Parallel Technology and Development in the Gaming Industry

Zixuan Xu[a]
School of Advanced Technology, Xi'an Jiaotong-Liverpool University,
111 Ren 'ai Road, Suzhou Industrial Park, Huqiu District, Suzhou, China

Keywords: Parallel Technology, Multi-Core, Multi-Thread, Game Industry.

Abstract: With the rapid development of the game industry, the traditional single-core simple logic games have been unable to meet the needs of teenagers, so the addition of parallel technology has changed the status quo. The use of multi-threading, multi-core, and the use of gpu and cpu improves graphics rendering, computing power, and physics simulation in the game industry. This article provides an overview of the application and development of parallel technical in the gaming industry. It gives readers a deeper understanding of parallelism and its impact on the game industry. The focus is on the underlying concept of parallel technique, for instance: multi-threading, multi-core, single instruction multiple data (SIMD), and the use of cpu and gpu. Furthermore, the article will also discuss about several applications of parallel techniques in gaming industry, for example: image rendering, route planning of Non-player characters (NPCS), and parallel processing of some large online games and so on. Finally, the future development of parallel technology will also be mentioned, such as how to combine with Artificial Intelligence (AI) to make AI more intelligent, and how to combine with AR technology to enhance the authenticity of the virtual world and reduce the delay of Augmented Reality (AR) devices.

1 INTRODUCTION

As the game industry rapidly evolves, traditional single-core simple logic games are increasingly inadequate in meeting the demands of teenagers; thus, the integration of parallel technology has transformed the existing landscape. Parallel computing represents a computational paradigm that harnesses multiple processing elements concurrently to address complex problems with greater efficiency than traditional serial computing methods. This approach is fundamentally driven by the necessity to manage large-scale computations and data processing tasks that surpass the capabilities of single-threaded processors. The origins of parallel computing can be traced back to the formative years of computer science; however, it gained substantial traction with the emergence of multi-core processors and distributed computing systems. In this framework, tasks are partitioned into smaller subtasks executed simultaneously across various processors or cores. This division of labor not only accelerates computation but also enhances application scalability. Parallel computing encompasses diverse architectures and models, including multi-core executing, muti-thread model, the together usage of CPU and GPU. With the continuous improvement of the demand for game processing performance, the demand for game picture quality improvement, parallel computing has become essential across game domains. As technology continues to advance, innovations in parallel computing promise to further augment performance while unlocking new potential within computational capabilities. So, this article will about the underlying definition of several parallel techniques as well as their applications in the game industry which will give readers a deeper understanding of parallelism and its impact on the game industry. Decades years ago, AlBahnassi, Mudur and Goswami have shown that in recent years, several existing games have begun to transition towards supporting multiple processors, so it seems that the game industry is moving toward parallelization (Albahnassi, Mudur, Goswami, 2012). The main reason why game manufacturers prefer to develop parallel games is not only because parallel

[a] https://orcid.org/0000-0001-5437-6650

Xu and Z.
Parallel Technology and Development in the Gaming Industry.
DOI: 10.5220/0013514900004619
In *Proceedings of the 2nd International Conference on Data Analysis and Machine Learning (DAML 2024)*, pages 232-238
ISBN: 978-989-758-754-2

technology can save a certain investment cost, but also because it can greatly improve the performance of the game and improve the processing efficiency of the computer. And Venu has demonstrated that various advanced techniques have been developed to enhance performance, including parallel processing, data-level parallelism, and instruction-level parallelism, all of which have demonstrated significant effectiveness (Venu, 2011). For example, for multi-core processing venu has also shown that multi-core processors were available for over a decade; however, their significance has increased recently due to the technological limitations faced by single-core processors, such as challenges in achieving high throughput and prolonged battery life with enhanced energy efficiency (Venu, 2011). Additionally, as for multi-thread Tulip, Bekkema, Nesbitte also found out that although current game engines are extensively optimized for efficient operation on single-processor architectures and have largely avoided multi-threaded design approaches. Nevertheless, the forthcoming generation of game engines must tackle the intricacies of multi-threaded programming to fully leverage the performance capabilities of emerging PC and gaming console platforms (Tulip, Bekkema, Nesbitte, 2006). Furthermore, the game industry also applied the parallel processing capabilities of GPUs, thereby markedly enhancing graphical performance and visual fidelity in gaming applications. In the remaining parts of the article, it will be divided into five small parts, firstly some Overview of parallel techniques, then the detailed development of parallel technology, after that some applications of parallel technology in game development, fourthly the prospect of future development and lastly the conclusion.

