 DOI 10.1109/TC.2020.2991002

LAWS: Locality-AWare Scheme for Automatic Speech Recognition

Reza Yazdani, Jose-Maria Arnau, and Antonio González, *Fellow, IEEE*
Department of Computer Architecture, Universitat Politècnica de Catalunya, Barcelona, Spain
Email: {ryazdani, jarnau, antonio}@ac.upc.edu

***Abstract*—Automatic Speech Recognition (ASR) systems are changing the way people interact with different applications on mobile devices. Fulfilling such user-interactivity requires not only a highly accurate, large-vocabulary recognition system, but also a real-time, energy-efficient solution. However, these ASR systems need high memory bandwidth and power budget, which may be impractical for most of small form-factor battery-operated devices.**

In this paper, we propose two combined techniques implemented on top of a state-of-the-art ASR accelerator in order to significantly reduce its energy consumption and memory requirements. First, by leveraging the locality among consecutive segments of the speech signal, we develop a Locality-AWare-Scheme (LAWS) which exploits the on-chip recently-explored data while removing most of the off-chip accesses during the ASR's decoding process. As a result, we remove up to 60% of ASR's workload.

As the second step, we introduce an approach to improve LAWS's effectiveness by selectively adapting the amount of ASR's workload, based on run-time feedback. In particular, we exploit the fact that the confidence of the ASR system varies along the recognition process. When confidence is high, the ASR system can be more restrictive and reduce the amount of work. The end design including both techniques provides a saving of more than 87% in the memory requests and 2.3x reduction in energy consumption, and a speedup of 2.1x with respect to a state-of-the-art baseline design.

***Index Terms*—Automatic Speech Recognition (ASR), Viterbi Beam Search, Hardware Accelerator, WFST, Memory-Efficient, Low-Power Architecture.**

I. Introduction

After achieving human parity in speech recognition [1], a main focus of Automatic Speech Recognition (ASR) systems is turning towards mobile, wearables and IoT devices. Such a high degree of accuracy comes at the expense of huge memory requirements and computational cost in terms of performance and energy [2], which is unaffordable in most of these devices. An ASR system traverses a huge graph-based recognition model that contains millions of states and arcs in order to decode the speech signal by searching among different alternatives to find the most likely sequence of words [3]. Several accelerators have been recently proposed to target some challenges of ASR [2], [4]–[9]. However, the main challenges of high energy-consumption and memory-bandwidth requirements still remain when deploying ASR in small form-factor battery-operated devices. [10]

Even though the End-to-End (E2E) ASR models based on standalone RNN or CNN are getting popular since they simplify the overall pipeline, the Kaldi's hybrid system [11] composed of DNN, Viterbi beam search, and RNN rescoring still achieves higher accuracy, especially for noisy audio. Furthermore, E2E systems also require a beam search based on a language model to achieve accuracy comparable to hybrid systems [12]. We compared the accuracy of Kaldi's ASR decoder [11] with different E2E systems: Baidu's Deep-Speech2 [12], a state-of-the-art LSTM network with attention mechanism [13] and Facebook's wav2letter++ [14]. Table I shows the Word Error Rate (WER) collected for one of our speech corpora, Librispeech, that includes several hours of challenging noisy speech.

TABLE I: WER of four ASR decoders for Librispeech dataset [11].

System	Type	WER(%)
Human	-	12.69
Kaldi's ASR (DNN + Viterbi)	**Hybrid**	**10.62**
DeepSpeech2 [12]	End-to-End	13.25
Deep bLSTM with attention [13]	End-to-End	12.76
wav2letter++ [14]	End-to-End	11.24

A state-of-the-art speech recognition accelerator requires an average memory bandwidth of 16 Gb/s [2] in order to run ASR on different speech corpora. On the other hand, IoT and wearable devices normally use low-power memory technologies with limited throughput, such as NAND/NOR flash memories [15]. As reported by Micron, these memory systems can achieve a maximum bandwidth ranging from 1 to 6 Gb/s [16]. As a result, the ASR's memory management needs significant improvement in order to be deployed for these devices. Furthermore, it has been shown that each DRAM memory access consumes nearly three orders of magnitude more energy than a typical computation or on-chip memory accesses [17]. Therefore, high memory requirement is also the main energy bottleneck for ASR systems.

Each step of an ASR system processes one frame of typically 10 ms of the speech signal. For each frame, it expands multiple nodes in a speech graph, called hypotheses, to decode the speech by finding the most likely path on this graph. Each node of this graph represents a partial hypothesis, i.e. a partial sequence of words from the beginning up until the current frame. From now on, we use the term hypothesis to refer to a partial hypothesis. Considering all possible hypotheses would grow the size of the dynamically explored graph exponentially, which is not feasibly tractable. Instead, ASR systems use a beam in order to control the search space. For each frame, we keep track of the best hypothesis and those hypotheses whose

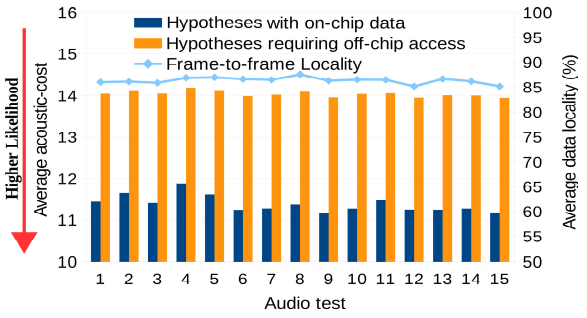


Fig. 1: The bars show the average acoustic-cost of hypotheses grouped based on their data availability on-chip. We also show the percentage of states and arcs that are the same in two consecutive frames (data locality) [18].

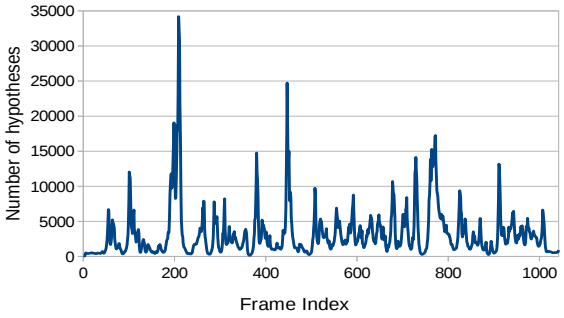


Fig. 2: The number of hypotheses along the frames of an audio test containing 1024 frames (10.42 seconds of speech) for the default fixed beam used by Kaldi.

score (likelihood) is lower than the best score minus the beam are discarded. The beam is selected in a conservative way to obtain the desired accuracy [19], whereas a more restrictive beam would improve performance and energy. In this work, we introduce a new way of managing ASR's workload which takes into account the hypotheses' data-locality besides using a lower beam distance, in order to reach an efficient trade-off in accuracy and energy-consumption.

ASR shows high correlation for the hypotheses evaluated in consecutive frames of speech, as they share quite similar data of the speech graph [18]. Based on the observation for Librispeech Corpus [20], Viterbi search expansion exhibits nearly 86% data-locality between consecutive frames [18] (see Figure 1). Moreover, we have seen that the "correct" hypothesis at each frame, i.e. the hypothesis that ends up being part of the final answer which is not known until the end of processing each utterance, often resides in the on-chip memory. Note that the best hypothesis at each frame can be different from the correct one, since the correct one may have lower likelihood at some intermediate frames while still being in the most likely full-hypothesis from the beginning to the end of an utterance.

Furthermore, we have observed that the correct hypothesis and the best one at each frame are often the same, and even when they differ, their scores (i.e. likelihoods) are quite close. This has motivated us to propose a scheme that uses a more restrictive beam when the explored hypotheses require off-chip memory accesses. The rationale behind it is that these accesses are very expensive and based on the above observations are very unlikely to contain the correct hypotheses. In this paper, we propose a Locality-AWare Scheme (LAWS) for ASR, which dynamically adjusts the beam distance based on the hypotheses' data-locality. In other words, our scheme uses the locality and hypotheses' likelihood to decide the search strategy, unlike previous schemes that use only likelihood information. This results in a more efficient expansion on the search graph in terms of energy consumption with negligible impact on accuracy.

