University of Ibadan
Journal of Science and Logics in ICT Research

Graphical Processing Unit and Central Processing Unit Performance in Human-Computer Interaction

[1]Adigun A. A., [2]Alabi O. O., [3]Ogundoyin I. K., [4]Adigun O. I., [5]Ibrahim M. A., [6]Ozoh P. O. [7]Abanikannda M. O.

[1, 2, 3, 5, 6,7] Osun State University, Osogbo, Nigeria
[4]University of Applied Science Wurzburg - Schweinfurt, Germany
[1]adepeju.adigun@uniosun.edu.ng [2]olusegun.alabi@uniosun.edu.ng [3]ibraheem.ogundoyin@uniosun.edu.ng
[4]olufemibrahim@gmail.com [5]ibrahima@uniosun.edu.ng [6]patrick.ozoh@uniosun.edu.ng
[7]mo.abanikannda@uniosun.edu.ng

Abstract

The Graphical Processing Units (GPU) and Central Processing Units (CPU) performance in Human-Computer Interaction (HCI) has improved in recent years. Many researchers have worked in comparative analysis of GPU and CPU, but a lacuna was discovered in the area of importance of HCI-which is securing user satisfaction. This paper aims to review the performance evaluation of CPU and GPU systems. Questionnaire and experimental models were used for evaluation. Questionnaires were distributed to evaluate the level of user satisfaction, using demands and usability parameters. In experimental models, tools of evaluation were downloading PDF and text files. Parameters used to assess satisfaction were speed, storage capacity and execution time. The results of the evaluated performances on GPU and CPU systems based on the evaluation parameters were derived from the questionnaire and experimental data. Questionnaire results have shown that a lesser number of users demand GPU due to its high price, and a large number of users demand CPU due to its low cost of finance, despite its disadvantages. Based on the experimental model, resources were tested for two weeks with different test data and environments which showed that the GPU works faster than the CPU in terms of speed and storage capacity. The results of the functionality and performance level of the two processors showed how GPU obeys the laws of the user interface design principle and satisfied the importance of HCI by securing user satisfaction. Users in the area of Photography, Cinema, Film Production companies, etc. should be engaged with the use of GPU than CPU systems.

Keywords: *Human-Computer Interaction, Graphical Processing Unit, Central Processing Unit, Questionnaire model, Experimental model*

1. Introduction

Human-Computer Interaction (HCI) is how people interact with computers and to what extent computers developed or not for successful interaction with human beings. A significant number of major corporations and academic institutions now study HCI. Through our immediate and outside environments, we discovered that the level of human-computer interaction (HCI) fails to showcase the importance of HCI with the users most of the time.

Many researchers have worked in this area and reviewed but did not consider the importance of HCI. Observations show that the CPU contributed a lot to the problem facing the users, and GPU almost gives users satisfaction. Due to the disparities between the two processors and the aim to reduce the disparities, performance evaluation of GPU and CPU activities came on board. This process will have an advanced effect on the user's activities in the areas of HCI. This research aims to review the performance evaluation of CPU and GPU systems. The objectives are to compare the performance

Adigun A. A., Alabi O. O., Ogundoyin I. K., Adigun O. I., Ibrahim M. A., Ozoh P. O. and Abanikannda M. O. (2022). Graphical Processing Unit and Central Processing Unit Performance in Human-Computer Interaction. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 8 No. 1, pp. 19 - 28
©U IJSLICTR Vol. 8 No. 1, June 2022

between CPU and GPU using downloaded PDF file with images on window 7 Operating System (OS); to compare the performance between CPU and GPU using downloaded word file without the image on window 7 OS; to analyze the results generated from objectives 1 & 2 and to determine the GPU and CPU user's demand.

A central processing unit (CPU) is the electronic device within a computer that carries out the instructions of a computer program by performing the following operations specified by the instructions. The operations are Basic Arithmetic, Logic, Controlling, and Input/output (I/O). The term "CPU" refers to a processor which has been in use by the computer industry since the early 1960s.