2 OVERVIEW OF PARALLEL TECHNIQUES

Nowadays, with the development of parallel technology, many different parallel technologies have been derived. Examples include the use of multiple cores, the use of multiple threads, or the combination of CPUs and GPUs. Most parallel techniques can greatly improve the speed, speed, and quality of a computer if they are used in the right amount which can definitely improve the quality of the game industry.

2.1 Multi-core

Multi-core technology refers to the integration of multiple processing cores on a processor chip. Each core is able to perform tasks independently, which can greatly improve computational performance. And nowadays, the processor may have two, four, six, eight or even more core to accelerate the processing quality. Each core can deal with the threads and tasks independently. Figure 1 below shows the simple architecture of a computer with 4 core processor.

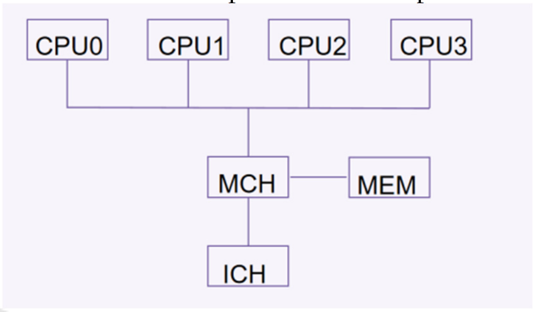

Figure1: a computer with 4 core processor.（Photo/Picture credit : Original ）

Memory control center (MCH), also known as the North bridge chip. Input/output control hub (ICH), also known as the South bridge chip. MCH mainly provides support for CPU, memory and other devices. ICH mainly supports peripheral devices such as keyboards and interfaces. And MEM is the memory which saves the data. The use of multi-core processors can significantly improve the multitasking ability and parallel computing performance. By sharing the workload, the individual cores can operate at a lower frequency, resulting in lower power consumption. The underlying working principle is that the single core in multi-core processor may not be powerful than the single-core processor. But it can enhance the overall performance by handling multiple tasks at the same time. A single-core processor will distribute different time slices to different program when dealing with multi-program. But once if one program takes longer time to complete, other processed start to lag behind. However, for multi-core, each task can be executed by separate core at the same time which can largely improve the performance as shown in Figure 2 (Geer, 2005).

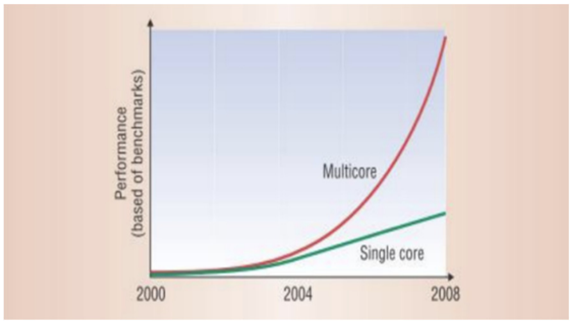

Figure 2: Multicore chips perform better – based on Intel tests using the SPECint2000 and SPECfp2000 benchmarks – than single-core processors (Geer, 2005).

Furthermore, the frequency is not high for multi-core in the chip, but the paralleling processing model can make greater performance and reduce the power consumption as shown in figure 3.

