# Technical Report: GPU- Based Parallel Computing

Kaali Lovell- UO1826674

Professor Nancy Turbé

CS610- 22597

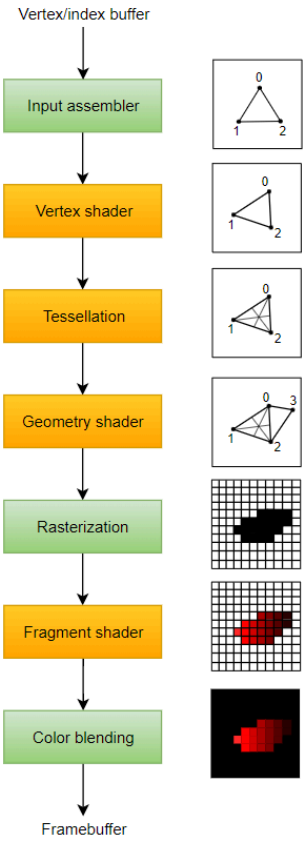
01 May 2024

Table of Contents

**History of Graphical Processing Units capabilities from 1990 to present**

**What is the GPU?**

The graphical processing unit(GPU) was derived from the Central Processing Unit(CPU). A CPU is a “general purpose” processor that runs programs that perform complex tasks. A CPU usually has 2- 16 cores. You can think of cores as the brains of the operation; they control how data is carried out throughout the system. CPU cores have a control unit, arithmetic-logic unit, and memory. The control unit allows the computer to communicate with other parts of the computer. The Arithmetic-logic unit executes arithmetic and logical operations. The memory holds data so it can be used for another time. When you increase the number of cores with the CPU, parallelism increases. Parallelism allows a machine to compute many tasks simultaneously. This is where the GPU comes into play. GPUs contain many more cores than CPUs. GPUs take the load off of the CPU so the CPU can focus on other tasks and the GPU act as a buffer between the CPU and the display.




GPUs are contained within graphics cards. A graphics card is a circuit board designed to handle graphics operations. It has its own memory (VRAM, Video RAM), which stores the data it's currently processing. Typically, a graphics card stores 8- 10 gigabytes of memory. The memory in a graphics card is similar to the computer’s main memory, except it is highly optimized for bandwidth, meaning it is faster, uses less power, and generates less heat. It contains capacitors that regulate the

voltage of the components of the card. The capacitors are connected to an external power supply. It also includes a cooling fan controlled by the graphics card. It turns on and off when needed. The graphics card is connected to the CPU and can be connected to other graphics cards by a scalable link interface(SLI) or crossfire technology. The graphics card is also connected to a monitor by different ports; the ports differ by how many times the pixel on the screen can be refreshed. The display ports and DVI(digital visual interface) have a refresh rate of up to 144 hertz(Hz). As mentioned, the GPU is stored inside the graphics card.

Within the GPU, all of the graphics processing occurs. GPU is made to handle the graphics or rendering pipeline. The pipeline consists of the following steps, illustrated to the right. The graphics pipeline is a series of stages like an assembly line. It takes in numerical data that consists of 3 coordinates, meaning the axis are (x,y,z), which represents a 3-dimensional area, and converts it into a 2-dimensional image that can be displayed on screen. Millions of calculations occur every second for interactive games or virtual reality. This happens because the scene constantly changes quickly in response to the input given, which is called real-time rendering.

In a 3D scene, each object visualized is made of simple polygons, primarily triangles. A triangle contains three vertices, and each vertice has a 3-point coordinate. The vertices are stored as vectors. The vectors are then manipulated using matrix arithmetic and linear algebra. Compared to the CPU mentioned before, the GPU has over a hundred more cores than the CPU. This allows the GPU to calculate much faster and calculate many operations simultaneously. Each core works on millions of data streams being processed simultaneously or in parallel. It contains a single instruction multiple data(SIMD), which allows the computation of various datasets with

one instruction. The great thing about having multiple cores is that if one core is lagging or processing data, the input can move to another core, which maximizes throughput. A GPU has many levels of cache memory. Some GPUs have cores and cache memory dedicated to specific tasks in the graphics pipeline. Some GPUs even have additional cores dedicated to real-time ray tracing, which calculates the paths of light rays in a 3D scene and how the light interacts with different materials. Now that we have an overall view of the GPU, let's understand how it all started.

