Analyzing GPU Performance Under System Stress

Advanced Computer Systems

Charles Clarke

12/6/2024

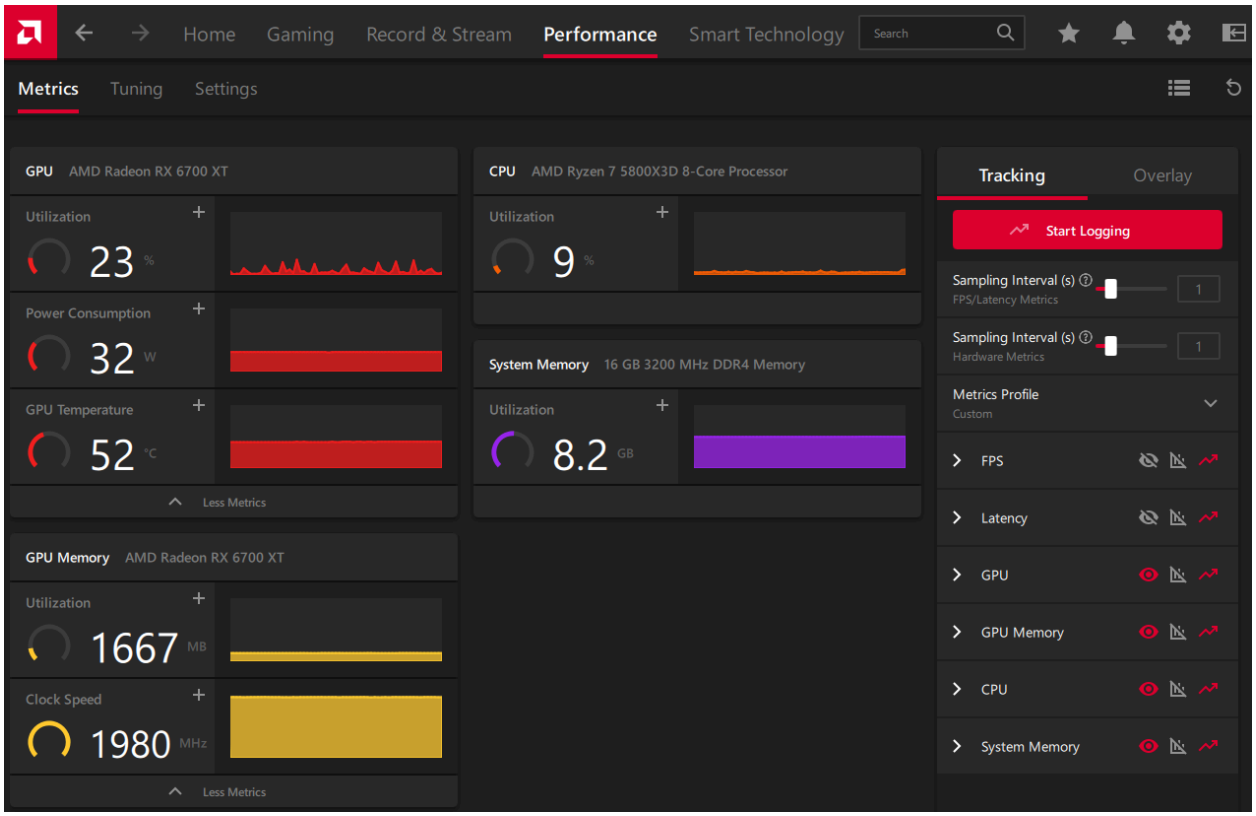
Introduction

The graphics processing unit or GPU is an essential piece of any modern computing system. It consists of many little ALUs that are able to complete lots of basic operations very quickly. This makes GPUs ideal for applications such as gaming, machine learning, and rending. The GPU can do a lot of work on its own, however would be nothing without the help of the CPU, memory, and external storages. This harmony between the components within the modern computer is what I will be exploring in this experiment. I intend to collect data on the GPU's performance in a few different scenarios that I will explain shortly. These scenarios will focus on stressing a component that assists the GPU and observe how the GPU reacts to this indirect stress.