LAWS uses a dynamic beam adaptation policy, as stated above. This policy not only takes into account the locality of the data, besides the likelihood scores, but also the confidence of the ASR system at each frame. There are some frames for which there are many hypotheses with scores very close to the best, whereas there are fewer in other frames. When the number of hypotheses close to the best one is high, we say that the ASR system has low confidence, since there are many alternatives similar to the best one, whereas we say that the confidence is high when this number is low.

Figure 2 shows an example of how the number of hypotheses that are close to the best one (using the default fixed beam in Kaldi [19] toolkit) changes during the evaluation of the frames of one sample audio test. We can observe that there are huge variations. For some frames, we have tens of thousands of hypotheses, whereas for others, we just have a few tens. LAWS exploits this information to adapt the beam search. We have seen that when the confidence is high, the correct hypothesis tends to be closer to the best one than when the confidence is low. This suggests that for high confidence intervals, we can be more restrictive with the beam to save energy without negatively impacting accuracy.

To summarize, we propose a new approach (LAWS) for implementing the Viterbi search in ASR systems that takes into account both the architectural and statistical features of the application in order to efficiently reduce the number of explored hypotheses with negligible impact on accuracy. By applying our techniques on a state-of-the-art ASR accelerator, we obtain 2.1x speedup and 2.3x energy savings with a small area overhead of less than 0.1% in the total accelerator's design. In addition, by removing most of the memory activity thanks to discarding most of the hypotheses requiring an off-chip memory access, we save around 74% of the memory bandwidth. The main contributions of this paper are as follows:

- We propose a novel scheme called LAWS, which combines the likelihood score and the data locality of the Viterbi's explored hypotheses, in order to reduce the ASR's workload by almost 60% with an accuracy loss of 1%.
- On top of the above scheme, we develop an adaptive approach to dynamically adjust the amount of Viterbi's workload based on the search confidence for each frame.
- By combining these techniques, LAWS achieves more than 2x improvement in both performance and energy consumption, while maintaining accuracy and reducing the main memory activity by more than 7.9x.

The remainder of this paper is organized as follows. Section II provides some background on speech recognition systems. We show some analysis of different run-time ASR's characteristics in Section III. Section IV presents our baseline accelerator's design for speech recognition. Section V introduces the two new techniques to efficiently adapt the Viterbi search workload. Section VI describes our evaluation methodology and Section VII shows the experimental results. Section VIII reviews some related work and, finally, Section IX sums up the main conclusions of this work.

II. Background

State-of-the-art ASR systems employ a pipeline that includes a DNN and a Viterbi search to decode the sequence of words from speech waveforms. First, the input audio signal, divided into several frames of normally 10 milliseconds, is fed into a signal processing algorithm such as Mel Frequency Cepstral Coefficients (MFCC) [21] to encode each frame as a feature vector. Next, these feature vectors are used as the input of a DNN network, which produces the acoustic scores, i.e. the probabilities of detecting different phonemes of the language at each frame. Then, the acoustic scores are used to run a Viterbi search on either one or several Weighted Finite State Transducers (WFSTs) [22] using the DNN-processed frames. Viterbi generates a dynamic graph of all the alternative hypotheses explored during the search. Each hypothesis represent an alternative transcription of the input speech signal. Finally, a backtracking step is performed to find the best path on the graph, representing the most likely sequence of words.

A WFST is a Mealy finite state machine which represents the mapping between input and output labels and applies a weight to each transition as the probability to traverse the arc between each two states, based on an offline training phase [3]. Regarding ASR systems, different knowledge is required in order to build their WFST, including an Acoustic Model (AM) and a Language Model (LM) [23]. AM represents the pronunciation of the different words in the vocabulary of a language, whereas LM scores the different hypotheses taking into account the probabilities of alternative sequences of words according to a given grammar. In order to merge multiple WFSTs into a single unified WFST, several arithmetic operations such as composition and graph minimization [3] are required. Although the size of both AM and LM WFSTs is normally in the order of 100 MB, the offline-composed WFST becomes large, requiring more than 1 GB in practice for large vocabulary systems, since multiplicative combinations of states and arcs occur in the composition process.

In order to run the Viterbi search on the speech graph(s), two different approaches have been proposed. The first scheme that consists in using the fully-composed WFST, is common in both software [19] and hardware [4], [6], [7]. In spite of the search simplicity due to exploring only one graph to decode the speech, the search has vast memory requirements as lots of hypotheses are expanded to explore the enormous search space. Therefore, the main bottleneck of such systems is the high memory requirements, as they typically require around 10 GB/s of bandwidth [2]. Furthermore, considering the acceleration of such system, the most important energy consumer is the main memory.

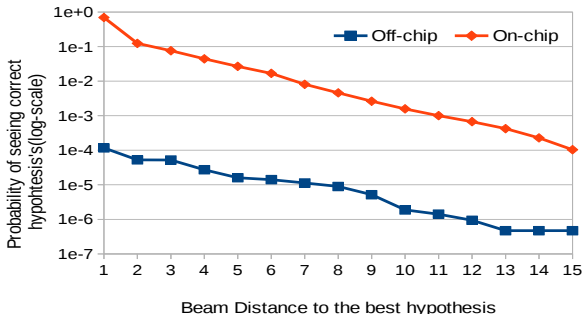


Fig. 3: The diagram shows the probability of seeing correct hypotheses at different beam distances. The distances are rounded up. Also, the likelihoods are shown based on whether or not hypotheses' data is on-chip.

As an alternative, Yazdani et al. presented UNFOLD [2], which significantly reduces the speech dataset by an average of 31x and proposed a novel accelerator architecture that performs the Viterbi search with on-the-fly composition of AM and LM. The technique of merging the separate speech graphs at run-time has been explored in several previous works [23]–[26]. A software implementation of the on-the-fly Viterbi takes an order of magnitude longer than the fully-composed [2], [27], since the composition is done during run-time whenever necessary. UNFOLD includes hardware specialized for on-the-fly WFST composition to achieve real-time performance by a large margin, while providing the benefits of reducing memory requirements by an order of magnitude and saving energy consumption by 28%.

Despite the benefits reported by UNFOLD, there is still one main requirement which prevents it to be entirely applicable for the small form-factor wearable and IoT devices: the memory bandwidth of almost 16 Gb/s which is much higher than the peak bandwidth of around 1 Gb/s to 6 Gb/s of either the NOR [15] or NAND [28] Flash memories. In this work, we will focus on this part of ASR as it represents the main power and memory bottleneck, in order to reduce the Viterbi workload in a way to minimize the energy consumption and memory usage, while maintaining accuracy.

III. Leveraging ASR Feedback At Run-Time

In this section, we present our analysis of different properties of the recognition process in ASR that we later exploit to built a highly efficient accelerator. First, we evaluate the locality between the processing of successive frames of speech during the Viterbi search. Then, we define the basics of LAWS and illustrate its efficiency compared against a naive solution that consists on simply reducing the beam. Finally, we elaborate on our selective approach of dynamically adapting the amount of ASR's workload based on the Viterbi search's confidence.

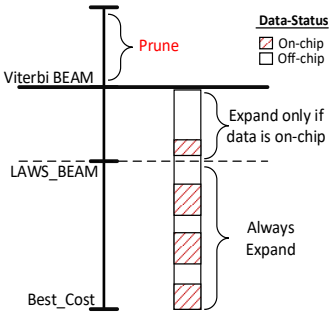


Fig. 4: Viterbi expansion under Locality-AWare Scheme.

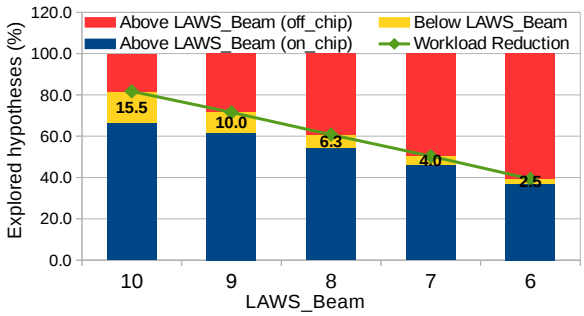


Fig. 5: Hypotheses distribution along the LAWS_Beam and Viterbi beam distance. Hypotheses whose cost are above LAWS_Beam are partitioned into on-chip and off-chip, depending on the location of their data.