## The 1990's

In 1993, SGI(Silicon et al.) released the RealityEngine board for graphics processing. The CPU completed the first half of the graphics pipeline, and the GPU completed the latter half. SGI was an American manufacturer of computer hardware and software. By the mid-90s, companies like NVIDIA and ATI started manufacturing and distributing 3D graphic boards. There was still the issue of "GPUs still only [outputting] one pixel per clock cycle, meaning CPUs could still send more triangles to the GPU than it could handle," which one can only imagine that it caused a lot of contention(McClanahan, 2010). The term "GPU" didn't exist during this time since the CPU was still handling some of the work. In 1999, the NVIDIA GeForce256 was released. It was one of the first cards to have the entire GPU pipeline operated. When this occurred, this was considered the "first true GPU." "This generation of cards was the first to use the new Accelerated Graphics Port (AGP) instead of the Peripheral Component Interconnect (PCI) bus and offered new graphics features in hardware, such as multi-texturing, bump maps, light maps, and hardware geometry transform and lighting"(McClanahan, 2010). With all of these advancements, there were still downfalls. There was the issue of the "fixed

function" pipeline. This means it couldn't be modified once data was sent to the GPU. The graphics cards were primarily developed for game visual rendering during this time. Companies like NVIDIA and ATI stood the test of time due to the constant adaption of consumers' appetites, but companies like SGI went bankrupt. It is assumed that SGI's credibility was damaged due to overpromoting what they were able to deliver. SGI promised more advanced migrations and features but couldn't provide them. SGI also tried to sell workstations running WindowsNT alongside workstations that ran IRIX but weren't able to justify their exorbitant prices(Silicon Graphics).

**The 2000's**

In 2001, NVIDIA released the GeForce 3. This GPU hardware allows a programmer to program parts of the GPU pipeline. Users didn't have to worry about the "fixed function" pipeline during this time. Of course, other companies, such as ATI and Microsoft, were also creating their own versions of the cards. In 2002, NVIDIA GeForce FX was released, and this graphics card was fully programmable. "In 2003, the first wave of GPU computing started to come about with the introduction of DirectX 9, which took advantage of the programmability of GPU hardware, but not for graphics. Full floating point support and advanced texture processing started to appear in cards"(McClanahan, 2010). In 2004, the GeForce 6 was released. The GPU had increased memory, high floating point precision, and texture accesses were implemented. By this time, GPUs were being used for more than graphics. "When viewed as a graphics pipeline, the GPU contains a programmable vertex engine, a programmable fragment engine, a texture engine, and a depth-compare/blending-data write engine. When viewed alternatively as a processor for non-graphics applications, a GPU can be seen as a large amount of programmable

floating-point horsepower and memory bandwidth that can be exploited for compute-intensive applications completely unrelated to computer graphics"(McClanahan, 2010). In 2006, NVIDIA released the GeForce 8; this GPU allowed for parallel processing. They implemented a Streaming Multiprocessor(SM), which handled vertex, pixel, and geometry computation, as well as a geometry shader. NVIDIA then released CUDA(Compute Unified Device Architecture), a physical processing core designed to take on multiple calculations simultaneously. The higher the core amount, the better the GPU performance. By this time, users were using graphics cards and GPUs for more than just gaming; they used them for applications that required complex imaging techniques such as medical imagery.

**2010 & Beyond**

In 2010, NVIDIA released an improved version of the Fermi gaming card, which they released in late 2009. This card contained 512 CUDA cores, which provided GPGPU(General Purpose Computing on Graphics Processing Units) computing. There were better and faster simultaneous calculations and higher memory bandwidth. GPUs have evolved from a single core, fixed pipeline, and graphical rendering technology to a multi-programmable core for general-purpose technology. The figure below shows how GPU technology(by NVIDIA) has advanced throughout the years. Companies like NVIDIA have been able to stand the test of time due to their current adaption of consumers' demands. I once watched an interview with NVIDIA's CEO, Jensen Huang, who said NVIDIA is not in competition with price; they compete with quality. If a developer has that mentality when releasing products, you can't go wrong. You know you'll get a high-quality product every time. I believe NVIDIA has been primarily focused on providing the most dynamic GPU rather than trying to provide a GPU that is affordable for

consumers. In the end, this is what consumers want. They want a graphics card that provides the most precise, smooth, fast, seamless visualizations possible.

Table 1. NVIDIA GPU technology development.

| Date | Product | Transistors | CUDA cores | Technology |
|---|---|---|---|---|
| 1997 | RIVA 128 | 3 million | — | 3D graphics accelerator |
| 1999 | GeForce 256 | 25 million | — | First GPU, programmed with DX7 and OpenGL |
| 2001 | GeForce 3 | 60 million | — | First programmable shader GPU, programmed with DX8 and OpenGL |
| 2002 | GeForce FX | 125 million | — | 32-bit floating-point (FP) programmable GPU with Cg programs, DX9, and OpenGL |
| 2004 | GeForce 6800 | 222 million | — | 32-bit FP programmable scalable GPU, GPGPU Cg programs, DX9, and OpenGL |
| 2006 | GeForce 8800 | 681 million | 128 | First unified graphics and computing GPU, programmed in C with CUDA |
| 2007 | Tesla T8, C870 | 681 million | 128 | First GPU computing system programmed in C with CUDA |
| 2008 | GeForce GTX 280 | 1.4 billion | 240 | Unified graphics and computing GPU, IEEE FP, CUDA C, OpenCL, and DirectCompute |
| 2008 | Tesla T10, S1070 | 1.4 billion | 240 | GPU computing clusters, 64-bit IEEE FP, 4-Gbyte memory, CUDA C, and OpenCL |
| 2009 | Fermi | 3.0 billion | 512 | GPU computing architecture, IEEE 754-2008 FP, 64-bit unified addressing, caching, ECC memory, CUDA C, C++, OpenCL, and DirectCompute |

