

# GPU / CPU Benchmark Test

Nathan Annoh-Kwafo

## Abstract

Some may argue you do not need a Graphics Processing Unit (GPU) but only need a Central Processing Unit (CPU) to view high quality videos, images and to some extent play some video games. Both sides of this argument may have a valid point, but the presence of a GPU makes a huge difference in the performance of a computer when displaying digital images. This benchmark program tests the hypothesis of this argument using a diverse sample size of high-quality videos on different caliber of computer systems.

When it comes to CPU usage, the size of the main memory (RAM), number of caches the computer system possesses and the clock rate of the CPU among others all play a role in its performance. When it comes to multimedia files, most CPUs are tasked with decoding the media format when it is being displayed and encoding when the digital media is being compressed.

Likewise depending on the computer's architecture and operating system, the GPU may either be responsible for giving the user the best graphical experience or may be managed with the CPU to produce the desired user effect. In this project we get to observe both instances previously stated and view data that show on the abstract level of how different companies such as Intel and Nvidia manage to give users the desired visual quality from a computer.

## Introduction

This project was initially aimed at testing system GPUs only, which took a slight turn to including CPU testing at a relatively low level since not all computers capable of playing videos contain a GPU. Although the initially considered language was C, the benchmark project was written in Java, and is compatible with numerous versions of currently active Windows Operating Systems. This project generally extracts relevant information concerning the computer hardware, before running 4 tests that contain 4 different qualities of ".mp4" videos.

The MP4 (MPEG-4 Part 14) file format is one of the earliest digital video formats introduced in the early 2000s. These files are generally compressed versions of video files, still images, audio files and in some cases text that maintain the quality of the digital media with a relatively small size which make them ideal web media file formats.

In this benchmark program, the information concerning the number of cores used for each test is appended to a unique report text file containing the current computer information. The data obtained is then manually accumulated in a Microsoft Excel worksheet and analysed with Tableau.

## **Objectives**

- Test the current performance and video display capabilities of target computer systems.
- Obtain data that depict the usage of both CPU and GPU (where applicable) when it comes to graphic presentation by different hardware manufacturers and different Operating systems.

## **Design and Development**

### **Some Specific System Requirements**

- Windows Operating System (preferably Windows 7 upwards)
- Java Runtime Environment
- JavaFX library
- A functional MP4 media player

### **Tools and Test Resources**

- Visual Studio code
- Microsoft Excel
- Tableau
- Windows Task Manager
- Test Videos:
  - 171124\_B1\_HD\_001.mp4
    - Size: 69.7 MB
    - Length: 00:00:19
    - Frame Dimension: 1920 x 1080
    - Frame Rate: 25.00 frames/second
    - Data Rate: 30465kbps
    - Total bitrate: 30786kbps
  - pexels-ambientnature-atmosphere-5956783.mp4
    - Size: 72.4 MB
    - Length: 00:00:24

- Frame Dimension: 3840 x 2160
- Frame Rate: 29.97 frames/second
- Data Rate: 25067kbps
- Total bitrate: 25324kbps
- pexels-bethe-observer-6906495.mp4
  - Size: 94.1 MB
  - Length: 00:00:31
  - Frame Dimension: 3840 x 2160
  - Frame Rate: 59.94 frames/second
  - Data Rate: 25224kbps
  - Total bitrate: 25481kbps
- production ID\_4913621.mp4
  - Size: 50.9 MB
  - Length: 00:00:17
  - Frame Dimension: 3840 x 2160
  - Frame Rate: 25.00 frames/second
  - Data Rate: 23758kbps
  - Total bitrate: 23758kbps

## Design and Development Process

To kick start the project, the necessary *.jar* files had to be added to the VS Code IDE library. As the java language requires, the necessary *awt*, *io* & *util* libraries were imported. At the beginning of the program, there is a general print statement displayed in the console to reassure the user the program is running until the first test video appears.

The initial major aspect of the program was obtaining the current computer system information which includes the OS Name, OS version, Total Physical Memory (MB) and Total Physical Memory (MB) among others. This was done by executing the *systeminfo* command in a process. The result of the previously invoked command is extracted with a buffer reader and assigned to a print writer which will be later logged into a report text file with a system time suffix. To keep up with modern practice and to ensure all resources used are properly closed, *try with resources* blocks are used to manage all resources. All subsequent system information were also obtained through system environment variables such as *PROCESSOR\_ARCHITECTURE* and *PROCESSOR\_LEVEL*.