The Central Processing Unit (CPU) has often been called the brain of the personal computer (PC). But increasingly, that brain is being enhanced by another part of the PC – the Graphical Processing Unit (GPU) its soul. All PCs have chips that render the display images to monitors, but not all these chips are created equal. Intel's integrated graphics controller provides basic graphics that can display only productivity applications like Microsoft PowerPoint, low-resolution videos and basic games. The GPU is in a class by itself. It is beyond the graphics controller functions and is a programmable and powerful computational device in its own right. The GPU's advanced capabilities are mainly used primarily for 3D game rendering. But now, those capabilities are being harnessed more broadly to accelerate computational workloads in areas such as financial modeling, cutting-edge scientific research, and oil and gas exploration. [4][8][12][16].

A graphical processing unit (GPU) is an electronic device designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. The application of GPUs is in embedded systems, mobile phones, personal computers, workstations, and game consoles. Graphics processing units (GPUs) also powered the display of images and motion on computer displays. GPUs are now powerful enough to do more than move images across the screen. They are capable of performing high-end computations that are the staple of many engineering activities. GPUs can perform engineering computations much faster than the traditional central processing units (CPUs) used in today's workstations sometimes as much as 20 times faster, depending on the operation.

The performance advantage in these benchmarks does not automatically make it a slam dunk for running engineering applications. [5]. The performance of Graphical Processing Units (GPU) has improved in recent years. Compared with the CPU, the GPU is better suited for parallel processing and vector processing. A high-performance computing environment is necessary for numerical computations like physics and earth environment simulations which require enormous computational power. [4][8][9][13].

2. Related Works

Many researchers have worked in this area and reviewed as follows:

i. Central Processing Unit-Graphic Processing Unit Computing Scheme for Multi-Object Tracking in Surveillance to present a novel central processing unit graphics processing unit (CPU-GPU) computing scheme for multiple objects tracking during a surveillance operation. The objectives are to dynamically divide the processing operations into parallel units; to reduce the communication between CPU-GPU processing units. Parallel execution of a computational program is the method adopted, which is divided into phases such as the parallelism phase, computation phase, and interaction phase between CPU and GPU operations. The results achieved were used to reduce the time of operation for the input and the output cycles. This is a new effective approach for the other jobs to collaborate and algorithms for parallel job divisions. The research finding/observation during the review discovered that, out of the two algorithm approaches to be employed in the current paper, only one was discussed in the work reviewed. The Grid-wise reduction of computation to communication ratio approach was not discussed [1].

ii. Performance Comparison between OpenCV Built-in CPU and GPU Functions on Image Processing Operations. The aims were to compute the performance of some commonly used Image Processing operations, and compare OpenCV's built-in CPU and GPU functions that use CUDA; the objectives are the comparison of OpenCV's built-in CPU and GPU, and to measure the time spent at some commonly used image processing operations with using OpenCV's built-in CPU and OpenCV's built-in GPU functions.

    The methods used are Canny Edge Detector for detecting edges of an image; Linear Interpolation for resizing images; and C++ is using OpenCV's built-in CPU and GPU functions. The image size and time were chosen as performance comparison criteria. Measurements showed that GPU functions provided a performance improvement because they run in parallel but effects of GPU appear especially when image size increases. The research finding/observation during the review shows that, instead of OpenCV's built-in GPU functions, CUDA Toolkit's native functions and libraries can be used to do image processing operations and performance evaluation.[3].

iii. Optimized Block-Based Algorithms to Label Connected Components on GPUs. The paper aims to optimize existing GPU solutions by introducing a block-based approach; the objectives are to propose two new 8-connectivity GPU-based connected components labeling methods; and to extend YACCLAB (Yet Another Connected Components Labeling Benchmark), a public benchmark to evaluate the performance of sequential CCL algorithms. Two new algorithms have been proposed, Block based on Union Find (BUF) and Block-based Komura Equivalence (BKE) together with the dataset. Experiments on a wide selection of both real case and synthetically generated datasets confirm that the proposals represent the state-of-the-art for GPU-based connected components labeling. The datasets cover most of the fields where CCL is commonly used and allow the evaluation of the correlation of performance to specific characteristics of the input. Among the two proposals, BKE demonstrated superior performance in every test case, except for images with very low density. In fact, on random images, BUF has better performance than BKE for density below 5 - 10, depending on the granularity. The collection of datasets has been enriched with new real case and synthetic datasets of three-dimensional volumes. During the review, it is observed that the reviewed paper is a proposed article that has not been implemented [19].

iv. A comparative study of the implementation of SJF and SRT Algorithms on the GPU Processor using CUDA; with the aims to define and implement the two algorithms, the SJF (Shortest Job First) algorithm and the SRT (Shortest Remaining Time) in a single-wire CPU environment using the C language. Objectives are to implement the GPU using the CUDA C language; to compare the different performances of the implementation of the two algorithms on GPU and CPU processors; to verify the efficiency of this study. Shortest Job First Algorithm and Shortest Remaining Time Algorithm are used as methods to execute average waiting time and average execution time and C language for the codes.