## Analysis of Present-day leading GPU Products(NVIDIA, Intel, and AMD)

### NVIDIA GeForce RTX

NVIDIA GeForce RTX "is a professional visual computing platform created by Nvidia, primarily used in workstations for designing complex, large-scale models in architecture and product design, scientific visualization, energy exploration, and film and video production, as well as being used in mainstream PCs for gaming"(Wikipedia). The GPU supporting the GeForce RTX is called the Ada Lovelace, named after the mathematician who became one of the first female computer scientists. It allows for real-time ray tracing, which uses physics to simulate how light *behaves* in the real world. It generates interactive images to react to lights, shadows, reflections, etc. We will review the current versions of the NVIDIA GeForce RTX.

The RTX 4060 version cores range from 3072-4352 and contain up to 16 GB GDDR6 vRAM of memory. vRAM, which stands for video random access memory, is the GPU RAM rather than the motherboard's RAM. It acts as temporary storage for the data in graphics rendering. The more vRAM, the better the frame rates and performance. It allows a user to see visuals at full HD or 1080p resolution. With this GPU, you can exceed 60 frames per second. The card is capable of rate tracing, which allows for advanced lighting effects. It lacks horsepower compared to the other versions. Another model of the GeForce series available is the RTX 4070 Series. Its core number ranges from 5888-8448 CUDA cores. It contains 12-16 GB GDDR6X vRAM with a memory bandwidth of up to 256-bit. It allows gamers to play most modern-day games with the maximum settings of 2K or 1440p resolution. Its FPS ranges from 80-100. FPS stands for Frames Per Second and measures the rate at which the computer can produce and render images. The higher the FPS, the smoother the visuals are. With this GPU, you can exceed 200 FPS. There also exists the RTX 4080. Its cores range from 9728-10240 CUDA cores, 16 GS GDDR6X vRAM, and 265-bit memory bandwidth. The best version of the RTX is the RTX 4090. It's considered the most powerful GPU on the market. It contains a CUDA core of 16384, 24 GB GDDR6X vRAM, and 384- bits of memory bandwidth. It's designed for 4K gaming and provides over *triple-digit* FPS(SoulOfTech).

**Intel Arc**

The Intel Arc is a brand of GPUs designed by Intel. They were primarily designed for the gaming market with a range of 1440- 1080p. The Intel Arc uses the Intel $X^e$ GPU architecture($X^e$ -HPG variant). Intel $X^e$ , which stands for "eXascale for everyone," is a current version of Intel's GPU architecture. The Intel $X^e$ contains a new ISA(Instruction Set Architecture). ISA is a set of

instructions that defines the operations a process can use to perform operations. The ISA is important because it defines the functionalities of a processor; developers can write programs that can run on different parts of the computer's architecture. It contains a family of microarchitectures such as $X^e$-LP, $X^e$-HPG, $X^e$-HP, and $X^e$-HPC. The $X^e$-LP is the low-power variant of the $X^e$ architecture, which includes features such as "Sampler Feedback, Dual Queue Support, DirectX 12 View Instancing Tier2, and AV1 10-bit fixed function hardware decoding." The $X^e$-HPG is the high-performance architecture variant specializing in performance and supporting *accelerated* ray tracing. The $X^e$-HP and the $X^e$-HPC specialized in high-performance computing and multi-tile scalability(Intel Xe).

The Intel Arc A580, compared to the A750, is 15% slower. The Intel Arc A580 has an average light rendering of 85.6 fps and a reflection of 132 fps. The Arc A750 has a light rendering of 88.9 fps and a reflection of 140 fps. The Intel Arc A770 has a slightly faster speed than the A750 by 9%. It has a light rendering of 92.2 fps and a reflection of 150 fps. Intel may not provide the best performance compared to NVIDIA and AMD, but you can get a decent-performing graphics card reasonably priced. Intel's GPU does live up to its name $X^e$-"eXascale for everyone."

**AMD Radeon RX**

AMD Radeon RX is a series of GPUs developed by AMD(Advanced Micro Devices). It's the first generation of AMD GPUSs that support accelerated real-time ray tracing. The GPUs I will compare are built on the AMD RDNA architecture. The RDNA(Radeon DNA) is a GPU microarchitecture and ISA. The AMD Radeon RX 7900 XTX has 6144 CUDA cores, 24 GB