A. *ASR's Frame-to-Frame Data-Locality*

In order to illustrate how useful data-locality is for the ASR's search, we compute the average probability of the hypotheses, considering whether or not their data is on-chip. In general, probabilities are represented in the negative log-space for ASR systems, in order to simplify operations by replacing multiplications with additions and also prevent arithmetic underflows [6]. Regarding the beam pruning of Viterbi search, we consider 15 as the beam value as specified for the Kaldi ASR toolkit [19], which is representative as the very low probability of 3.1e-7. In other words, hypotheses whose distance to the best hypothesis is higher than 15, up to the current frame, are discarded (pruned) in order to keep the search space tractable. Figure 1 shows the average acoustic-cost, i.e. negative log-probability, of the different hypotheses, for 15 audio tests selected from different speakers of the Librispeech corpus [20]. In addition, we measure the average frame-to-frame locality for all audio tests using the UNFOLD's architecture parameters [2]. We can see that on average 86% of the WFST states and arcs used in a given frame are reused in the next frame. Besides, we can also observe that hypotheses whose data is on-chip have considerably lower cost, i.e. higher probability, than the ones explored for the first time in each frame, whose data requires an off-chip memory access.

Furthermore, we extend our analysis to show that on-chip hypotheses are more likely to be in the correct path at the end of Viterbi search. Considering the decoding of the entire Librispeech corpus (5.4 hours of speech), we have measured the likelihood of seeing the correct hypotheses regarding their data location. Figure 3 depicts the probability that the correct partial hypotheses have scores at a given beam distance (distances are rounded up), distingushing between hypotheses whose data is on-chip and those whose data is off-chip. We can see that the vast majority of of correct hypotheses are stored on-chip, about three to four orders of magnitude more than off-chip, and most of them are at a beam distance lower than 1.

Based on the above data, we propose to prioritize the expansion of the hypotheses that exhibit good temporal locality due to two main reasons. First, they are less expensive from a computational point of view, which is beneficial for performance and energy. Second, they are much more likely to be in the correct path, which is beneficial for accuracy. On the other hand, we are more selective with the hypotheses that require off-chip memory accesses, by using a more aggressive beam. Figure 4 illustrates the Viterbi search expansion for both the conventional beam search and our Locality-AWare Scheme (LAWS). In the conventional approach, the hypothesis with the best cost is identified on every frame. Hypotheses whose distance to the best hypothesis is smaller than a given beam are explored, whereas the rest are discarded since they are very unlikely. Our scheme is different since we consider both the scores (likelihood) of the hypotheses and their temporal locality to prune the search space.

LAWS defines an additional beam, named LAWS_BEAM in Figure 4, smaller than the original beam. Hypotheses whose distance to the best hypothesis is smaller than the LAWS_-BEAM are still expanded, since the likelihood that they end up in the correct path is significant. Hypotheses whose distance is larger than the original beam are still discarded as in the conventional search, since they are very unlikely to be correct. However, hypotheses between LAWS_BEAM and the Viterbi beam are only explored if their data is on-chip. Note that those are hypotheses that are always explored in the conventional search, but they are not very likely to be in the correct path. Therefore, we consider that the cost of accessing main memory is excessive for such unlikely hypotheses and, hence, we discard them. Figure 5 shows that a significant percentage of the hypotheses are above the LAWS_BEAM and have their data off-chip and, hence, our scheme is effective at reducing workload. More specifically, LAWS is able to save between 20% and 60% of the Viterbi search workload depending on the selection of the LAWS_BEAM.

Note that our scheme is better than a naive solution that just lowers the beam used in the Viterbi beam search, since hypotheses whose distance is between the LAWS_BEAM and the Viterbi beam are still explored provided that their data is on-chip. Furthermore, exploring these hypotheses is cheap as they do not require off-chip memory accesses. Hence, from the point of view of main memory, our scheme is equivalent to a conventional Viterbi beam search with a more aggressive beam, but we achieve much better accuracy since we explore more hypotheses using already avaiable on-chip data. Figure 6 shows the ASR accuracy, i.e. Word-Error-Rate (WER), for decoding the entire Librispeech corpus, using different values for the Viterbi beam and the LAWS_BEAM. As we can see,

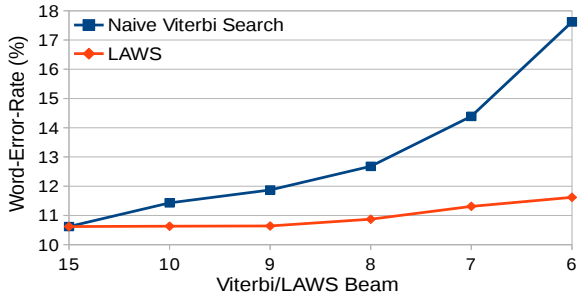


Fig. 6: Word-Error-Rate (WER) with respect to different beams for LAWS and naive Viterbi search.

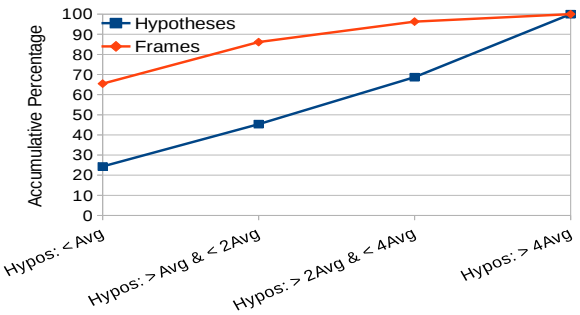


Fig. 7: Cumulative percentage of hypotheses and frames explored during the entire 5.4 hours of audio in Librispeech corpus, for different groups of frames based on their number of hypotheses.

lowering the beam has a large impact on the accuracy of the conventional Viterbi search, whereas LAWS is able to use more aggressive beams with smaller impact on WER, because the aggressive beam is used only for the hypotheses with bad temporal locality, i.e. whose data is not on-chip. In the next subsection, we show that by dynamically adjusting the LAWS_BEAM, we can further reduce the impact on accuracy while obtaining large performance and energy improvements.

B. Improving LAWS Using Viterbi's Confidence

LAWS is highly effective at saving energy consumption and main memory bandwidth by significantly reducing ASR's workload. However, its main issue is the potential increase of WER which can be non-negligible for some applications that require a high level of accuracy. In order to handle this problem, we show that we can leverage a measure of confidence at each frame to build an adaptive approach for dynamically selecting the amount of workload, and reach a high level of accuracy. We use the number of expanded hypothesis at each frame as a measure of confidence of the search. That is, when the number of hypothesis is low, this implies that the search has high confidence whereas a large number of hypothesis implies that the search has low confidence.

Viterbi search exhibits very different levels of confidence when decoding different frames of speech, as illustrated with an example in Figure 2). Therefore, in order to decide the degree of pruning based on the confidence of the search, we partition audio frames in a way that each group has approximately the same amount of work during the Viterbi search, that is, a similar number of total hypotheses. To do so, we dynamically compute the average number of hypotheses explored per frame and use several multiples of this average in order to balance the total number of hypotheses in each group, as shown in Figure 7. In this study we use four categories but more groups can be used for a finer granularity in controlling the ASR workload.

Next, we measure the impact on ASR accuracy when changing the beam for each group, so as to validate our initial hypothesis that when the number of hypothesis is large, confidence is low and thus correct hypotheses tend to be farther from the optimal score (in other words, a larger beam should be used), and vice versa. For this purpose, Figure 8 shows the cumulative percentage of correct hypotheses expanded for each group of frames when varying the beam distance from 1 to 15 (distances are rounded up). We can observe that for any beam the number of correct hypotheses is higher for groups with higher confidence (i.e., lower number of hypothesis). In other words, if we want to reach a certain level of accuracy, we need to choose a different beam value for each group: the higher the confidence of the frames, the lower is the beam-distance required.