Setup

I'm lucky enough to have quite a competent computing system in my room here at RPI. This system contains a GPU perfect for some stress testing, thanks to the AMD Adrenaline software. This software allows me to do a few things. First of all I can monitor a few key metrics. In relation to the GPU, Adrenaline measures utilization, power consumption and temperature. I can also monitor CPU and system memory

utilization which are key for my experiment. The Adrenaline Software can be seen below, while my computer isn't working too hard.



Now that I can measure the performance of the GPU using a few key metrics, and verify that I'm stressing my CPU and memory, I can start to collect data. The question is, how will I consistently test my GPU the same amount repeatedly. A possible choice is to render something, or play a game. However I believe that these methods won't keep the test constant enough to be sure that the data recorded was from the stress on other components.

I will be using a tool called Heaven benchmark to test the GPU repeatedly. This benchmarking tool plays a scene designed to test the GPUs ability to render all the frames. After 4 or so minutes of this test, it returns with a score, based on how many frames the GPU was able to produce over the 26 benchmarking scenes. This score will be a large part of the data comparison seen later in this report, as it gives a raw metric of how well the GPU performed. A screenshot from the Heaven benchmark can be seen below to help visualize the tool.

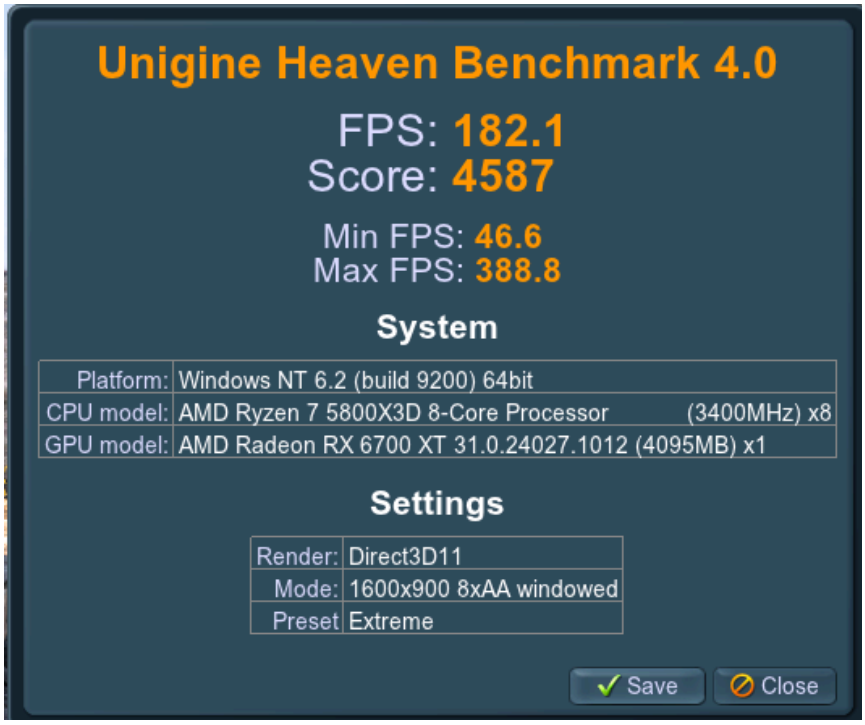
Now that I have a way to track the GPU's performance, and a way to stress the GPU repeatedly, I need a way to introduce stress into the other components I'm looking to test. I will be focusing on mainly the CPU and the memory in this experiment. To stress these components I will be using two C++ programs. One will stress the CPU by

computing many multi-threaded floating point operations. The other will stress the memory by completing lots of random accesses over the course of the benchmark's runtime. These two programs can be found on my github page.

Putting all of these pieces together, I can run the Heaven benchmark to obtain a score detailing how well my GPU performed. While that is running I can record the data using AMD Adrenaline software to obtain metrics like GPU temperature and utilization. In the background I can run either one of my programs to see a noticeable difference in CPU or memory utilization in the Adrenaline software which will affect the score received on heaven benchmark. All of these programs work together to help me analyze the GPU's performance under system stress.

