

Visualization in the Cloud

Memory and Processor Utilization on Nvidia K2200 and K5000

Blessing Leonard

University of Baltimore

Baltimore, MD

blessingleonard@yahoo.com

ABSTRACT

This paper focuses on the effect of visualization on two Linux machines, each equipped with different Nvidia GPUs. The study used: two of Nvidia's GPUs (the K5000 and K2200), ParaView 5.3.0 for the visualization processes, and the nvidia monitoring tool in Linux command line for recording the amount of resources being used.

CCS CONCEPTS

• **Networks** → Network reliability;

KEYWORDS

ParaView, benchmark, streaming, GPU, dataset, sm%

ACM Reference format:

Blessing Leonard. 2017. Visualization in the Cloud. In *Proceedings of The International Conference for High Performance Computing, Networking, Storage and Analysis, Denver, Colorado USA, November 2017 (SuperComputing '17)*, 4 pages.

https://doi.org/10.475/123_4

1 INTRODUCTION

The modern world is plagued with a sea of information but not enough time to navigate through to find the most important details. This is where large scale visualization is useful, it relays the most important insights using high resolution images so that all distractions are eliminated thereby reducing the time spent on reviewing large databases and eradicating confusions about their key values. With the movement of data storage to the cloud, large scale data visualization can be used to foster collaboration in real time. This possibility gives rise to the need to stream the visualized data in real-time. The goal of this research was to identify the roadmap to the challenges involved in, and the requirements for implementing visualization in the cloud that allows for real-time streaming. To tackle these rising needs, a collaboration was made with the Clemson University visualization lab to leverage their resources and expertise on the subject matter. The methodologies applied were split into two: hands-on experiments and systematic literature study.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SuperComputing '17, November 2017, Denver, Colorado USA

© 2017 Copyright held by the owner/author(s).

ACM ISBN 123-4567-24-567/08/06...\$15.00

https://doi.org/10.475/123_4

2 PROBLEM STATEMENT

Understanding how computer components affect visualization, focusing on GPUs because they are vital to the visualization process.

3 PURPOSE OF STUDY

ParaView loads datasets using the CPU of the local machine and resorts to the GPU when processing data interactivity such as filter applications. The aim of the research is to study different visualization factors and their relationship to performance such as resolution and compute time. In other words, understanding resource usage on different GPUs and the Palmetto Cluster.

3.1 Research Questions

The following questions will be addressed through this research: What technology is currently available? What are the roadmap, needs, and challenges of cloud visualization? Are there any software or hardware compatibility issues? What is the difference between running on a CPU vs on a GPU?

- What technology is currently available?
For this research, we focused on the two Nvidia GPUs that were already installed on the Linux machines in the visualization lab. Their features can be seen in table 2 below. Both machines that were used for this study had x64 ubuntu 16.04 LTS installed on them. However, in addition to their different GPUs, they had different memory, disk, and processor capabilities (table 1 shows both machines' setup side by side).
- What are the roadmap, needs, and challenges of cloud visualization?
Figure 1 depicts the applied roadmap for cloud visualization. The connectivity between the machines and the Palmetto Cluster which served as the cloud infrastructure in our study is illustrated for a more comprehensive approach. One of the issues encountered with the cluster during the study was the inability to successfully load larger datasets in GB due to a system upgrade that was completed just before the study commenced. The largest file that was successfully loaded and was used for the test was 11.9MB.
- Are there any software or hardware compatibility issues?
Due to the limitations with the Palmetto Cluster, a switch was made midway through the study to an IBM Power8 but the Power8 did not possess drivers that support OpenGL, thus, testing was moved back to Palmetto. According to an IBM representative, a driver that supports OpenGL is expected to be released by the end of July 2017.