Separate methods were created for individual tests that returned a string value of the number of processor cores used during the test. Before coming across the Desktop class

in the Java API, creating a JFrame to play the test videos was highly considered but was later aborted because creating and maintaining will have added unnecessary CPU usage which would not reflect its usage in playing the test video.

After each test video was initialized, the thread running the benchmark application is put to sleep for the first relative half of the video and reloading to obtain the current number of processor cores used to run the test video. The thread is then put back to sleep till the test video ends. The number of processors obtained is returned and logged into the report file. The above process is repeated for all tests.

Subsequently, all the significant data obtained is manually added to an excel report sheet and later imported to a Tableau workbook for analysis.

## **Testing and Analysis**

When it came to testing, the general format used on each test computer system was to:

1. Download and configure JavaFX if it was not already configured on the target computer system.
2. Open the benchmark program's source code in an IDE of the user's choice.
3. Open the Windows Task Manager and run the benchmark program.
4. Observe and record the maximum and minimum CPU and GPU values.
5. Repeat steps 3 and 4 multiple times, preferably 3 or more times and record the average maximum CPU and GPU values in the project Excel worksheet.

Once all the data was obtained from the 4 test computers (Huawei MACH-WX9, Huawei MACH-WX10, Dell Inspiron 7573 and Dell Precision Tower 3620) the Excel worksheet was imported and analysed in Tableau. See [Appendix Tableau Analysis](#) for results.

## **Discussion and Conclusion**

Computers do not solely rely on graphics cards to be able to play videos. Depending on the Operating System installed and the manufacturer of the computer. A large portion of the of the CPU or the GPU may be used to display certain kinds of videos, but the CPU is ultimately used to decode the encoded video before it is being processed and displayed hence creating a spike in the percentage of CPU used by the computer system at the beginning of videos.

## **Recommendation**

- For similar future developments, it would be more ideal to write the benchmark program in C or a similar low-level programming language to easily get access to the computer hardware and live CPU and GPU values without relying on the Task Manager.
- Expand the benchmark program to accommodate other Operating Systems.
- Create videos by using either DSLR camera or any mobile device that could give more information concerning the encoding algorithm used for the video.
- Automate the information logging procedure to excel to prevent manually inserting the data obtained by the benchmark program.