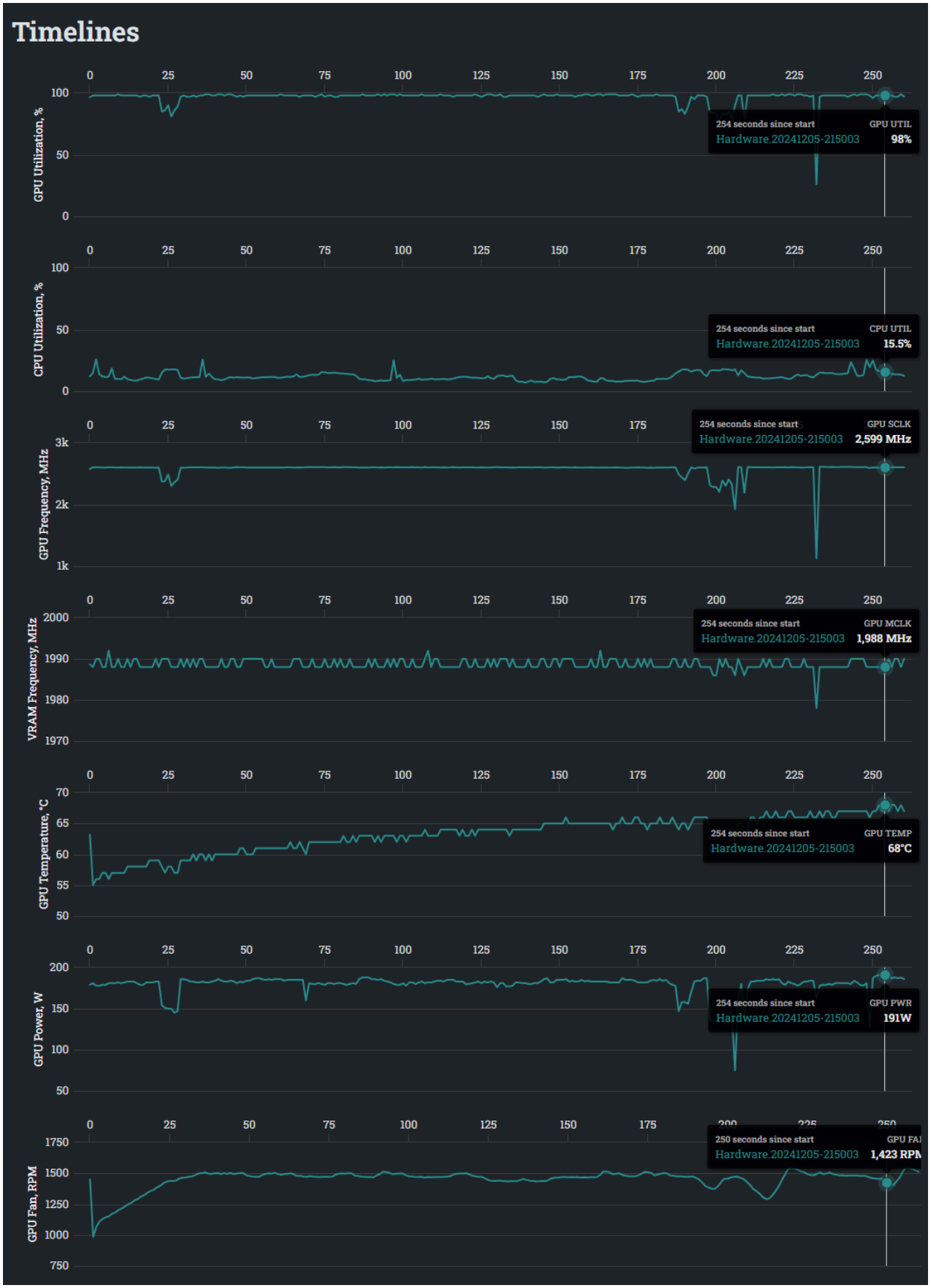
Control Test

The first step in any experiment is to obtain some control data. In this case I will run the Heaven benchmark with no stress on the system whatsoever, no programs running, no spotify, no google chrome in the background. This will give me a baseline measure of how the GPU performs when completing this task. In order to keep the experiments consistent, I will use default extreme settings on Heaven for each test to ensure the only factor being changed is the stress on other parts of the system. The first test went great, and my GPU received a score of 4587 as you can see in the screenshot below.



