

Improving Game Performance Without the Need of Buying Better Hardware Equipment

Kevin Le

December 12 2020

Abstract

Every year or so, large tech companies such as NVIDIA and Intel release new and better versions of their computer hardware's, mainly the CPUs and GPUs. Many consumers, especially gamers, buy these hardware's to improve game performances on their PC. Newer CPUs and GPUs nowadays however cost a great deal and it is believed that buying newer computer hardware is the only solution for better game performances. Although it is true that superior hardware's improve game performances, there are other ways to enhance game performances aside from buying better equipment. In this paper, I found that users can increase the overall frame rate and game performances of the game they want play by using a game cloud service that uses multiple GPU and CPU servers. Game developers can also help improve game performances by implementing unique algorithms and techniques into their game's. This paper will go over multiple methods that can overall boost game performances without the need of buying better hardware.

1 Introduction

Technology is ever growing and will only continue to improve and grow as time goes on. We now live in a society where hardwares gets constant enhancements and improvements every year. Better hardware means better performance and as consumers we know that means more money. This very notion can be applied to central processing units (CPU) and graphics processing units (GPU) when it comes to game performance. Better CPU and GPU means better game performances, but for gamers that also means spending a good deal of money. This is a potential problem because not everyone has the money to buy the latest GeForce RTX 3080 graphics card or the latest AMD Ryzen 9 5900X CPU processor. Many game consumers believe that buying superior computer hardware is the only solution to getting better gaming performances. The current trend in computer gaming is higher performance and realism and to achieve these, game consumers are required to buy or upgrade to the latest GPU or PPU (physics processing unit) [5]. There are even mentions of users switching

over from hard disk drives (HHDs) to solid state drives (SSDs) because of its overall improvement in game reading, writing and boot time [3]. Buying better computer hardware is a definite solution to getting better game performances, however there is no need to do to that. There are other ways of achieving greater game performances without the need to spend large amounts of money. Users who switch to cloud gaming services that handle multiple GPU and CPU servers can see an overall improvement in response time and frame rates in video games. Game creators can also help improve game performances by implementing efficient techniques and algorithms into their games. By carrying out better techniques in certain aspects of their game, game developers will be able to see overall better game performances such as faster load time and frame rates. These outcomes showed that there are multiple ways of improving game performances without the need of buying newer expensive computer hardware.

2 Cloud Gaming with GPU and CPU Resources

Cloud gaming services is a way for gamers to play actual games without the need for an actual computer hardware. Games are run through a remote server instead then gets streamed onto a device screen. In other words, games can be played remotely through a cloud without any sort of hardware needed. However the real question comes into play when playing large complex video games. If someone has a PC that could not handle large 3D games and switched over to cloud gaming, will that person be able to play these complex games let alone get better game performances? The answer to this question depends on the amount of computer hardwares being used in a cloud gaming server. If a cloud server only contained one piece of CPU and GPU and multiple users are using it, the chances of being able to stream a large complex game let alone get better performances out of it is very slim. One team decided to investigate a method called GPU pass-through to see if it was capable for cloud gaming. GPU pass-through is the idea of running games through a virtual machine (VM) by directly accessing a machine's physical GPU. Since virtual machines are accelerated by the GPU, it was assumed that games running through VMs will perform on par with its non virtualized counterpart. However this theory was proven false and the team discovered that games performed more poorly virtualized compared to its baseline [10]. One game that was tested on both the baseline and virtual machine was Doom 3. The experiment showed that Doom 3 had a frame rate of 120 when it ran on the physical machine. But once the game switched to playing on the virtual machine, the frame rates dropped down to 40. What we learn from this is that games that are virtualized don't perform well compared to its physical counterpart. In a way this makes sense because when a game is being streamed or virtualized, data is always constantly being moved and processed between the virtual machine and the actual device itself. Hence in order for a game to run and perform well virtually, it would need multiple GPUs and CPUs to efficiently handle all the tasks accordingly. This is where cloud servers with multiple GPUs and CPUs come into play.