As illustrated in Figure 7, groups have different number of frames, in contrast with the even distribution of the hypotheses among different groups. We can observe that nearly 65% of the frames have a number of hypotheses less than average. On the other hand, a very small percentage of frames have their hypotheses-count higher than 4 times the average. Figure 8 suggests that low confident frames require higher beam distance than the high confident ones in order to reach a particular level of accuracy, but we can further refine the selection of the beam by taking into account that there are more frames with high confidence than with low confidence (as shown in Figure 7). This suggest that we can decrease the beam for low confident regions, which will affect very few frames and thus cause a minor penalty on accuracy but a large reduction in workload, since these frames have many hypotheses, while slightly increasing it for high confident regions, which will affect many frames and thus compensate for the loss of accuracy with a small increase in workload, since these frames have few hypotheses. Using this way, we get the same global accuracy but with a significant reduction in workload.

IV. Basic Design of ASR's Accelerator

The proposed ASR system is built on top of UNFOLD accelerator [2] to achieve high- performance and energy-efficient Viterbi search. Figure 9 illustrates UNFOLD's architecture. It includes several pipeline stages and various on-chip memories. The modified components are marked with dashed line, and all the memory components are depicted in gray color. Regarding the pipeline stages, several components are employed to fetch WFST's states and arcs, i.e. State and Arc Issuer, fetch DNN acoustic scores, i.e. Acoustic-likelihood Issuer, compute the likelihood of a hypothesis, i.e. Likelihood Evaluation, and

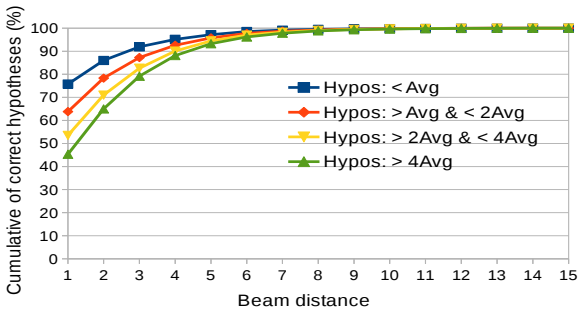


Fig. 8: Cumulative percent of the correct hypotheses explored under different beam distances in Viterbi search, for the different groups of frames based on their generated hypotheses.

write the information of the new hypothesis, i.e. Hypothesis Issuer. Hypothesis Issuer stores both on-chip and off-chip, the intermediate results and backtracking data of the new hypotheses, respectively.

Two hash tables are used by UNFOLD to store the intermediate results of the different hypotheses for the current and next frames of speech. These hash tables are direct mapped using an XOR hash function of the WFST's states of hypotheses' information. Regarding the reminding mechanisms of the hash structure, like handling collisions and overflows, they are managed as described in [6]. Moreover, the accelerator provides several on-chip caches to speed up the accesses to the different datasets used in ASR and an Offset Lookup Table for storing the result of the on-the-fly composition of the AM and LM WFSTs.

The accelerator's pipeline starts at the State Issuer. By going through all the hypotheses using the current Hash table, State Issuer firstly checks the condition regarding the beam pruning, and it fetches the state's information of each non-pruned hypothesis using State Cache. Next, Arc Issuer generates new hypotheses by expanding all the arcs outgoing from each state that is received from State Issuer, by fetching their data from memory through the Arc Cache. After that, Acoustic-likelihood Issuer reads the DNN-computed probabilities for the sub-phoneme associated with each arc. Then, Likelihood Evaluation Unit computes the scores of the new hypothesis using the previous hypothesis's scores, arc's weight and the DNN likelihood. Finally, the Hypothesis Issuer stores the new hypothesis's intermediate results in the next Hash table and also stores its backtracking information in memory using the Word Lattice Cache. At the end of each frame's evaluation, the Hash tables are swapped to continue with the new hypotheses formation in the next frame. At the end of search, a backtracking phase is executed to locate the best sequence of words on the word lattice generated by the search.

To add our Locality-AWare Scheme, we modify the accelerator's design in a way to add another level of beam pruning at State Issuer. Also, we change the State Cache to notify State Issuer of the locality of hypotheses' data using an status-signal. Our proposed scheme is added at the initial stage of the pipeline, after the beam pruning and before fetching the state's information.

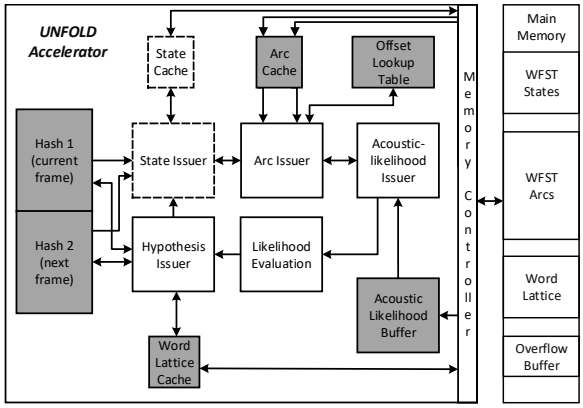


Fig. 9: Architecture of UNFOLD [2], a state-of-the-art Viterbi search accelerator. All the memory components are in gray color. Modified parts are shown with dashed lines.

V. Locality-Aware Scheme for ASR Acceleration

In this section, we elaborate on the implementation details of our Locality-AWare Scheme on top of UNFOLD. Then, we provide some initial results to show the benefits of this approach. Next, we evaluate different strategies in order to determine the pruning beam based on the search confidence. Finally, we introduce several configurations by combining our two proposed techniques, providing an efficient trade-off between accuracy, performance and energy of the ASR system.

A. LAWS Mechanism

The analysis in section III-A showed the benefit of LAWS for efficiently expanding the Viterbi search for ASR systems. Unlike the previous schemes, our approach considers both the likelihood of the evaluated hypotheses and the locality of their data on the on-chip memory components as an important aspect for driving the search. LAWS's mechanism is implemented in the State Issuer of the UNFLOD's architecture, which is processed after fetching the information of each hypothesis from the current Hash table (see Figure 9).

Figure 10 shows the speedup achieved in the Viterbi search with respect to the baseline (LAWS_Beam = 15) using different LAWS_Beam values. We assume the same size of 256 KB for the *State Cache* as reported for UNFOLD [2]. As seen in this figure, we can obtain nearly 55% performance improvement over UNFOLD by using LAWS_Beam of 9, without losing any accuracy. However, if higher speedup is required, ASR loses some accuracy up to 1% when using LAWS_Beam = 6. To solve this issue, we combine LAWS with a new method of controlling ASR accuracy, by dynamically adapting the beam during runtime, based on the confidence of the search at each frame's evaluation.

The size of *State Cache* is an important parameter when combining hypotheses' likelihood with their data-locality. Thus, we have tested several cache sizes in order to accurately measure LAWS's benefits. Figure 11 shows the speedup achieved for each cache size versus different LAWS_beams. As illustrated, using a larger cache than UNFOLD's base configuration would result in approximately similar speedup

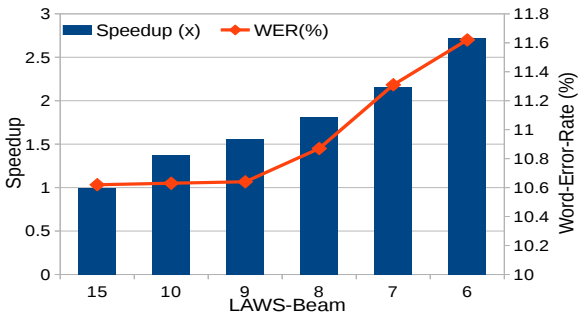


Fig. 10: Speedup and Word-Error-Rate (WER) of LAWS using different LAWS_Beam. LAWS achieves 55% speedup without losing any accuracy. Further performance improvement causes some accuracy loss.

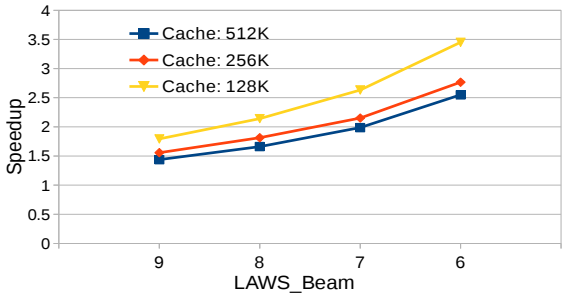


Fig. 11: LAWS speedup versus beam width, for different sizes of cache. The larger the cache, the higher the locality it provides and the more hypotheses are explored, resulting in lower speedup.

for LAWS. On the other hand, by decreasing the cache size, we see that speedup grows significantly by applying a more aggressive LAWS_beam. However, these improvements come at the cost of losing some accuracy, due to discarding some correct hypotheses because of their poor locality.