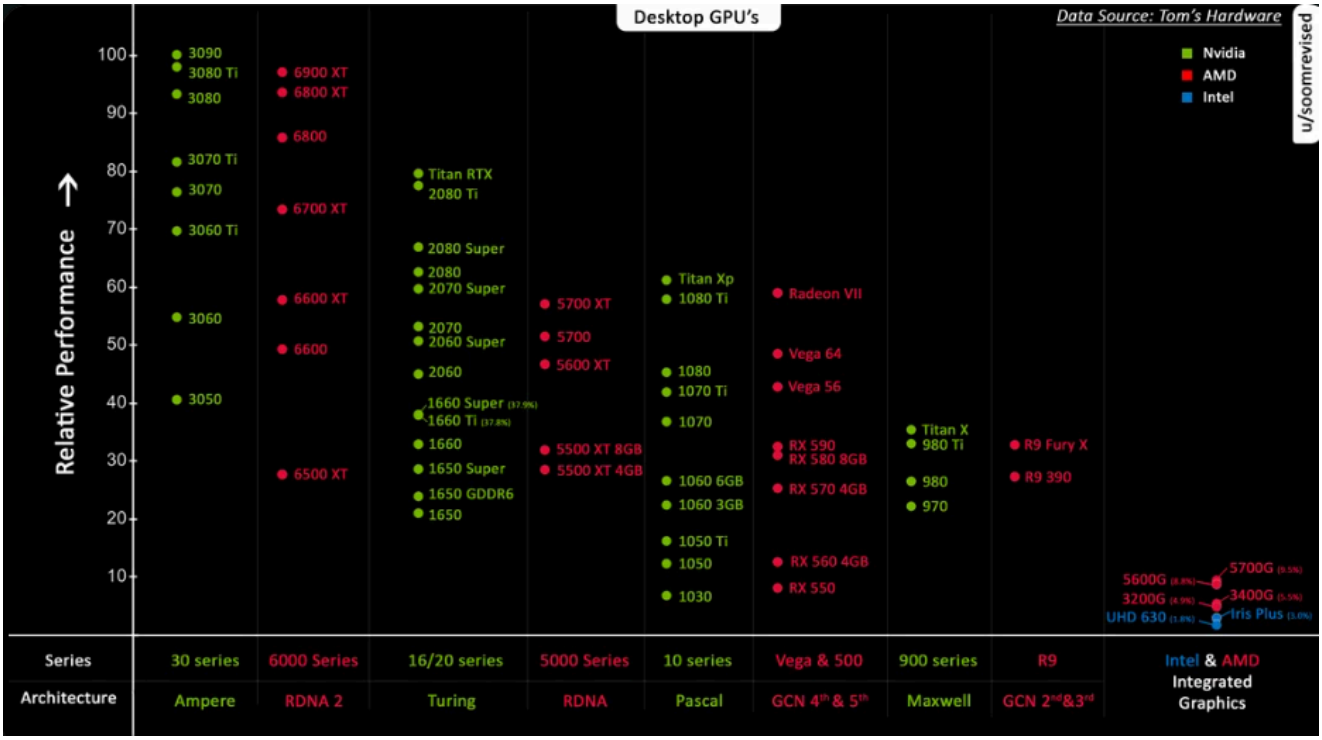
GDDR6 memory, a clock speed of 2270 MHz, 57,700 million transistors, and supports a memory bandwidth of up to 960 GB. It renders up to 252 fps. The AMD Radeon RX 7900 XT has 5376 CUDA cores, a clock speed of 2000 MHZ, 57,700 million transistors, and a memory bandwidth of up to 800 GB. The AMD Radeon RX 6950 XT has 5120 cores, a clock speed of 1925 MHz, 26800 million transistors, renders up to 220 fps, and supports a memory bandwidth of up to 567 GB. The RX 7900 XT is 11% faster than the 6950 XT. It also renders more FPS by up to 26%. The RX 7900 XTX, compared to the RX 7900 XT, is 17% faster. It also renders more FPS by up to 21%.

**NVIDIA GeForce RTX v. Intel Arc v. AMD Radeon**

Now that we have stats on the current competitive GPUs let's compare their best GPU versions with each other. The GeForce RTX 4090 outperforms the Radeon RX 7900 XTX by 25%. The GeForce has more cores, faster clock speed, transistors, and memory bandwidth. The Intel Arc A770 is 81% slower than the Radeon RX. The GeForce RTX 4090 has sold over 160K units with a whopping price tag of about $1800. Compared to NVIDIA's customers, the price tag is nothing for Meta Platform Inc., Microsoft Corp, Amazon.com Inc., and Alphabe Inc. Meta and Amazon uses NVIDIA's chip for their research systems to power their machine learning systems. Amazon and Google also use their chips for their cloud computing services. The AMD Radeon RX 7000 series has sold over 2.3 billion units for a price ranging from $270 to $1000. Yes, that sounds amazing, but they have sold this amount since 2000. Also, more of these versions have sold as many as they did because they're more consumer-friendly to the public. AMD's customers also include Microsoft, Meta, and Oracle. Oracle uses AMD for its cloud services, just like Microsoft and Amazon. I think the Intel Arc does not even stand a competitive chance

against the other two graphics cards. Intel Arc sales haven't been as successful as NVIDIA and AMD. In 2022, there were many issues with their new release of GPUs, but Intel has recently debugged those issues and is selling units for a humble price tag of $350. Dell, Lenovo, and HP are some of Intel's customers.

Below is a chart of comparisons between NVIDIA, Intel, and AMD's GPU performance.




