A GPU Accelerated Speech Recognition System

Daniel May, Mississippi State University

ISCR, John Johnson, Lawrence Livermore National Laboratory

GAIA Project: David Bremer, John Johnson, Holger Jones, Yang Liu, Daniel May, Jeremy Meredith, Sheila Vaidya

Abstract - Graphics Processing Units (GPUs) have become increasingly programmable over the past few years and are able to accomplish more than the specific graphics tasks for which they were designed. Their relatively low price, inherent parallelism, and the fact that their overall performance is increasing faster than for CPUs make them ideal co-processors for certain applications other than graphics. The purpose of this project is to utilize the power of a GPU for a speech recognition application, decreasing the overall processing time while maintaining the same level of recognition performance.

Introduction

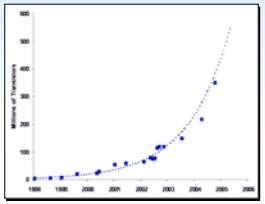
Why use GPUs for general purpose applications?

- low cost
- programmable hardware
- inherent parallelism
- performance is increasing faster than for CPUs

Performance Trends for GPUs

Year	Performance	I/O Speed	Memory Speed
2005	160GFLOPS	2GB/s	30GB/s
2006	400GFLOPS	8GB/s	60GB/s
2007	1TFLOP	-	-
2014	10TFLOPS*	-	3TB/s

* Worlds fastest computer, BlueGene/L at Lawrence Livermore National Laboratory, is 360TF



Performance of Algorithms on GPUs

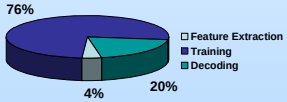
GPUs have shown dramatic speedups of various algorithms, many of which are applicable to speech recognition.

Algorithm	Speedup*
Convolution	100x
Support Vector Machines	18x
Viterbi	17x
Discrete Cosine Transform	17x
Correlation	10x
Fast Fourier Transform	6x

* D. Bremer, J. Johnson, H. Jones, Y. Liu, J. Meredith, and S. Vaidya, "Graphics Processing Units (GPUs) for General Purpose High Performance Computing," SC'05, Seattle, WA.

Three Steps of Speech Recognition

The speech recognition process is made up of three major steps. The graph to the right illustrates a comparison of the time required by each step.



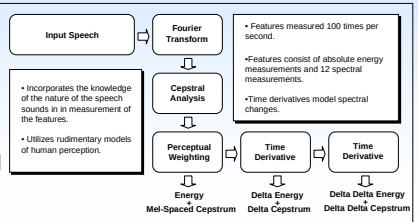
Approach

The Automatic Speech Recognition (ASR) system used for this project was developed by the Institute for Signal and Information Processing at Mississippi State University.

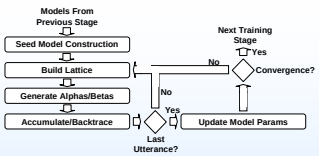
(http://www.cavs.msstate.edu/hse/ies/projects/speech/)

Feature Extraction

All signal frames are independent and their features can be extracted in parallel. These conditions are ideal for implementation on a GPU.



Training

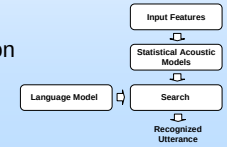


Baum-Welch is used to re-estimate the acoustic models.

Training can be batched into multiple jobs, and the GPU can process these in parallel, generating accumulators for each batch which can be combined after all batches are complete.

Decoding

The decoder uses a Viterbi beam search to hypothesize the transcription of an utterance. The GPU can assist state probability calculations and the state-level search.



Results

The timing results for feature extraction are detailed in the table below. Features were extracted from a subset of the TIDigits corpus.

Algorithm	CPU Speed (sec)	GPU Speed (sec)
Download Texture Memory	N/A	0.40
Absolute Energy	0.13	0.04
Pre-emphasis/Hamming	0.15	0.19
FFT Magnitude	8.40	1.78
Mel-Scale/Cepstrum	6.00	0.46
DCT/Liftering	6.30	0.04
Derivatives	0.07	N/A
Entire Front-end	**21.15**	**2.98**
Speedup GPU/CPU		**7.1x**

Training and decoding are still being implemented.

Discussion

- How portable is the software?
- How constrained is the programming environment in comparison to high-level languages like C++?
- How difficult would it be to implement a Large-Vocabulary Continuous Speech Recognition (LVCSR) application?
- In what other areas of scientific computing can the assistance of GPUs be utilized?

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.