B. Adaptive Beam-Selection Approach

As discussed in Section III-B, the confidence of Viterbi search is one of the important ASR's characteristics that LAWS leverages. To measure confidence during the search, we count the number of hypotheses that are expanded at each frame. Using this number, we score the confidence of each frame with respect to the different confidence regions specified by the four groups of frames (see Figure 7). As mentioned in Section III-B, by choosing a range of beam distances rather than just one for the different frame regions, we can obtain higher accuracy while significantly reducing the ASR's workload. Therefore, we define a multi-beam selection scheme for LAWS that can adapt the LAWS_Beam dynamically based on the search confidence.

Figure 12 shows different models of controlling ASR's workload by selecting the beam distance based on the search confidence. The Single-beam scheme considers only one beam regardless of the confidence of each frame. However, as mentioned in Section III-A, in order to maintain ASR accuracy, we are limited to almost 30% of potential performance improvement when using a single LAWS_Beam. On the other hand, the Multi-beam model is more flexible as it chooses the beam values proportional to the search confidence. Nevertheless, we have seen that we can obtain most of the benefits of this scheme by using two or three beam values. In the last model, called Dual-Beam, we use two beams, high and low, for low and high confident regions, respectively.

As the Dual-beam approach is simple and highly efficient, we implement it in the Viterbi accelerator to improve LAWS's accuracy. To do so, we add the required logic at State Issuer to compute the average number of hypotheses at each frame (multiply+add+division). Moreover, we use a table that contains the LAWS_Beams and a table for storing the confidence threshold for the different groups of frames. We use these tables at run-time, in order to choose between LAWS_Beams based on the confidence of the Viterbi search, by comparing the number of hypotheses being processed at each frame with the confidence thresholds. Note that we consider the hypotheses at the beginning of processing each frame for their confidence measurement, but not the ones that are generated at the end of each frame. The confidence thresholds are dynamically updated during the search, by constantly updating the dynamic average number of hypotheses. We completely overlap these operations with the processing of the best hypothesis at the beginning of each frame.

Regarding Librispeech corpus, we choose 6 as the low LAWS_Beam since it provides the highest reduction in workload, and 8 as the high LAWS_Beam to compensate for the accuracy with negligible loss compared to the baseline. We call this scheme $Small_8$-Big_6, which selects the beam 8 and 6 for the frames with the number of hypotheses smaller and bigger than the specified threshold, respectively.

Figure 13 shows the performance and accuracy of $Small_8$-Big_6 for the different confidence thresholds. Furthermore, it also shows the LAWS's single-beam approach for several beam widths. As depicted, the single-beam scheme loses between 0.7% to 1% accuracy to achieve the same performance as what $Small_8$-Big_6 offers. For instance, $Small_8$-Big_6 obtains a speedup of 2.45x with a negligible increase of 0.32% in WER, whereas by using a single beam, we lose almost 0.85% of accuracy to achieve the same performance improvement. As illustrated, by using beam 6 at more frames with low confidence, we can obtain better speedup while losing some accuracy. Therefore, based on each application's requirements, we can choose either of these thresholds to both maintain the accuracy and gain high performance benefits. Furthermore, we have tested several configurations using three LAWS_Beams to improve the trade-off between accuracy and workload reduction. However, we observed very slight improvement in speedup while losing some accuracy. Therefore, we conclude that it is better to use only two beams as it results in a simpler scheme.

We have also evaluated our scheme for two more ASR decoders, Tedlium [29] and Voxforge [30]. Tedlium is a more challenging benchmark than Librispeech, because it includes spontaneous speech in noisy environments and, hence, the WER is larger (22.6%). On the other hand, Voxforge decodes

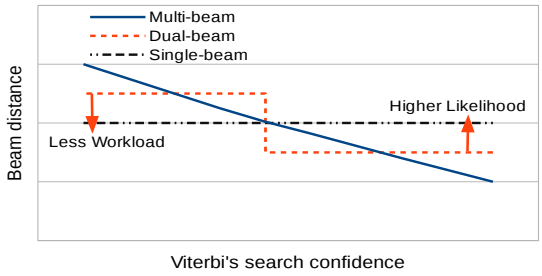


Fig. 12: Different beam selection models based on the confidence of Viterbi search, i.e. number of hypotheses. Single-beam uses one beam independent to the search confidence.

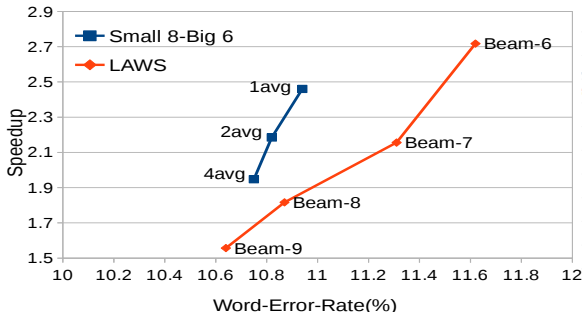


Fig. 13: Speedup versus Word-Error-Rate (WER), for the Dual-beam selection scheme, called $Small_8$-Big_6. High confidence threshold provides better speedup, while maintaining accuracy.

speech over a smaller vocabulary, which results in a simpler task and higher accuracy of 13.3% for the baseline system. Figure 14 depicts Tedlium ASR workload reduction versus WER, when applying each part of our proposal separately. By applying LAWS, the number of hypotheses decreases significantly with respect the baseline. Regarding the case of using Beam 9, we shrink the workload to less than 0.68 with negligible change in accuracy (less than 0.01% increase in WER). However, we incur some accuracy loss for achieving higher performance benefit when considering beams smaller than 8. By using the $Small_8$-Big_6 approach, we can alleviate the problem by obtaining a better tradeoff between performance and accuracy. For instance, by setting the threshold of switching between beams 6 and 8 to $2avg$, we achieve almost the same hypotheses reduction as LAWS with Beam-7, whereas improving WER by 0.3%. Regarding Voxforge, we achieve up to 40% workload reduction by using LAWS, with an accuracy loss of 1.26%. We reduce this accuracy loss to just 0.55% using the adaptive beam-selection scheme, while maintaining LAWS's efficiency.

VI. Evaluation Methodology

The overall ASR system comprises a EIE [17] and a modified UNFOLD [2] accelerators for the DNN and Viterbi search respectively. The integration between them is performed as described in [7]. The input speech is split into batches of N frames and the the two accelerators work in parallel: the EIE computes the acoustic scores for the current batch while the UNFOLD performs the decoding of the previous batch. The EIE communicates the acoustic scores through a shared buffer in main memory. Our simulations account for the time and energy required by the accesses to this shared buffer.

In order to evaluate our scheme, we developed a cycle-accurate simulator which models the on-the-fly Viterbi search accelerator described in [2]. The simulator also works as a functional emulator, which produces the most likely sequence of words for the input speech to evaluate the Word-Error-Rate (WER). Furthermore, we have implemented in the Viterbi simulator all the hardware extensions necessary for the LAWS. Regarding the parameters of the accelerator's architecture, we use the same configuration as UNFOLD, shown in Table II. Regarding the DNN accelerator, we use EIE [17] configured with parameters shown in Table III. We use the 70%-pruned Kaldi network with its weights quantized to 12-bit. Finally, regarding our datasets, we employ the Librispeech corpus [20], which includes 2620 utterances from 87 different speakers (more than 5.4 hours of speech).

All the accelerator's pipeline stages are implemented in Verilog and synthesized to obtain the delay and power using the Synopsys Design Compiler, the modules of the DesignWare library and the technology library of 28/32nm from Synopsys [31]. On the other hand, we characterize the memory components of the accelerator by obtaining the delay, energy per access and area using CACTI-P [32]. For both the Design Compiler and CACTI, we use the technology library and the typical configurations with a supply voltage of 1 V. Finally, to model the off-chip main memory, we use the Micron power model for an 8-GB LPDDR4 [33], [34]. The simulator provides the activity factors for the different components and the total cycle count, which are then used to compute execution time, and dynamic and static energy by combining them with the estimations of the Design Compiler, CACTI and MICRON's power models.