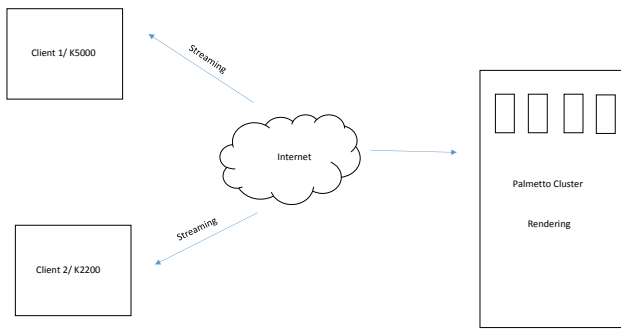


Figure 1: Client/Server Connectivity Setup

- What is the difference between running on a CPU vs on a GPU?

ParaView is a resource-intensive visualization tool, thus it is favorable to run it on a device that has a GPU since unlike CPUs, GPUs perform better with such programs. For instance, Paraview was run on a Windows 10 laptop with a 4GB RAM and an AMD E2-1800 processor (a CPU). Not only did it slow down the PC to a crawl but it also crashed when a filter was applied.

The research hypotheses are:

- An increase in dataset size which equals an increase in resource usage will lead to an increase in GPU memory usage and compute time.
- Understanding how GPU resource is used will help in determining what GPU can be used to reduce compute time.

4 METHODOLOGY

As previously stated, the methodologies for this research will be split into two categories: measurement experiments and literature review.

4.1 Measurement Experiments

This portion of the research will focus on how ParaView can be used to setup a visualization environment and streaming. Due to the demanding nature of media, the initial plan was to take measurements to establish a memory, speed, and load time benchmark. However, due to challenges faced and the limited time for the study, the test paid the most attention to memory use. Two Linux Desktops and the Palmetto Cluster were used for testing purposes. One of the Linux PCs was equipped with an Nvidia K2200 GPU while the other had an Nvidia k5000. The Palmetto cluster took care of the heavy lifting such as the computation and then streamed the rendered images to both clients. Measurements were taken on all devices to have a complete overview of the resources being affected on all ends.

Table 1 shows both Clients' resources side by side.

4.2 Literature Review

Prior to the project commencement and through the course of the project, pre-existing literary content were reviewed for better concept understanding and for discovering the best technology to

be implemented for a successful outcome. One of such papers that were reviewed was by Cherkasova and Staley [2] and it detailed their streaming benchmark study for a media sever in a utility data center. This paper helped with picturing one of the end goals because their setup is similar to future proceedings of this project as the lab hopes to be able to increase collaboration by streaming multiple instances of interactive ParaView sessions in real-time.

5 DATA COLLECTION AND ANALYSIS

A Pre-collected dataset from the protein data bank [1] was used in the experiments, ParaView was used to visualize the dataset, and pvsrver to stream from the server to the client. The mindset at the beginning of this research was to find at least one application that will be used in measuring the capacities of the GPUs on the client and the server nodes while ParaView is being run and interacted with. However, it was soon discovered that there was no readily available application-specific benchmark software. Majority of what was found were geared towards holistic system benchmark tests as against specific processes. An example, is the Heaven Benchmark tool [9]. The papers that addressed our approach, stated the use of proprietary software such as the Yahoo YCSB [3] [10] to successfully measure process-specific resource usage. Based on this discovery, the Nvidia monitoring tool[4] in the Linux command line was used to extract the required information. "By default, the monitoring data for each process includes the pid, command name and average utilization values for SM, Memory, Encoder and Decoder since the last monitoring cycle." [4]¹

It should be noted that the tests were not run simultaneously on both machines, however, the measurements were. That is, the tests were completed on one client before moving to the other, but the Nvidia monitoring tool was run on a client and the server simultaneously during the streaming portion of the study. The tests involved applying different filters, mapped variables, and representations to a protein dataset in ParaView (Figure 2 shows the visualization pipeline in ParaView)[6].

- Data Analysis: data is prepared for visualization.
- Filtering: filters are used to select what portion of the data would be visualized.
- Mapping: geometric primitives (such as points, lines) and their attributes (e.g., color, position, size) are applied to the data.
- Rendering: geometric data becomes image data.

