GPU-CPU Multi-Core For Real-Time Signal Processing

Saraju P. Mohanty
Department of Computer Science and Engineering
University of North Texas, Denton, TX 76203.
Email-ID: saraju.mohanty@unt.edu

Abstract

- Modern graphics cards are supported with powerful computational facilities for fast computation of vertex geometry and realistic rendering of 3D.
- Introduction of programmable pipeline in the graphics processing units (GPU) has enabled configurability.
- GPU which is available in every computer has a tremendous feat of highly parallel SIMD processing, but its capability is often under-utilized.
- We analyze computation of general algorithms on GPU.

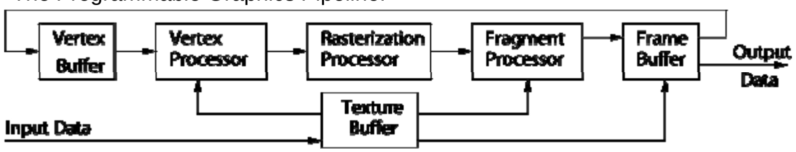
Introduction and Motivation

- Modern GPU architectures are providing ways of configuring the graphics pipeline.
- We want to examine power of the GPUs for general purpose computing.
- GPUs are designed to perform a specific set of operations on large amounts of data.
- In last decade the computing power of the GUPs is increasing much faster than Moore's law.
- The performance gap of GPU and CPU is widening.
- It is shown that a GeForceFX 5900 processor operating at 20 GigaFlops is equivalent to a 10GHz Pentium 4 processor.
- As the GPUs are becoming faster and evolving to incorporate additional programmability, the challenge becomes to provide new functionality without sacrificing the performance advantages over the conventional GPUs.
- GPUs use a different computational model than the classical von Neumann architecture used by the CPUs.
- The questions arises can the GPU and CPU of a PC be used together for high-performance and low-cost computing.
- This can enable low cost high performance computing.

Graphic Processing Unit (GPU) – A Broad View

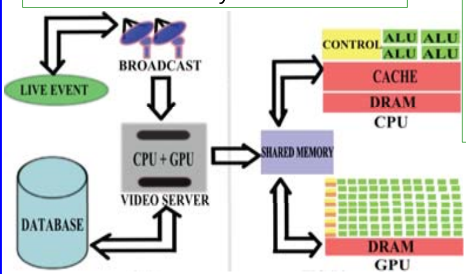
- The architecture of GPU offers a large degree of parallelism at a relatively low cost through the well known vector processing model known as Single Instruction, Multiple Data (SIMD).
- GPU includes 2 types of processing units: vertex and pixel (or fragment) processors.
- The programmable vertex and fragment processors execute a user defined assembly level program having 4-way SIMD instructions.
- The vertex processor performs mathematical operations that transform a vertex into a screen position.
- This result is then pipelined to the pixel or fragment processor, which performs the texturing operations.

The Programmable Graphics Pipeline.



An Application Scenario – Internet Protocol Television

- In IP-TV that broadcasts live events, a server containing GPU and CPU can perform tasks, like compression and copyright protection.
- The data, which arrives at shared memory, is directed by CPU and sent to GPU, then data is mapped into GPUs memory.
- The GPU processes the video data. After GPU finishes, the control from CPU initiates to copy the data back to CPU and the data is stored.
- This approach ensures faster processing for vast amount of data and will not add extra hardware cost to either video providers or receivers.
- Data is sent to GPU as textures, i.e. that data is treated as group of pictures.



IP-TV Broadcasting Showing a Shared Architecture for GPU-CPU Multi-Core Processing in a Video Server.

Computation of Discrete Cosine Transformation (DCT) – A Case Study

- We used a 256×256 gray scale image to compute the DCT using OpenGL API. The input data are eight bit integers. The computation involves block-wise DCT of 8 × 8 pixels.
- The computation of DCT on GPU is performed using one or more rendering passes.
- The working of a rendering pass can be divided into 2 independent stages.
- 1st stage: A number of source textures, the associated vertex streams, render target, the vertex shader and pixel shader are specified.
- 2nd stage: It is the rendering stage which is triggered issuing the DrawPrimitives call.

Experimental Results for Execution Time

Test Cases	CPU Time	GPU Time	CPU Release Time
CPU only	$4.376ms$	Free	Fully Occupied
GPU only	Free	$44.789ms$	Fully Free
CPU + GPU	$2.15ms$	$21.157ms$	50.87% Free

(a) Original Image

(b) DCT Coefficient Image

Computing DCT in GPU.

Conclusions

- GPU is an excellent candidate for performing data intensive signal processing algorithms.
- A direct application of this work is to perform DCT and IDCT on GPU in real-time signal processing to be used for broadcasting where real-time video compression is needed.
- The disadvantage in GPU is in the fact that the data transfer rate from the graphics hardware to the main memory is very slow.
- This bottleneck degrades the performance as it is needed to bring the results to the main memory in the general purpose computation.