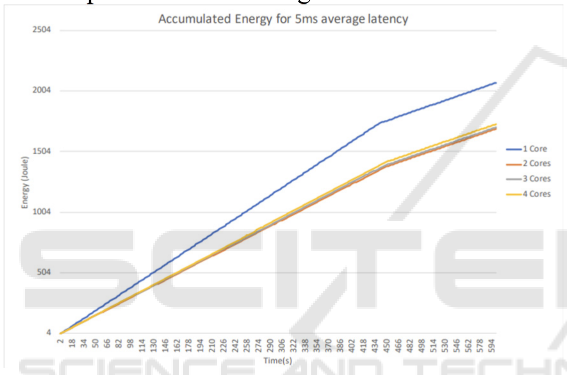

Figure 3: Accumulated energy consumption for different numbers of cores during network average latency test with a 5 ms limit (Oliveira, Xavier-De-souza, Silveira, 2021).

So, this explains why numerous game developers favor multi-core processing.

2.2 Multi-threading

Multi-threading technology encompasses the concurrent execution of multiple threads within a single processor core. A thread represents an autonomous path of execution for a program, while multi-threading facilitates the sharing of resources—such as memory and file descriptors—among multiple threads operating under the same process. There are two main models for multi-threading. One is multi-threading working on a single processor. In a single-core processor, the thread scheduler plays a crucial role in distributing CPU time among multiple threads. Given that there is only one core available, true simultaneous execution of multiple threads is not feasible; therefore, the scheduler rapidly alternates between thread executions, ensuring that each thread has an opportunity to utilize the CPU. Although this model is not really working in parallel, it still improve the usage of the cpu that is why a single processor model can still accelerate the processing efficiency. For the second model, multi-threading working on multiple processor, it can realize parallel operation in the real sense. Tulip, Bekkema and Nesbitt also found that for a fixed number of cpu processor, the user can increase the number of threads to further accelerate the processing speed. However since each of the cpu processor has maximum processing bottlenecks, the user needs to increase the number of cpu to enhance the processing speed and energy efficiency which shown in Figure 4 and Figure 5 below (Zecena, Burtscher, Jin et al, 2013).

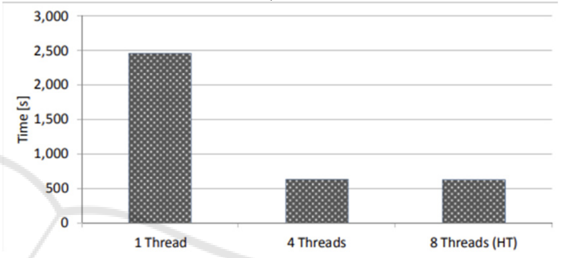

Figure 4: Runtime of NBOMP with 100,000 bodies and 10 timesteps on System 1 (Zecena, Burtscher, Jin et al, 2013).

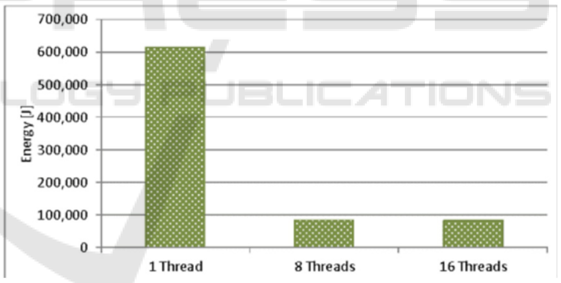

Figure 5: Energy consumption of NBOMP with 100,000 bodies and 10 time steps on System 1 (Zecena, Burtscher, Jin et al, 2013).

2.3 CPU-GPU hybrid technology

CPU-GPU hybrid technology leverages the synergistic collaboration between central processing units (CPUs) and graphics processing units (GPUs) to significantly enhance computational performance. Due to their extensive parallel processing capabilities, GPUs are particularly adept at handling tasks that demand substantial parallel computation. In real life, computers usually improve the processing speed of computers by reasonably allocating computer tasks to CPU and GPU, such as uploading some content with large memory or for deep learning and scientific

