GPU Computing and Its Applications

Bhavana Samel[1], Shubhrata Mahajan[2], Prof.A.M.Ingole[3]

[1] *Student, Dept. of Computer Engineering, BVCOEL Pune, Maharashtra, India*
[2] *Student, Dept. of Computer Engineering, BVCOEL Pune, Maharashtra, India*
[3] *Professor, Dept. of Computer Engineering, BVCOEL Pune, Maharashtra, India*

---***---

Abstract - *The graphics processing unit has become important part of today's mainstream computing system. GPU-accelerated computing is defined as the use of a graphics processing unit (GPU) together with a CPU (central processing unit). It developed in 2007 by NVIDIA. Graphics Processing Units (GPUs) have obtained an increasing popularity as a set up for High Performance Computing (HPC) applications. Recent GPUs are powerful graphics engines and also highly threaded parallel computing processors that can achieve long time speedup as compared with CPUs.*

Key Words: GPU's, Rasterization, Pipeline, Composition, Bioinformatics

1.INTRODUCTION

GPU computing is the use of a GPU to do general purpose scientific and engineering computing. CPU and GPU together in a variety of computing model. In our application contain sequential and computationally accelerated part so the sequential part of the application runs on the CPU and the computationally-accelerated part runs on the GPU. The application just runs faster because it is using the high-performance of the GPU to improve performance.

GPU Computing is a type of several of computing – that is, parallel computing with multiple processor architectures. In GPU computing, multicore CPUs are combined with many-core GPUs to achieve higher performance. Today's CPUs have 4, 6, 8 or even 12 cores, although GPUs have up to 512 cores in a single chip. Today's computing applications typically have data intensive regions with a large amount of inbuilt parallelism. GPUs speed up these parallel Regions by spreading the calculation over hundreds of cores. This can deliver significant speed up for many applications. Our time is precious, which is why performance is critical. So you need to focus on what's important. To add GPU acceleration, simply identify and accelerate the "performance-critical areas" in your application. The rest can run as-is on the CPU. There's no need for an entire application to be ported to the GPU. The important point here is that in a hybrid architecture where GPUs and CPUs are working together, developers accelerate performance-critical functions on the GPUs in the system.

2. ARCHITECTURE OF GPU COMPUTING

2.1 GPU Pipeline

A list of geometric primitives, triangles, in a 3-D world coordinate systems are the input to the GPUs. By using many steps, those primitives are shaded and mapped onto the screen, where they are assembled to create a final picture. Figure 1 is shows the pipeline process in the GPU. The GPU pipeline block description is given below.[1]

2.1.1. Vertex Operations: In Vertex operations raw 3D geometry transform into the 2D plane of monitor. Vertex pipelines also used to eliminate unnecessary geometry by detecting parts of the scene that are hidden by other parts and simply discarding those parts.[1]

2.1.2 Primitive Assembly: In primitives assembly the vertices are assembled into triangles, the fundamental hardware-supported primitive in the GPUs.[1]

2.1.3 Rasterization: In Rasterization contain the process of determining which screen-space pixel locations are covered by each triangle. Each triangle generates a primitive. That primitives called a "fragment" at each screen-space pixel location that it covers. Each pixel's color value may be calculated from several fragments, because many triangles may overlap at any pixel location[1]

2.1.4 Fragment Operations: Each fragment is shaded to determine its final color using color information from the vertices and possibly fetching additional data from global memory in the form of textures (images that are mapped onto surfaces),. Just as in the vertex stage, each fragment can be calculated in parallel. This stage is typically the most computationally demanding stage in the graphics pipeline.[1]

2.1.5 Composition: In composition fragments are assembled into a final image with one color per pixel, usually by keeping the closest fragment to the camera for each pixel location.[1]

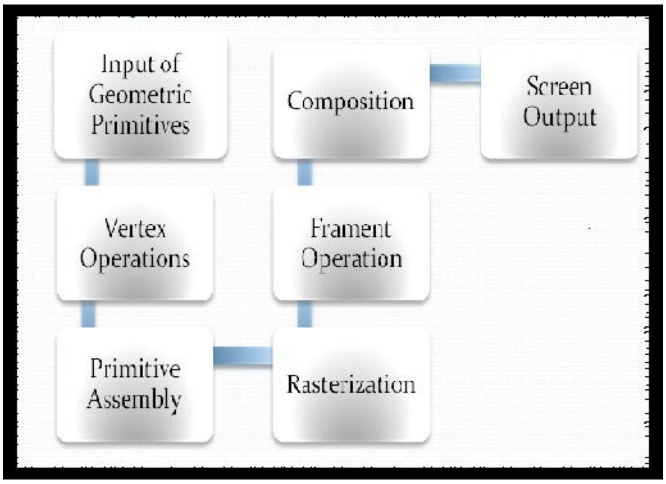

Fig -1: GPU Pipeline

3. APPLICATIONS OF GPU COMPUTING

3.1 Bioinformatics

Sequencing and protein docking are very calculating intensive tasks that see a larger performance benefit by using CUDA enabled GPU. There is a relatively bit of ongoing work on GPUs for a range of bioinformatics and life sciences codes.