This score will act as the baseline for the future tests I perform on the GPU. Alongside this score, AMD Adrenaline provides the user with a CSV file containing very insightful data that was recorded during the benchmark. This file isn't too helpful unless there is a way to visualize the data. Thankfully someone in the past created a website to turn the CSV file into interactive graphs. These can be seen below.

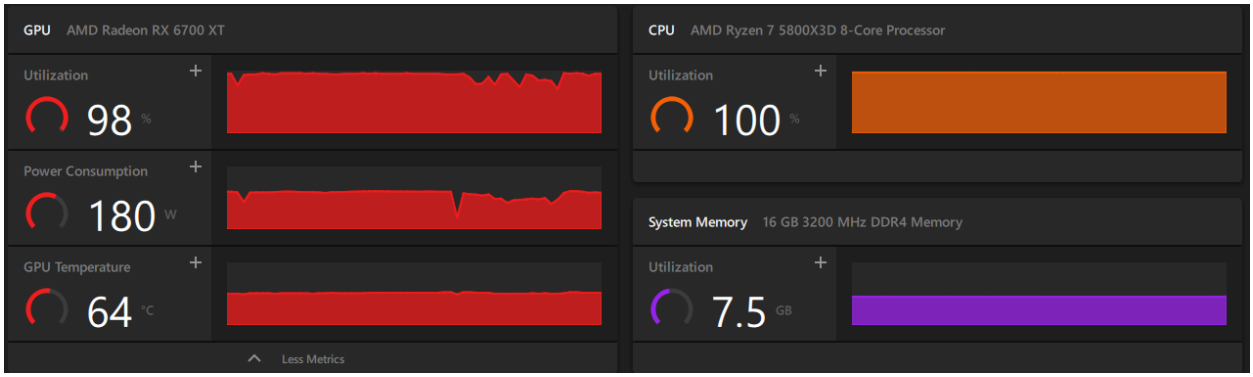
Timelines


GPU Utilization, %
254 seconds since start
Hardware.20241205-215003
GPU UTIL
98%
CPU Utilization, %
254 seconds since start
Hardware.20241205-215003
CPU UTIL
15.5%
GPU Frequency, MHz
254 seconds since start
Hardware.20241205-215003
GPU SCLK
2,599 MHz
VRAM Frequency, MHz
254 seconds since start
Hardware.20241205-215003
GPU MCLK
1,988 MHz
GPU Temperature, °C
254 seconds since start
Hardware.20241205-215003
GPU TEMP
68°C
GPU Power, W
254 seconds since start
Hardware.20241205-215003
GPU PWR
191W
GPU Fan, RPM
250 seconds since start
Hardware.20241205-215003
1,423 RPM