## GPU Developers Tools CUDA, OpenCL, OpenMPI

Compute Unified Device Architecture(CUDA), introduced by NVIDIA in 2006, is a computing platform that allows you to use your GPU in many ways. It is a parallel computing platform and API(application programming interface) that allows software to use certain GPUs for GPGPU. It allows you to access the GPU's instruction set and parallel computational elements. CUDA is

designed to work with C/C++ by ‘CUDA C/C++,’ Fortran by ‘CUDA Fortran,’ and Python by ‘CUDA Python.’ CUDA provides a low and high-level API, which allows for two or more computer components to communicate with each other. CUDA is more advantageous than other GPUs because it provides scattered reads, unified memory, fast shared memory, faster downloads, and readbacks to and from the GPU. It also supports whole integer and bitwise operations. “CUDA programming paradigm is a combination of both serial and parallel executions and contains a special C function called the kernel, which is in simple terms a C code that is executed on a graphics card on a fixed number of threads concurrently”(Exterman, 2021). CUDA is, unfortunately, only supported by NVIDIA products(CUDA).

Open Computing Language(OpenCL) is a low-level API for parallel computing of heterogeneous computing that runs on CUDA GPUs. Apple and the Khronors group launched it. It was created because they wanted a version of CUDA that wasn't restricted to NVIDIA’s products. Using OpenCL, users can write programs that execute on the GPU without sending their algorithms to a graphics API. “penCL offers a portable language for GPU programming that uses CPUs, GPUs, Digital Signal Processors, and other types of processors. This portable language is used to design programs or applications that are general enough to run on considerably different architectures while still being adaptable enough to allow each hardware platform to achieve high performance”(Exterman, 2021). Three components of OpenCL are the kernel, the program, and the application queue kernel execution instances. The kernel executes OpenCL devices and host programs that execute the host. The host programs manage the executions of the kernels. The host defines the collection of OpenCL devices to be used by the host, the functions that run on the OpenCL device, the program source that implements the

kernels, and the set of memory objects visible to the host and the OpenCL devices. (Munshi 2009). OpenCL provides a language that resembles the C language to work directly with the GPU.

OpenMPI, which stands for Message Passing Interface implementation, was designed by researchers to achieve inter-process communication between multiple computers in a network. It represents the merger of three MPI implementations(FT-MPI from the University of Tennessee, LA-MPI from Los Alamos National Laboratory, and LAM/MPI from Indiana University). OpenMPI is compatible with CUDA because it allows communication between distributed processes used in HPC(High-Performance Computing). Users usually combine OpenMPI with CUDA because they need to be able to work with large datasets that are too large to fit into GPU's memory or to scale across multiple node GPU applications. We will discuss this in the CUDA-aware MPI section.

We will provide a more in-depth analysis of Open CL and CUDA. OpenCL can be run on more devices than CUDA. OpenCL can also be compiled at runtime, making OpenCL's runtime longer. CUDA, however, has computing characteristics similar to those of GPU, which in turn provides better performance. CUDA's only vendor is NVIDIA, while OpenCL has many vendors, such as AMD, NVIDIA, and Apple. With this being said, OpenCL is more portable than CUDA since CUDA only runs on NVIDIA GPUs, and OpenCL is an open industry standard. OpenCL works with many operating systems (OS) and hardware, while CUDA only works on OS that uses NVIDIA hardware. OpenCL provides CPU fallback, while CUDA does not. CUDA users use if-else statements in their code to alert them of the presence or absence of their GPU device at runtime. CUDA is also owned and developed by NVIDIA, while OpenCL is

open-source. CUDA has libraries that support high-performance math routines. One can say the CUDA library is more mature and sophisticated than the OpenCL library(Exterman, 2021). Below, you will find a comparison table of CUDA and OpenCL that simplifies the differences between both technologies.

| Comparison | CUDA | OpenCL |
|---|---|---|
| **Performance** | No clear advantage, dependent code quality, hardware type and other variables | No clear advantage, dependent code quality, hardware type and other variables |
| **Vendor Implementation** | Implemented by only NVIDIA | Implemented by TONS of vendors including AMD, NVIDIA, Intel, Apple, Radeon etc. |
| **Portability** | Only works using NVIDIA hardware | Can be ported to various other hardware as long as vendor-specific extensions are avoided |
| **Open Source vs Commercial** | Proprietary framework of NVIDIA | Open Source standard |
| **OS Support** | Supported on the leading Operating systems with the only distinction of NVIDIA hardware must be used | Supported on various Operating Systems |
| **Libraries** | Has extensive high performance libraries | Has a good number of libraries which can be used on all OpenCL compliant hardware but not as extensive as CUDA |
| **Community** | Has a larger community | Has a growing community not as large as CUDA |
| **Technicalities** | Not a language but a platform and programming model that achieves parallelization using CUDA keywords | Does not enable for writing code in C++ but works in a C programming language resembling environment |

CUDA-Aware MPI combines CUDA and MPI, which “allows developers to pass pointers of GPU memory to MPI APIs without explicitly handling the data movement between CPU and GPU at application level”(Manian 19). CUDA-Aware MPI makes working with CUDA and MPI applications easier. “All operations required to carry out the message transfer can be pipelined, and acceleration technologies like GPUDirect can be utilized by the MPI library transparently to the user”(Kraus, 2013).

