



NVIDIA's New Data Science Workstation— a Review and Benchmark

By George Seif

Data science is hot. | The past several years have seen a massive surge in interest for Data Science, so much so that many companies are reorienting their business strategy and branding themselves as “data driven.”

There's a good reason for this and it's no secret: more data gives us bigger opportunities to extract insights and create bigger business value.

Big datasets require big computing power. You'll need a good CPU, ideally with many cores like a Xeon type processor. Lots of RAM is a definite requirement— it's not uncommon to see Data Scientists purchasing 64 or even 128GB of RAM, and sometimes that's not even enough!

On top of that, after becoming very popular in Deep Learning, GPUs are now making their way into Data Science. GPUs can now also be used to accelerate the processing and visualization of large dataframes (similar to Pandas). The always popular [XGBoost](#) library comes with its own built-in GPU acceleration.

NVIDIA has recently released their [Data Science Workstation](#), a PC that puts together all the Data Science hardware and software into one nice package. The workstation is a total powerhouse machine, packed with all the computing power—and software—that's great for plowing through data.

I recently had the chance to test and review the Data Science Workstation, which we'll go through in this article.

The Specs

The workstation I tested was built by [BOXX](#). It comes with all the hardware pre-built and the software pre-installed, plus some extra cabling just in case. Having the machine pre-built was great since it's something that might take a good few hours. Here are the main hardware specs:

CPU | 2 Intel Xeon SP Gold 5217's, 8 core / 16 thread each @ 3.0Ghz

RAM | 192GB DDR4-2933 MHz ECC

Storage | 1.0TB SSD M.2 PCIe Drive

GPU | 2 NVIDIA Quadro RTX 8000s, 48GB VRAM each

Plus a mouse, keyboard, and 3 year warranty.

Building a custom machine is always the way to go when it comes to getting the most bang for your buck. The parts that came with this machine are picked just right: 32 total threads coming from the CPUs for parallel processing, large amounts of high-speed RAM, a PCIe SSD (much faster than a standard SATA drive, [example](#)), and 2 GPUs with 48GB of VRAM each.



The Data Science Workstation from [BOXX](#). Although it's designed for Data Science, it still comes with all the standard ports including USB, audio, mic, and a disk drive.

Then comes the often underrated part: the software. The workstation comes pre-loaded with Ubuntu 18.04 and literally every Data Science and Machine Learning library and software you could ever want installed from source. Just to name a few:

Data and ML | pandas, numpy, scipy, numba, matplotlib, xgboost, dask, scikit-learn, h5py, cython

Deep Learning | TensorFlow, Keras, PyTorch, NLTK

NVIDIA | GPU Drivers, CUDA 10, cuDNN, [RAPIDS](#)

All of those Python libraries are packaged in a [Python virtual environment](#) to avoid any future conflicts.

Installing TensorFlow from source can be a major challenge. Getting the GPU drivers and CUDA to work with any of the above libraries can be even tougher. So having all the software pre-installed was a big relief to avoid the big hassle.

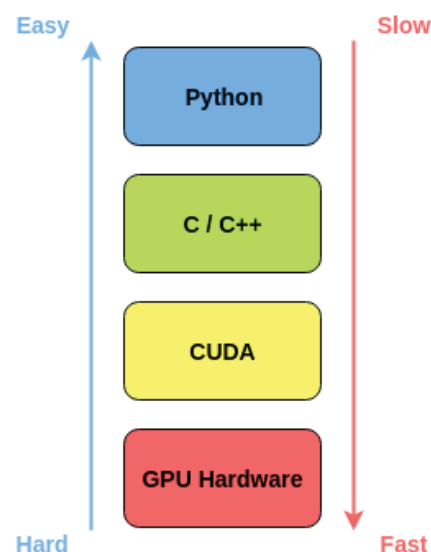
So what can all this awesome sounding hardware and software do? Let's find out.

XGBoost Benchmark

XGBoost is an open source library providing a high-performance implementation of gradient boosted decision trees. An underlying C++ codebase combined with a Python interface sitting on top makes it super fast and easy to use. It's the go-to library for Data Scientists, especially in Kaggle competitions.

We're going to run a benchmark test to see how the work station performs when processing a very large dataset using XGBoost. We'll be using a [demo notebook](#) from [Rapids.ai](#) to run the test.

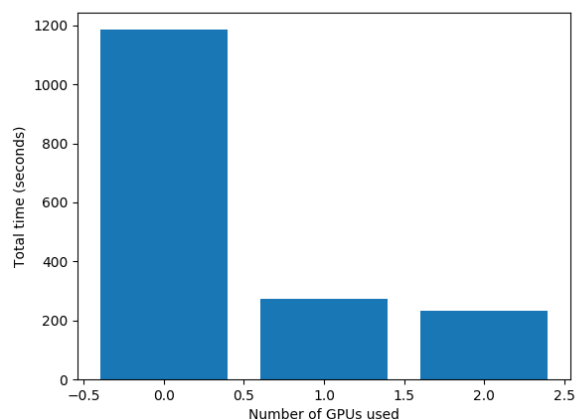
Rapids is a suite of software libraries designed for accelerating Data Science by leveraging GPUs. It uses low-level CUDA code for fast, GPU-optimized implementations algorithms while still having an easy to use Python layer on top.