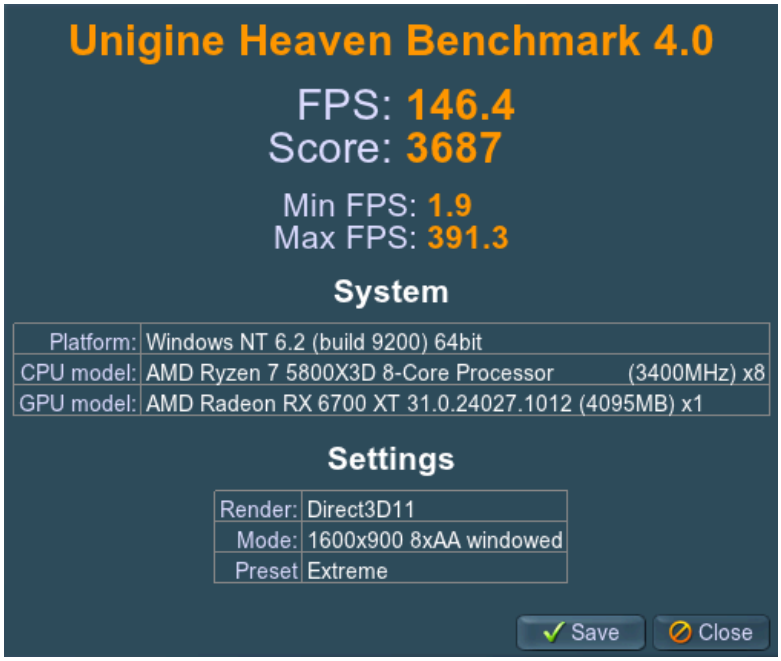
I absolutely adore this website as it gives the user an incredible way to visualize the metrics recorded during the benchmark. I think it's important to take note of the GPU utilization, temperature, and power graphs, as they show just how much it is working. These metrics will be important for comparisons once the CPU or memory are stressed in later tests.

CPU Stress

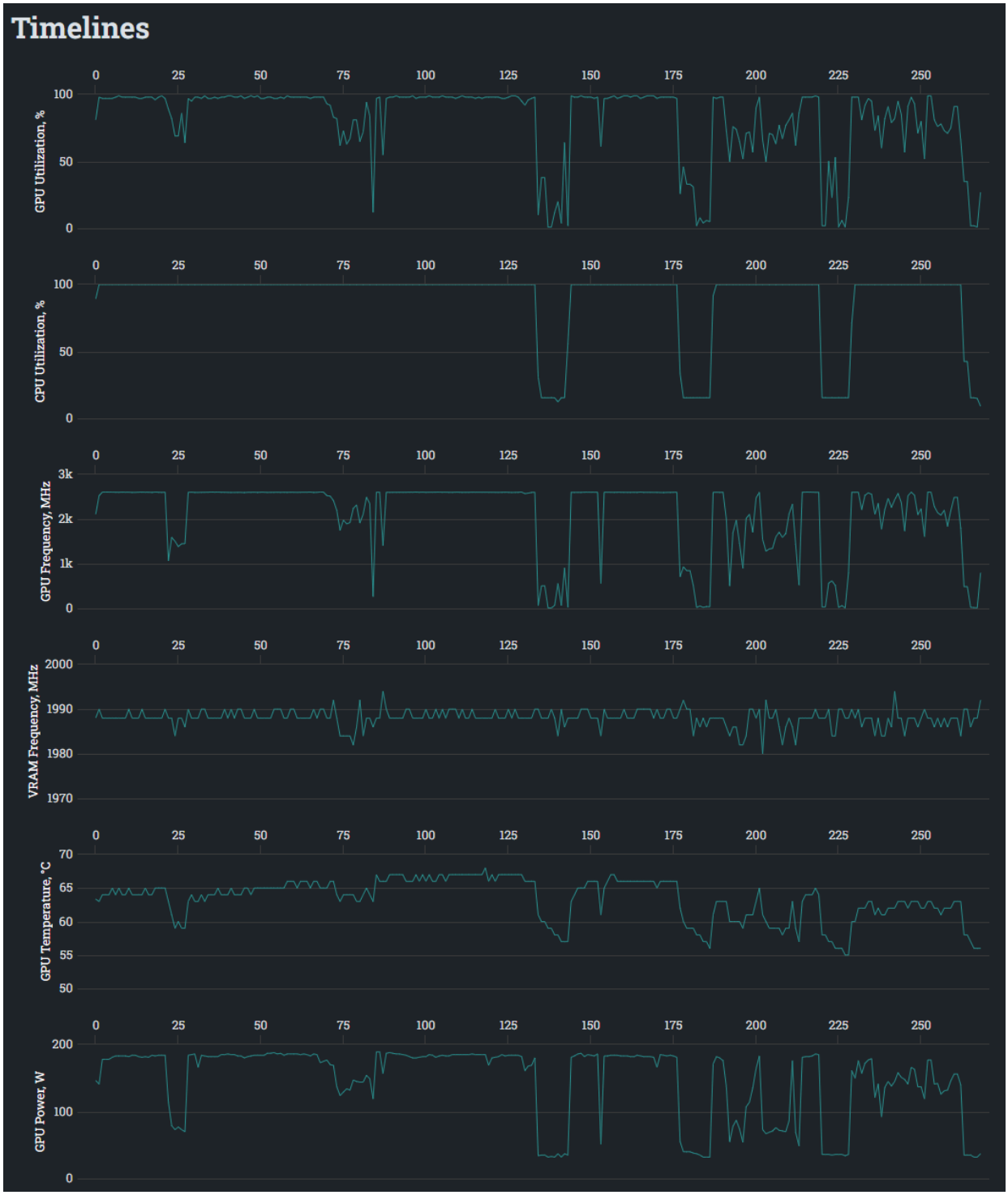
Next I conducted the GPU performance test while stressing the CPU, recording data using the same method as the previous control test. The program I'm using to stress the CPU does so by using all of the cores and threads available to compute floating point operations. This makes sure the CPU's utilization is very high as seen below in the Adrenaline screenshot.