    The results of the implementation show that the execution time achieved on the GPU for both algorithms is faster than on the CPU; the acceleration factor of the SRT algorithm is higher than that of the SJF algorithm, showing that the performance of SRT algorithms is more efficient than that of SJF algorithms. This shows the efficiency of using GPUs for parallel computing and obtaining the best performance. Despite this implementation by CUDA C, Nvidia still faces many challenges to keep CUDA C flexible to parallel task programming on GPUs [22].

v. Accelerating the Canny Edge Detection Algorithm with CUDA/GPU; with aims to have an efficient implementation of the Canny edge detection algorithm on GPU using CUDA. The objectives are to implement the Canny Edge Detection Algorithm on GPU using CUDA and to improve the quality of Edge Detection of an image. A Matlab script file is responsible for reading images and writing content in a text file. Two kernel functions are used to implement the non-maximum suppression step.

The first kernel function performs the interpolation, for each image pixel two additional pixels are obtained, which are used to create two respective images, keeping the same position of the pixel analyzed. The second kernel function uses three images to determine if the pixel takes the value 0 or maintains its value. A CPU-GPU hybrid approach is used to determine whether the pixels previously classified as possible edges become definitive borders or not. The interpolation in the Non-Maximum Suppression step is used to improve the quality of edge detection in the image The Canny edge detection algorithm on GPU using CUDA is implemented. Experimental results show that with the Canny algorithm implemented in GPU an acceptable speedup is achieved regarding CPU implementations. This speedup value increases as there is an increase in the size of the processed image; this indicates that the GPU devices for processing images in real-time.

At the review level, the research work observed that the Hysteresis Thresholding step occupies a considerable percentage of the total execution time. Therefore, it is advisable to consider other approaches for the implementation and improve the implementation done in this paper [2].

Nonadiabatic Molecular Dynamics on Graphics Processing Units: Performance and Application to Rotary Molecular Motors. The aim was to investigate the use of graphics processing units (GPUs) in addition to central processing units (CPUs) to efficiently calculate those properties at the time-dependent density functional theory (TDDFT) level of theory. The objectives are: to examine the use of graphics processing units (GPUs) for the evaluation of two-electron integrals in excited-state energies and properties; and to investigate a series of newly designed rotary molecular machines. Implementation was done based on the FermiONs++ program package that uses the J-engine and a pre-selective screening procedure to calculate Coulomb and exchange kernels, respectively. Good speed for small and large molecular systems achieved and reduced scaling behavior for the system size. The results were able to present efficient NAMD simulations of a series of newly designed light-driven rotary molecular motors and compare their S1 lifetimes. The review shows that the current implementation needs to extend toward decoherence corrections and triplet states [14].

3. Methodology

Two data collection models were used in carrying out this paper, such as questionnaire and experimental models while MS excel application package also used for the data analysis.

3.1 Questionnaire Model

Questionnaires were distributed among participants to generate a dataset. The CPU and GPU Research were carried out in four (4) different environments for data collection. Based on the formulated dataset, two hundred (200) questionnaires were produced; one hundred and one (101) questionnaires used for CPU, sixty-two (62) questionnaires used for GPU, and thirty-seven (37) questionnaires were returned unused. The gathered data determined the test value for both GPU and CPU as established in section 4.2. Questionnaire administration was used to establish the GPU/CPU users' acceptance testing and performance evaluation validation.

