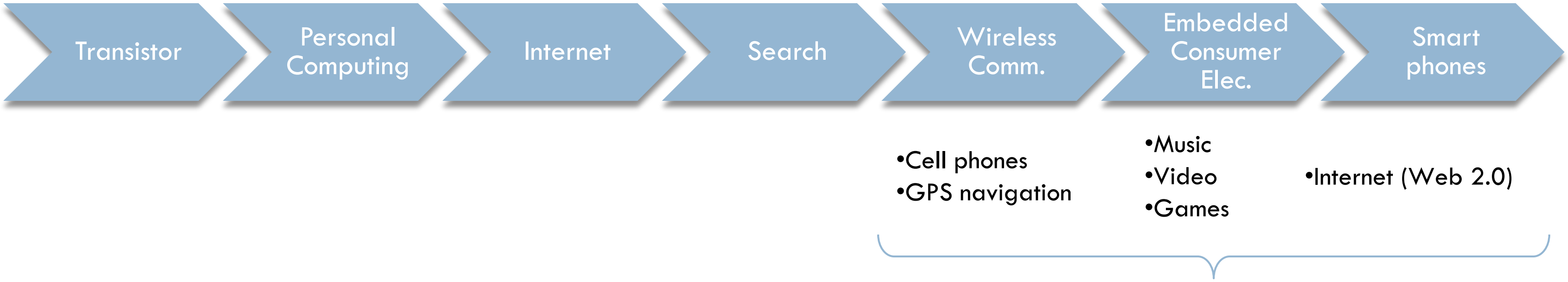
Hello GPU: High-Quality, Real-Time Speech Recognition on Embedded GPUs

Kshitij Gupta [/shi/ /tij/]
UC Davis

www.KshitijGupta.com

Three Trends

Trend #1: Technology



Transistor

Personal Computing

Internet

Search

Wireless Comm.

Embedded Consumer Elec.

Smart phones

- Cell phones
- GPS navigation

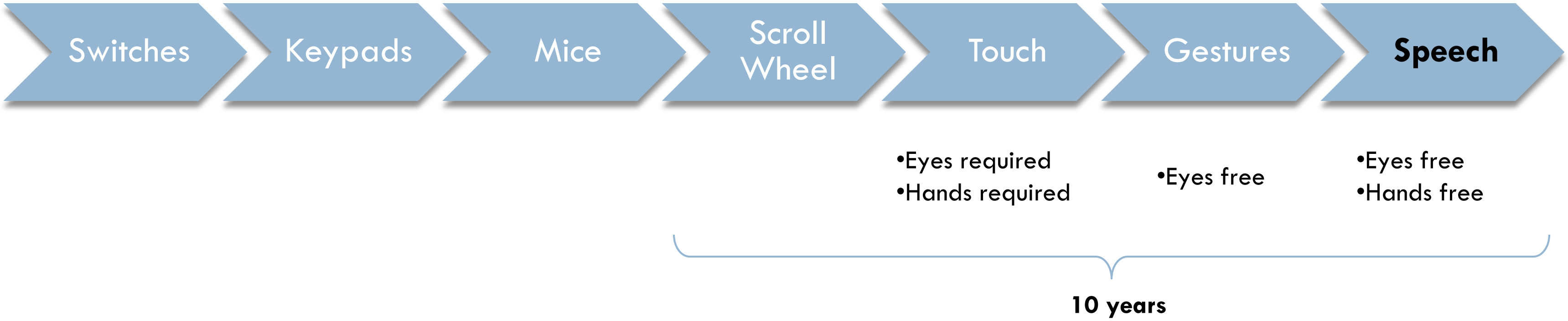
- Music
- Video
- Games

- Internet (Web 2.0)

Embedded/Mobile & Entertainment

Mobile + Convergence

Trend #2:
User Interface



Switches → Keypads → Mice → Scroll Wheel → Touch → Gestures → **Speech**

Touch:
- Eyes required
- Hands required

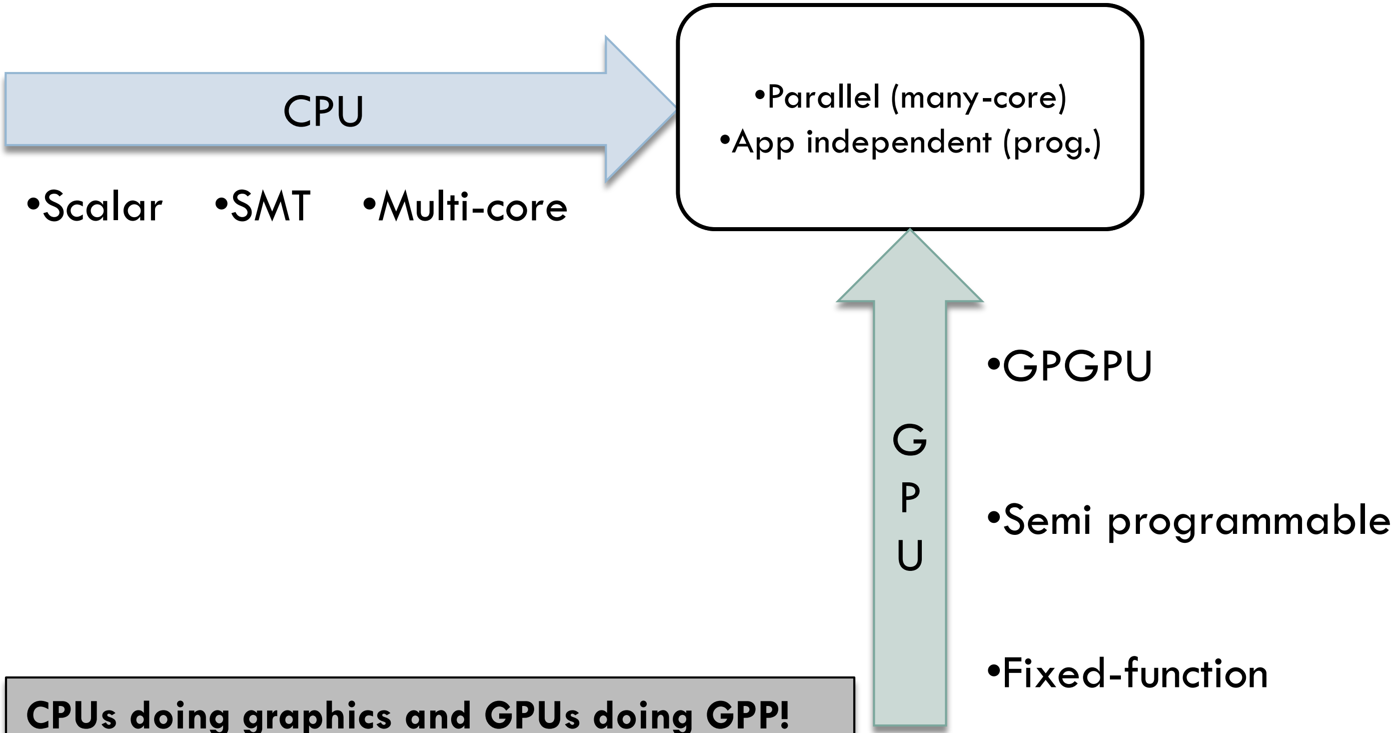
Gestures:
- Eyes free

Speech:
- Eyes free
- Hands free

10 years

User Interface has proven to be a key enabler

Trend #3(a): Processor Architecture (Desktop)



CPU

- Scalar
- SMT
- Multi-core

- Parallel (many-core)
- App independent (prog.)

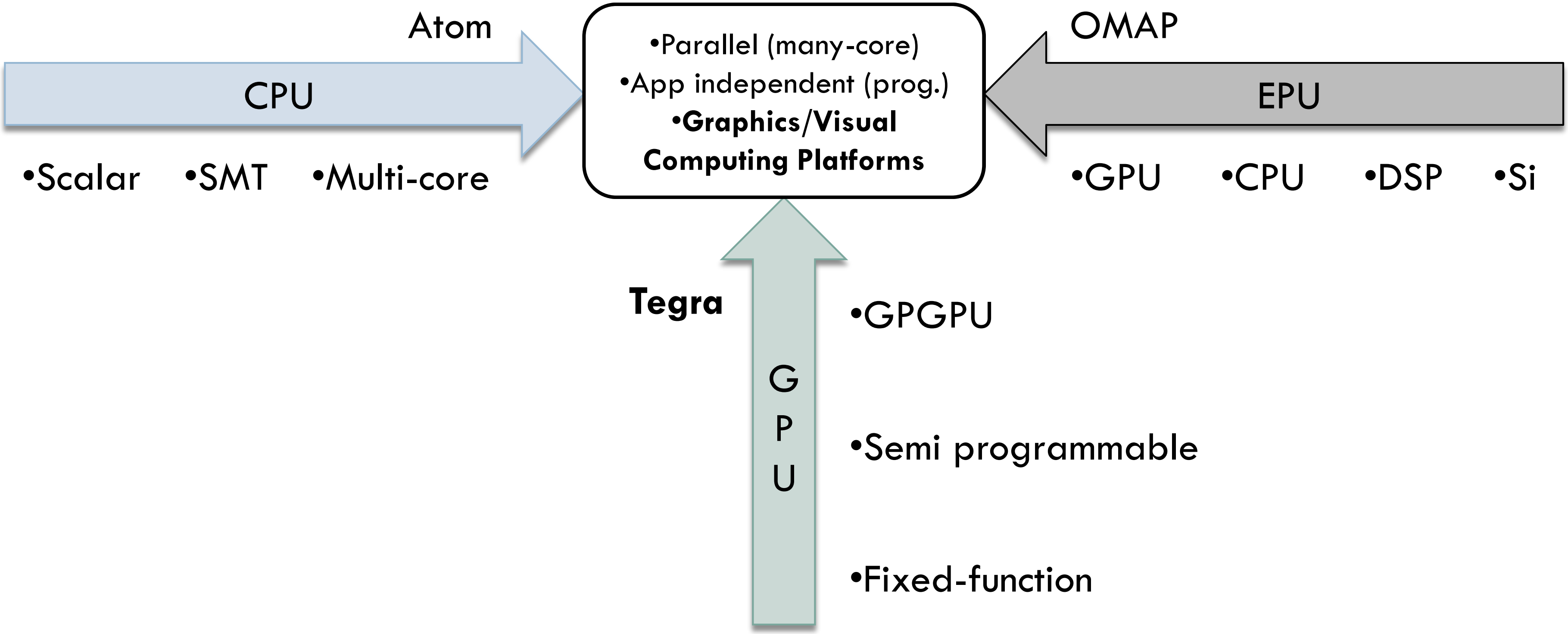
GPU

- GPGPU
- Semi programmable
- Fixed-function

CPUs doing graphics and GPUs doing GPP!

- CPU to run Aero-class graphics on Windows
- GPU evolving from “kernels” to “applications”

Trend #3(b): Processor Architecture (Embedded)



Atom

CPU

- Scalar
- SMT
- Multi-core

OMAP

EPU

- GPU
- CPU
- DSP
- Si

- Parallel (many-core)
- App independent (prog.)
- **Graphics/Visual Computing Platforms**

Tegra

GPU

- GPGPU
- Semi programmable
- Fixed-function

Looking Ahead…

Mobile

+

UI

+

Parallel, programmable

Outline

- Introduction
- Motivation
- Overview & Characterization
- Design Goals & Principles
- Acoustic Modeling Lookahead
- Future Directions

Why so hard?

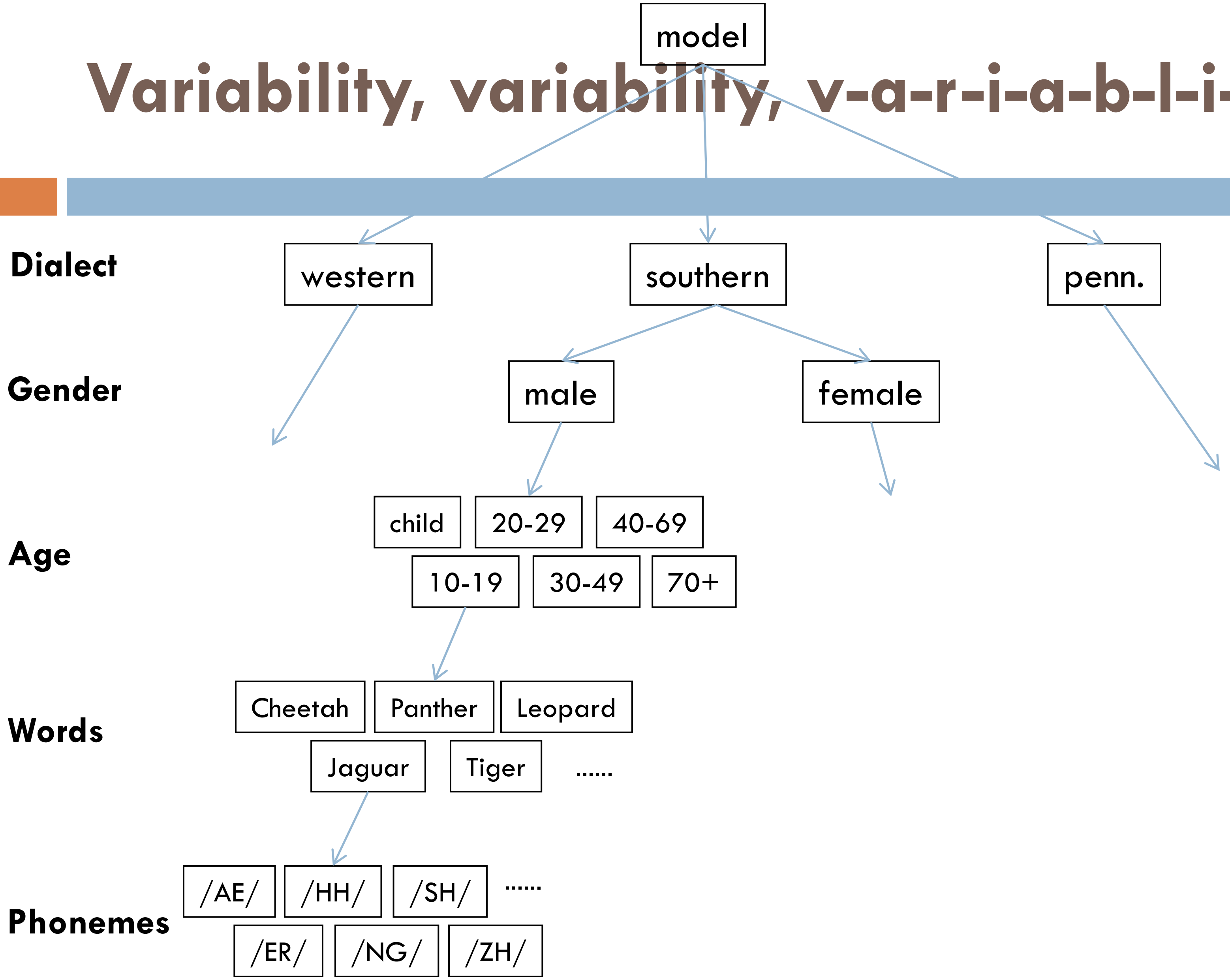
- The Holy Grail…
  - accurate
  - real-time
  - continuous
  - naturally spoken
  - noisy conditions
  - large set of words
  - speaker-independent
  - real-time!

