

# Implementing Parallel Speech Decoding in HTK Toolkit by Exploiting Manycore GPU and SIMD Technology

*Jun Cai*

Laboratory of Images, Signals & Telecommunication Devices, Faculté des Sciences Appliquées,  
Université Libre de Bruxelles, Av. F. D. Roosevelt 50, CP 165/51, 1050 Brussels, Belgium

## Introduction

Large vocabulary continuous speech recognition (LVCSR) systems are usually based on continuous density HMMs [1], which are typically implemented using Gaussian mixture distributions. Such statistical modeling systems tend to operate several times slower than real-time, largely because of the heavy computational overhead of the likelihood evaluation and the decoding search [2, 3]. A wide variety of techniques based on different strategies have been devised to speed up the likelihood computation as well as the search procedure [4, 5, 6]. The objective of this research is to investigate how to utilize the improvements of modern computer architecture, especially the manycore graphics processing unit (GPU) and Single-Instruction-Multiple-Data (SIMD) technology, to facilitate high-speed speech decoding. By launching a large amount of threads on a manycore GPU, data-level parallel computation can be easily implemented. Therefore, it is suitable to use general-purpose GPU instead of CPU to perform parallel computations in LVCSR systems [7, 8] to improve the real-time performance without degrading the recognition accuracy. In this research, the speech decoding module HVite in the open source LVCSR toolkit HTK 3.4 [9] has been modified to implement the parallel speech recognition.

## Methods

Two parts of computation, namely the likelihoods evaluation of each speech frame against all active HMM states, and the Viterbi beam search implemented by using a token passing scheme, play the central roles in HVite. Since the number of the active model states usually ranges from 2000 to 6000, the state likelihoods evaluation is computation-intensive and takes typically about 60% of the total decoding time. Reducing the time consumption of likelihoods computation is an important issue in improving real-time performance of the system. Likelihoods of a certain observation frame should be computed successively at different levels. For a specific input frame, computations on an object at a certain level are normally independent of the computations on other objects at the same level [4]. Therefore, the likelihoods can be computed in parallel within each of these levels. This intrinsic parallelism lays the foundations for implementing the likelihoods evaluation in parallel computing schemes on the manycore GPU.

GPUs excel at arithmetic-intensive computations such as dense matrix multiplication. To facilitate the parallel computation on the GPU, a new parallel algorithm has been designed to express the likelihoods evaluation as matrix multiplications. New data structures are correspondingly designed to represent the acoustic model parameters as matrices [8]. A batch scheme is adopted in the parallel algorithm to transfer a bundle of successive observation feature vectors from CPU to GPU during each communication session to reduce the communication frequency between the two processing units. CUDA technology [10] has been used to implement the parallel algorithm. During the decoding, the CPU first transfers a bundle of feature vectors to the GPU, and then creates a CUDA stream to launch a large amount of threads on the GPU to perform the likelihoods computation, upon the completion of which the acoustic scores are fed back to the CPU. After the invocation of the stream, the program control returns immediately to the CPU so that the CPU can operate in parallel with the GPU: based on the acoustic scores fed by the GPU, the CPU performs the Viterbi

beam search operations such as token passing and path pruning on the current window of observations while at the same time the GPU is computing the acoustic scores on the next observation window. In this asynchronous manner, the computational power of both CPU and GPU can be exploited to the maximum extent. Within each thread which runs on the GPU, SSE instructions are employed to enable the cores in the GPU to perform data-level parallel computation based on their SIMD architecture, so as to further speed up the likelihood evaluation.

## Results

The HVite module in HTK3.4 has been modified to implement the parallel speech decoder. Two large vocabulary continuous speech corpora of English, TIMIT and WSJ0, are used to evaluate the performance. The performance is assessed by calculating the time for completing the recognition task. All results are evaluated on a 2.66GHz Intel Core 2 Duo machine with 4GB RAM. The GPU is NVIDIA GeForce 9800 GTX, containing 128 stream cores. The system platform is Ubuntu 9.04 Desktop. It shows that, for the speech decoding by using 32-Gaussian HMMs, the parallel decoder works about 3 times faster than the original version.

## Conclusions

A parallel speech decoder based on the open source LVCSR system HTK 3.4 has been implemented on the PC platform with NVIDIA manycore GPU, by using the CUDA programming model and the SIMD technology. In this parallel decoder, the likelihoods evaluation is performed on the GPU, while the other tasks of Viterbi beam search are performed asynchronously on the CPU. Compared to the serial computing implementation, the parallel decoder achieves a significant speed-up without degrading the recognition accuracy. Therefore, the GPU-based computation is an effective and practical way to improve the real-time performance of LVCSR systems.

## References

- [1] Jurafsky, D. and Martin, J. H., "SPEECH and LANGUAGE PROCESSING: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (2nd Edition)", Prentice Hall, May 2008.
- [2] Gales, M.J.F., Knill, K.M., Young, S.J., "State-based Gaussian Selection in Large Vocabulary Continuous Speech Recognition Using HMM's", IEEE Trans. on Speech and Audio Processing, 7(2):152–161, 1999.
- [3] Ravishankar, M., et al., "The 1999 CMU 10X real time broadcast news transcription system", Proceedings of Speech Transcription Workshop, Maryland, 2000.
- [4] Cai, J., et al., "Efficient Likelihood Evaluation and Dynamic Gaussian Selection for HMM-based Speech Recognition", Computer Speech and Language, 23: 147–164, April 2009.
- [5] Ou, J. L., Cai, J., Lin, Q., "Using SIMD Technology to Speed up Likelihood Computation in HMM-based Speech Recognition Systems", Proc. of ICALIP 2008: 123-127, July 2008
- [6] Fleury, M., Downton, A.C., Clark, A.F., "Parallel Structure in an Integrated Speech-Recognition Network", Lecture Notes in Computer Science, 1685/1999: 995-1004, Jan. 1999.
- [7] Cardinal, P., et al., "GPU Accelerated Acoustic Likelihood Computations", Proc. of Interspeech 2008: 964-967, Sept. 2008.
- [8] Dixon, P. R., Oonishi, T., Furui, S., "Harnessing graphics processors for the fast computation of acoustic likelihoods in speech recognition", Computer Speech and Language 23(4): 510–526, Oct. 2009.
- [9] "HTK Web-Site", <http://htk.eng.cam.ac.uk>. Accessed July 2, 2009.
- [10] "NVIDIA CUDA 2.2 Programming Guide", [http://developer.download.nvidia.com/compute/cuda/2\\_21/toolkit/docs/NVIDIA\\_CUDA\\_Programming\\_Guide\\_2.2.1.pdf](http://developer.download.nvidia.com/compute/cuda/2_21/toolkit/docs/NVIDIA_CUDA_Programming_Guide_2.2.1.pdf). Accessed July 2, 2009.