SAMPLE OF QUESTIONAIRE

Please complete the following questions with specific regards to the below enquiry by ticking in the appropriate box ☑

QUESTIONS	Strogly Agree	Agree	Uncertain not applicable	Disagree	Strongly disagree
1. CPU is general purpose resource	☐	☐	☐	☐	☐
2. GPU is specific purpose resource	☐	☐	☐	☐	☐
3. CPU can perform manipulation of computer graphics and image processing without slowing down the operation	☐	☐	☐	☐	☐
4. GPU can perform manipulation of computer graphics and image processing without slowing down the operation	☐	☐	☐	☐	☐
5. CPU can perform plain text without slowing down the operation	☐	☐	☐	☐	☐
6. CPU (Central Processing Unit) best option solution for performing parallel computing base on their scalability issues and cost	☐	☐	☐	☐	☐
7. GPU can perform plain text without slowing down the operation	☐	☐	☐	☐	☐
8. GPU (Graphical Processing Unit) best option solution for performing parallel computing base on their scalability issues and cost	☐	☐	☐	☐	☐
9. CPU and GPU has different architectural different	☐	☐	☐	☐	☐
10. CPU system operation encouraged more user's demand with reason	☐	☐	☐	☐	☐

Figure. 1: PDF Downloading Procedures for GPU/CPU

3.2 *Experimental Model*

The experimental model was the downloading of PDF and text files. During the experimental process, we observed that GPU was available on the video card or embedded on the motherboard, the CPU is available only on the motherboard, and the same procedures were acquired with both the CPU and GPU. The process to store is also achieved as part of the downloading procedures. The downloading processes occurred on the typical systems showcased in figures 2a and 2b. The procedures are available in figure 1.

Step 1. Download a free PDF reader.
Step 2. Find the PDF that you want to download.
Step 3. Right-click inside the PDF.
Step 4. Click Save As or Save Page As.
Step 5. Choose the location where you want to save the file.
Step 6. Type a name for the PDF

Figure. 1: PDF Downloading Procedures for GPU/CPU

Step 1: Download a free PDF reader.
A PDF reader allows users to open the PDF files once it has been downloaded. Examples of free PDF readers are Foxit Reader, Nitro PDF Reader, and PDF XChange Editor.

Step 2: Find the PDF to download.
The websites usually display the PDF within the web browser opens full-screen as part of home page.

Step 3: Right-click inside the PDF.
This step enables user to download the file in PDF format.

Step 4: Click Save As or Save Page As.
This opens a file browser to save and select files for use.

Step 5: Choose location to save the file.
Click the Quick Access folders in the left-sidebar of the file browser to select a location.

Step 6: Type a name for the PDF (optional).
Creation of file name or labelling.

Step 7: Click save
This will save the PDF file to the hard drive in the location specified in step 5.
Note that from figure 1, steps 4 to step 7 are the processes to store the PDF on both the CPU and GPU.

.

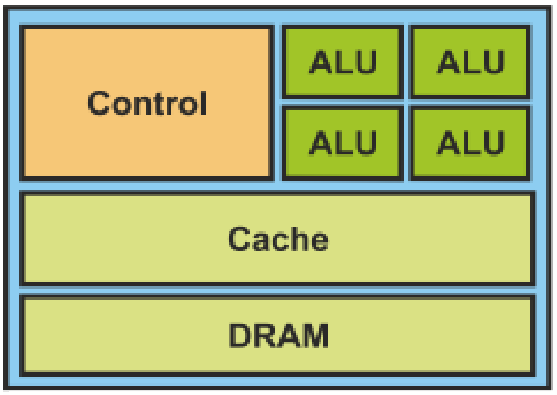

Figure. 2a: CPU

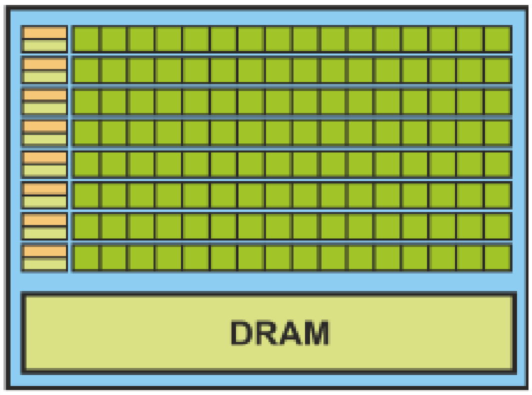

Figure. 2b: GPU

Experimental observations:

i. CPU has low computed density, while GPU has high computed density.

ii. CPU consists of complex control logic; GPU consists of simple computations per memory Access.

iii. Large caches for the CPU but GPU Built for parallel operations.

iv. In CPU, Optimization for serial operations is performed but GPU performed many parallel execution units (ALU).

v. Fewer execution units (ALU) are available with the CPU; Graphics is the best-known case of parallelism in GPU.

vi. Shallow pipelines for the CPU; deep pipelines for the GPU.

vii. CPU has low latency tolerance; GPU has high latency tolerance.