Hard limit: Real-time response
Soft(er) limit: Accuracy!

A few examples of ‘continuous’ speech

- **thisnewdistplaywillrecognizespeech**
  - This new display will recognize speech
  - This nudist play will wreck a nice beach
- **greytape**
  - Grey tape
  - Great ape
- **hesgone**
  - He’s gone.
  - He’s gone?
- **Lets not go, ummm, ok, errr, fine, lets do this!**
  - Was that a ‘yes’ or a ‘no’?
  - What’s the context here?

Variability, variability, v-a-r-i-a-b-l-i-t-y!



model

Dialect: western, southern, penn.

Gender: male, female

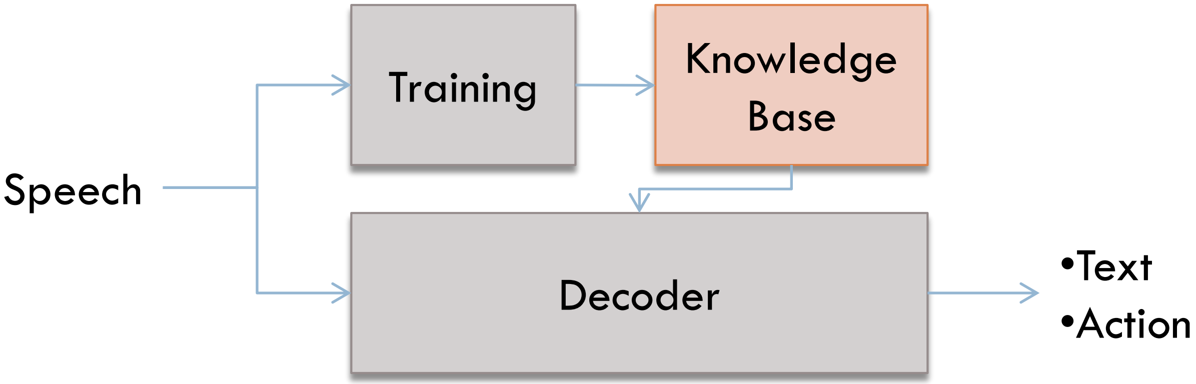
Age: child, 20-29, 40-69, 10-19, 30-49, 70+

Words: Cheetah, Panther, Leopard, Jaguar, Tiger ……

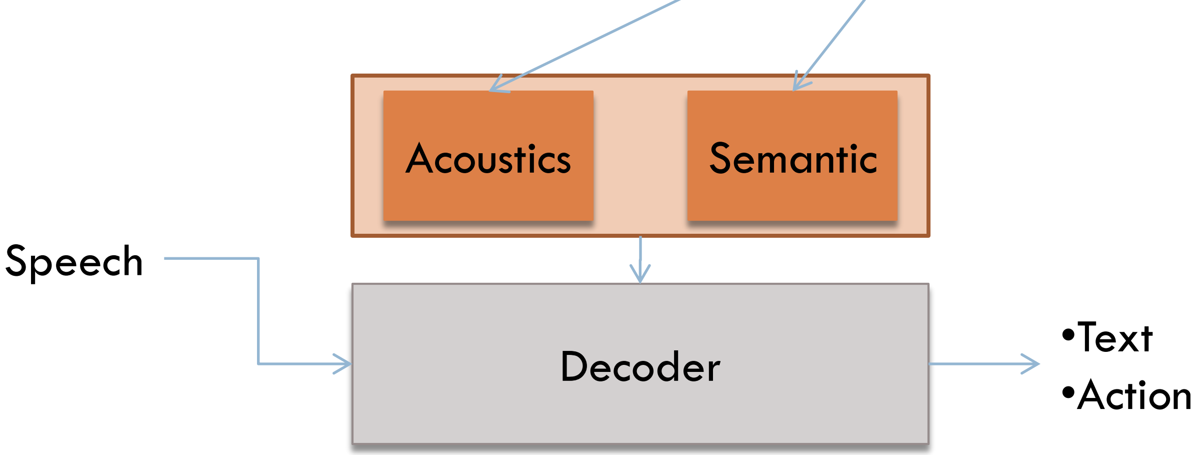
Phonemes: /AE/, /HH/, /SH/ …… /ER/, /NG/, /ZH/

Good speech models are BIG!

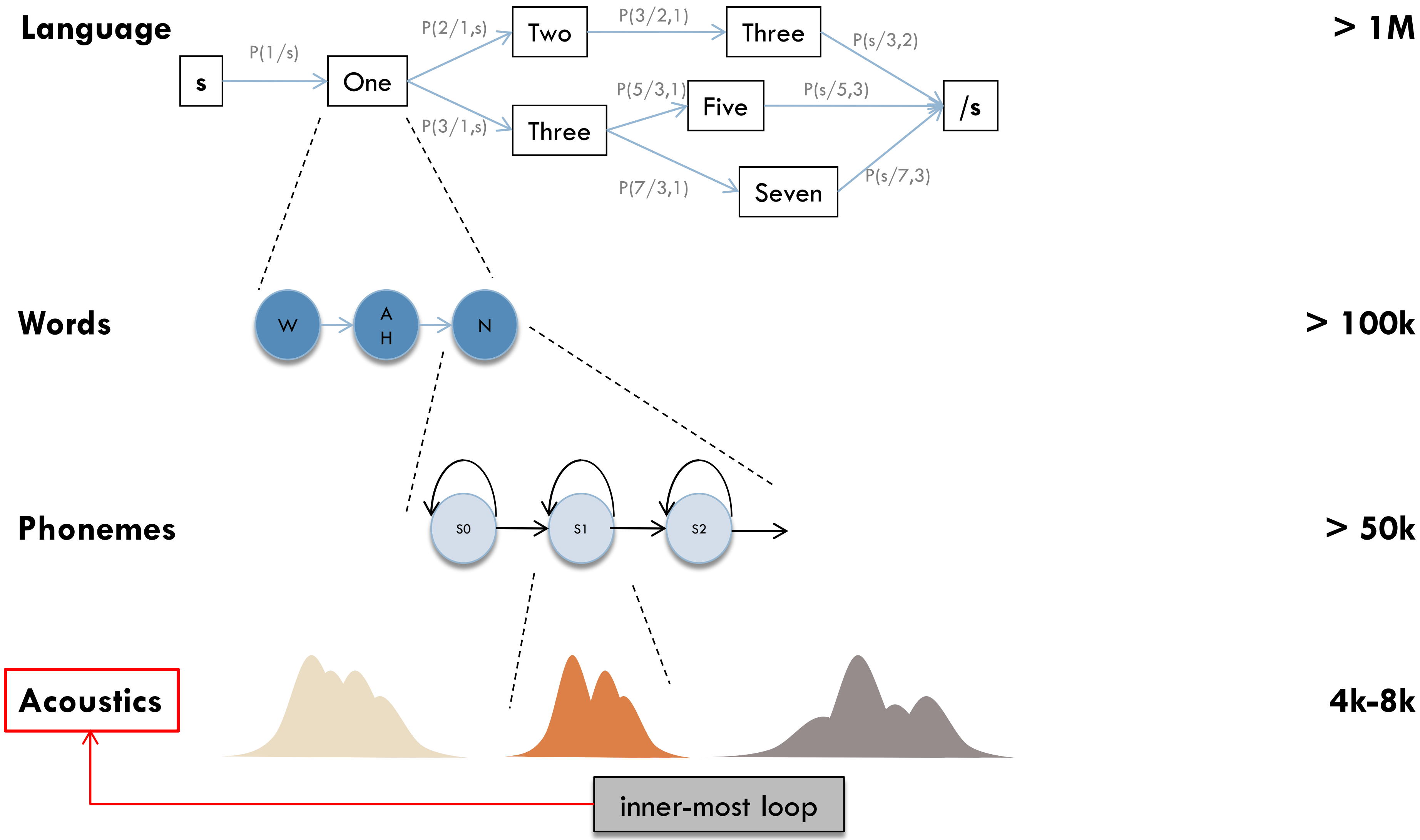
Automatic Speech Recognition: A high-level view



$$W_{best} = \arg\max_W \left[P(\overline{O}/W) * P(W) \right]$$



ASR:
Knowledge-Base View



Language

s

P(1/s)

One

P(2/1,s)

Two

P(3/2,1)

Three

P(s/3,2)

P(3/1,s)

Three

P(5/3,1)

Five

P(s/5,3)

/s

P(7/3,1)

Seven

P(s/7,3)

> 1M

Words

W

A H

N

> 100k

Phonemes

S0

S1

S2

> 50k

Acoustics

4k-8k

inner-most loop

ASR:
Knowledge-Base View (GMM)

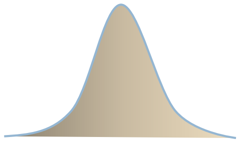
2M – 80M

□□

Equation $\frac{1}{\sqrt{(2\pi)\ \ |\sigma^2|}}\exp\left({(x-\mu)^2}/{2\sigma^2}\right)$ **2**