3.2 Computational Finance

NVIDIA TESLA GPU acceleration offers financial service firms a competitive advantage by enabling applications, such as the widely deployed Monte Carlo simulation, to run faster. Calculating pricing and risk for compound options and OTC derivatives within seconds versus hours allows you to run more simulations thereby increasing the quality of results and more confidence in data.

3.3 Computational Fluid Dynamics

Several ongoing projects on Navier-Stokes models and Lattice methods have shown very large speedups using CUDA-enabled GPUs. This work is highlighted below starting with some charts.

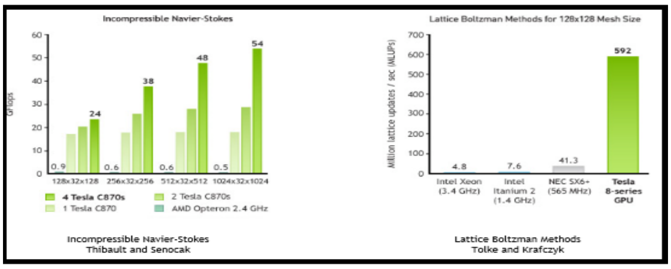

Fig 2. Computational Fluids Dynamics

3.4. Data Science, Analytics, and Databases

A increasing number of customers are using GPUs for big data analytics to make better, real –time business decisions.

3.5. Defense and Intelligence

The defense and intelligence community heavily rely on accurate and timely information in its strategic and day to day operations. Intelligence gathering and assessment are essential parts of these activities that comprise data coming from the number of move away sources such as satellites, surveillance cameras, UAVs, and radar. Converting the collected raw data into actionable information required a significant infrastructure-people, computer hardware and software, power and facilities –all of which are limited. NVDIA graphics card shows a "Game Changing" technology that dramatically increases productivity while reducing cost, power and facilities. Using GPU to augment existing processing systems is an established practice computing centers and research institutes around the globe to close the gap between the growing demands of their scientist, engineers and the compute capacity of current IT infrastructure. In the following chart, we highlight work completed on NPP and CuFFTs

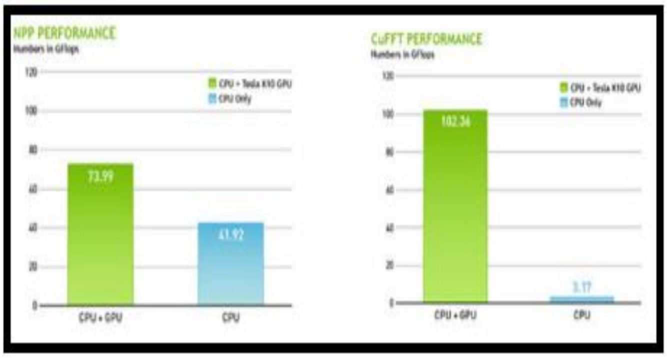

Fig 3. NPP Performance and CuFFT Performance

3.6 Electronic Design Automation

EDA involves a varied set of software algorithms and applications that are required for the design of complex next generation semiconductor and electronics products. The increase in VLSI design complexity poses significant challenges to EDA; application performance is not scaling effectively since microprocessor performance gains have been hampered due to increase in power and manufacturability issues, which accompany scaling. Digital systems are usually validated by distributing logic simulation tasks among huge compute farm for weeks at a time. Yet, the performance of simulation often falls behind, leading to incomplete verification and missed function bugs. It in indeed no surprise that the semiconductor industry is always seeking for faster simulation solutions.

Recent trends in HPC are increasingly exploiting many core GPUs to a competitive advantage through the use of such GPUs as massively-parallel CPU co-processor to achieve speedup of computationally intensive EDA simulations including verilog simulation, signal integrity & Electromagnetic, computational Lithography, SPICE circuit simulation and more.

3.7. Imaging and Computer Visions

Computer vision and image processing algorithms are computationally intensive. With CUDA acceleration, applications can achieve interactive video frame-rate performance.

3.8. Machine Learning

Data scientists in both industry and academia have been using GPUs for machine learning to make innovative improvements across a multiplicity of applications including image classification, video analytics, speech recognition and natural language processing. In particular, deep learning-use of difficult, multi level "deep" neutral networks to create system that can perform feature detection from that has been seeing significant investment and research.