All the observations were measured and generated during experimental process based on the evaluation parameters.

Newer CPUs have more parallelism, also, the newer GPUs which are better flow control logic and gradually becoming more CPU-like as presented in figure 3.

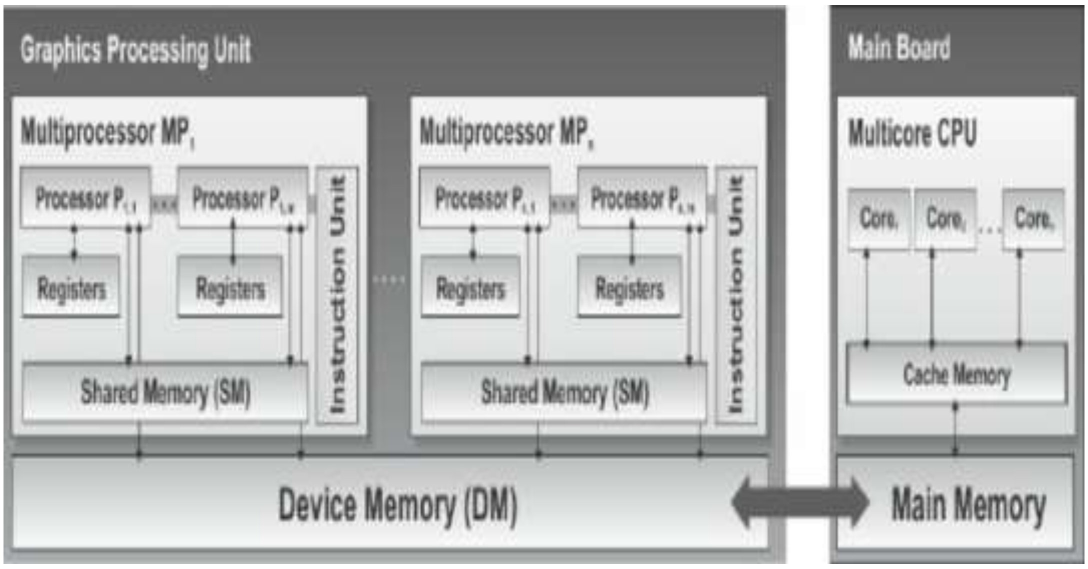

Figure 3: Newest versions of GPU/CPU

4. Results and Discussion

4.1 Questionnaire Results

Questionnaires were distributed among participants to generate a dataset. The CPU and GPU Research were carried out in four (4) different environments (Osun State, Oyo State, and Lagos State (two different locations) where GPU and CPU systems available for data collection. Based on the formulated dataset, two hundred (200) questionnaires were produced; one hundred and one (101) questionnaires used for CPU, sixty-two (62) questionnaires used for GPU, and thirty-seven (37) questionnaires were returned unused.

For CPU, out of one hundred and one (101) participants collected, one (1) participant voided the questionnaire, and twenty-one (21) participants accepted the use of CPU without any reason. Seventy-nine (79) participants rejected the CPU due to its performance based on low speed, image data stored on a vast volume of memory space, and a higher rate of time taken to download PDF and word (text only) files.

In the case of GPU, sixty-two (62) participants responded. Two (2) participants rejected the use of GPU without any reason, and three (3) participants attempted but had no valid conclusion. Fifty-seven (57) participants accepted the use of GPU due to its performance based on its high speed, image data stored on the minimum volume of memory space, and a lesser rate of execution time.

4.2 Experimental Results

The process by which data are tested using the GPU and CPU resources of the computer system to know which performed better based on the output produced for a given dataset is shown in Figures 4 and 5. These resources are tested for two weeks with different test data and environments shown in Tables 1 and 2.