*

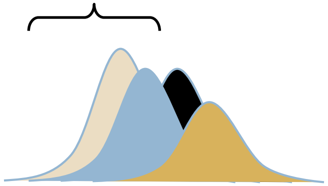
Dimensions



39

*

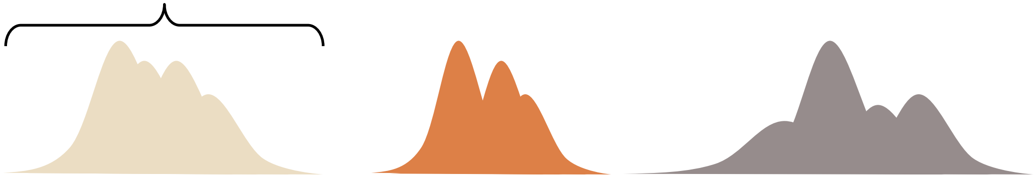
Mixtures



8-128

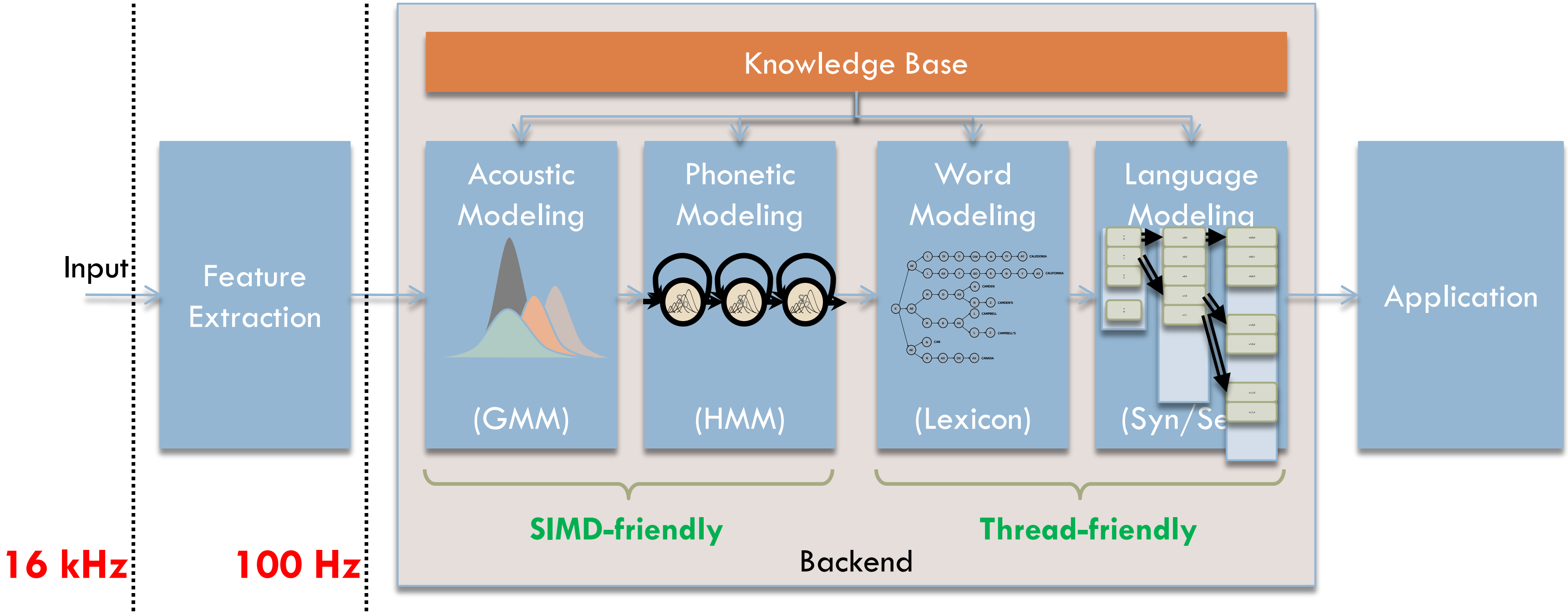
*

Acoustics



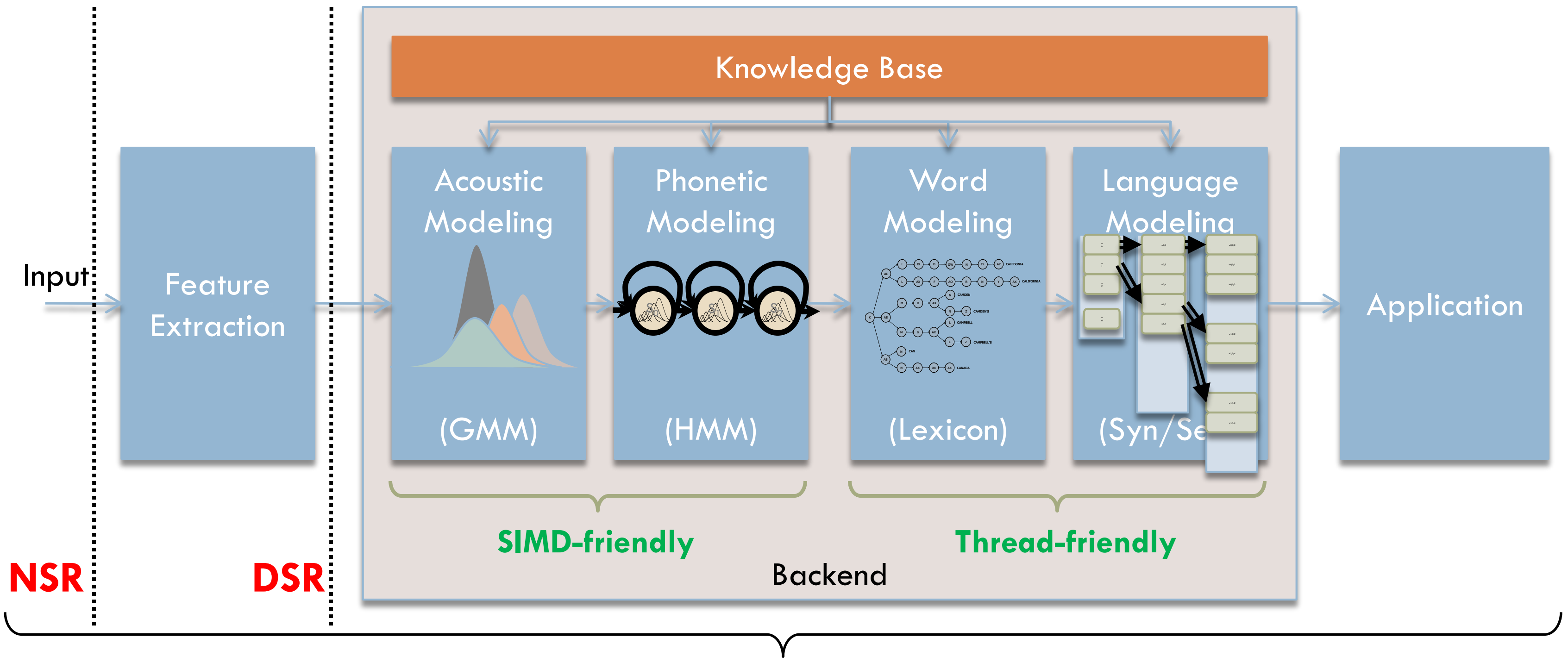
4k-8k

ASR:
Block Diagram View



ASR: State-of-the-art, Today

- Offload processing to the **'cloud'**
  - **Drawbacks:** Latency, Accuracy, Power



ESR ← **The Challenge**

NSR/DSR are the only solution today for supporting ASR on embedded devices

Characterization of ASR algorithms

		Frontend		Backend			
		Feature Extraction		**Acoustic Modeling**		**Language Modeling**	
Core kernels		FFT, DCT		GMM computation & HMM state traversal		Layered graph search	
Memory	Footprint	Very small	++	Medium	+	Very large	- -
	Bandwidth	Low	++	Very high	- -	Medium	+
	Access pattern	N/A		Spatial locality (for mini-datasets)	+	Temporal locality (non-sequential)	+
Compute		Very low	++	Very High	- -	Low	++
Data-structure		N/A		Regular: Dense	+	H. irregular: Sparse	- -
Time	System	< 1%		**50-90%**		10-50%	

Focus of this talk

Bottleneck

Application Domains for ASR

*anything not plugged into the power socket

		Server	Desktop	Embedded*
		Off-line & On-line	On-line & Off-line	On-line
Real-Time constraint		N/A & Soft	Soft	Hard
Application domain		Transcription	Desktop control	Search
		Data mining	Dictation	Dictation
		Customer support	Game consoles	SMS/Chatting
		Distributed Speech Recognition	Home automation (multi-stream)	Command & Control
			Data mining	Automotive
Hardware	#	10s-1,000s + CPU/GPU	CPU + GPU	CPU + GPU + acc. Si
	Compute	PFLOP	TFLOP	GFLOP
	Memory	~ (TB/PB)/s	~ GB/s	~ (GB/MB)/s
Vocabulary size		1M +	~ 50k	10+

The Challenge

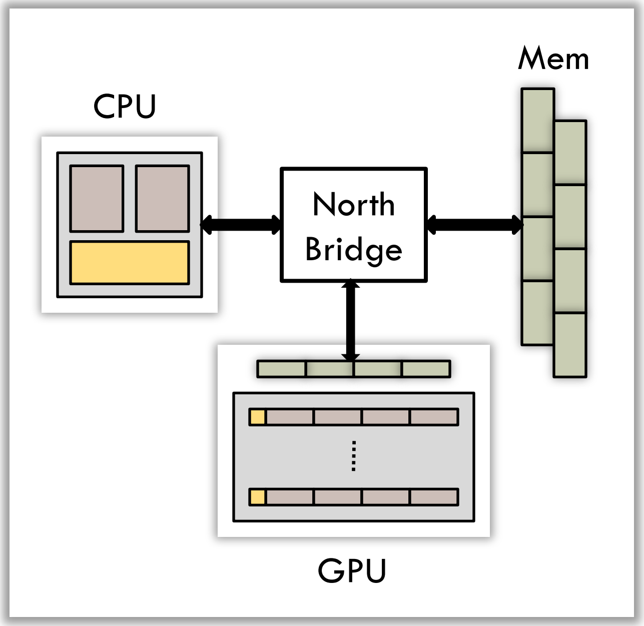
*anything not plugged into the power socket

		Server	Desktop	Embedded*
		Off-line & On-line	On-line & Off-line	**On-line**
Real-Time constraint		N/A & Soft	Soft	**Hard**
Application domain		Transcription	**Desktop control**	**Search**
		Data mining	**Dictation**	**Dictation**
		Customer support	**Game consoles**	**SMS/Chatting**
		Distributed Speech Recognition	**Home automation (multi-stream)**	**Command & Control**
			Data mining	**Automotive**
Hardware	#	10s-1,000s + CPU/GPU	CPU + GPU	CPU + **GPU** + acc. Si
	Compute	PFLOP	TFLOP	**GFLOP/MFLOP**
	Memory	~ (TB/PB)/s	~ GB/s	**~ (GB/MB)/s**
Vocabulary size		1M +	**~ 50k**	**10+**

"Desktop-class ASR on Embedded devices"

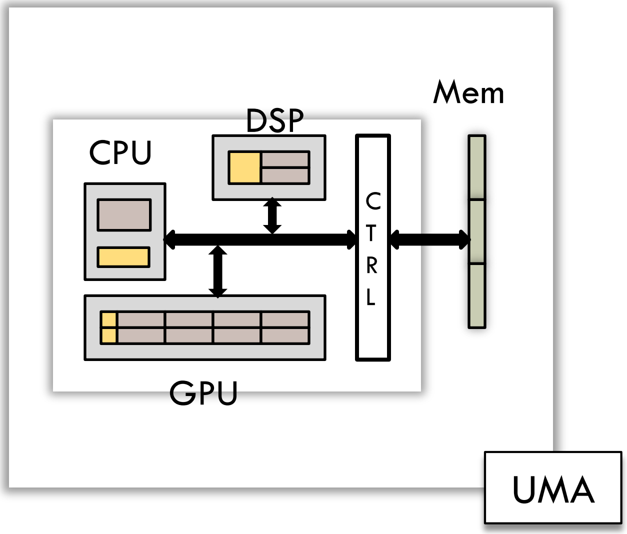
The Challenge:
Desktop v/s Embedded System Architectures

Desktop System Architecture



- Processor
- Cache
- Memory

Embedded System Architecture



	Desktop (480GTX)	Embedded (9400M)
# of SMs	16 x 32	2 x 8
Compute	TFLOP	**GFLOP**
Memory	~ 100’s of GB/s	**< 10 GB/s**
	Discrete	Integrated

Vastly different architectures & constraints: **Memory & Compute** resources are limited

Design Goals

- Target : GeForce 9400M
  - # of SMs: 2
  - Shared memory: 16kB/SM
  - Registers file: 8k/SM
  - Compute Capability 1.1
    - Stringent memory coalescing constraints
  - OpenCL-capable
- Speed : Faster than real-time
- Accuracy: Any optimizations should impact accuracy 'marginally'

- HOW?
  - Re-visit traditional ASR pipeline
  - Extract intra-module parallelism!

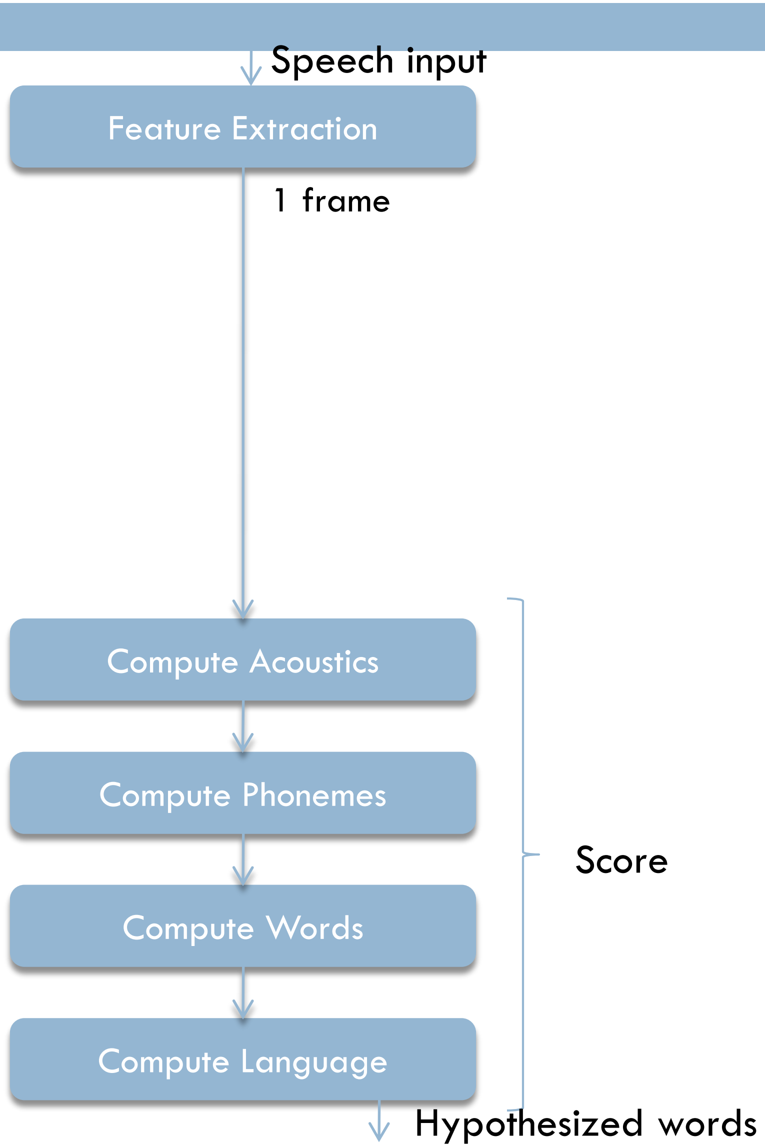
Design Principles: CPU v/s GPU (1)

- #1
  - CPU: Dynamisim is fine; remove every state that is not needed
  - GPU: Regular structure, consistency important; extra work OK
    - Compute is cheap, main memory accesses are expensive
    - Static; memory allocation/de-allocation user-managed
- #2
  - CPU: Branches are fine; HW support
  - GPU: Branches may lead to serialization
    - Carefully organize your data-structures
    - Avoid branches and reduce access to branch-able code

Design Principles: CPU v/s GPU (2)

- #3
  - CPU: Repetitive computation over time is OK
  - GPU: Repetitive computation staggered over time has a huge cost
    - Small/non-existant on-chip memories
    - Increase ‘arithmetic intensity’ of computations
- #4
  - CPU: Multiple optimization layers are fine
  - GPU: Hand-pick few optimizations that map well to the arch.