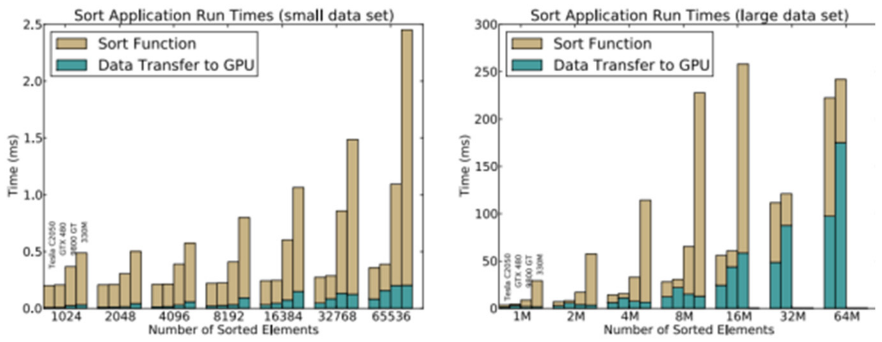

Figure 6: Sort benchmark. Faster GPUs are affected more by the memory transfer overhead. For instance, when sorting 64M values, the application time on the GTX 480 is 3.6x slower than the kernel itself (Gregg, Hazelwood, .2011).

Table 1: GPUs tested.

GPU Type	Compute Capability	Cores	Memory(MB)	Clock(MHz)	Host-Dev BW(Mb/s)	DDev-Host BW(Mb/s)
Tesla C2025	2.0	480	3072	1150	2413.9	2359.2
GTX 480	2.0	480	1024	1401	1428.0	1354.2
9800 GT	1.1	112	1024	1500	2148.8	1502.5
330M	1.2	48	256	1265	2396.2	2064.7

All GPUs are from Nvidia and run the Cuda programming language. the 330m GPU is a laptop GPU and the others are desktop GPUs. "host-dev" shows transfer times from main memory to GPU memory, and "dev-host" shows transfer times from GPU memory to main memory (Gregg, Hazelwood, 2011).

computing. As shown in figure 6 and table 1 below, they demonstrate that cpu and gpu work together and show that gpu with more cores has higher processing efficiency to further improve data processing speed.

3 APPLICATION OF PARALLEL TECHNOLOGY IN GAME DEVELOPMENT

There are numerous applications of parallel technology in the game industry. This article will only discuss three main states: graphic rendering, AI route or instruction planning and massive online gaming.

3.1 Graphic Rendering

Various forms of parallelism can be employed in the rendering process, including data decomposition, task assignment and the acceleration of GPU. For data decomposition, a prevalent strategy involves partitioning the image into smaller segments. Each segment is processed independently by distinct processing units, such as CPU cores or GPU threads. This approach effectively leverages computational resources and mitigates the burden on individual processing units. While for complex 3d scene, the scene will be divided into several parts, each part will be rendered in separated processing unit to achieve the parallel processing. Then for the task assignment, during the rendering process, geometric processing tasks—such as transformations, clipping, and lighting calculations — can be decomposed into multiple parallelizable components. For instance, each task associated with geometric processing for a given model may be allocated to distinct threads or processing units. The rasterization process transforms geometric shapes into pixels, enabling parallel execution across multiple threads. As for the GPU acceleration. The developer usually applies GPU parallel architectural CUDA or OpenCL to deal with numerous threads at the same time which will largely increase the speed of rendering. Figure 7 below shows two different types of parallel rendering model and Figure 8 demonstrates the frame improvement of different number of GPUS.

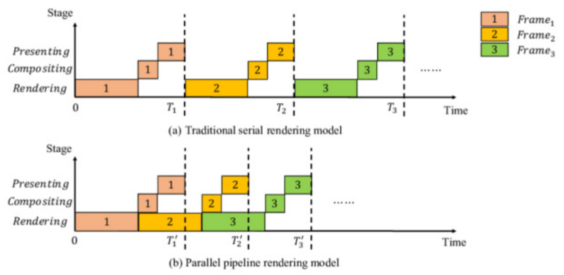
Stage
Presenting
Compositing
Rendering
Frame$_1$
Frame$_2$
Frame$_3$
Time
(a) Traditional serial rendering model
(b) Parallel pipeline rendering model

Figure 7: The rendering time variation under different rendering models (Zhang, Ma, Qiu, et al, 2023).