Readings were taken every second for each change (started at the application of a filter, mapped variable, or representation, and stopped at the end of each application) and again as a continued test without breaks. To ensure only the resources used by ParaView were used in interpreting findings, the GPU to be measured was specified in the Nvidia command and the Linux grep command was used to extract only the ParaView process readings. These precautions were applied on both the clients and the server.

6 INITIAL FINDINGS

- Application-specific benchmarking software is currently unavailable for public use

¹SM stands for streaming multiprocessor which contains the GPU cores. sm% shows how much of the GPU's multiprocessors are being used at a time.

Feature	K2200 Client	K5000 Client
Operating System	x64 Ubuntu 16.04 LTS	x64 Ubuntu 16.04 LTS
Memory	62.8 GiB	15.6 GiB
Processor	Intel Core i7-5960x CPU @ 3.00GHz x 16	Intel Xeon(R) CPU E5-2603 @ 1.80GHz x 4
Graphics	Quadro K2200/PCIe/SSE2	Quadro K5000/PCIe/SSE2 1
Disk	168.6 GB	44.9 GB

Table 1: Client Resources

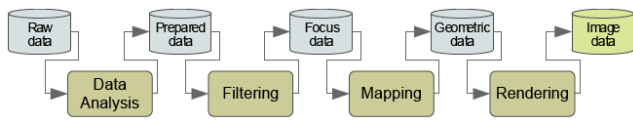


Figure 2: ParaView Visualization Pipeline
[6]

Feature	K2200	K5000
GPU Memory	4GB	4GB
Memory Interface	128 bit	256 bit
Memory Bandwidth	80 GB/s	173 GB/s
Nvidia CUDA Cores	640	1536 parallel
Diver Version	375.66	370.28

Table 2: GPU Specifications

- Different ParaView versions offer different filter values
- Specific filter (Point Gaussian) causes latency in ParaView
- Cannot load large files on client with K5000 and server (smallest GB dataset tried was 3.5GB)
- K2200 client can load large files and apply filters but with noticeable lag time
- Palmetto reduced the latency when applying changes but had no effect on the sm% and mem% use

Table 2 shows the GPU specifications side by side.[7] [8]

7 TEST PROCESS

The 5x8p-pdb-bundle (11.9MB) from the RCSB Protein Data Bank was used for the measurement experiments.

Mapped Variables and Representations applied include:

- Mapped Variables: rgb colors, Chain, atom types
- Representations: 3D Glyphs, Point Gaussian, and Surface
- Zoomed in and out (slower with point gaussian)

Filters applied were: Slice, Clip, Contour, Group Datasets, and Stream Tracer.

8 LIMITATIONS

Some limitations experienced by the study include: inability of the client with the K5000 GPU and server to load datasets larger than that used for the study, and an 8-week time constraint. Also, the default open source driver used by Ubuntu for Nvidia GPUs, Nouveau, lacks adequate support for 3D acceleration in newer Nvidia cards.[5]



Figure 3: Continuous Test on K5000 with no Breaks

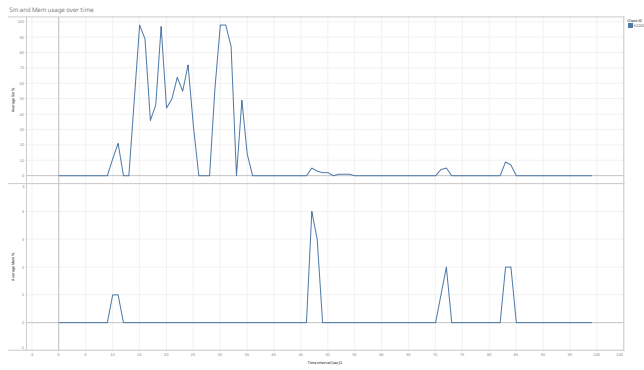


Figure 4: Continuous Test on K2200 with no Breaks

9 RESULTS

In all but one of the non-continuous tests (contour filter + zoom in and out), the K5000 attained a higher sm% than the K2200. Similarly, except for the Point Gaussian representation and the contour filter, the K2200 recorded less mem%. One observation that stood out throughout the research period was that the K2200 was faster in processing the changes than the K5000 was. This discovery was unexpected due to the more favorable features of the K5000 over the K2200. Figure 3 shows the sm% and mem% reading on the K5000 GPUs for the test without breaks. Figure 4 shows the sm% and mem% reading on the K2200 GPUs for the test without breaks.