Task List: Brute Force Feed-forward Loop



Speech input

Feature Extraction

1 frame

Compute Acoustics

Compute Phonemes

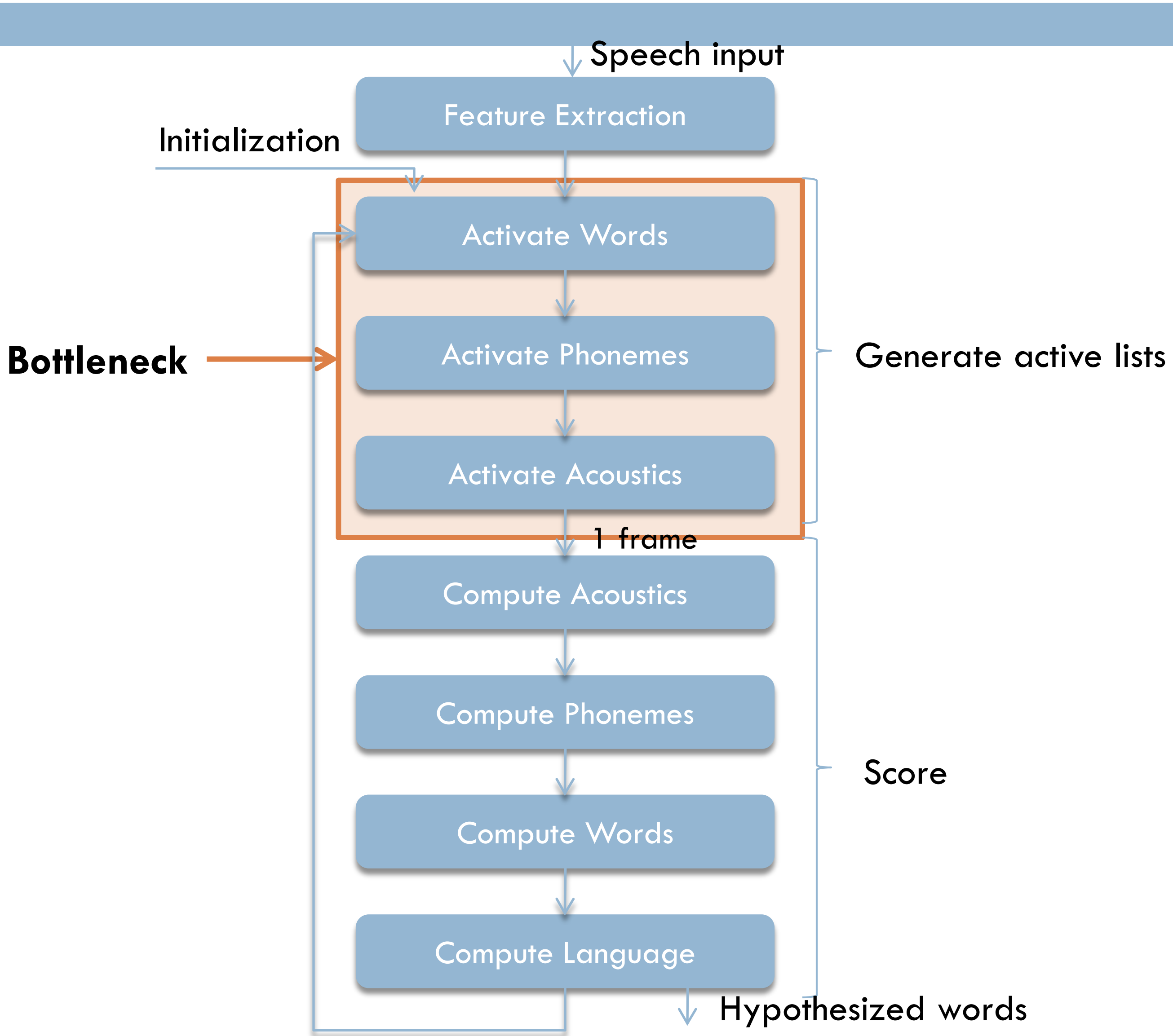
Compute Words

Compute Language

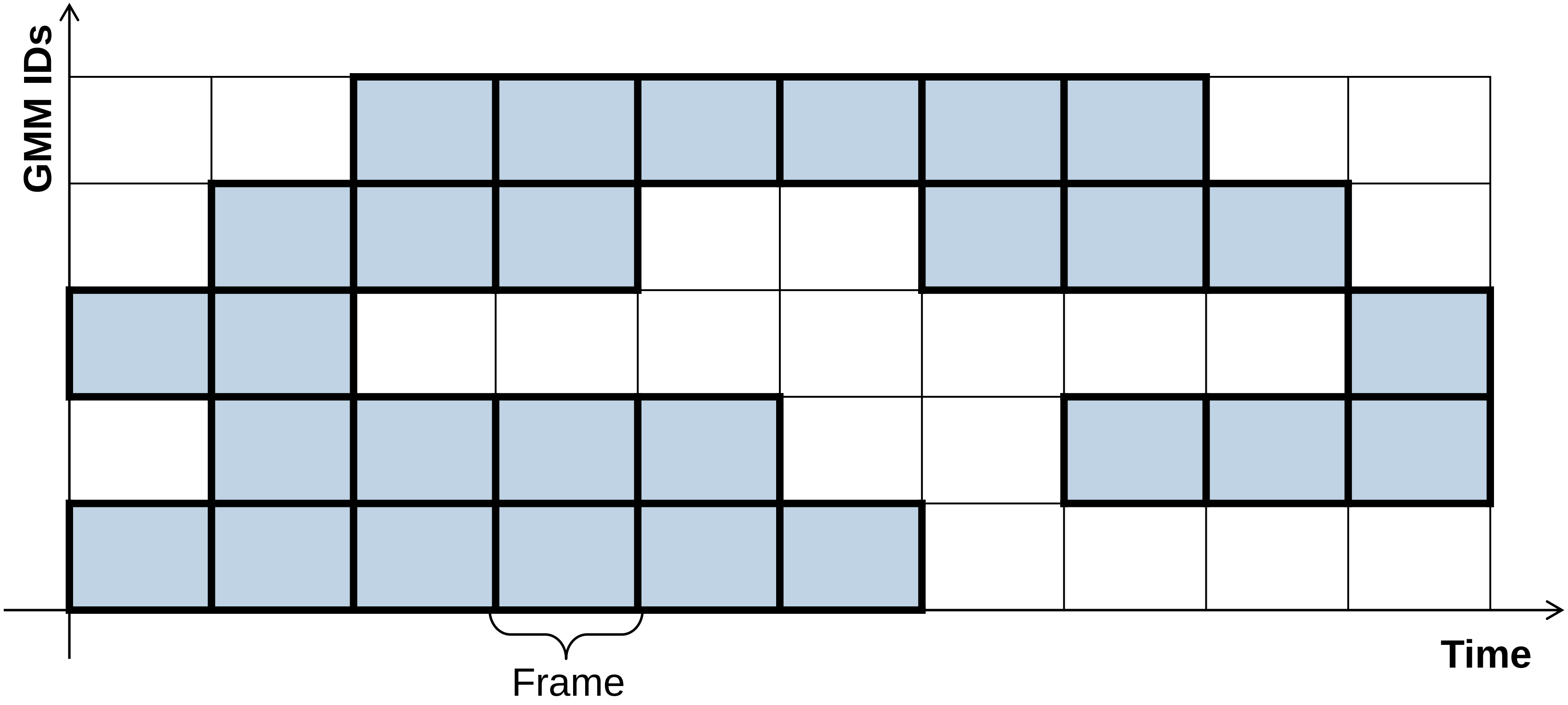
Score

Hypothesized words

Task List: Prune, prune, p-r-u-n-e Feedback Loop

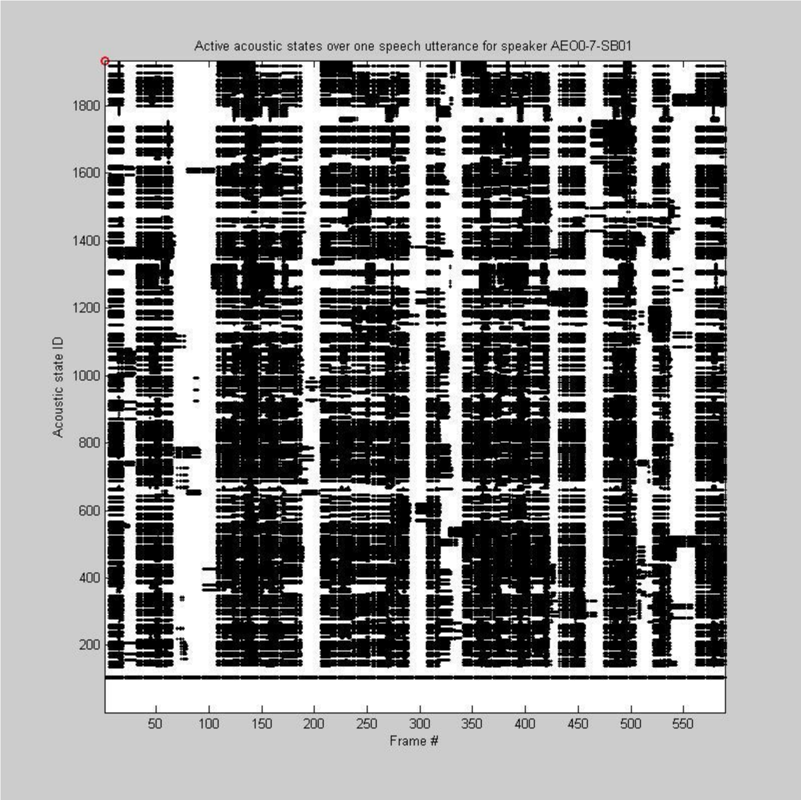


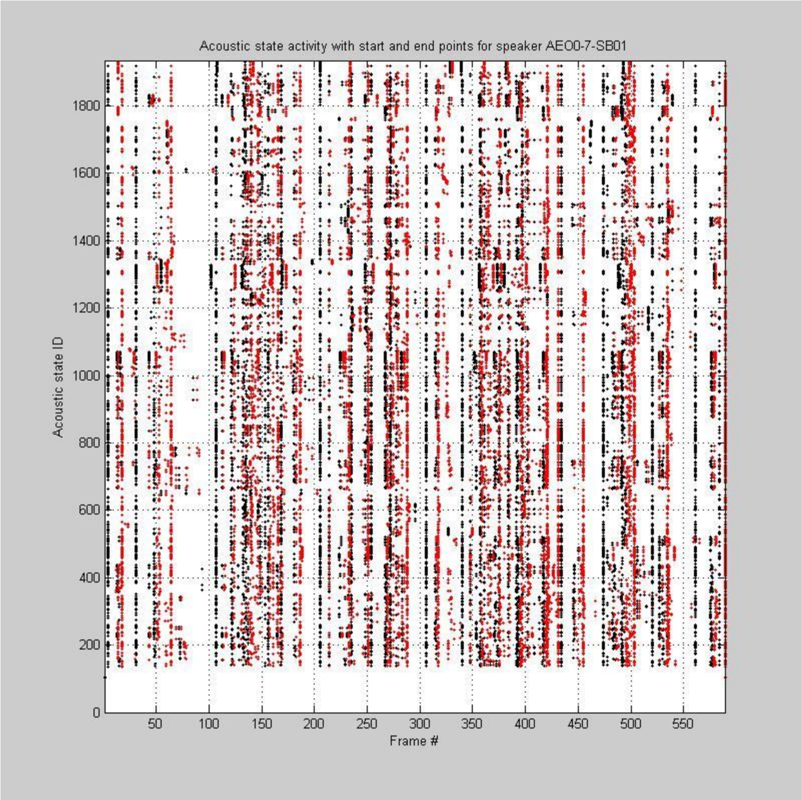
Active Acoustics



Memory bandwidth intensive

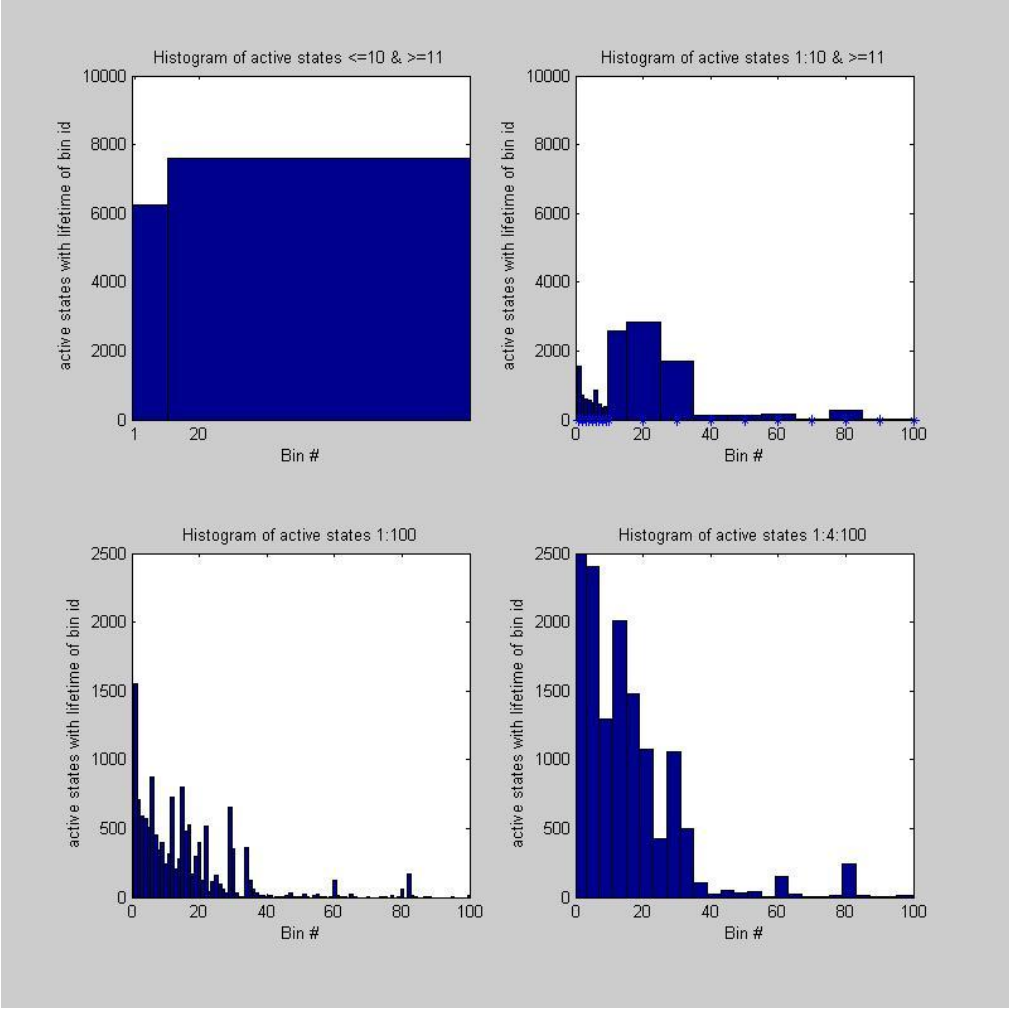
Active Acoustics: Observation (1)



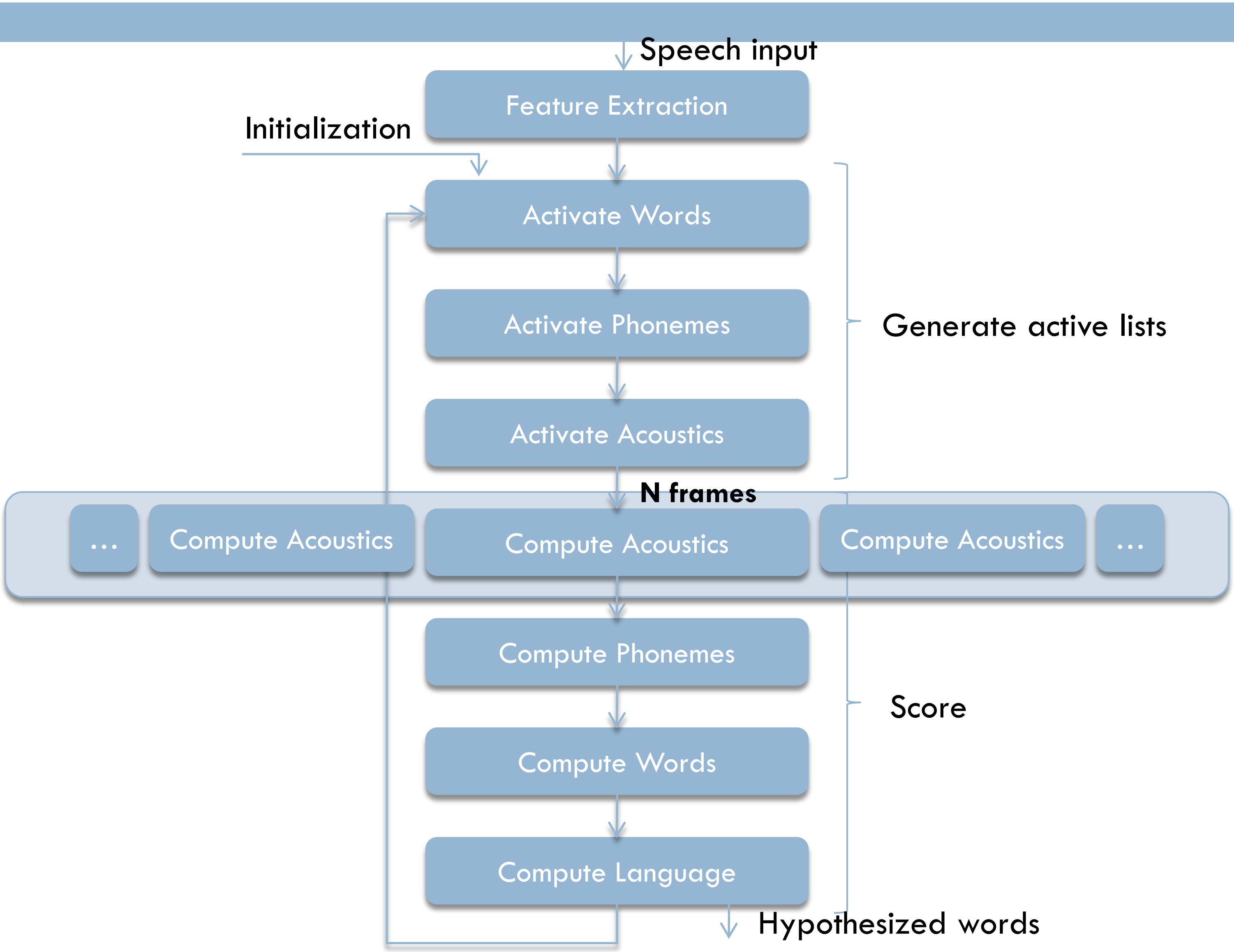


“show locations and c-ratings for all deployed subs that were in their home ports april five"