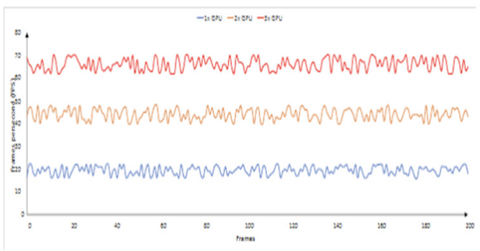

Figure 8: performance of the multi-GPU system under different number of GPUS (Zhang, Ma, Qiu, et al, 2023).

3.2 AI Route or Instruction Planning

There are various parallel methods that can be applied to Ai route or instruction planning. For example, multi-thread and multi-progress. In addressing complex path planning challenges, such as navigation tasks within extensive environments, multi-threading enables the simultaneous computation of multiple potential paths to identify the optimal route. Each thread can manage a specific sub-region or explore various algorithms, thereby reducing overall computational time. In decades years ago, Sanci has also shown the result of the acceleration of route algorithm by applying multi-threads in figure 9. In systems requiring the simultaneous execution of multiple instructions, such as automated production lines, multi-process architectures can manage distinct tasks or instructions in parallel, thereby enhancing the overall operational efficiency of the system. For some NPC in the game, the constructor usually applies reinforce learning to train the ai to explore and study independently. In the training of agents such as those used in gaming or robotics, parallel computing facilitates the simultaneous execution of multiple environment instances, thereby expediting the learning process. Each instance operates independently to explore and learn, with subsequent results aggregated for model updates.

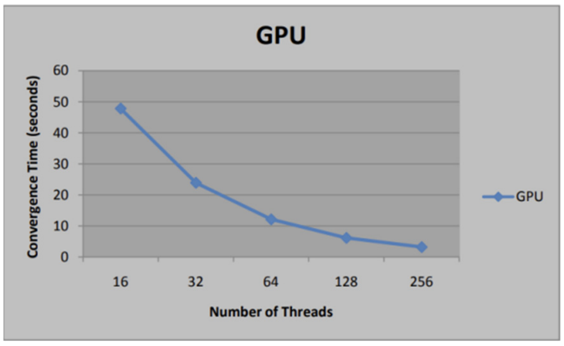

Figure 9: Effect of Using Different Number of Threads for the Genetic Algorithm in the GPU (Sanci, 2010).

3.3 Massive Online Games

For large online games, it usually requires handling complex game logic, such as the analysis of players methods, the update of players' mission, or some event trigger. This progress will be divided into several processors or various threads in order to improve the speed of game logic processing and reduce the latency due to the large number of players. Abdelkhalek and Bilas have found out that for more threads or more processor server, the average time of response is lower than those which have reach the bottleneck of the peak value of player. But when both servers not achieving the bottleneck, the respond time will be almost the same (Abdelkhalek, Bilas, 2004). And Figure 10 shows the response rate of sever due to the player. Figure 11 demonstrates the average respond time. In each figure different line means different numbers of threads.

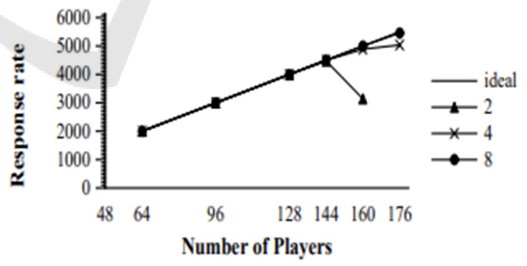

Figure 10: Total server response rate (Abdelkhalek, Bilas, 2004).

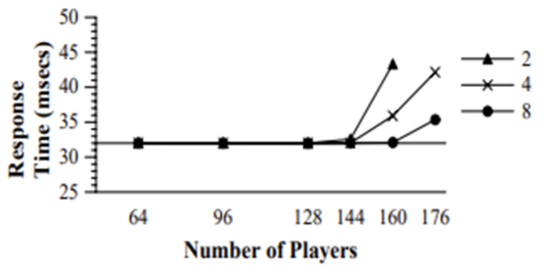

Figure 11: Average server response time (Abdelkhalek, Bilas, 2004).