Table 1: Execution Data Analysis Table for Week 1

DAY	START	ITEMS	SIZE			EXECUTION DATA ANALYSIS TABLE	
				GPU	CPU	GPU	CPU
SUN	6:05 AM	CLAY MINERALS (PDF)	2.89BM	6:30	6:50	25	45
MON	8:30 PM	CLAY MINERALS (PDF)	2.89BM	8:59	9:20	29	50
TUE	10:01 AM	CLAY MINERALS (PDF)	2.89BM	10:28	10:42	27	41
WED	12:00 PM	CLAY MINERALS (PDF)	2.89BM	12:26	12:38	26	38
THUR	7:40 AM	CLAY MINERALS (PDF)	2.89BM	8:03	8:09	23	29
FRI	1:00 PM	CLAY MINERALS (PDF)	2.89BM	1:30	1:42	30	42
SAT	9:30 AM	CLAY MINERALS (PDF)	2.89BM	10:02	10:20	32	50

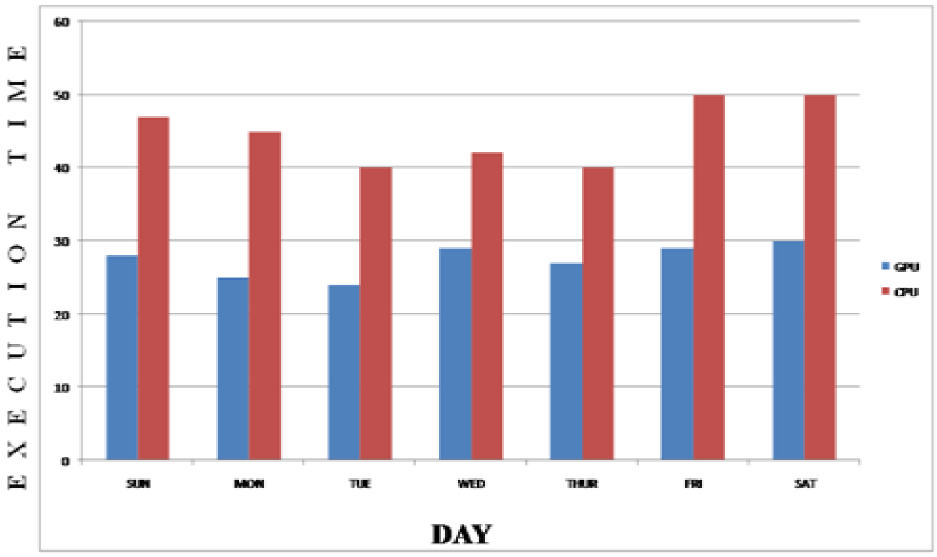

Figure 4: EXECUTION DATA CHART FOR WEEK 1

The execution-time data analyses (Tables 1 & 2) were used to generate the chart (Figures 4 & 5) that shows the differences between GPU and CPU operations. The analysis of the dataset through questionnaire results shows that a lesser number of users demand GPU due to its high price, and a large number of users demand CPU due to its low cost of finance, despite its disadvantages.

Table 2: Execution Data Analysis Table for Week 2

DAY	START	ITEMS	SIZE			EXECUTION DATA ANALYSIS TABLE	
				GPU	CPU	GPU	CPU
SUN	7.05 AM	GEOMAGNETIC (PDF)	1.89MB	7:33	7:52	28	47
MON	9:05 PM	GEOMAGNETIC (PDF)	1.89MB	9:30	9:50	25	45
TUE	8:10 AM	GEOMAGNETIC (PDF)	1.89MB	8:34	8:50	24	40
WED	10:00 PM	GEOMAGNETIC (PDF)	1.89MB	10:29	10:42	29	42
THUR	10:08 AM	GEOMAGNETIC (PDF)	1.89MB	10:35	10:48	27	40
FRI	6:00 PM	GEOMAGNETIC (PDF)	1.89MB	6:29	6:50	29	50
SAT	11:30 AM	GEOMAGNETIC	1.89MB	12:00	12:20	30	50