The great thing about the internet is that there are many resources where an academic can learn about these GPU developer tools for free. Some of these tools are open-sourced, so there is a community out there that can help you get started. There are also Parallel computing courses an

academic can take. Even NVIDIA provides education and training for those yearning to learn about CUDA.

## Analysis: Applied Use of Modern-day GPU-based parallel programming

### . Virtual and Augmented Reality in Massively Multiplayer Online games (MMOs)

Virtual reality(VR) is all in the name; it's the human reality of a virtual scene. It is an entirely virtual, three-dimensional, computer-generated environment. Augmented reality(AR) combines the three-dimensional digital world and the actual physical world. AR involves the real world to some extent, and VR is entirely digital. VR and AR are provided on many devices, such as VR headsets, glasses(Google Glass), and phone apps(Pokemon Go). The author of the "What is Virtual Reality" post said it best, "virtual reality entails presenting our senses with a computer-generated virtual environment that we can explore." From personal experience, using the Meta Virtual Reality headset dramatically interferes with your senses, such as balance and perception. Although you know, based on the graphs, that it is a game, your brain perceives some parts of the game as reality. Going into the neurological research for this topic would be exciting, but for now, we'll stick to how GPU-based parallel programs are applied to VR and AR. The platform I will focus on is Pokemon Go, which uses AR.

Modern smartphones are more advanced than ever. Some are more advanced than the computers that helped Neil Armstrong land on the moon almost 55 years ago. Modern-day smartphones contain multi-cores, can have speeds of more than 3 GHz, and have a memory capacity of 12GB or higher. For example, the Apple iPhone 15 Pro, which uses the A17 Pro chip(Apple's GPU), can perform up to 35 trillion operations per second with 16 cores. They are also built with sensors that can be used to detect their environment. For example, you have

GPS(Global Positioning System), which detects your location; facial detection sensors(your iPhone can sense when your face is too close/far from the screen); light detection sensors(your phone dims its screen lighting if the room is dark and vice versa). Pokemon Go took advantage of these advanced phone features. It combined "augmented reality, edge computing, ubiquitous smartphone usage, and location-based massively multi-player features" to create a successful platform for users(Shea 2017). Going forward, I would like to note that for most of the discussion of the Pokemon Go application on Android devices, I will be using much of the research from the Shea 2017 article unless noted otherwise. Pokemon Go works by having a player interact with the physical world while looking for virtual characters.

The researcher of the "Location-Based Augmented Reality" article compared phone usage of the augmented reality game to other phone applications. They tested the AR usage on the Moto G 3rd Generation Android device for 30 minutes and found that Pokemon Go used a power consumption of 3544 mW. The figure to the right illustrates the apps that consumed power and by how much percent. These stats do depend on the device hosting the AR platform. The Moto G red generation included a " Quad-core 1.4 GHz Cortex-A53 CPU, an Adreno 306 GPU, 2 GB of RAM and 16 GB of internal flash memory"(Shea 2017). Although

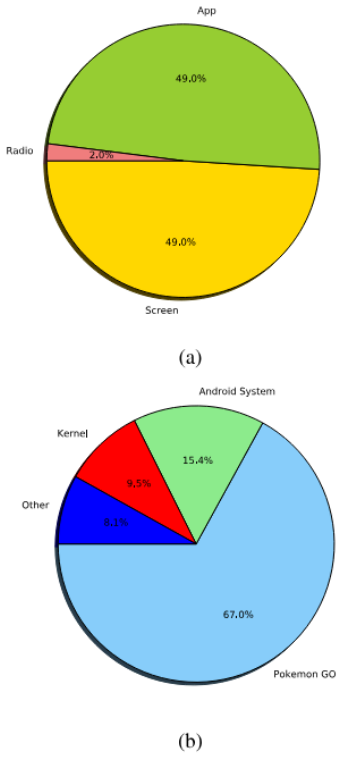



**FIGURE 3.** **MotoG: Power Consumption (Total: 3544 mW). (a) Power total. (b) Power app.**

this is just one example of AR using one device, it shows that AR has a high energy consumption. One may ask, why does using the screen for AR use up so much power? The researchers used Android Studio to help profile the application. They found that 80% of the CPU used the function called UnityPlayer.nativeRender, “which is responsible for processing 3D objects for display”(Shea 2017). The function call is assumed to compose the Pokemon characters to the screen. As you can see, this function call is computationally expensive due to the high percentage of the CPU cycles used. The remaining 20% was used for a function called ContextService, which gathers data from sensors on the phone. The sensors in the phone are used to update the game world continually.