The notebook generates a dataframe using NumPy's `random.rand()` function. To really push the limit, I set the dataframe size to be 1,000,000 rows by 5000 columns of float32 numbers; that's as big as I could make it in the notebook to use up all the memory. You might be able to squeeze in even more if you read the data in more efficiently, as the RAM usage on the machine seemed to settle down once the training commenced.

The notebook then runs the training for our XGBoost model on the data.

Num GPUs	GPU Memory Taken	Time for 30 Epochs
0	0	19min 45s
1	26GB	4min 32s
2	2x 13GB	3min 54s



Even with such a massive dataset, using CPU only, the 2 Xeon processors totaling to 16 cores / 32 threads handle it pretty well. As soon as the training starts up, all 32 threads fire up to 100% usage. We've also got plenty of RAM spaces left and could run another smaller training job if we wanted to.

Once we add in the GPUs, the speed of XGBoost seamlessly accelerates about 4.5X with a single GPU and 5X with 2 GPUs.

Deep Learning Benchmark

Deep Learning has its own firm place in Data Science. Almost all of the challenges in Computer Vision and Natural Language Processing are dominated by state-of-the-art deep networks.

One of the big reasons one might want to buy a machine with this kind of GPU power is for training Deep Learning models.

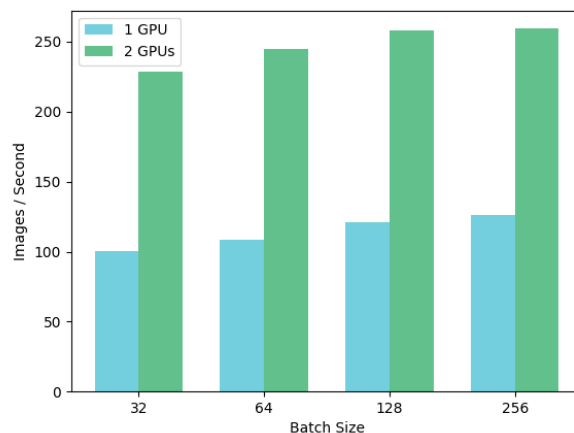
With big Deep Learning datasets, the more GPU memory you have the better. Those deep networks that are hundreds of layers deep need a good amount of memory space, especially if you want to increase the batch size to help speed up the training.

The high-end consumer GPUs, like the 2080 Ti and the 1080 Ti, come with 11GB of memory. They're very powerful cards, but 11GB is often not enough to fit a big neural network in memory. You could go with something more powerful like a V100 GPU on the cloud, but that'll come in at \$3.06 per hour on demand!

To test how much we can get out of these RTX 8000s, we'll use the official [tf_cnn_benchmarks](#) from TensorFlow. The repository contains scripts that run a training of standard image classification models on the ImageNet classification dataset. Training is run for a set number of iterations while the average speed is measured in images / second.

To truly test the workstation's capabilities, we'll run the benchmark with various batch sizes and number of GPUs. The results I got when training a [ResNet152](#) on this benchmark are shown in the table down below.

GPU	Total Batch Size	Images / Second
1x RTX 8000	32	100.23
1x RTX 8000	64	108.28
1x RTX 8000	128	120.99
1x RTX 8000	256	126.36
2x RTX 8000	32	228.80
2x RTX 8000	64	245.12
2x RTX 8000	128	257.93
2x RTX 8000	256	259.42



Even a single RTX 8000 can process more than 100 images / second, very fast for a ResNet152. We also have plenty of room to increase the batch size which further speeds things up, up to size 256.

Larger batch sizes have been shown to consistently [reduce the training time](#) since we are able to fully utilize our computing power. In some applications which require large amounts of memory, such as 3D CNNs for video processing, even a batch size of 1 can't fit on a standard consumer 11GB GPU! The 48GB in the RTX 8000 gives a ton of breathing room and enough space to play with the batch size and model size.

Final Points



All in all, the Data Science Workstation is a great machine. Really it's hard for it not to be with all the high-quality hardware.

The best part of it was really that everything "just works." All the libraries and software are updated to the latest versions and fully installed. It's almost surprising to power up a PC and find that you can run TensorFlow code that normally requires a source code installation without any trouble at all. That alone is a huge time, money, and frustration saver!

The hardware selection is done right. The flexibility that comes with having so much RAM and GPU memory is very convenient. No extra work or issues that normally come with running many GPUs. Getting more and more GPUs also has diminishing returns as you get up past 4 cards. If you do prefer a different hardware setup, you can customize your CPU, RAM, and GPU selections when ordering the machine to suit your needs.

Getting an in-house workstation can be a pretty big upfront investment—you're buying a lot of hardware in one shot. But the convenience of working locally with no cloud costs and having everything perfectly set up more than make up for the upfront price. It's a Data Science package that "just works."