10 CONCLUSION

Based on the results from the study, it can be inferred that a GPU's memory and core capabilities should not be solely relied on for determining its ability to implement visualization of large datasets. Administrators should also factor in available driver support. Future work include: using Nvidia drivers (instead of the default open source Nouveau drivers used by Ubuntu) to test for improved performance, and improving the real-time collaboration on, and the availability of streamed media (which are usually resource intensive) stored in the cloud on mobile devices and eventually virtual reality gears.

ACKNOWLEDGEMENT

I would like to thank Dr. Oyewole Oyekoya, Director of Visualization, Advanced Visualization Division, Clemson University, for this REU opportunity and the support and guidance offered throughout the program; Dr. Kuangching Wang, Holcombe Department of Electrical and Computer Engineering, Clemson University, for the opportunity to work on this particular project; and Mehrdad Yousefi, Graduate Research Assistant, Advanced Visualization Division, Clemson University, for sharing his ParaView and Palmetto expertise. I would also like to acknowledge Clemson University's Computing and Information Technology Department for their various workshops on the Palmetto Cluster.

REFERENCES

- [1] T. Ahmed, S. Bhusan, and J. Shi. 2017. Unique localization of the plastid specific ribosomal proteins in the chloroplast ribosome small subunit provides mechanistic insights into the chloroplastic translation. <https://www.rcsb.org/pdb/explore/explore.do?structureId=5X8P>. (2017). <https://doi.org/10.2210/pdb5x8p/pdb> [Online; accessed June-2017].
- [2] Ludmila Cherkasova and Loren Staley. 2002. Measuring the capacity of a streaming media server in a utility data center environment. In *Proceedings of the tenth ACM international conference on Multimedia*. ACM, 299–302.
- [3] Brian F Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, and Russell Sears. 2010. Benchmarking cloud serving systems with YCSB. In *Proceedings of the 1st ACM symposium on Cloud computing*. ACM, 143–154.
- [4] NVIDIA Corporation. 2016. nvidia-smi.txt. <http://developer.download.nvidia.com/compute/DCGM/docs/nvidia-smi-367.38.pdf>. (2016). [Online; accessed June-2017].
- [5] Help.ubuntu.com. 2017. BinaryDriverHowto/Nvidia. <https://help.ubuntu.com/community/BinaryDriverHowto/Nvidia>. (2017). [Online; accessed July-2017].
- [6] InfoVis:Wiki. 2007. Visualization Pipeline. http://www.infovis-wiki.net/index.php/Visualization_Pipeline. (2007). [Online; accessed 21-July-2017].
- [7] NVIDIA. 2012. PROFESSIONAL GRAPHICS PERFECTED NVIDIA QUADRO K5000. http://www.nvidia.com/docs/IO/140231/NV_DS_QUADRO_K5000_11_05_NV_US_LR.pdf. (2012). [Online; accessed June-2017].
- [8] NVIDIA. 2016. ACCELERATE YOUR CREATIVITY NVIDIA QUADRO K2200. http://images.nvidia.com/content/pdf/quadro/data-sheets/75509_DS_NV_Quadro_K2200_US_NV_HR.pdf. (2016). [Online; accessed June-2017].
- [9] Unigine.com. 2009. Heaven. <https://benchmark.unigine.com/heaven>. (2009). [Online; accessed June-2017].
- [10] Yahoo. 2010. Yahoo! Cloud Serving Benchmark. <https://research.yahoo.com/news/yahoo-cloud-serving-benchmark/>. (2010). [Online; accessed June-2017].