Pokemon Go uses Google Cloud to handle traffic and scalability. Although modern phones are made to handle video processing, to make video streams more efficient, developers host the platform in the cloud rather than locally, reducing hardware requirements. It also reduced the need for the developers to test the game on different devices and removed the fact that they must update the game every time the hosted application is updated. The figure below, to the left, depicts the architecture of the Cloud-based streaming platform. On the server side, there are two modules: MetaData processor and ClientInteraction. The application logic takes the data from the client and the data from the phone sensors and updates the game world based on those inputs. ScreenRendering performs the rendering. “The rendered scenes are passed to the Video Encoder module that contains a video encoder and a discreet framer”(Shea 2017). The encoder consists of an H.264 encoder, which uses a discreet framer for real-time streaming support. You can request live frames from the encoder at a desired stream frame rate. The video stream is sent to UDP, TCP, or HTTP.

The figure below, on the right side, depicts how Rhizome, the cloud gaming platform, works. Rhizome is a cloud gaming framework. The streaming engine is built on NVIDIA GRID GPU(Amazon EC2 G2 instance). “To accomplish ultra-low latency and battery-friendly gaming experience on any client device, Rhizome optimizes the device-level decoding, configuration, and interaction modules”(Fu).

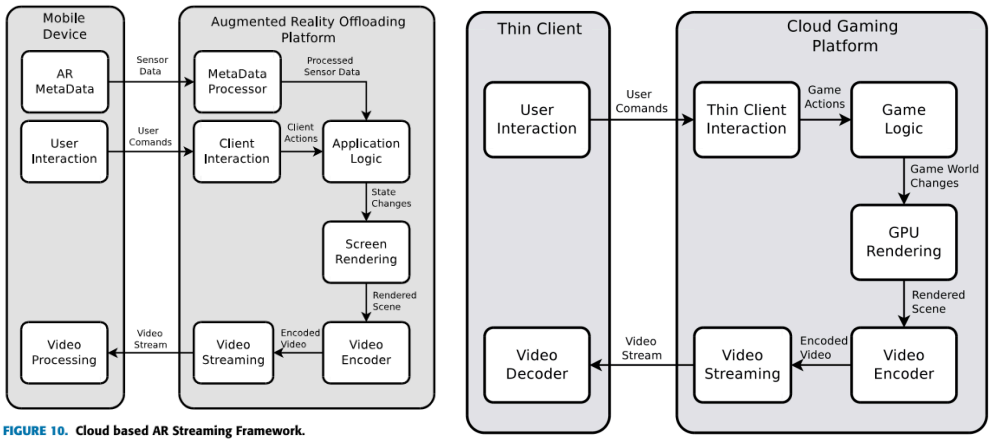



**FIGURE 10.** **Cloud based AR Streaming Framework.**

NVIDIA GRID GPU(let's make up an acronym for this, NGG) is a virtual GPU made for cloud gaming. It “allows multiple virtualized systems to utilize a dedicated GPU by placing several logical GPUs on the same physical GPU board”(Shea 2015). It contains a complete developer’s kit, allowing you to capture and process images. To discuss the details of the Rhizome architecture, we will reference the research provided by Shea's 2015 article. The user interaction modules take the user's input from their device and send the inputs to the server over TCP and to the gaming window. The game logic computes the game changes based on the user’s constant inputs while using the application. The game is rendered by the GPU NVIDIA GRID. As mentioned before, the encoder uses the discreet framer module as support. The encoded video is transported by TCP or HTTP and sent to the gamer's device. “NVIDIA GRID-K520 board contains a GK104 GPU with 1536 CUDA cores and 4 GB of video memory. The GRID’s

on-board hardware video encoder supports up to eight live HD video streams (720p at 30 frames/s) or up to four live Full HD (FHD) video streams (1080p at 30 frames/s), as well as low-latency frame capture for either the entire screen or selected rendering objects, enabling a G2 instance to offer such high-quality interactive streaming as game streaming, 3-D application streaming, or other server-side graphics tasks"(Shea 2015). When the AR game is locally rendered, power consumption is increased due to consistent calls to the CPU and GPU. Cloud games such as Pokemon Go reduce power consumption. Cloud gaming does falter due to queuing and response delays. Based on the figure to the right, one can observe that the GPU is mostly more efficient than the CPU. It is assumed that using both the CPU and GPU to process images can provide better performance than using only the GPU. (Baek 2013)

TABLE I. PROCESSING SPEED OF IMAGE PROCESSING ALGORITHMS.

| *Algorithms* | *CPU* | *GPU* |
|---|---|---|
| 3x3 erosion | 24 ms | 3.0 ms |
| 5x5 erosion | 31 ms | 8.5 ms |
| RGB to XYZ | 11 ms | 0.6 ms |
| RGB to HSV | 18 ms | 1.7 ms |
| Binary threshold | 1.2 ms | 0.7 ms |
| Canny Edge Detection | 42.7 ms | 12.9 ms |
| SURF(determinant) | 44.84 ms | 714.28 ms |

## References

A. -R. Baek, K. Lee, and H. Choi, "CPU and GPU parallel processing for mobile Augmented Reality," 2013 6th International Congress on Image and Signal Processing (CISP), Hangzhou, China, 2013, pp. 133-137, doi: 10.1109/CISP.2013.6743972.