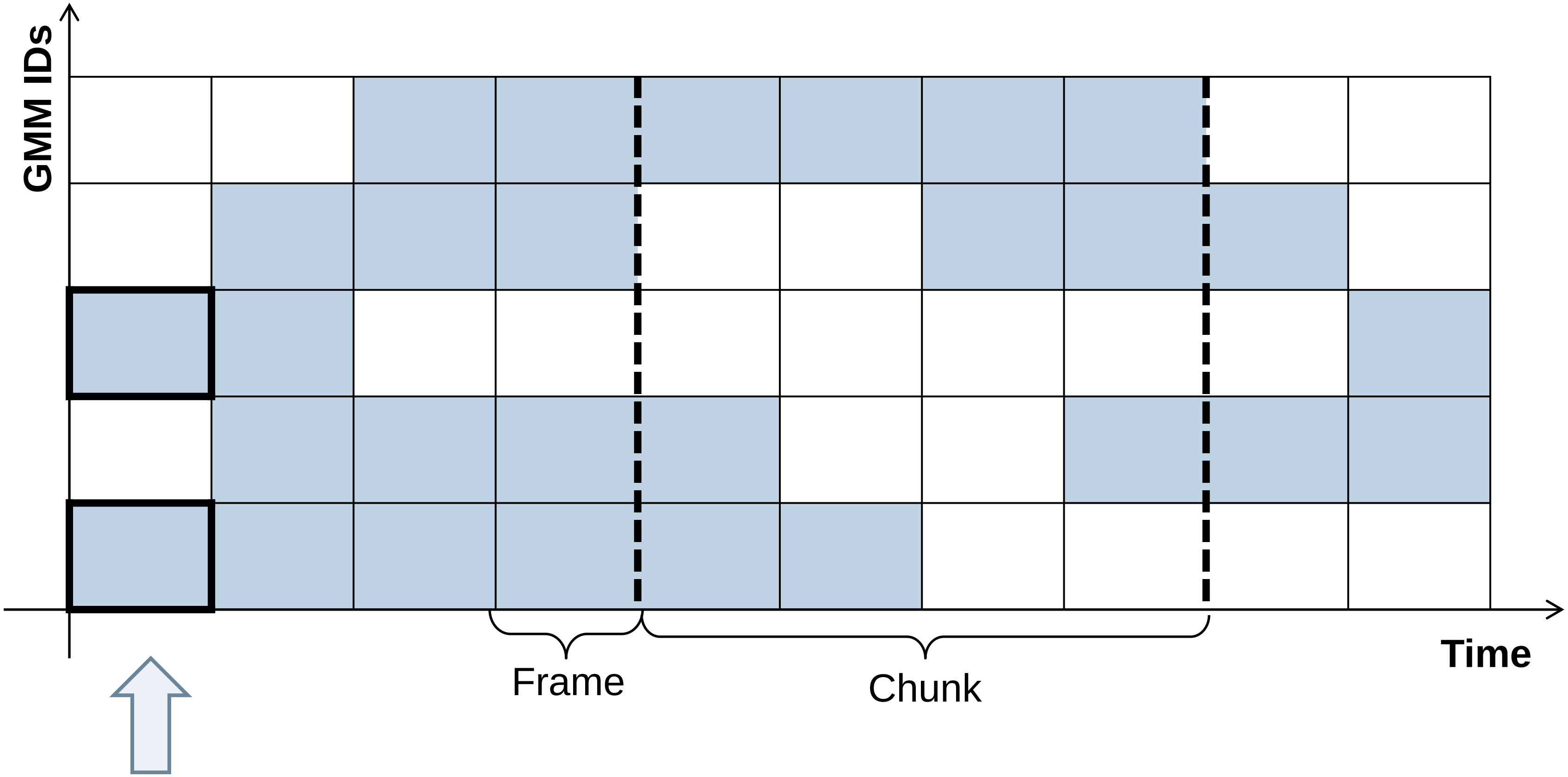
Active Acoustics: Observation (2)



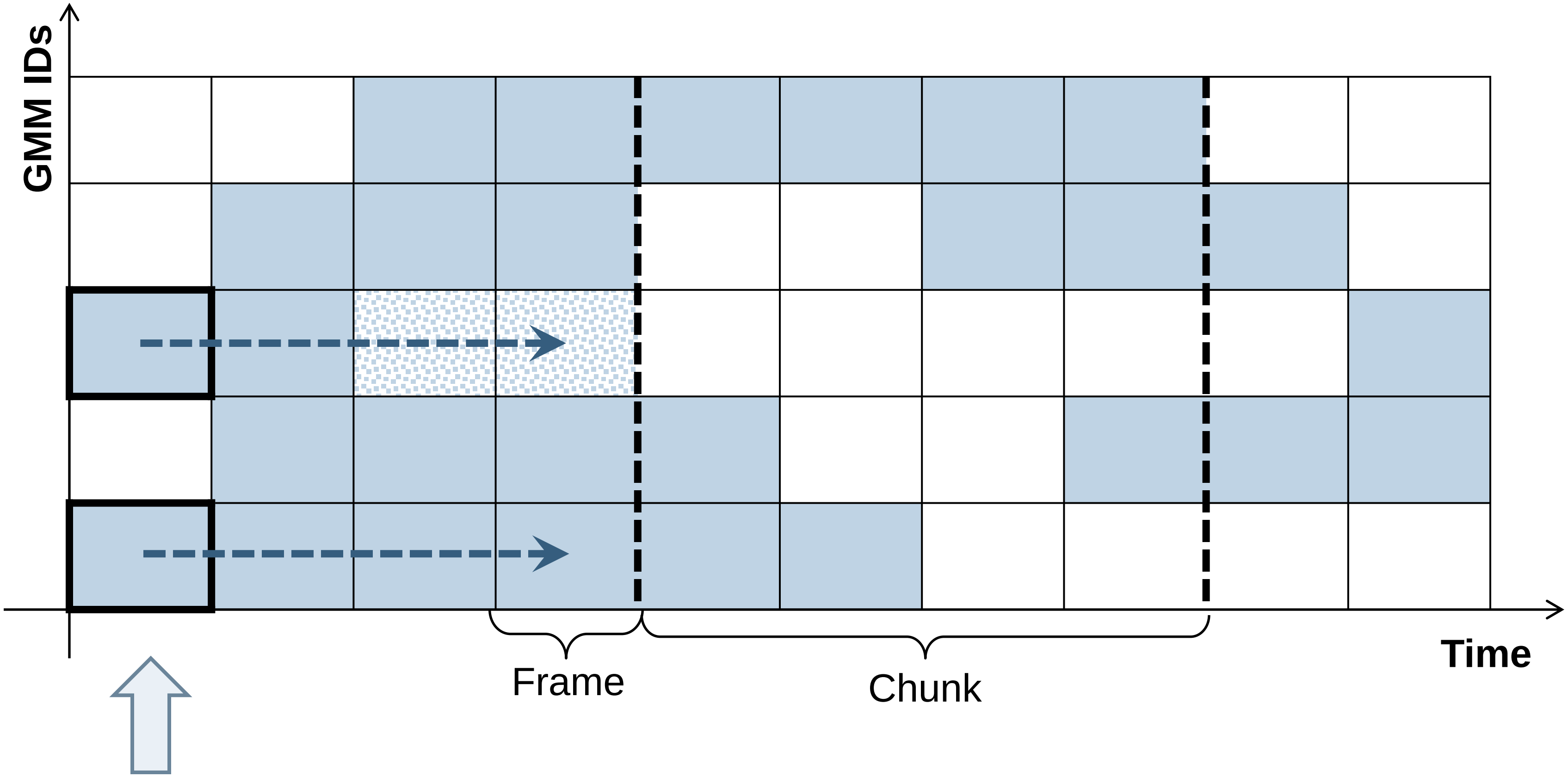
Solution: Feedback (w/ intra-module parallelism)