Now that the CPU is being used at full capacity, I can run the GPU benchmark and start tracking metrics again to eventually compare the data. After completing this test I received the following score.



As you can see by the scores above from the heaven benchmark, the GPU performed much worse when the CPU was stressed 100%. The GPU received a score of 3687 which is roughly 20% less than the control score. This just shows how important the CPU is in helping the GPU do its job. The real time metrics of the test can be seen below.

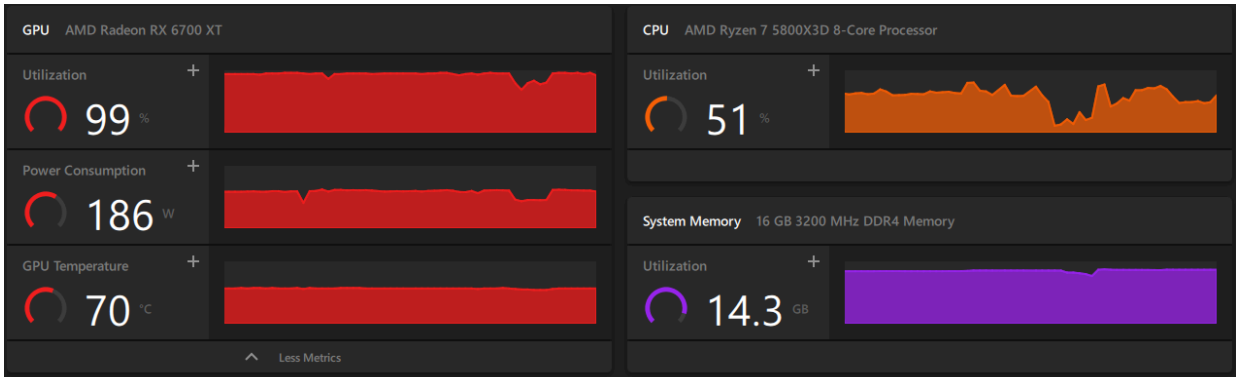


As you can see above, these graphs are much less stable than the control test, with many instances where both the CPU and GPU utilization drop down to almost