Appendix

Tableau Analysis

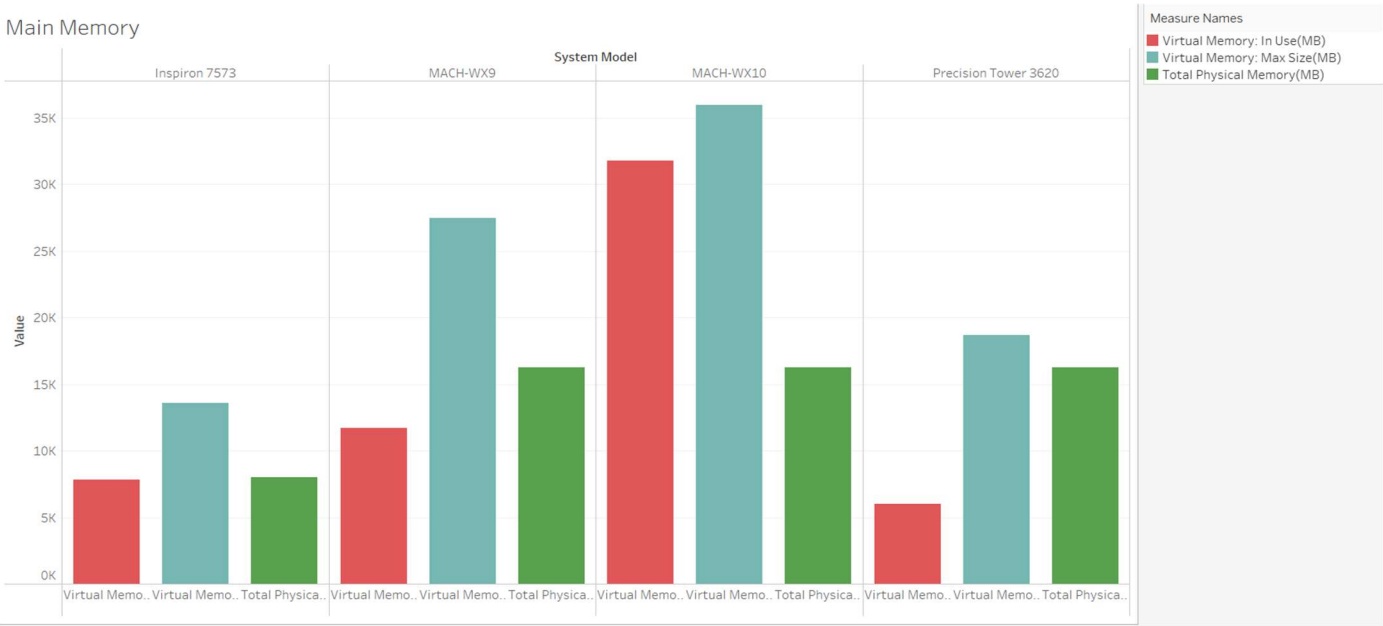


FIG 1



FIG 2

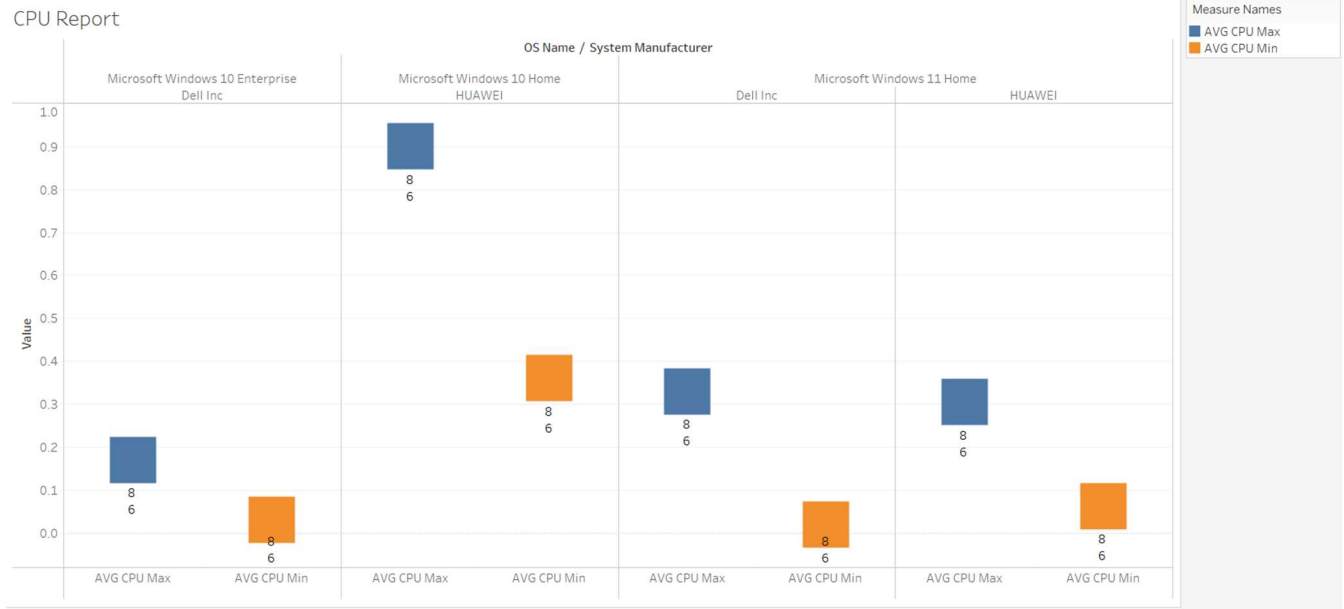


FIG 3

## References

- Project Github
  - [https://github.com/nannohk/GPU\\_Benchmark.git](https://github.com/nannohk/GPU_Benchmark.git)
- Java Requirements
  - <https://java.com/en/download/manual.jsp>
  - <https://openjfx.io/>
- Test video sources
  - <https://www.pexels.com/videos/>
  - <https://www.videvo.net/>
- Other information sources
  - <https://docs.oracle.com/en/java/javase/15/docs/api/index.html>
  - <https://stackoverflow.com/>
  - <https://docs.microsoft.com/en-us/windows/win32/apiindex/windows-api-list>
  - [https://en.wikipedia.org/wiki/MPEG-4\\_Part\\_14](https://en.wikipedia.org/wiki/MPEG-4_Part_14)
  - <https://www.google.com/webhp?hl=en&sa=X&ved=0ahUKEwio0onulJH3AhUIkYkEHQZRBI4QPAgI>
  - <https://www.computer.org/publications/tech-news/trends/8-best-video-file-formats-for-2020>