Furthermore, it is also significant for massive online games to make multiple data synchronization. So when applying parallel processing technique, the server can deal with numerous data packages at the same time which can reduce the internet latency and maintain the synchronization of the data. Additionally, in a massive online game, the artificial intelligence needs to process the behavior decisions of multiple NPCS (non-player characters) at the same time. So parallel computing is of great importance to evaluate the behavior logic of multiple NPCS at the same time to improve the intelligence level and response speed of AI.

4 FUTURE DEVELOPMENT

The past decades years of this new kinds of technology (parallel technology) was mainly applied on some basic online games. And nowadays with the development of artificial intelligence, the game industry has a trend to develop smarter Ai and smart computing. it is also realizable for users to play game without the touch of their fingers with the development of virtual reality in the future. There also has been some progress in virtual reality (VR). By applying DLoVe and other parallel systems, Deligiannidis and Jacob have already found that the application of DLoVe can largely improve the overall performance of applications and increase the average framework, it can also be used to provide mechanisms for the implementation and transformation of single-user programs into multi-user applications which is useful when developing some large online game on virtual world in the future (Deligiannidis, Jacob, 2005).

Moreover, this article only discussed about several parallel technologies and some applications on game industry, a more detailed description of these contents will be improved and supplemented in the future.

5 CONCLUSION

In conclusion, this article has discussed several basic parallel technologies for instance: multi-core, multi-threads and cpu-gpu hybrid technology. It also talked about some applications in the game industry. For example, graphic rendering, ai route or instruction planning and large online games. Moreover, this article also explores the future integration of artificial intelligence, virtual reality, and parallel computing. Ultimately, several drawbacks emerged that require resolution in the future, including addressing compatibility issues between parallel techniques and hardware, as well as ensuring data consistency.

REFERENCES

Abdelkhalek, A., & Bilas, A. 2004. Parallelization and performance of interactive multiplayer game servers. In *Proceedings of the 18th International Parallel and Distributed Processing Symposium* (pp. 72-). IEEE.

AlBahnassi, W., Mudur, S. P., & Goswami, D. 2012. A design pattern for parallel programming of games. In *2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems* (pp. 1007-1014). IEEE.enu, B. (2011). *Multi-core processors: An overview*. arXiv.

Deligiannidis, L., & Jacob, R. J. K. 2005. Improving performance of virtual reality applications through parallel processing. *Journal of Supercomputing, 33*(2), 155–173.

Geer, D. 2005. Chip makers turn to multicore processors. *Computer, 38*(5), 11-13.

Gregg, C., & Hazelwood, K. 2011. Where is the data? Why you cannot debate CPU vs. GPU performance without the answer. In *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software* (pp. 134-144). IEEE.

Oliveira, T. F., Xavier-De-souza, S., & Silveira, L. F. 2021. Improving energy efficiency on SDN control-plane using multi-core controllers. Energies, 14(11). https://doi.org/10.3390/en14113161

Sanci, S. 2010. A parallel algorithm for flight route planning on gpu using cuda [M.S. - Master of Science]. Middle East Technical University.

Tulip, J., Bekkema, J., & Nesbitt, K. 2006. *Multi-threaded game engine design*. In *Proceedings of the 3rd Australasian Conference on Interactive Entertainment* (pp. 9–14). Murdoch University.

Zecena, I., Burtscher, M., Jin, T., & Zong, Z. 2013. Evaluating the performance and energy efficiency of n-body codes on multi-core CPUs and GPUs. 2013 IEEE 32nd International Performance Computing and Communications Conference (IPCCC), Performance Computing and Communications Conference (IPCCC), 2013 IEEE 32nd International, 1–8. https://doi.org/10.1109/PCCC.2013.6742789

Zhang, H., Sheng, B., Bi, L., Kim, J., Magnenat-Thalmann, N., & Thalmann, D. (Eds.). 2024. *Multi-GPU parallel pipeline rendering with splitting frame*. In *Advances in computer graphics: CGI 2023* (Vol. 14496, pp. 225-238). Springer.