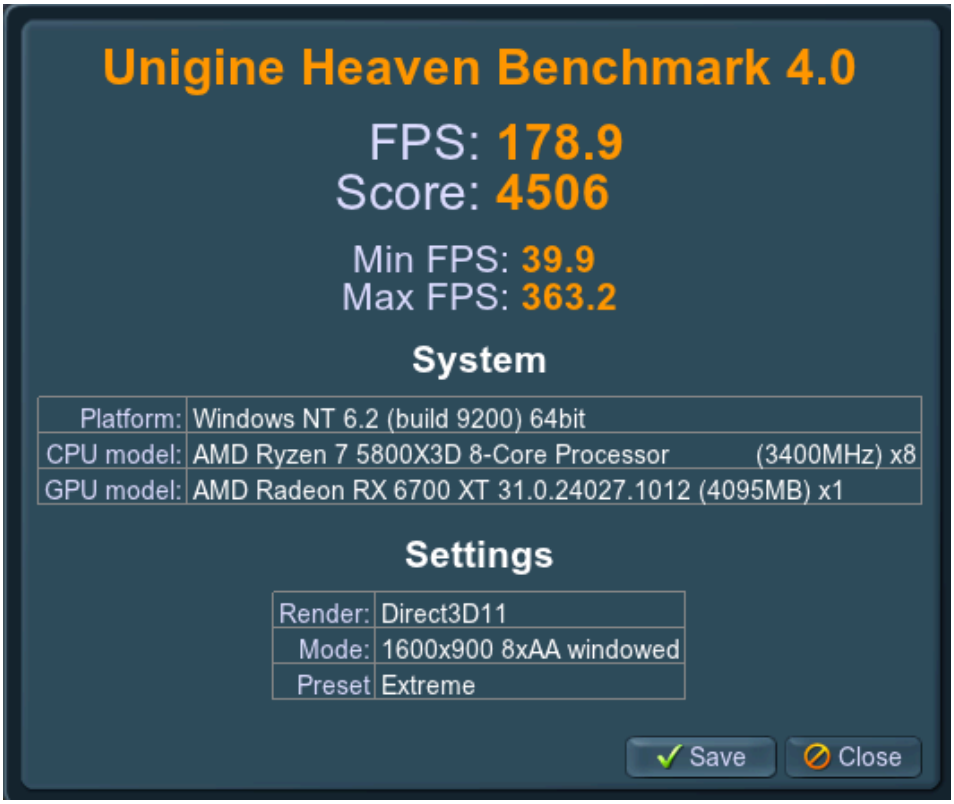
nothing. This is because the system is being worked too hard while both of them are stressed as in this case.