To set the frequency of the system, we consider the critical path-delay and access times reported by Design Compiler and CACTI respectively. We take the maximum delay among the different components, which is 1.25 ns for accessing Arc Cache, resulting in 800 MHz of frequency.

VII. Experimental Results

In this section, we evaluate the benefits of our scheme, called LAWS, when implemented on top of a state-of-the-art accelerator for speech recognition, UNFOLD [2]. We consider several configurations using different beams. First, we report the decoding time and energy consumption of the proposed approach compared to the baseline accelerator. Next, we present the reduction in memory accesses by exploiting the locality in ASR. Finally, we show the power-dissipation benefits achieved by LAWS. We refer to our baseline design as UNFOLD, and the different configurations of LAWS using different beams as LAWS-Beam-8, LAWS-Beam-7, and LAWS-Beam-6, respectively. Also, we use analogous terminology for the $Small_8$-Big_6 scheme that uses different confidence thresholds in order to adjust the ASR's workload in LAWS's mechanism.

Figure 15 shows the Viterbi and DNN breakdown of the ASR's decoding time per one second of speech for the different approaches. We group different configurations of LAWS

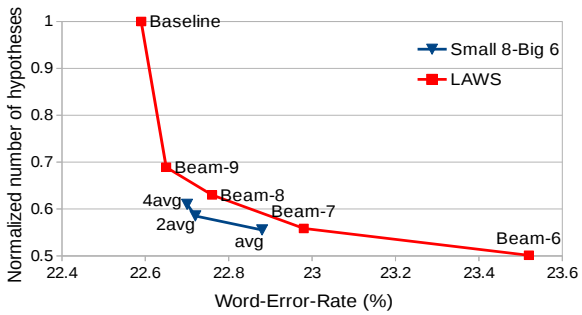


Fig. 14: ASR workload reduction versus WER increase, for the LAWS single-beam approach and $Small_8$-Big_6. We significantly improve performance with a minor overhead in accuracy.

TABLE II: UNFOLD's configuration.

Technology	32 nm
Frequency	800 MHz
State Cache	256 KB, 4-way, 64 B/line
Arc Cache	768 KB, 8-way, 64 B/line
Word Lattice Cache	128 KB, 2-way, 64 B/line
Acoustic Likelihood Buffer	64 Kbytes
Hash Table	576KB, 32K entries
Offset Lookup Table	192KB, 32K entries
Memory Controller	32 in-flight requests
Likelihood Evaluation Unit	4 FP adders, 2 FP comparators

TABLE III: EIE's configuration.

Technology	45 nm
Frequency	800 MHz
Number of PEs	16
PE FIFO Size	8
Quantization	12-bit
DNN model size	2 MB

based on whether they use either single or dual beams. As shown, Viterbi is by far the main bottleneck in ASR, representing 95.2% of the execution time in the baseline configuration (UNFOLD+EIE). By using our scheme, we achieve more than 1.5x speedup compared to UNFOLD in all the LAWS configurations. The main reason of this improvement is the decrease of ASR's workload, which is achieved by combining the hypotheses' data-locality with their likelihood in driving the Viterbi search. With the conservative selection of LAWS beam width of 9, we can reduce the decoding time by 1.52x. On the other hand, by aggressively pruning hypotheses with bad temporal locality, LAWS largely reduces workload with small impact on accuracy.

When using a single beam, LAWS reaches up to 2.5x speedup by using the most aggressive configuration. However, by having only one beam, LAWS suffers some accuracy loss near to 1% (see Figure 10). This is because LAWS discards the hypotheses requiring off-chip accesses at all the frames similarly. On the other hand, by applying two different beams and selecting between them based on different confidence thresholds, we can adjust the workload reduction in various speech frames and achieve high speedup with better accuracy. The more aggressive we set the threshold on the number of hypotheses, the higher the performance improvement and WER. For instance, considering *2avg* as the threshold, we obtain 2.1x speedup with a negligible increase of 0.2% in WER (see Figure 13).

Another main benefit of LAWS, as a results of the ASR's workload reduction, is a significant saving in energy consumption. Figure 16 shows the energy consumed for decoding one second of speech in the baseline and different LAWS's configurations. Regarding the breakdown, Viterbi takes the bulk of energy consumption (more than 97%), whereas the DNN represents just around 3%. Compared with UNFOLD, our scheme saves energy by more than 1.62x in all the configurations. Similar to the performance improvements, $Small_8$-Big_6 provides the best trade-off between accuracy and energy reduction. When using the threshold of *2avg*, the energy-consumption decreases by 2.3x. Additionally, we can save 26% more energy reduction using a more aggressive threshold as *1avg*. Doing so increases the WER by 0.32%, which may be acceptable for most applications. Thus, by means of an more intelligent beam pruning that leverages data-locality, we achieve significant improvements in energy consumption and performance of ASR systems. Furthermore, we maintain the ASR accuracy by adapting the workload reduction using a measure of the Viterbi search confidence.

We have also compared LAWS with a recent ASR acceleration scheme [8]. In that work, the authors employed a DNN accelerator designed for dense models that is not particularly efficient for pruned (sparse) DNNs. In our work, we employ a state-of-the-art accelerator for sparse DNN models (EIE) combined with UNFOLD. For this reason, the execution time of the DNN is very small in our baseline compared to the one reported in [8]. This difference in the methodology makes it difficult to directly compare our numbers with the ones reported in [8]. In any case, our work is orthogonal to [8], since they propose a new hash structure to restrict the hypotheses to 1024, but they are still generating more hypotheses during the search expansion as there are collisions in the hash table. When implementing LAWS with LAWS_beam 7 on top of [8], we obtain a 1.7x speedup at the cost of increasing WER by 0.35%. We can further reduce accuracy loss to around 0.2%, by leveraging our dynamic beam selection technique.

In addition to the benefits in energy reduction and performance, LAWS solves another main challenge of ASR systems, by reducing the memory requirements by almost an order of magnitude. Figure 17 depicts the normalized memory requests and bandwidth required for UNFOLD and LAWS using the different configurations. Regarding the DNN, all the weights (2 MB for 70%-pruned model) are stored on-chip and memory traffic is mainly consumed for fetching speech frames' feature vectors. However, this amount only stands for less than 1% of the total ASR memory bandwidth requirement. As shown in this figure, by using a single beam, LAWS reduces memory activity to less than 22%. In addition, the dual beam approach shows between 87% to 89% decrease in memory requests depending on the confidence thresholds. We achieve this huge reduction in memory requirements since LAWS removes most of the memory fetches required for the hypotheses whose data is off-chip. Regarding memory bandwidth, we save more than 64% in all the configurations. By reducing the beam, bandwidth requirement gets as low as 381 MB/s, reduced by

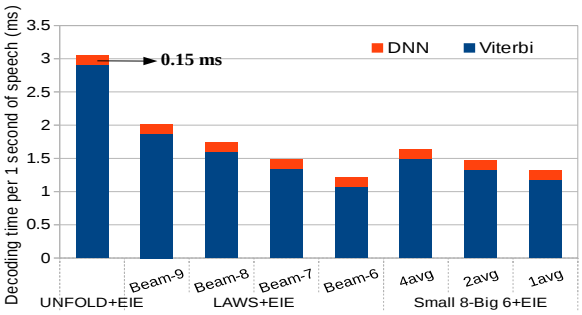


Fig. 15: ASR's decoding time per one second of speech for the baseline and different configurations of LAWS using single and dual beams.