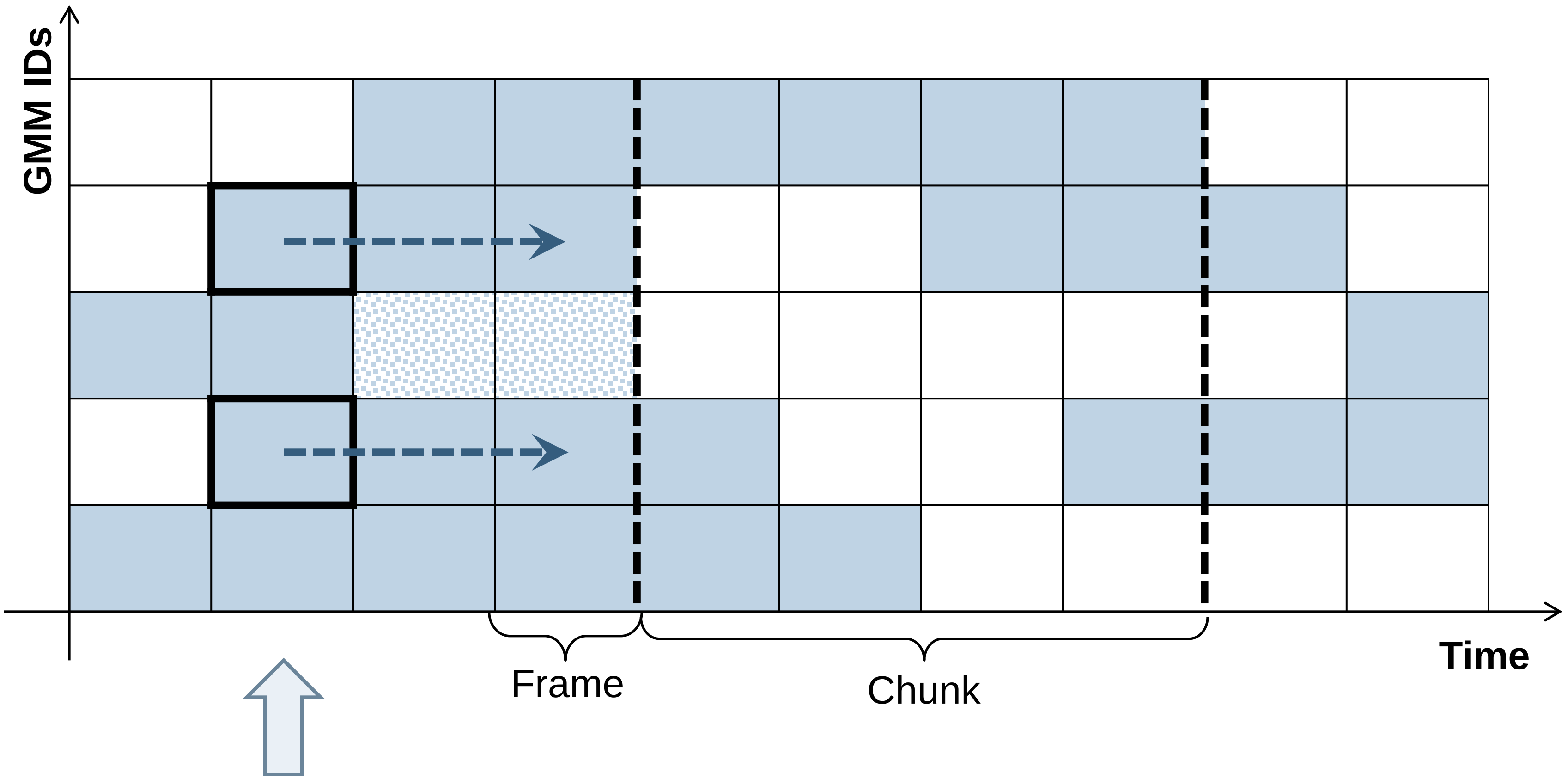
Acoustic Model Look-ahead: Frame #1



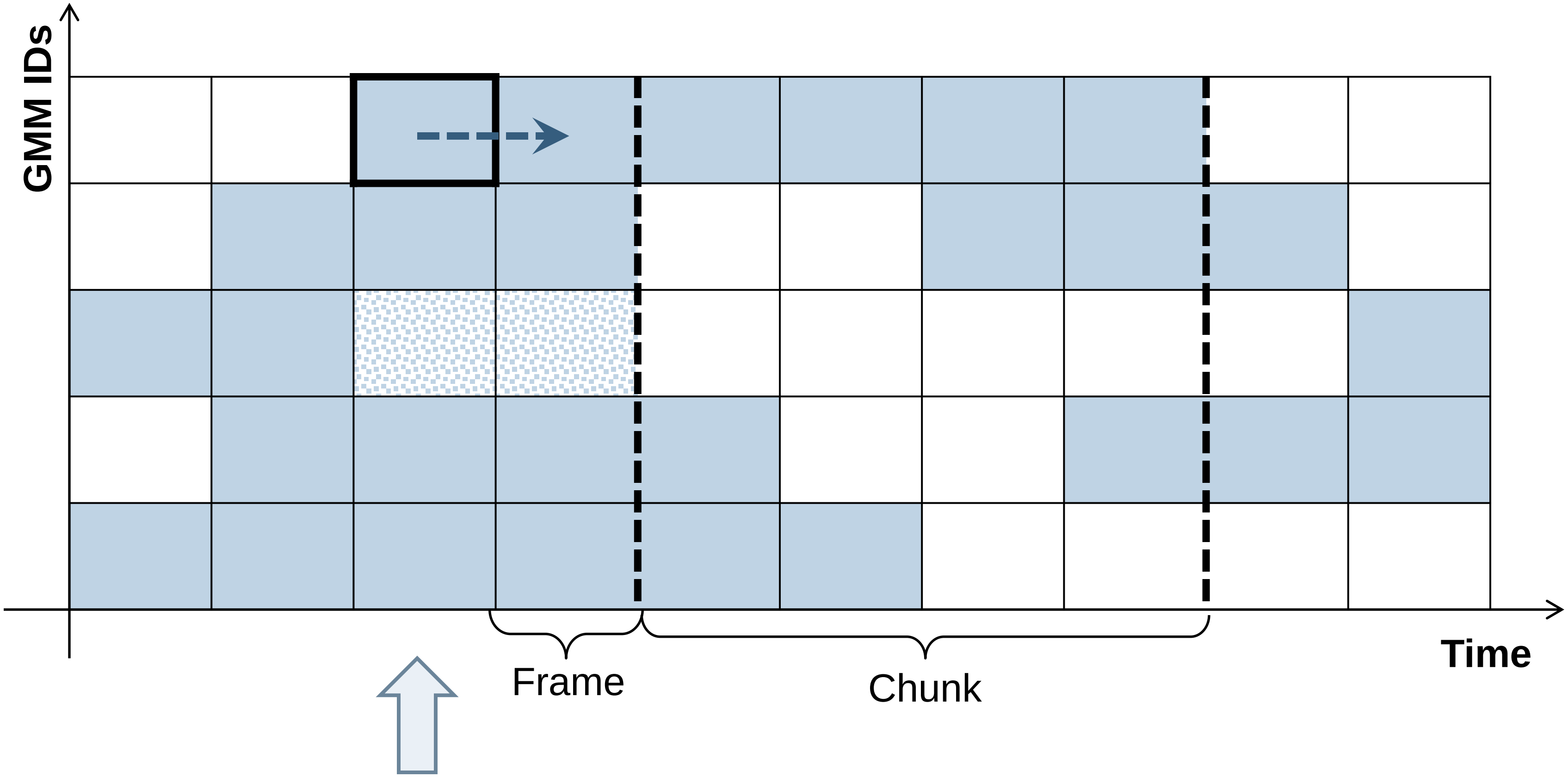
Acoustic Model Look-ahead: Frame #1



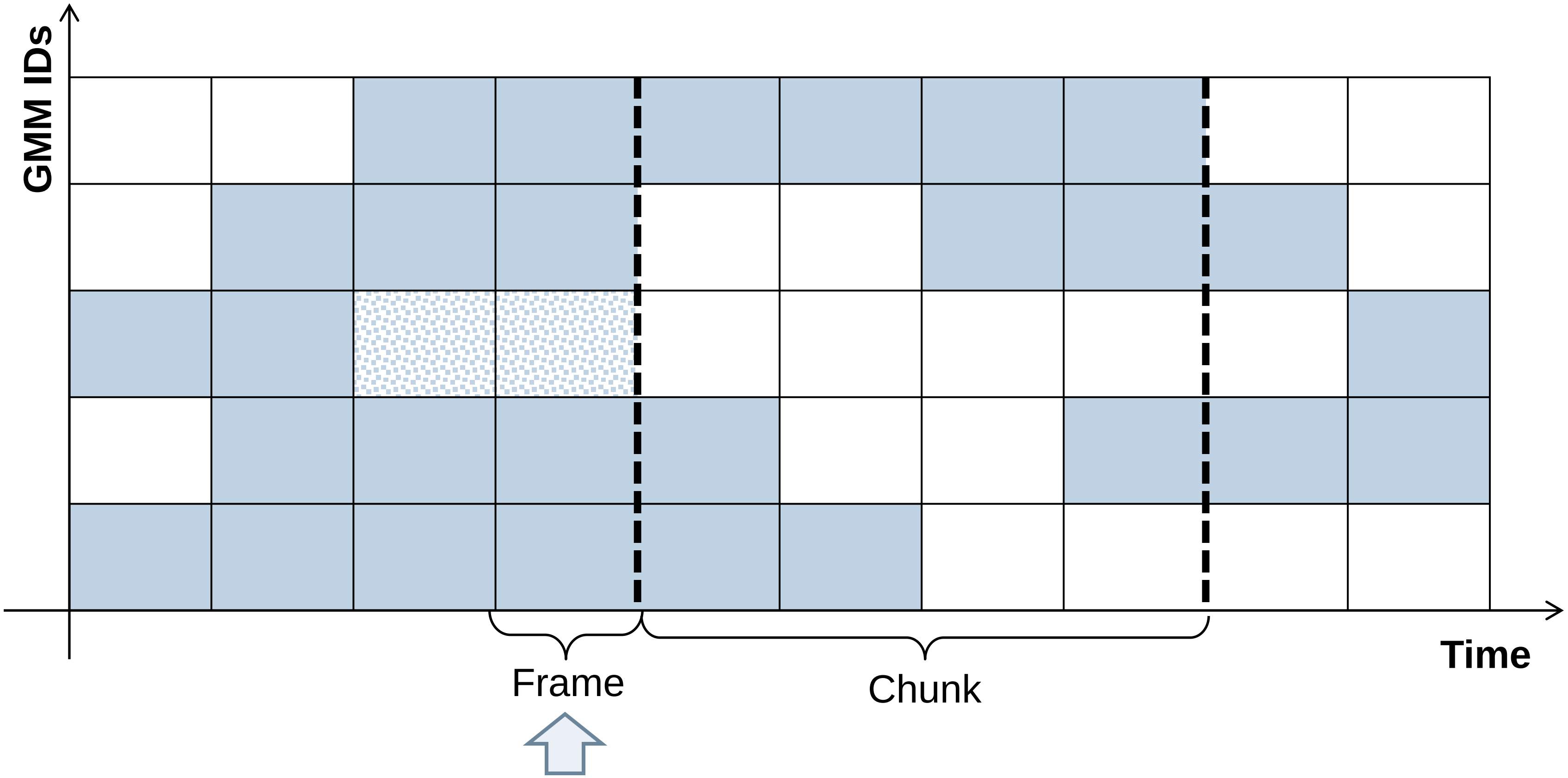
Acoustic Model Look-ahead: Frame #2



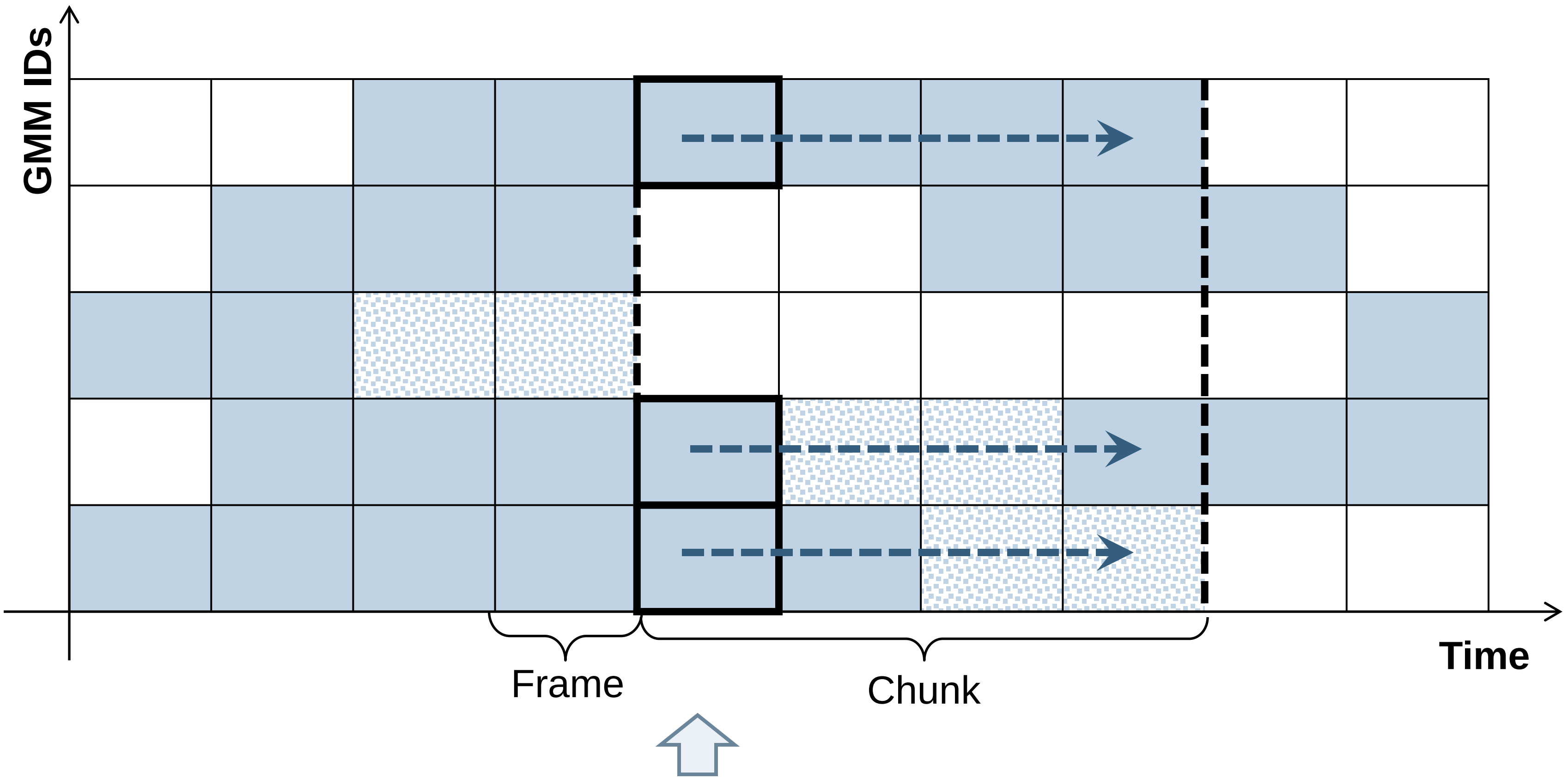
Acoustic Model Look-ahead: Frame #3



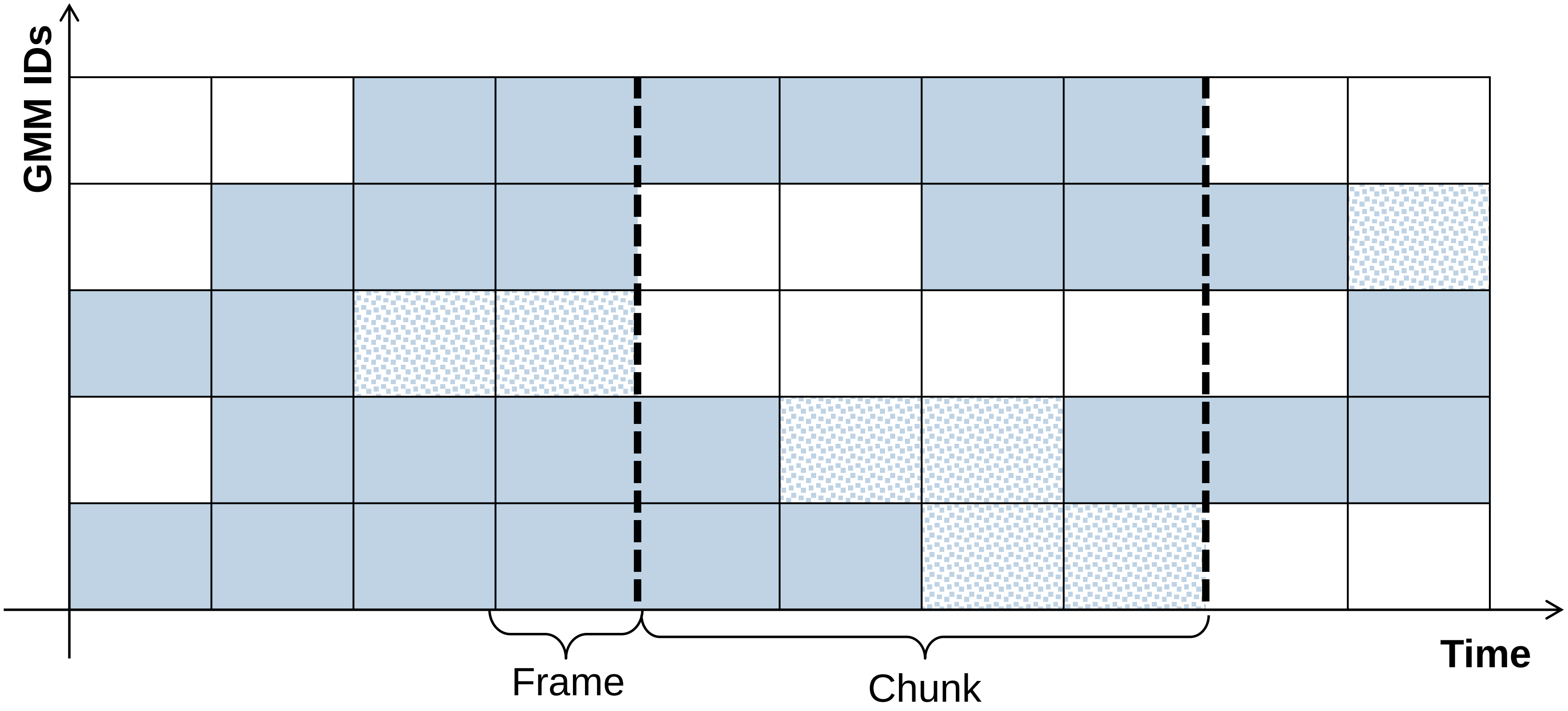
Acoustic Model Look-ahead: Frame #4 (do nothing)