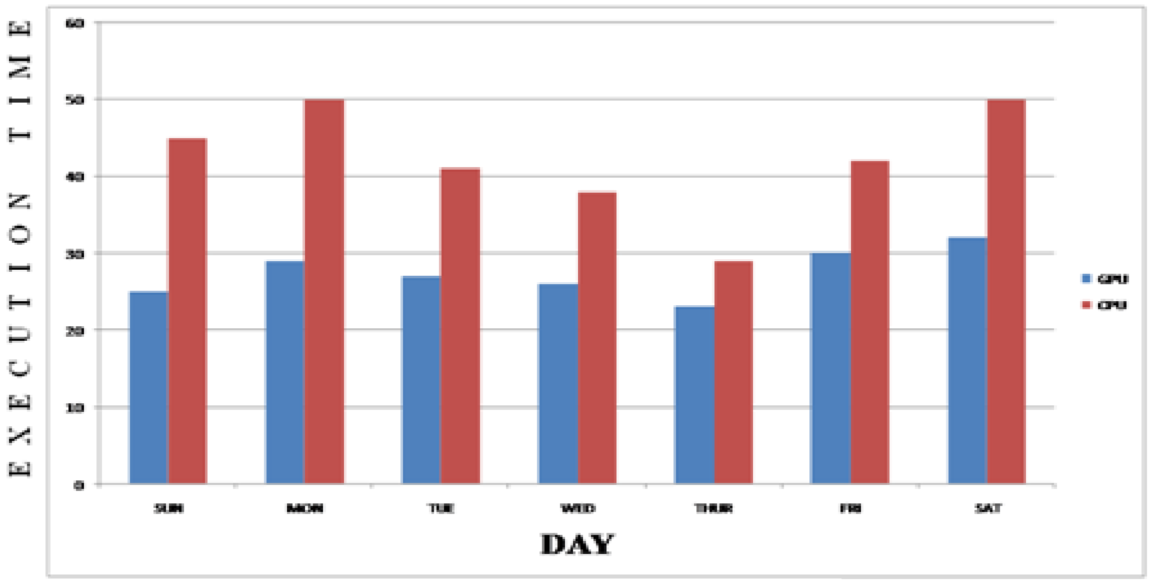

Figure 5: EXECUTION DATA CHART FOR WEEK 2

4.2 Discussion

Based on the results generated the questionnaire model shows that a lesser number of users demand GPU due to its high price, and a large number of users demand CPU due to its low cost of finance, despite its disadvantages; while the experimental model shows that the GPU works faster than the CPU in terms of speed and storage capacity. The results also show the functionality and performance level of the two processors that allow the GPU to obey the laws of the user interface design principle, which states that "the computer shall not waste the user's time (first law); the computer shall not harm the user's work (second law)". Also, the results show that GPU satisfied the importance of human-computer interaction (HCI) by securing user satisfaction.

The size of the two test data used shown that GPU has more capacity for image storage without affecting its functionality and performance level, unlike the CPU. While downloading the file with image and without image for the GPU and CPU, the result showed that the GPU was faster than the CPU, and also, GPUs can perform computations much faster than the traditional central processing units (CPUs) used in today's workstations.

5. Conclusion

The data collected shows that weather interference during execution affects the computations to get an accurate result. These will have an advanced effect on the values of data generated. It was also discovered that the GPU covered up the reading ability, the CPU has no opportunity, because, the CPU will read puzzle by puzzle, that is why the CPU is not good enough for the various organizations.
During the research, the framework for the workflow model was required; questionnaires on CPU and GPU were analyzed.

The analysis of the dataset obtained through a questionnaire, results show that there is low GPU demand within our local environment, due to its high-cost implication and lack of awareness of its advantages among users. With GPU, all PCs have chips that render the display images to monitors such as Intel's integrated graphics controller. Intel's integrated graphics controller provides basic graphics that can display only productivity applications like Microsoft PowerPoint, low-resolution video, and basic games. GPUs are capable of performing high-end computations that are the staple of many engineering activities showing that GPUs can perform faster than the CPU.

With all of these, this paper recommends that users in the area of the Photography industry, Cinema and Film Production companies, etc.

should be engaged with the use of GPU rather than CPU for more efficient execution of tasks and minimize the storage devices.

References

[1] Ankush Rai, Jagadeesh Kannan R (2017). - “Central Processing Unit-Graphics Processing Unit Computing Scheme for Multiobject Tracking in Surveillance”, Asian Journal of Pharmaceutical and Clinical Research. DOI: 10.22159/ajpcr.2017.v10s1.19651.

[2] Bárbaro Maykel López-Portilla Vigil (2015)- “Accelerating the Canny Edge Detection Algorithm with CUDA/GPU” Telecommunications and Electronics Department, University of Pinar del Río, Cuba.