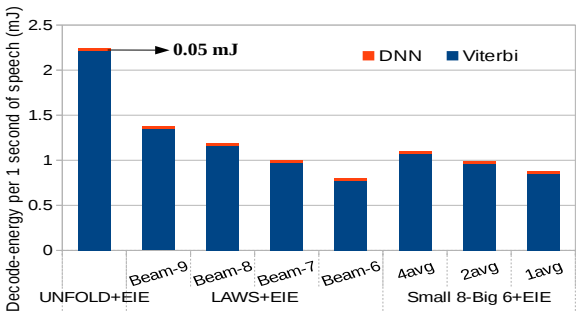


Fig. 16: ASR's energy-consumption for decoding one second of speech for the baseline and different configurations of LAWS using single and dual beams.

almost 74% compared to the baseline. Furthermore, $Small_8$-Big_6 achieves between 73% to 74.5% saving in bandwidth. As LAWS performs much faster than UNFOLD, the results of bandwidth savings show lower improvement than the reduction in memory requests.

Figure 18 shows the breakdown of the power dissipation in different components for the $Small_8$-Big_6 LAWS's configuration and UNFOLD. Furthermore, the DNN power is shown in both cases. As illustrated, the main power bottleneck is Viterbi, which dissipates nearly 70% of the total power. The power is almost similar for most of the accelerator's components, except for main memory, which has been reduced by 73% in LAWS. The reason for a total moderate power reduction is that LAWS improves performance and at the same time it saves energy by a large amount. Overall, it achieves a power reduction between 3.6% and 4.2%. The total power of the system is 1.06 W.

Finally, we have evaluated the area overhead of our design, with respect to the UNFOLD's baseline architecture. Our synthesis results show an area increase of 7.5% in the State Issuer to implement the LAWS's mechanism and less than 1% in the State Cache, which results in less than 0.1% increase in total Viterbi accelerator's area. The total area including LAWS and EIE is 46.86 mm^2.

VIII. Related Work

Accelerating ASR systems in hardware has attracted the attention of architectural community recently. There have been two main types of acceleration for small [5], [35], [36] and large [2], [4], [6], [7], [37] vocabulary speech recognition systems. The former designs use very small speech models to avoid the main memory bottlenecks, whereas the latter ones search very large WFSTs to provide higher accuracy. As the various applications of mobile systems and wearable devices require highly accurate ASR systems supporting a large vocabulary, several research works focused on accelerators to solve different challenges such as achieving real-time [4], [7], reducing memory-footprint [2], [38], [39], and lowering the power dissipation and energy-consumption [6], [37], [40].

Although a state-of-the-art ASR accelerator [2] significantly reduces the speech data-set footprint, there is still high bandwidth requirement for the dynamic memory activity. IoT and wearable devices have the tightest memory bandwidth limitation as they normally operate on battery and need to be ultra low-power [15]. To deal with this problem, there have been several proposals on reducing the ASR's workload [4], [27], [41]. The most well-known approach is to prune the hypotheses that have lower likelihood than the best one minus a fixed beam width. Johnston et.al [4] presented a thorough analysis on how to use the beam pruning and proposed a preemptive pruning which discards the hypotheses in advance during each frame's evaluation. Our baseline scheme includes this method and our LAWS proposal is completely orthogonal.

Other schemes of pruning are more complex and inefficient for hardware implementation [27], [41]. For instance, Ortmanns et.al [41] proposed the histogram pruning, which only expands the K-best hypotheses and needs to sort the hypotheses on each frame. Kaldi [19] uses the same technique, called absolute pruning, on top of the relative preemptive pruning to keep the search space manageable. In absolute pruning, only the K-best hypotheses are kept whereas the rest are discarded. Price et. al. [37], [42] propose a hardware implementation of absolute and relative pruning that does not require a full search of the hypotheses. However, this scheme requires stalling the pipeline of the accelerator to re-prune the hypotheses when the on-chip resources are exceeded, introducing non-negligible overheads as the re-pruning process may be repeated multiple times. Our proposal is different since we introduce a novel method for reducing ASR's workload combining hypotheses's likelihood with their data-locality. Moreover, we have shown that by taking into account the search confidence at each frame, we efficiently reduce the workload while maintaining the accuracy by using several beam-widths depending on the search confidence.

Regarding the adaptive beam selection scheme, there have been some previous proposals [43], [44]. However, they have mainly evaluated several pruning techniques and used different values of the beam width in order to choose the one that best suits the trade-off between recognition accuracy and performance. There is also some similar measurement of confidence in [44] that is used as a complementary pruning phase to the relative likelihood preemptive pruning in a system that combines different pruning schemes to achieve higher speedup. Our confidence-based approach is different due to two main reasons. First, it dynamically sets the beam width. Second, it is much simpler as it only uses one beam pruning

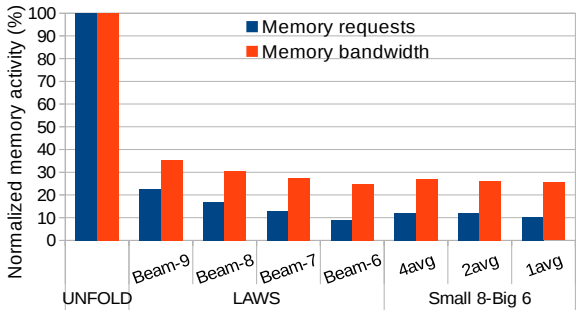


Fig. 17: Normalized memory requests and memory bandwidth for the baseline accelerator and the different configurations of LAWS. The baseline memory bandwidth is 1.43 GB/s. The reduction in memory bandwidth is smaller than the reduction in memory traffic due to the speedup provided by LAWS.

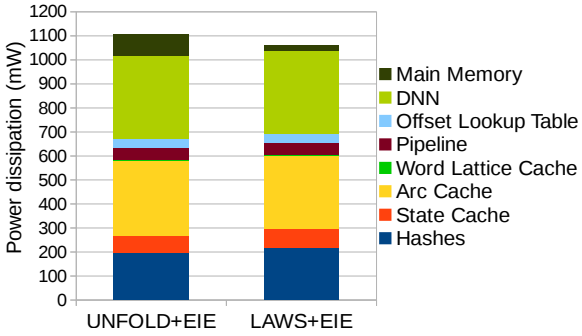


Fig. 18: ASR's power-dissipation of the baseline and LAWS using dual 8 and 6 beams ($Small_8$-Big_6(2avg)).

policy without adding the extra complexity of having multiple pruning strategies and choosing between them, being more amenable for a hardware implementation.

IX. Conclusions

In this paper, we target one of the main challenges for ASR systems in mobile, IoT and wearable devices, which is the high memory and energy requirements to perform speech recognition. Previous schemes try to solve this issue by discarding the unlikely hypotheses expanded by the Viterbi search using a beam pruning. Although reducing the beam width decreases ASR's workload significantly, it causes an important loss of accuracy. We present LAWS, a scheme that obtains high benefits in workload reduction without compromising accuracy by a new approach that combines novel insights about data-locality and confidence of the Viterbi search with the statistical scores computed by during the search. LAWS removes over 87% of the off-chip memory activity, which improves performance and energy-consumption by 2.1x and 2.3x, respectively, with negligible impact in accuracy.

X. Acknowledgements

This work has been supported by the the CoCoUnit ERC Advanced Grant of the EU's Horizon 2020 program (grant No 833057), the Spanish State Research Agency under grant TIN2016-75344-R (AEI/FEDER, EU), and the ICREA Academia program.

References

[1] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *CoRR*, vol. abs/1610.05256, 2016. [Online]. Available: http://arxiv.org/abs/1610.05256

[2] R. Yazdani, J.-M. Arnau, and A. González, "Unfold: A memory-efficient speech recognizer using on-the-fly wfst composition," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-50 '17. New York, NY, USA: ACM, 2017, pp. 69–81. [Online]. Available: http://doi.acm.org/10.1145/3123939.3124542

[3] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 69 – 88, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230801901846

[4] J. R. Johnston and R. A. Rutenbar, "A high-rate, low-power, asic speech decoder using finite state transducers," in *2012 IEEE 23rd International Conference on Application-Specific Systems, Architectures and Processors*, July 2012, pp. 77–85.

[5] J. Choi, K. You, and W. Sung, "An fpga implementation of speech recognition with weighted finite state transducers," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2010, pp. 1602–1605.

[6] R. Yazdani, A. Segura, J.-M. Arnau, and A. González, "An ultra low-power hardware accelerator for automatic speech recognition," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Oct 2016, pp. 1–12.