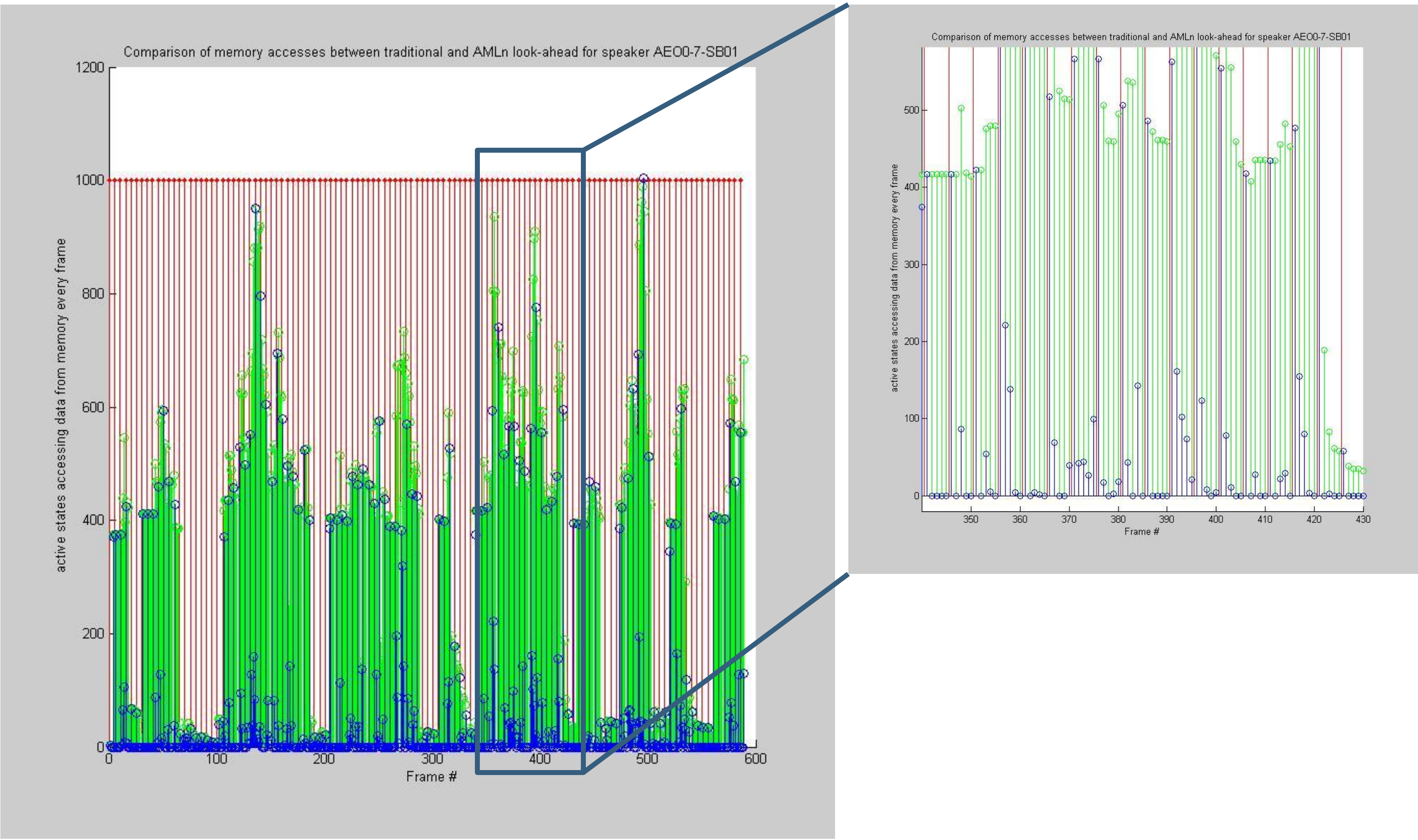
Acoustic Model Look-ahead: Frame #5



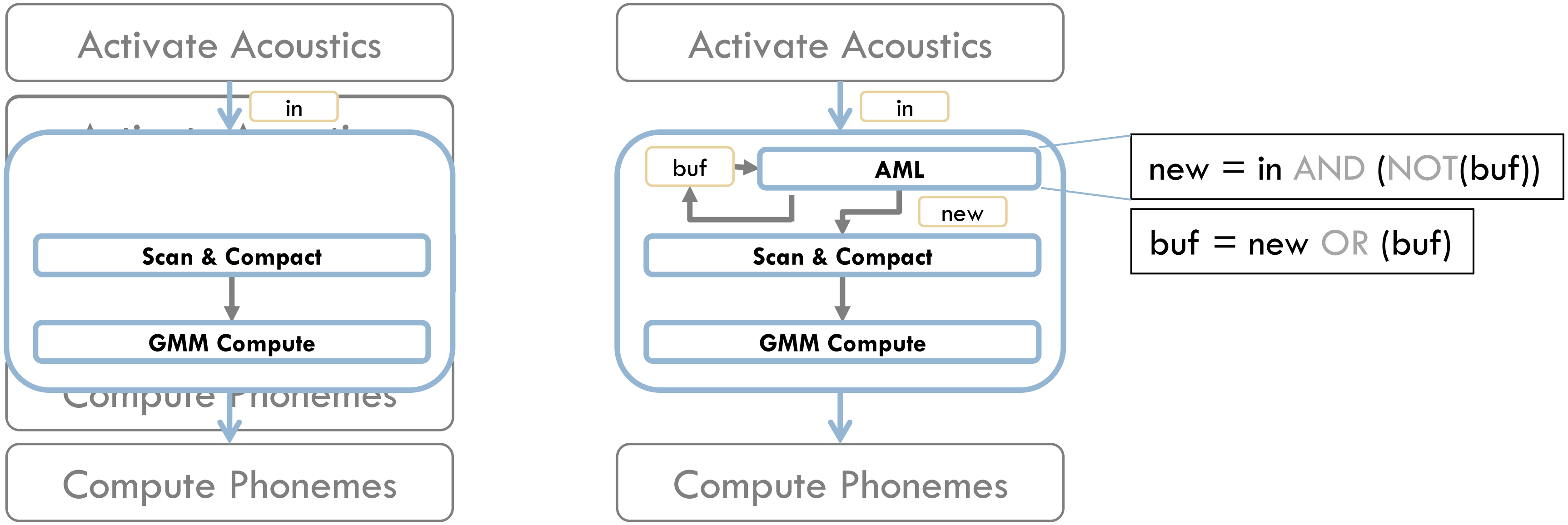
Acoustic Model Look-ahead: All Frames



Result:
Significant savings in Memory Bandwidth



Acoustic Model Look-ahead (#1)



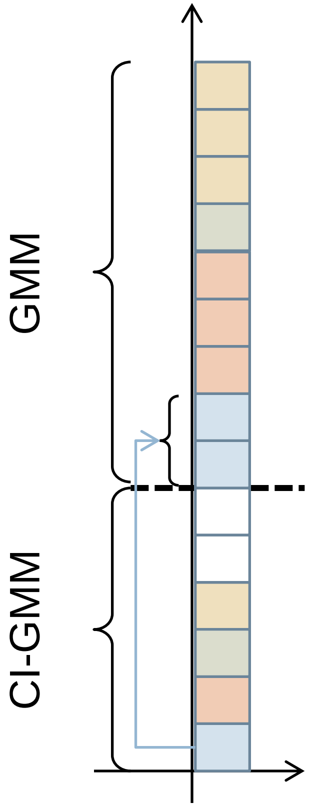
new = in AND (NOT(buf))

buf = new OR (buf)

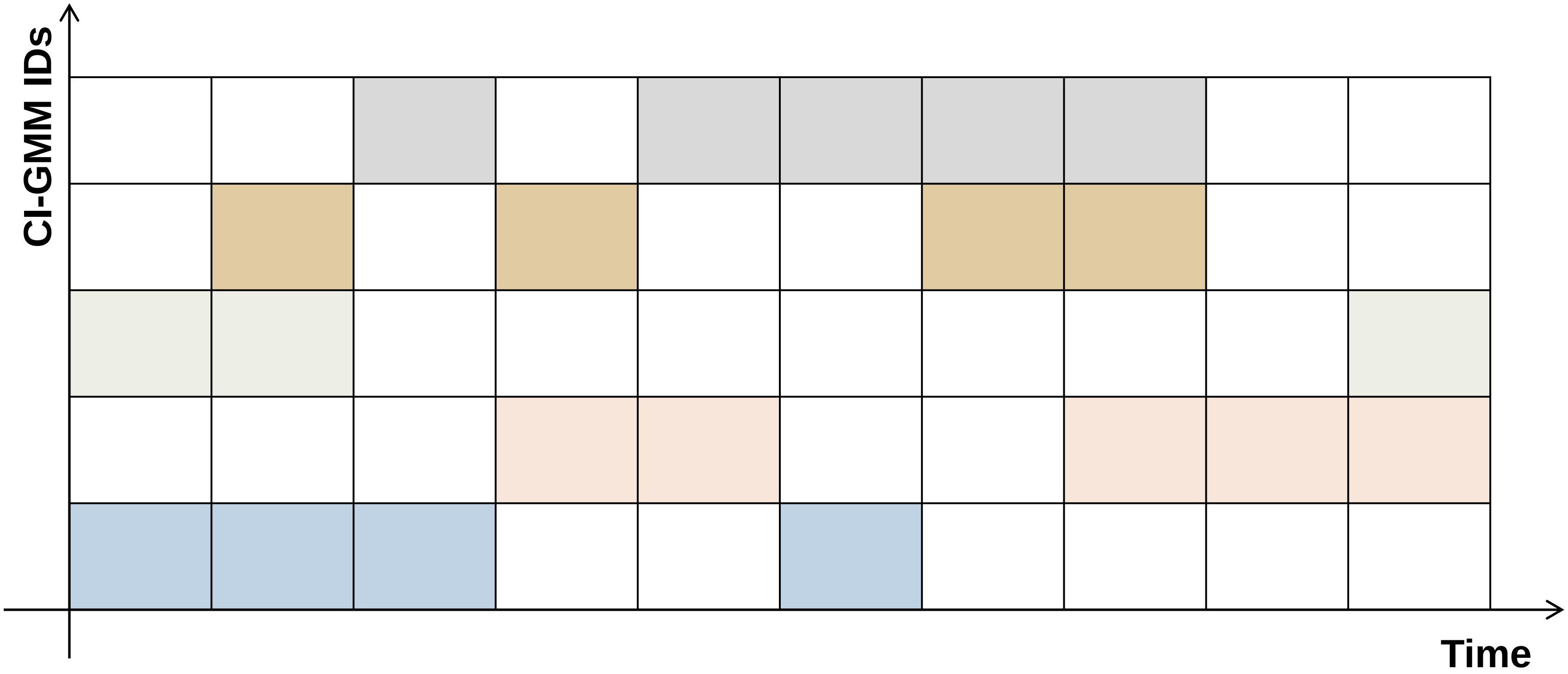
Results

Chunk	WER	Comp. Ovrd (%)	BW Saved (%)	RTF 260 GTX	RTF 9400 M (ION)	
1	6.86	0	0	14.38	**1.50**	**360 MB**
2		3.46	43.76	20.30	**2.70**	
4		9.76	67.46	25.34	**3.27**	
8		**20.64**	**79.90**	**32.36**	**3.96**	**70 MB**