The key to getting better game performances through cloud gaming services is to have servers full of CPUs and GPUs. By having multiple GPUs run simultaneously, it can create high parallelism which in turn increases game performance and can make a game run better. One group created their own cloud architecture called “GamePipe” which had a cluster of CPU and GPU servers [13]. By having a bunch of servers filled with CPUs and GPUs, the team was able to assign different roles to each server which made handling multiple tasks easier. For example, the team was able to develop backend virtual machines that handled different types of functions. GPU servers dealt with game graphics and physics based stuff while CPU servers handled frames and audio. By taking advantage of high parallelism and distributed VMs, the team was able to demonstrate that strength of cloud gaming. When the game “Destroy the Castle” was executed on the GamePipe cloud, the team found the following results. The overall execution time was reduced, the frame rates ended up being 35 percent higher, the processing time was 8 times faster. These game improvements happened because of parallelism which was done through the teams “GamePipe” cloud architecture. Another group was able to demonstrate better game performance through cloud gaming by making their own resource management framework. Their framework was called VGRIS (Virtualized GPU Resource Isolation and Scheduling) and its main goal was to employ multi task scheduling to improve utilization of GPUs [12]. Three scheduling algorithms were created within this framework to help achieve this goal. Service Level Agreement (SLA) aware scheduling, proportional share scheduling, and hybrid scheduling. Each of these scheduling algorithms has a unique task of allocating GPU resources accordingly. Once executed, the team was able to regulate GPU resource usage and achieve GPU resource management in virtual machines. This in turn results in better game performances for games in cloud gaming. When the group tested out Portal 2 using their framework for example, they were able to find an average FPS increase of 65 percent and an overall latency drop of 3 percent. These papers showed that it is possible for games to perform better through cloud gaming if done effectively.

3 Game Creators: Improve Game Performance through Algorithms and Techniques

Computer hardware is not the only factor that affects game performance, game implementation also plays an important factor as well. The way a game is designed and implemented plays an essential factor in how it ends up performing once it is finished. If a finished game is not optimized or programmed well by the game creators, then the game performance issues lie within the program or game itself. This is why game designers also play an essential role in game performances as well. In order for game creators to improve their game’s performances, they have to implement efficient algorithms and techniques to certain aspects of their game. One group for instance implemented a simple precal-

calculation technique which helped improve their game's performance in terms of frame rates and computation time. The team knew that 3D games do massive amounts of calculations on things like lightings, shadows, and dynamic animations before each rendering. So in order to minimize the number of calculations done, the team implemented a precalculation data structure [1]. The precalculation technique found that static objects and polygons of scenes both play an important role in rendering frames. This led the team to implement a Binary Space Partition tree (BSP) in their game code to help deal with static polygons. The team only wanted specific static polygons to be rendered within a current frame to help reduce calculations, so in order to do that the BSP tree was implemented. The team also added light maps in their game to make lighting effects more efficient. By the end of the study, the team discovered that their game overall performed better when BSP tree and lightmaps were implemented. One result showed that their game was able to load 13 seconds faster with BSP. By adding in techniques such as pre calculations into the games program, the game designers were able to make their game perform better.

Realistic effects can heavily affect a game's performance. Effects such as fire spread and smoke can take up lots of data and slow down frame rates if not implemented correctly. Salwala and her team wrote a paper that focused on analysing games that use realistic environments and make use of ornaments like flow of water [9]. To produce such a desired effect in games without causing problems to the performance, two techniques were implemented. The first technique called cellular automata (CA) simulation is used to simulate events in the game, such as fire spread, and assigns them with cell properties. These cell properties are usually grouped up to represent the environmental event, but the group made changes to that concept. Instead of grouping the properties together, each property will have its own associated variable and will be allocated separately instead. This was primarily done to help improve mutation and crossover operations in the team's second key technique which is the generic algorithm. The core idea was to implement and improve the genetic algorithm so that parallel computing can be applied to the GPU. By achieving parallelism, the team was able to improve performances of emergent environments in video games. Now there is a potential way for game developers to be able to implement emergent environments into their game without the fear of poor performance. Another key component in large 3D games that can affect game performance is the map. Processing a map, especially a large one, can be a problem since level of details (LOD) can affect the workload of the processor. One group saw this potential problem and created an algorithm called Continuous Level of Detail (CLOD) which is essentially a dynamic LOD. CLOD basically changes the level of detail in real time and uses the CPU to optimize the map rendering process [7]. There are many different techniques game designers can implement in their games to improve game performance.