"CUDA." *Wikipedia*, Wikimedia Foundation, 27 Apr. 2024, en.wikipedia.org/wiki/CUDA. Accessed 27 April 2024.

Exterman, Dori. "CUDA Vs. OpenCL: Which to Use for GPU Programming." *Incredibuild*, 7 Jun. 2021 www.incredibuild.com/blog/cuda-vs-opencl-which-to-use-for-gpu-programming. Accessed 26 April 2024.

Fu, Silvery. "Gaming on Demand in the Public Cloud." *GitHub*, 18 Oct. 2021, github.com/zenodflow/rhizome. Accessed 1 May 2024.

"Graphics Processing Unit (GPU)." *YouTube*, uploaded by Computer Science, 22 Oct. 2020, www.youtube.com/watch?v=bZdxcHEM-uc.

H. Chen, Y. Dai, H. Meng, Y. Chen and T. Li, "Understanding the Characteristics of Mobile Augmented Reality Applications," 2018 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Belfast, UK, 2018, pp. 128–138, doi: 10.1109/ISPASS.2018.00026.

"Intel Arc." *Wikipedia*, Wikimedia Foundation, 25 Apr. 2024, en.wikipedia.org/wiki/Intel_Arc#Alchemist. Accessed 30 Apr. 2024.

"Intel Xe." *Wikipedia*, Wikimedia Foundation, 15 Apr. 2024, en.wikipedia.org/wiki/Intel_Xe#Xe-HPG_(High_Performance_Graphics). Accessed 30 Apr. 2024.

Kraus, Jiri. "An Introduction to CUDA-Aware MPI." *NVIDIA Developer*, 13 Mar. 2013, developer.nvidia.com/blog/introduction-cuda-aware-mpi/. Accessed 1 May 2024.

K. V. Manian, A. A. Ammar, A. Ruhela, C.-H. Chu, H. Subramoni, and D. K. Panda. 2019. Characterizing CUDA Unified Memory (UM)-Aware MPI Designs on Modern GPU Architectures. In Proceedings of the 12th Workshop on General Purpose Processing Using GPUs (GPGPU '19). Association for Computing Machinery, New York, NY, USA, 43–52. https://doi-org.rlib.pace.edu/10.1145/3300053.3319419

Larabel, Michael. "Intel Arc Graphics Vs. AMD Radeon Vs. NVIDIA GeForce For 1080p Linux Graphics In Late 2023." *Phoronix*, 7 Nov. 2023, www.phoronix.com/review/1080p-linux-gaming-late-2023/4. Accessed 29 Apr. 2024.

McClanahan, Chris. "History and evolution of GPU architecture." *A Survey Paper* 9 2010: 1–7.

Munshi, Aaftab. "The opencl specification." *2009 IEEE Hot Chips 21 Symposium (HCS)*. IEEE, 2009.

"NVIDIA RTX Graphics Card Comparison (40 Series) Explained." *YouTube*, uploaded by SoulOfTech, 28 Jan. 2024, www.youtube.com/watch?v=leicT1SZez0.

"NVIDIA RTX." *Wikipedia*, Wikimedia Foundation, 17 Jan. 2024, en.wikipedia.org/wiki/Nvidia_RTX. Accessed 29 Apr. 2024.

"Please Buy Intel GPUs. - Arc A750 & A770 Review." *YouTube*, uploaded by Linus Tech Tips, 5 Oct. 2022, www.youtube.com/watch?v=6T9d9LM1TwY.

R. Shea, D. Fu and J. Liu, "Cloud Gaming: Understanding the Support From Advanced Virtualization and Hardware," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 12, pp. 2026-2037, Dec. 2015, doi: 10.1109/TCSVT.2015.2450172.

R. Shea et al., "Location-Based Augmented Reality With Pervasive Smartphone Sensors: Inside and Beyond Pokemon Go!," in IEEE Access, vol. 5, pp. 9619–9631, 2017, doi: 10.1109/ACCESS.2017.2696953.

"Silicon Graphics." *Wikipedia*, Wikimedia Foundation, 28 Apr. 2024, en.wikipedia.org/wiki/Silicon_Graphics#:~:text=SGI's%20premature%20announcement%20of%20its,SGI's%20credibility%20in%20the%20market. Accessed 1 May 2023.

"Speed Test Your GPU in Less than a Minute." *UserBenchmark*, gpu.userbenchmark.com/. Accessed 28 Apr. 2024.

T. Garrett, R. Radkowski and J. Sheaffer, "GPU-accelerated descriptor extraction process for 3D registration in Augmented Reality," 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 2016, pp. 3085–3090, doi: 10.1109/ICPR.2016.7900108.

"What Is Virtual Reality." *Virtual Reality Society*, www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html. Accessed 1 May 2024.

W. J. Dally, S. W. Keckler, and D. B. Kirk, "Evolution of the Graphics Processing Unit (GPU)," in IEEE Micro, vol. 41, no. 6, pp. 42–51, 1 Nov.-Dec. 2021, doi: 10.1109/MM.2021.3113475.