Although machine learning has been around for decades, two relatively recent trends have sparked general use of machine learning :the availability of massive amounts of training data, and powerful and efficient parallel computing provided by GPU Computing. GPUs are used to train these deep neutral network using far larger training sets, in an order of magnitude less time, using far less datacenter infracture. GPUs are also being used to run these trained machine learning models to do classification and prediction in the cloud, supporting far more data volume and throughput with less power and infrastructure.

Previous times adopters of GPU accelerators for machine learning include many of largest web and social media companies, along with top tier research institutions in data science and machine learning. Along thousands of computational cores and 10-100x application throughput compared to CPUs alone, GPUs have become the processor of choice for processing big data for data scientist.

3.9. Media and Entertainment

Digital video professionals are always looking for easiest, quick ways to deliver great results. For example, working with more video streams as well as high resolution camera format, adding effects and not slowing the system to a crawl, or working interactively with compound modeled characters or scenes that contain real world physics and effects.

Using the most powerful workstation can make a remarkable difference in delivering the fastest results with the smallest amount possible production costs.

Multi-GPU systems address these challenges by combining the horsepower of two or more professional GPUs to deliver both high performance graphics and parallel processing. This is truly unique technology that helps transform video editing, effects, digital animation, rendering and media creation workflows and helps you realize your final project on a fraction of the time of a traditional single-GPU setup.

3.10. Medical Imaging

Medical Imaging is one of the initial applications to take advantage of GPU Computing to get acceleration. The use of GPUs in this field has developed to the point that there are several medical modalities shipping with NVIDIA's Tesla GPUs now.

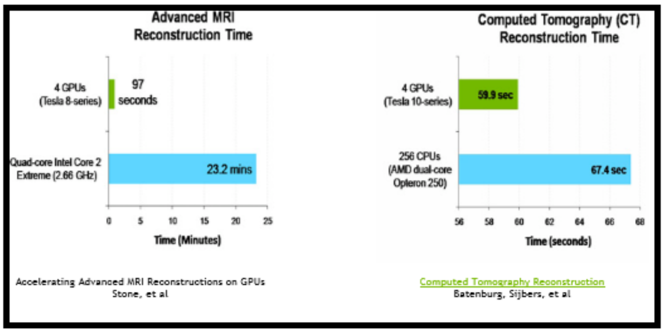

Fig 4. Advanced MRI reconstruction time and Computed Tomography Reconstruction time

With incorporation of GPU computing into the image processing workflow Automated segmentation algorithms gets the most benefit. There are also new visualization paradigms, such as stereoscopic 3D, which have been made possible by the continued increase in computational capacity of GPUs. These two key functions of modern GPUs will enable medical images to keep pace with the increasing size of scan data sets while allowing for new and innovative analysis and interaction paradigms.[2]

3.11. Structural Mechanics

By using CUDA-enabled GPUs ANSY, Abaqus, MSC Nastran, IMPETUS Afea and additional structural mechanics applications profit very much.

3.12. Weather and Climate

Work computation fluid dynamics application such as WRF. Full form of WRF is Weather Research and Forecasting model and tsunami simulations. WRF has shown tremendous speedups that enable savings in time and improvements in correctness.

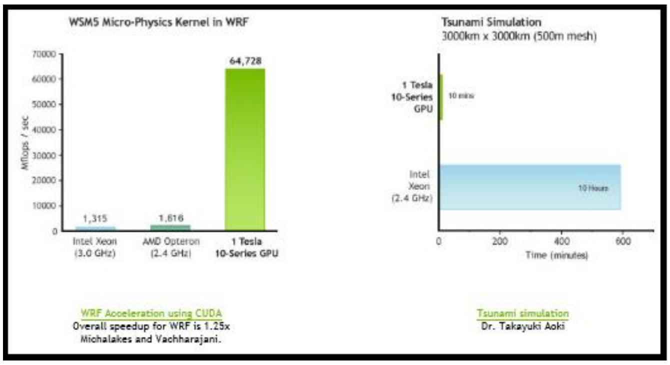

Fig 5. Acceleration using CUDA and Tsunami Simulation

4. CONCLUSIONS

GPU computing is at the tipping point. Using GPU computing our system performance is increases. Our application runs faster. Single-threaded processor performance is no longer scaling at significant rates. Thus, we must use parallelism for the increased performance required to deliver more value to users. A GPU that's optimized for throughput delivers parallel performance much more powerfully than a CPU that's optimized for latency.

REFERENCES

[1] John Nickolls, William J. Dally,"The GPU Computing Era" Published by the IEEE Computer Society in 2010
[2] S. Moulik, W. Boonn, M.V. Knopp, PACS3D, Proc RSNA Informatics (2010).