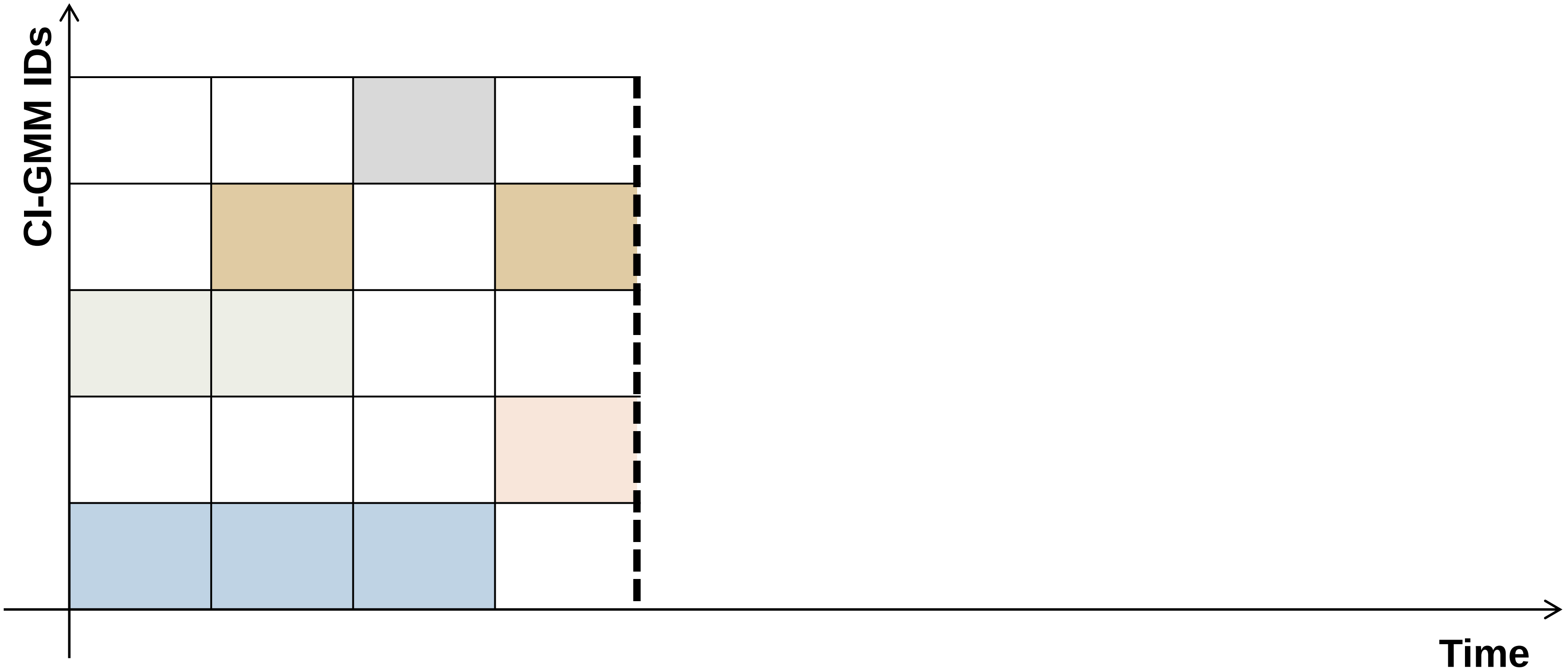
Context-Independent Acoustics



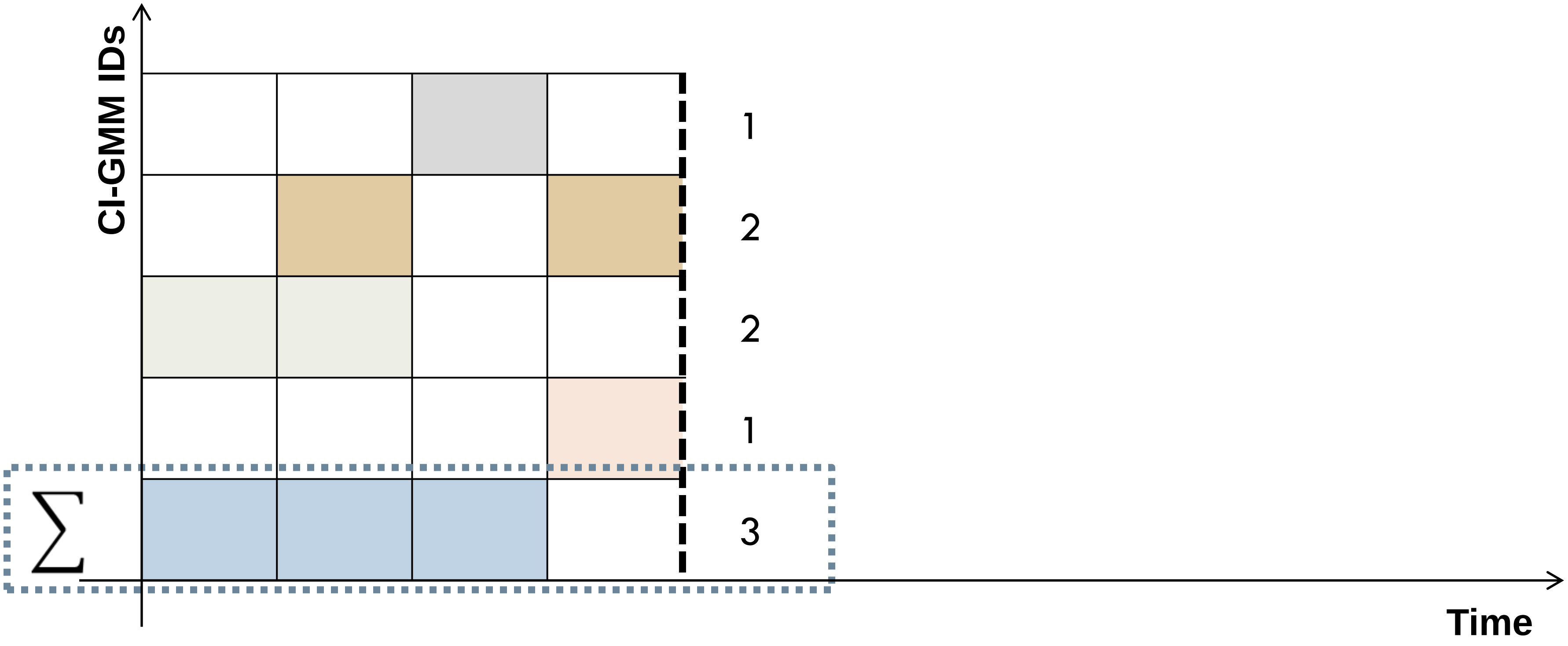
Context-Independent Acoustics: Lifetime



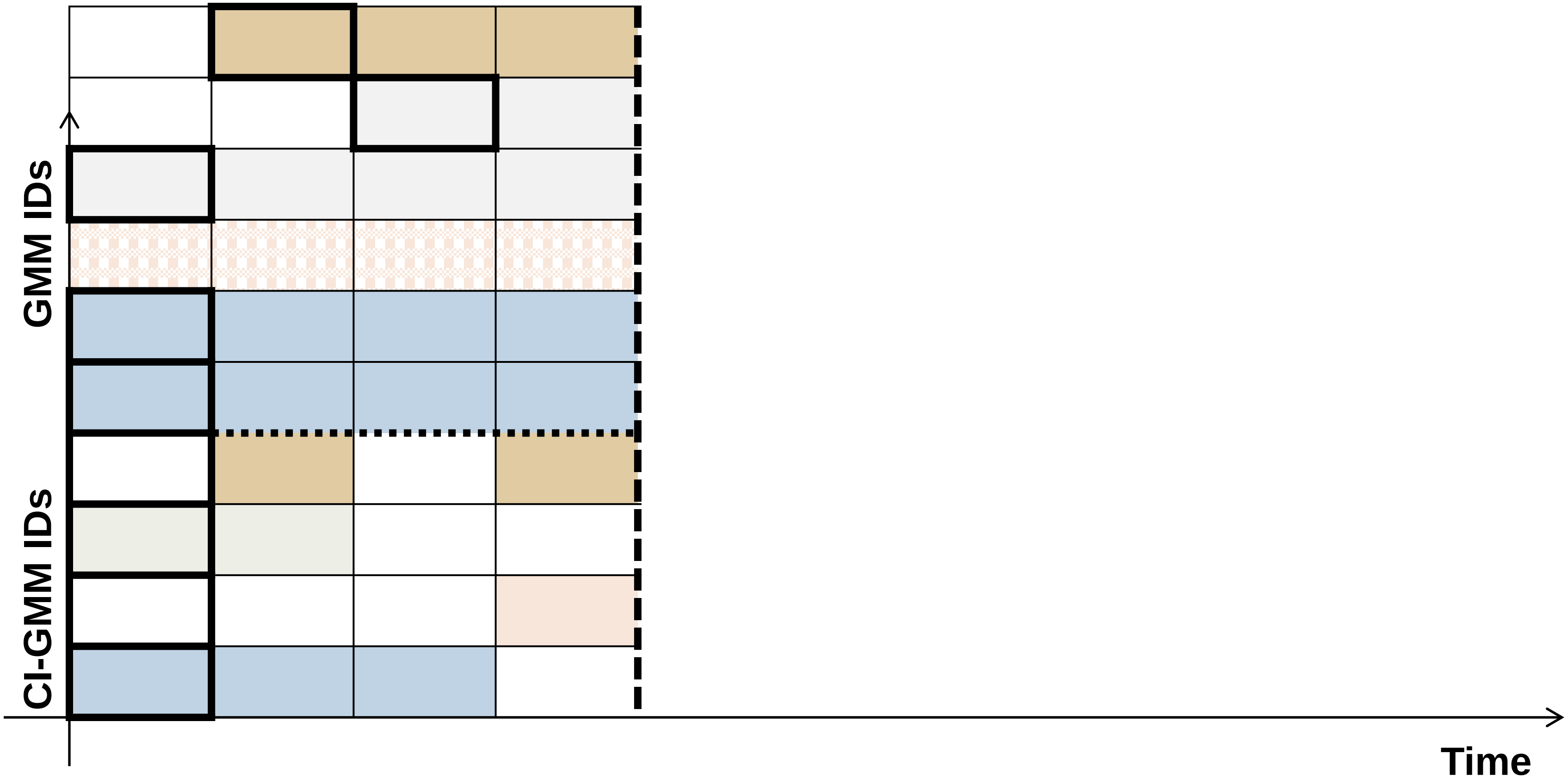
Context-Independent Acoustics: Chunk-based processing



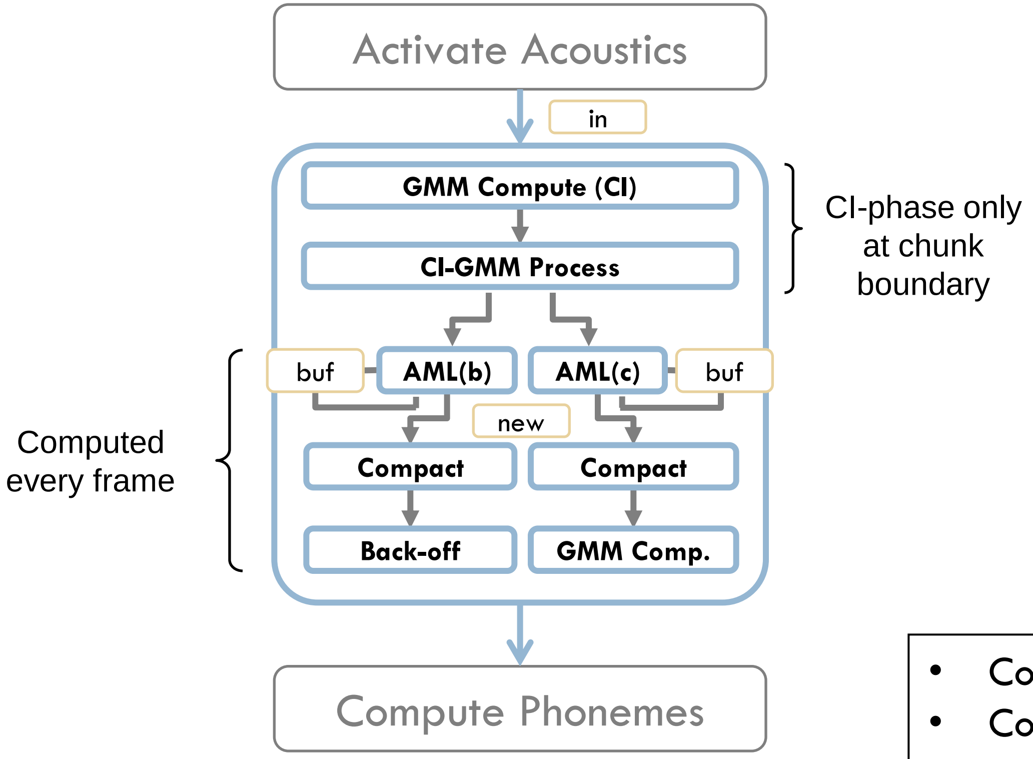
Context-Independent Acoustics: Chunk-based processing



Context-Independent Acoustics: Chunk-based processing



Acoustic Model Look-ahead (#2)



- Compute CI-GMMs
- Compute Maximum for beam pruning

- If (CI-GMM > CIGMM Threshold) {
  score corresponding GMMs
  }

Results

Chunk	CI-GMM Thresh	WER	Comp. Saved (%)	BW Saved (%)	RTF 260 GTX	RTF 9400 M (ION)
4	1	7.27	24.04	79.47	23.52	**4.32**
4	2	7.72	36.81	82.95	24.93	**4.85**
4	3	8.67	48.81	86.21	26.58	**5.40**
8	1	7.23	11.78	86.05	33.23	**4.95**
8	2	7.31	23.57	87.75	34.68	**5.37**
8	**3**	**7.81**	**34.05**	**89.27**	**36.25**	**6.18**

36 MB

Faster than real-time; with savings in both compute & memory bandwidth

In Summary

- High-end & Low-end systems vastly different in
  - Architectures
  - Constraints
- Re-visit traditional application pipeline
- Memory is a key bottleneck
  - Extraction of temporal locality is critical

- Acoustic Modeling Look-ahead is ‘critical’ in …
  - Enabling faster than real-time performance
  - Saving bandwidth
  - Saving compute
  - … at a marginal loss in accuracy

Future Directions

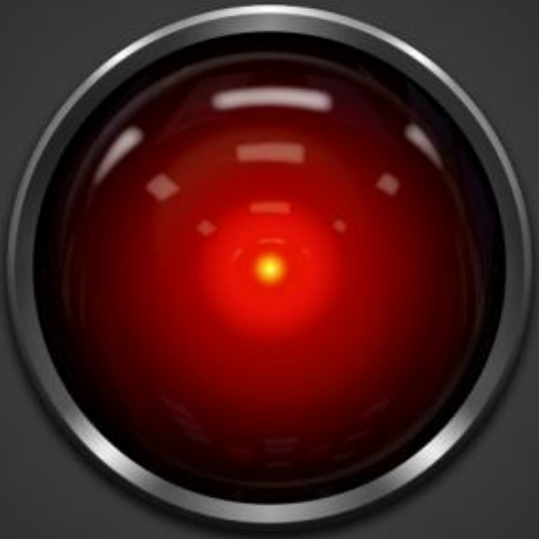
We're just getting started…

- Multi-stream Speech Recognition
  - Home automation
- Transcription
  - Minutes of meetings
- Language Translation
  - Tour guides
- Today's killer-app
  - Dictation!
- …

The Final Frontier in Speech Recognition...

- The Holy Grail
  - accurate
  - real-time
  - continuous
  - naturally spoken
  - noisy conditions
  - large set of words
  - speaker-independent

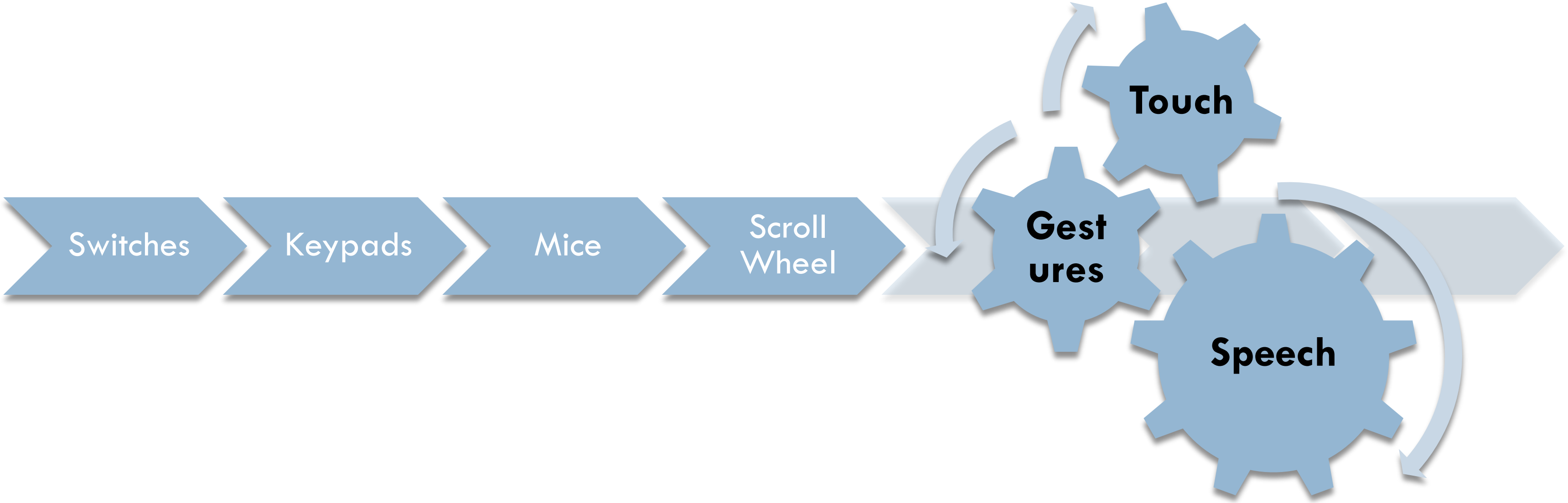
- **Using speech recognition not just for a few selective, non-critical tasks, but for all tasks, including 'mission-critical' ones.**



HAL 9000

“Perfect” voice-driven interfaces are not possible with today’s algorithms

The Future: ‘Complimentary’ UIs!



Thank You