4 Alternatives for Better Game Performance

Besides cloud gaming and better game implementation from game developers, there are other various ways to improve game performances. Prakash and his team managed to improve mobile game performances for instance by implementing a cooperative CPU - GPU thermal management technique [8]. During their research, the group discovered that the CPU and GPU lacked coordination which can degrade app performance. To solve this problem, the team proposed a dynamic thermal management algorithm which controlled the CPU and GPU settings to help maintain their temperatures. By maintaining the CPUs and GPUs internal temperature, the group was able to achieve an average 20 percent improvement to frame rates in various android mobile games. Although the thermal management technique was done on mobile devices only, further work and add ons to it can make it potentially work on desktop computers in the future. The thermal algorithm handles tasks within the CPU and GPU, so implementing it on PC is very possible. If this kind of technique was applied to all desktop computers in the very future, there might not be a need for cooling fans anymore.

Another simple way to improve game performance is to avoid using ray tracing rendering in 3D graphics. One group wrote a paper that compared two rendering techniques and their effects on game engine performances. The two rendering methods were rasterization and ray tracing. Rasterization is the default rendering approach used in game engines while ray tracing is another rendering approach used to make games look realistic. Because ray tracing is used to make games look more realistic, we know it will take a toll on game performances. In fact throughout the paper, Chang and their team stated multiple times that the rasterization rendering method is better than ray tracing because it runs much faster and is supported by the GPU. This information most likely encouraged the group to write a paper that focused on improving the ray tracing technique. In order to improve ray tracing and close its gap performance against rasterization, the team used a GPGPU approach and applied GLSL shaders in the game engine [2]. This slightly increased game performance and made the run time for ray tracing more reasonable. Despite the decent improvements made to ray tracing, Chang and his team still concluded that rasterization was the best rendering technique overall. So instead of switching over to ray tracing and improving it in game engines, game designers should stick to using the default fast rasterization rendering technique. These are just a few more methods users and game designers can do to improve their game performances.

5 Discussion

The main purpose of this paper was to show that there are other ways to improve game performances without the need of buying new and expensive computer hardware every year. Gamers are always on the lookout to make their computer

better so that they can get the ultimate gaming experience. To achieve the best gaming experience, games need to be able to run fast and well and the key to doing that is to buy the latest CPU, GPU, cooling fan, etc. It is good to upgrade and replace old computer parts once in a while to improve game performance, but most game consumers don't follow this concept. Many just go straight to buying the new and latest hardware every year like most consumers with new releases of mobile phones. The problem is not everyone has the ability to do this due to multiple reasons, primarily financial reasons. It was this very reason that pushed this paper to happen. It is important for game consumers to know that there are other ways to achieve better game performances on games they want to play like using powerful cloud gaming services. There are some users out there however who might not want to do that and just want to stick to using their desktop computer. If that's the case, then it's strongly encouraged that gamers test and fully analyze their CPU and GPU performance before inevitably succumbing to buying new computer hardwares. By having a better understanding of their current computer hardware, users can find and research their own methods and ways of improving their game performances. J. Issa [6] and O. Dedehayir [4] wrote papers about the importance of understanding a GPUs performance. Computer hardware plays an important part in gaming performance, but so do game developers.

When game developers put in dedicated time and work into their games, it will play and perform well no matter what hardware it is on. However if a game developer relies on the latest computer hardware to make their game play and perform well, it can have potential problems. Instead, game creators should script their game carefully and implement unique algorithms and techniques to optimize it. One article highlighted the importance of writing better scripts when it comes to creating a game [11]. If a game programmer for instance coded in lots of iterations in their program, it could significantly degrade the game's performances. An advantage for making a game that performs and plays well is that most people will be able to play it without a problem. Not only that, but the game would also be playable under most computer hardware build as well. This means consumers with less powerful computer hardware would be able to play and experience the game too. It is a win situation for both sides if game developers program their game well and put hard work into it.