[7] R. Yazdani, A. Segura, J. M. Arnau, and A. González, "Low-power automatic speech recognition through a mobile gpu and a viterbi accelerator," *IEEE Micro*, vol. 37, no. 1, pp. 22–29, Jan 2017.

[8] R. Yazdani, M. Riera, J. Arnau, and A. González, "The dark side of dnn pruning," in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, June 2018, pp. 790–801.

[9] R. Yazdani, J. Arnau, and A. González, "A low-power, high-performance speech recognition accelerator," *IEEE Transactions on Computers*, vol. 68, no. 12, pp. 1817–1831, 2019.

[10] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[11] Y. Wang, V. Panayotov, I. Edrenkin, D. Povey, and G. Chen, "Kaldi speech recognition results," 2019.

[12] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Y. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Y. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," *CoRR*, vol. abs/1512.02595, 2015. [Online]. Available: http://arxiv.org/abs/1512.02595

[13] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," *CoRR*, vol. abs/1805.03294, 2018. [Online]. Available: http://arxiv.org/abs/1805.03294

[14] N. Zeghidour, Q. Xu, V. Liptchinsky, N. Usunier, G. Synnaeve, and R. Collobert, "Fully convolutional speech recognition," *CoRR*, vol. abs/1812.06864, 2018. [Online]. Available: http://arxiv.org/abs/1812.06864

[15] H. Sian, "Iot and wearable devices mean rethinking memory design," *Micron Technology*, October 2014. [Online]. Available: https://www.embedded.com/design/mcus-processors-and-socs/4436137/wearable-devices-mean-rethinking-memory-design

[16] Micron, "Nor/nand flash guide," *Micron Technology*, 2017. [Online]. Available: https://www.micron.com/media/documents/products/product-flyer/nor_nand_flash_guide.pdf

[17] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "Eie: Efficient inference engine on compressed deep neural network," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, June 2016, pp. 243–254.

[18] R. Yazdani, J. Arnau, and A. González, "Poster: Leveraging run-time feedback for efficient asr acceleration," in *2019 28th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, Sep. 2019, pp. 463–464.

[19] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011*

Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, April 2015, pp. 5206–5210.

[21] R. Vergin, D. O'Shaughnessy, and A. Farhat, "Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 7, no. 5, pp. 525–532, Sep 1999.

[22] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[23] H. J. G. A. Dolfing and I. L. Hetherington, "Incremental language models for speech recognition using finite-state transducers," in *Automatic Speech Recognition and Understanding, 2001. ASRU '01. IEEE Workshop on*, 2001, pp. 194–197.

[24] D. Caseiro and I. Trancoso, "On integrating the lexicon with the language model," 2001.

[25] I. Trancoso and D. Caseiro, "Transducer composition for "on-the-fly" lexicon and language model integration," in *Automatic Speech Recognition and Understanding, 2001. ASRU '01. IEEE Workshop on*, 2001, pp. 393–396.

[26] D. Willett and S. Katagiri, "Recent advances in efficient decoding combining on-line transducer composition and smoothed language model incorporation," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, May 2002, pp. I–713–I–716.

[27] T. Hori, C. Hori, Y. Minami, and A. Nakamura, "Efficient wfst-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1352–1365, May 2007.

[28] B. Cole, "Micron shows off memory muscle with new iot/m2m mcps," *Micron Technology*, November 2014. [Online]. Available: https://www.edn.com/print/4436958

[29] A. Rousseau, P. Deléglise, and Y. Estève, "Enhancing the ted-lium corpus with selected data for language modeling and more ted talks," in *In Proc. LREC*, 2014, pp. 26–31.

[30] V. S. Corpus, "http://www.voxforge.org," 2009.

[31] "Synopsys," https://www.synopsys.com/, accessed: 2017-07-20.

[32] S. Li, K. Chen, J. H. Ahn, J. B. Brockman, and N. P. Jouppi, "Cacti-p: Architecture-level modeling for sram-based structures with advanced leakage reduction techniques," in *IEEE/ACM ICCAD*, 2011.

[33] TN-41-01, "Calculating memory system power for ddr3, micron technology, tech. rep," Tech. Rep., 2007.

[34] TN-53-01, "Lpddr4 power calculator, micron technology, tech. rep," Tech. Rep., 2016.

[35] K. You, J. Choi, and W. Sung, "Flexible and expandable speech recognition hardware with weighted finite state transducers," *Journal of Signal Processing Systems*, vol. 66, no. 3, pp. 235–244, Mar 2012. [Online]. Available: https://doi.org/10.1007/s11265-011-0587-9

[36] M. Price, J. Glass, and A. P. Chandrakasan, "A 6 mw, 5,000-word real-time speech recognizer using wfst models," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 102–112, Jan 2015.

[37] M. Price, "Energy-scalable speech recognition circuits (doctoral dissertation)," in *Massachusetts Institute of Technology*, 2016. [Online]. Available: http://hdl.handle.net/1721.1/106090

[38] M. Price, A. Chandrakasan, and J. R. Glass, "Memory-efficient modeling and search techniques for hardware asr decoders," in *INTERSPEECH*, 2016, pp. 1893–1897.

[39] M. Riera, J.-M. Arnau, and A. González, "Computation reuse in dnns by exploiting input similarity," in *Proceedings of the 45th Annual International Symposium on Computer Architecture*, ser. ISCA '18. Piscataway, NJ, USA: IEEE Press, 2018, pp. 57–68. [Online]. Available: https://doi.org/10.1109/ISCA.2018.00016

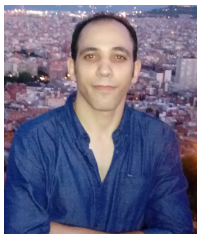
[40] F. Silfa, G. Dot, J.-M. Arnau, and A. González, "E-pur: An energy-efficient processing unit for recurrent neural networks," in *Proceedings of the 27th International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT '18. New York, NY, USA: ACM, 2018, pp. 18:1–18:12. [Online]. Available: http://doi.acm.org/10.1145/3243176.3243184

[41] S. Ortmanns, H. Ney, and X. Aubert, "A word graph algorithm for large vocabulary continuous speech recognition," *Computer Speech & Language*, vol. 11, no. 1, pp. 43 – 72, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230896900224

[42] M. Price, J. Glass, and A. P. Chandrakasan, "A low-power speech recognizer and voice activity detector using deep neural networks," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 1, pp. 66–75, Jan 2018.

[43] M. Freitag and Y. Al-Onaizan, "Beam search strategies for neural machine translation," *CoRR*, vol. abs/1702.01806, 2017. [Online]. Available: http://arxiv.org/abs/1702.01806

[44] S. Abdou and M. S. Scordilis, "Beam search pruning in speech recognition using a posterior probability-based confidence measure," *Speech Communication*, vol. 42, no. 3, pp. 409 – 428, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639303001432



Reza Yazdani is a PhD researcher at the Universitat Politècnica de Catalunya. He is currently pursuing the fourth year of his study on designing high-performance hardware accelerators for the low-power mobile systems. His research interests include the design of high-performance and low-power accelerators for cognitive computing architectures, fault-tolerant design, embedded systems, and VLSI design. He received an MS in computer architecture from the University of Tehran.

Jose-Maria Arnau received Ph.D. on Computer Architecture from the Universitat Politecnica de Catalunya (UPC) in 2015. He is a postdoctoral researcher at UPC BarcelonaTech and a member of the ARCO (ARchitecture and COmpilers) research group at UPC. His research interests include low-power architectures for cognitive computing, especially in the area of automatic speech recognition and object recognition.

Antonio González (PhD 1989) is a Full Professor at the Computer Architecture Department of the Universitat Politècnica de Catalunya, Barcelona (Spain), and the director of the Architecture and Compilers research group. He was the founding director of the Intel Barcelona Research Center from 2002 to 2014. His research has focused on computer architecture and compilers, with a special emphasis on cognitive computing systems and graphics processors in recent years. He has published over 370 papers, and has served as associate editor of five IEEE and ACM journals, program chair for ISCA, MICRO, HPCA, ICS and ISPASS, general chair for MICRO and HPCA, and PC member for more than 130 symposia. He is an IEEE Fellow.