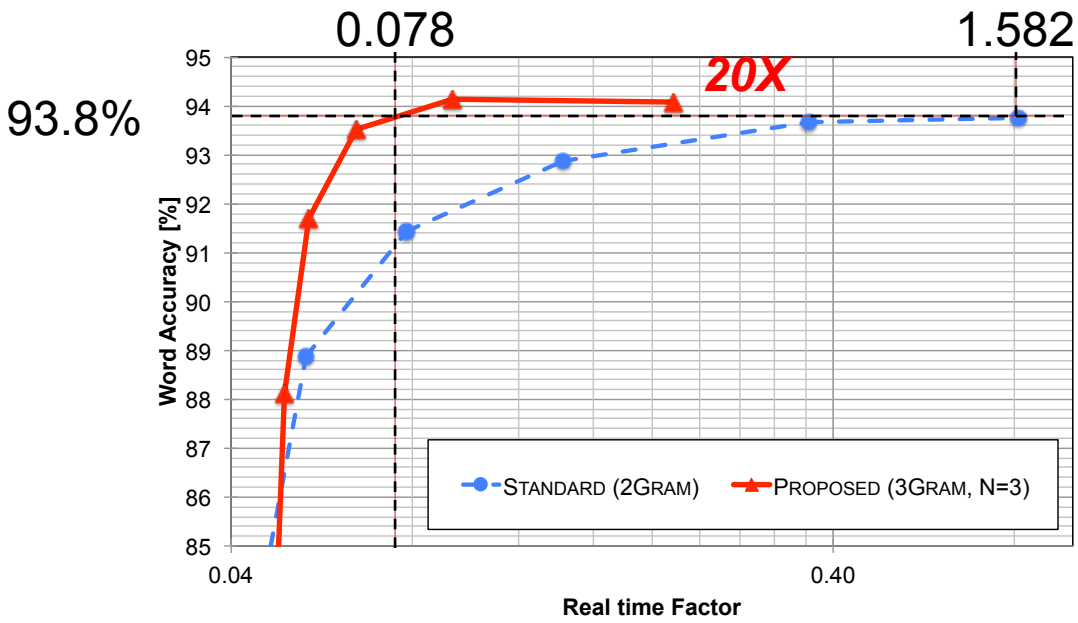
At word boundary update graph weight with linguistic context from CPU

On-The-Fly Hypothesis Rescoring



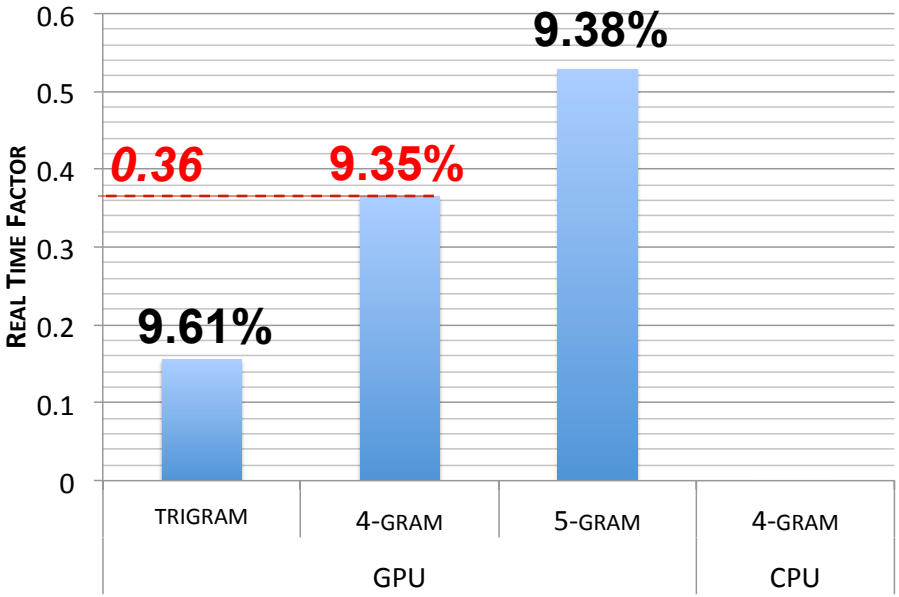
At word boundary update graph weight with linguistic context from CPU

Evaluation Results *(5k vocab.)*



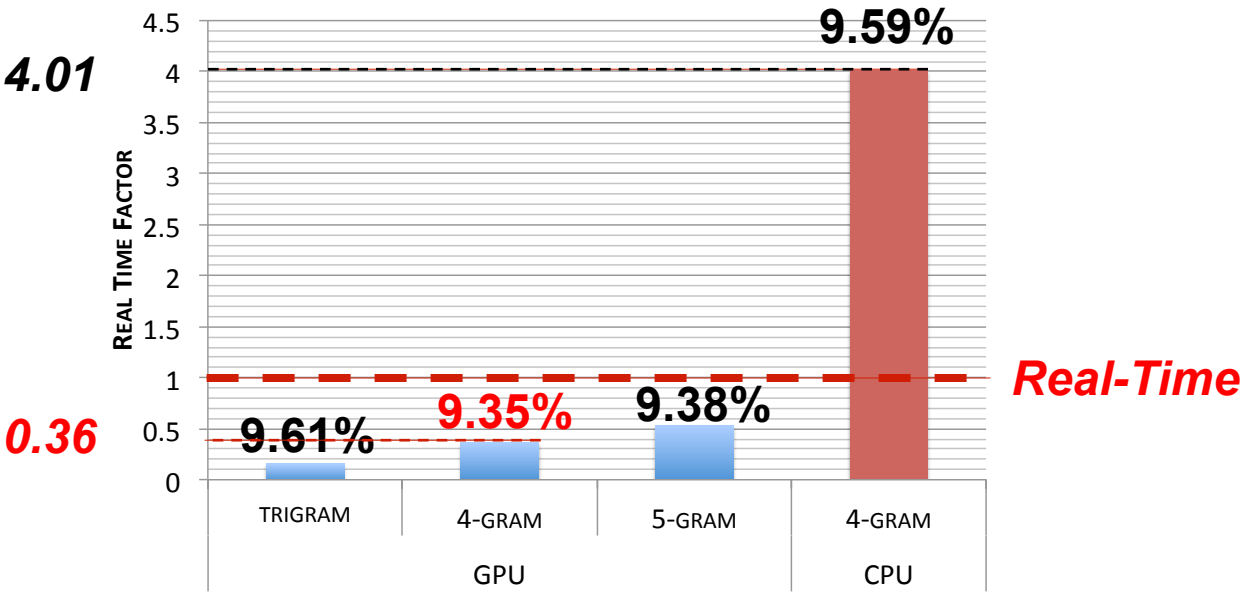
- ***20X speed-up*** compared to standard WFST decoding on CPU at word accuracy 93.80%
- ***95.4%*** maximum accuracy is achieved

Evaluation Results *(1M vocab.)*



- ***2.74X*** faster than real-time when the WER is ***9.35%***

Evaluation Results *(1M vocab.)*



- ***2.74X*** faster than real-time when the WER is ***9.35%***
- ***10X*** faster than CPU decoder.

Q&A

Thank you for your attention.