6 Conclusion

There are other ways of achieving better game performances without the need to spend large amounts of money on new and superior hardware. For game users, one possible solution would be to switch over to a powerful cloud gaming service. Games that run on cloud servers full of CPUs and GPUs will perform well if not better than its hardware counterpart due to high parallelism. Computer hardware is not the only main factor in achieving better game performance, game developers play an essential factor too. If game designers implement unique algorithms and techniques into their game and optimize it, game performance

will overall increase as well.

References

- [1] A. Al Mazed, M. H. Asif, and E. Ahmed. Increasing performance of 3d games using precalculation. In *2005 Pakistan Section Multitopic Conference*, pages 1–6, 2005.
- [2] C. Chang, K. Chen, and C. Chuang. Performance comparison of rasterization-based graphics pipeline and ray tracing on gpu shaders. In *2015 IEEE International Conference on Digital Signal Processing (DSP)*, pages 120–123, 2015.
- [3] Mark Claypool, Jared Hays, Alex Kuang, and Thomas Lextrait. On the performance of games using solid state drives. In *Proceedings of the 10th Annual Workshop on Network and Systems Support for Games*, NetGames '11. IEEE Press, 2011.
- [4] O. Dedehayir and S. J. Makinen. Does technological performance gap drive product performance?: An explorative study of personal computer gaming. In *2008 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 895–899, 2008.
- [5] D. Geer. Vendors upgrade their physics processing to improve gaming. *Computer*, 39(8):22–24, 2006.
- [6] J. Issa and S. Figueira. A performance estimation model for gpu-based systems. In *2012 2nd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA)*, pages 279–283, 2012.
- [7] A. E. Minarno, A. R. Agisna, W. A. Kusuma, W. Suharso, and H. Wibowo. Optimizing game performance with dynamic level of detail mesh terrain based on cpu usage. In *2020 3rd International Conference on Intelligent Autonomous Systems (ICoIAS)*, pages 93–98, 2020.
- [8] Alok Prakash, Hussam Amrouch, Muhammad Shafique, Tulika Mitra, and Jörg Henkel. Improving mobile gaming performance through cooperative cpu-gpu thermal management. In *Proceedings of the 53rd Annual Design Automation Conference, DAC '16*, New York, NY, USA, 2016. Association for Computing Machinery.
- [9] C. Salwala, V. Kotrajaras, and P. Horkaew. Improving performance for emergent environments parameter tuning and simulation in games using gpu. In *2010 3rd International Conference on Computer Science and Information Technology*, volume 2, pages 37–41, 2010.
- [10] R. Shea and J. Liu. On gpu pass-through performance for cloud gaming: Experiments and analysis. In *2013 12th Annual Workshop on Network and Systems Support for Games (NetGames)*, pages 1–6, 2013.

- [11] Walker White, Christoph Koch, Johannes Gehrke, and Alan Demers. Better scripts, better games: Smarter, more powerful scripting languages will improve game performance while making gameplay development more efficient. *Queue*, 6(7):18–25, November 2008.
- [12] Miao Yu, Chao Zhang, Zhengwei Qi, Jianguo Yao, Yin Wang, and Haibing Guan. Vgris: Virtualized gpu resource isolation and scheduling in cloud gaming. In *Proceedings of the 22nd International Symposium on High-Performance Parallel and Distributed Computing*, HPDC '13, page 203–214, New York, NY, USA, 2013. Association for Computing Machinery.
- [13] Zhou Zhao, Kai Hwang, and Jose Villeta. Game cloud design with virtualized cpu/gpu servers and initial performance results. In *Proceedings of the 3rd Workshop on Scientific Cloud Computing*, ScienceCloud '12, page 23–30, New York, NY, USA, 2012. Association for Computing Machinery.