[3] Batuhan Hangün, Önder Eyecioğlu (2017) - “Performance Comparison Between OpenCV Built-in CPU and GPU Functions on Image Processing Operations”, International Journal of Engineering Science and Application, 1(2).

[4] Blackford, L, S., Hammarling, S., Cleary, A., Petitet, A., Whaley, R.C., Demmel, J., Dhillon, I., Ren, H., Stanley, K., Dongarra, J. (1997) Practical experience in the numerical dangers of heterogeneous computing. ACM Transactions on Mathematical Software (TOMS)23: 133-147.

[5] Fernando, R. (2004) GPU Gems: Programming Techniques, Tips and Tricks for Real-time Graphics. Addison-Wesley Pub (Sd) Gpgpu.org: General-purpose computation on GPUs (GPGPU), http://gpgpu.org/

[6] Higham, N.J. (1990) Exploiting Fast Matrix Multiplication within the Level 3 BLAS. ACM Transactions on Mathematical Software 16 352-368.

[7] Hillesland, K., Lastra, A.s (2004): GPU floating-point paranoia. In: Proceedings of GP2. http://msdn.microsoft.com/directx/

[8] Jesse D. Hall, Nathan A. Carr, J.C.H. (2003) Cache and Bandwidth Aware Matrix Multiplication on the GPU. Technical report, University of Illinois Dept. of Computer Science.

[9] Jiang, C., Snir, M. (2005) Automatic Tuning Matrix Multiplication Performance on Graphics Hardware. In: Proceedings of the 14th International Conference on Parallel Architectures and Compilation Techniques (PACT'05). 185-196.

[10] John Montrym (2005), H.M. THE GEFORCE 6800. IEEE MICRO 2005, 25(2).

[11] K.Fatahalian, J.Sugerman, P.Hanrahan. (2004) Understanding the Efficiency of GPU Algorithms for Matrix-Matrix Multiplication. In: Graphics Hardware.

[12] Kruger J., Westermann, R. (2003) Linear Algebra Operators for GPU Implementation of Numerical Algorithms. In: Proceedings of ACM SIGGRAPH 2003.908—916.

[13] Larsen, E., McAllister, D. (2001) Fast matrix multiply using graphics hardware. In: Proceedings of the 2001 ACM/IEEE conference on Supercomputing.

[14] Laurens D. M. Peters, Jörg Kussmann, and Christian Ochsenfeld (2019) - “Journal of Chemical Theory and Computation” DOI: 10.1021/acs.jctc.9b00859.

[15] Mark Harris (2007). High-performance computing with CUDA – Optimizing CUDA.

[16] Moreland, K., Angle, E. (2003) The FFT on a GPU. In: Proc. SIGGRAPH/EUROGRAPHICS Workshop Graphics Hardware. 112-119.

[17] Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Kruger, J., Lefohn, A.E., Purcell, T.J. (2005) A Survey of General-purpose Computation on Graphics Hardware. In: Eurographics 2005, State of the Art Reports. 21-51.

[18] Shinomoto, y., miwa, s., shimada, h., mori, s.i., nakashima, y., tomita, s. (2005) Consideration for Speculative Rendering in PVR. In: IPSJ SIG Technical Reports. 2005-ARC-164 145-150.

[19] Stefano Allegretti, Federico Bolelli, Costantino Grana (2019) - “Optimized Block-Based Algorithms to Label Connected Components on GPU”, IEEE Transactions on Parallel and Distributed Systems. DOI: 10.1109/TPDS.2019.2934683.

[20] Thompson, C.J., Hahn, S., Oskin, M. (2002) Using Modern Graphics Architectures for General-purpose Computing: A Framework and Analysis. In: Proceedings of the 35th annual ACM/IEEE International Symposium on Microarchitecture. 306-317.

[21] Whaley, R.C., Petitet, A., Dongarra, J.J. (2001) Automated Empirical Optimization of Software and the ATLAS Project. Parallel Computing 27 3-3.

[22] Youness Rtal, Abdelkader Hadjoudja (2021). “A Comparative Study of the Implementation of SJF and SRT Algorithms on the GPU Processor Using CUDA”, EAI Endorsed Transactions on Internet of Things. DOI: 10.4108/eai.8-2- 2021.168689.