Memory Stress

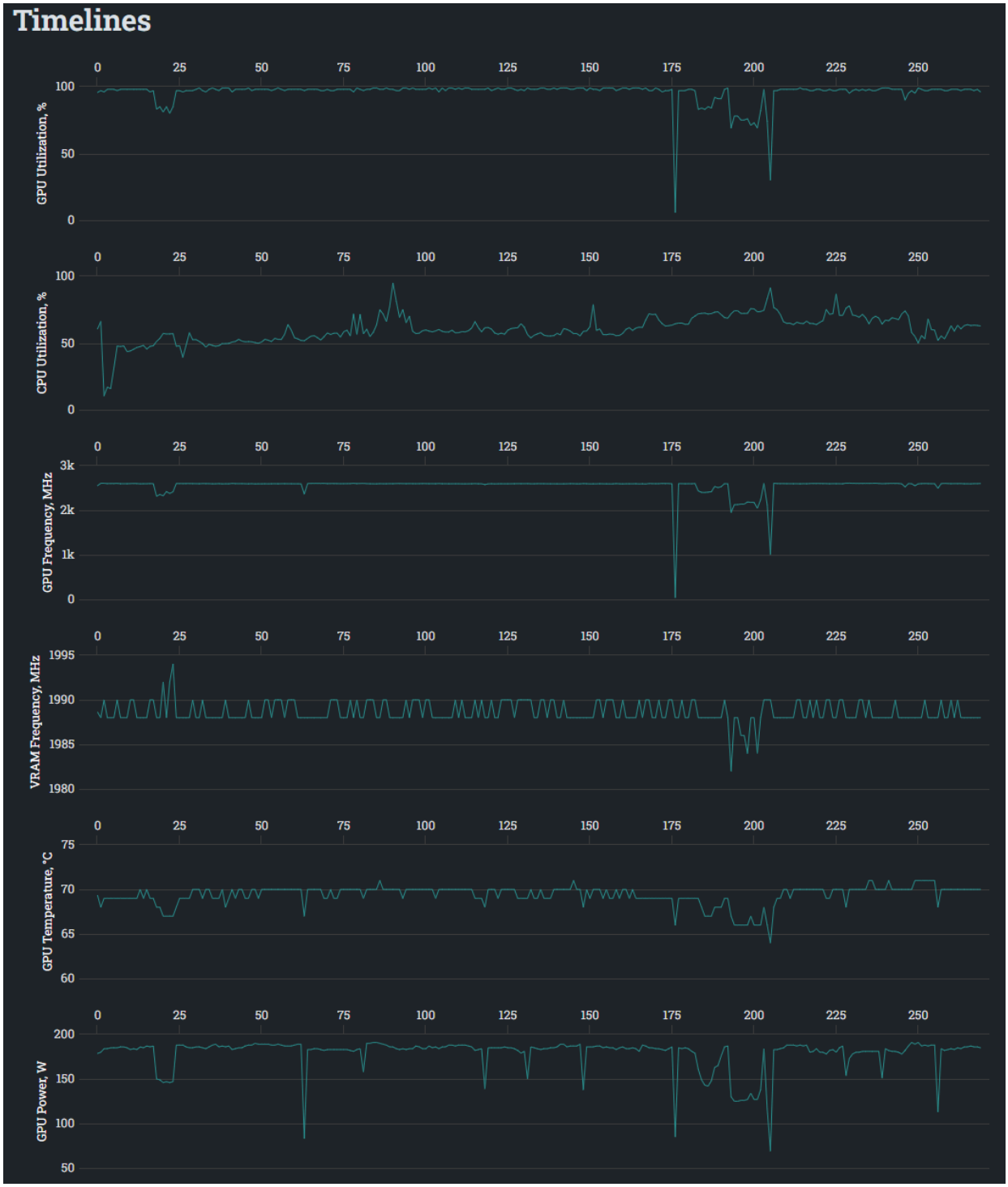
The final test will be very similar to the last one, however instead of stressing the CPU, I will be stressing the system memory. This test shouldn't interfere with the GPU as much however, as the CPU plays a much greater role in the GPU's performance than the memory. AMD Adrenaline can be seen below as the memory stressing program was running. This program works by executing many random memory accesses.



Here you can see that the system memory usage is drastically higher than seen in the previous tests. This means that the program is working as expected and I can continue with the test. The method to run this test will be the same as before, run the stressing program and the benchmark in parallel while recording data using Adrenaline. The results from the benchmark can be seen below.



From the results, it's clear that there was very little effect on the GPU when the memory was stressed. I expected there to be a response more similar to that of the CPU, however we only see about a 2% deviation in score and FPS compared to the 20% from the previous test. From this data it is expected that the graphs won't look too different from the control test, and this can be seen below.



This image above concurs with the theory that the memory wouldn't have much of an effect on the GPU's performance. Memory stress mainly affects shared memory

bandwidth and latency. This is why it only really impacts GPU workloads if it depends on frequent data transfers to and from the RAM.

Conclusion

The GPU is an incredibly important component of any computing setup, especially those attempting to render, play games, or run machine learning algorithms. This is because they excel at brute forcing large amounts of data, thanks to many many ALUs. This is in contrast to the CPU which is proficient in all types of operations, however struggles to do many of one operation as fast as the GPU can manage.

In this experiment I looked into the GPU's performance under different types of system stress. Specifically the CPU and system memory. After running the heaven benchmark in three scenarios, unstressed, with CPU stress, and with Memory stress, I received some great data. The GPU performed about 20% slower while the CPU was being fully utilized, and 2% while the memory was utilized. This means the CPU has about 10x the effect on the GPU's performance than the memory. This is due to many different factors such as the CPU's dependency in GPU workflows, or different stress patterns.

Overall I think this experiment gave me a deeper understanding and look into how the GPU works in its environment, the system. It heavily relies on some components, and less on others, depending on the task at hand. I learned that for a simple graphics benchmark like this, the memory had nowhere near the effect of the

CPU. I hope to try this again in the future using different types of GPU performance tests, such as gaming or machine learning.