DELLTechnologies

Dell Precision Data Science Workstation: Benchmarks and Best Practice for AI, R2.0

Authors: Steven Starrett, Raed Hijer, Brandon Kranz, Kyle Harper

NVIDIA
RTX

As the vast number-crunching enabled by artificial intelligence (AI) becomes a critical part of most organizations' business models, finding the right tools, technologies and techniques to do the job effectively becomes ever more vital. To demonstrate how selecting the right components can maximize the effectiveness of AI tasks, Dell has benchmarked various Dell Precision Data Science Workstation configurations against both deep learning and machine learning workflows.

Dell's analysis clearly demonstrates the substantial benefits of GPU acceleration and includes all original data, so testing and validation of the findings is possible by third parties. One machine learning model training benchmark reveals that running on a CPU takes 6.4x longer than on a GPU configuration. While another deep learning benchmark shows up to 4.74x in speedup due to multi-GPU acceleration.

The second half of this paper uses these findings to provide a simple best practice guide for selecting the optimum components for any workflow.

Introduction: The AI landscape

Business has changed rapidly over the last decade, with the Internet of Things and its industrial counterpart producing vast quantities of data. Estimates from IDC suggest these will reach 175 zettabytes worldwide by 2025 – a tenfold increase on 2017.

 This in turn, has resulted in an increased need for processing power, with concurrent advancements in algorithms, open source software and specialized hardware accelerators, driving an explosive adoption of Artificial Intelligence (AI).

Machine learning and deep learning each have their own unique use cases and challenges. Machine learning is less sophisticated and is generally used on structured data – such as tabular data – processed using well-known algorithms like linear regression, logistic regression, naïve Bayes and XGBoost.

These may run on CPU-only or GPU-accelerated compute platforms, depending on the use case.

The field of AI is large and the AI subsets of machine learning and deep learning are considered the most state-of-the-art data-driven approaches to solving many of today's complex problems.

These techniques unlock the value hidden in huge quantities of data streaming in from sources as diverse as customer spending, YouTube videos and machines on the factory floor. Yet things are never simple and it takes technology, skill and the right equipment to deliver actionable insights from this raw material.

Deep learning, on the other hand, is one of the most effective techniques in the AI arsenal and works particularly well on unstructured data – such as images, video or speech. It is deeply sophisticated – built on artificial neural network approaches, inspired by the structure and activity of neurons within the human brain.

Complexity and the role of data scientists

The rise in these AI techniques and their differing challenges has led to a rush of companies hiring data scientists. These individuals, tasked with the job of ingesting and curating the data to uncover its concealed value, have a complex job on their hands.

This is a multi-disciplinary field – covering a variety of domains including Analytics and Machine Learning, amongst others. And the upshot is it requires powerful, validated yet flexible hardware bundled with ready-to-use software environments and tools. This is easier said than done, however, and throws up two main complexity challenges.
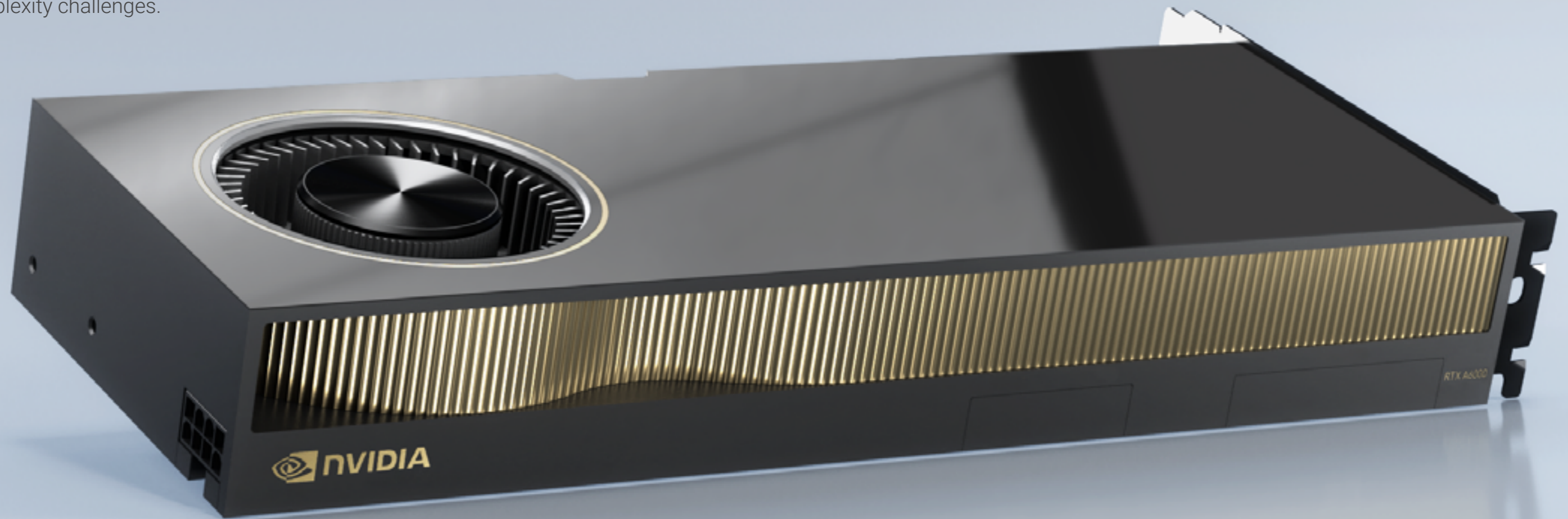
The first challenge is around integration.

Manually installing and integrating hardware and software components is cumbersome, tedious, and time consuming. On the one hand, hardware accelerators (e.g. NVIDIA RTX GPUs) are introducing new specialized enhancements with every generation. On the other, software tools are constantly being optimized to adopt the latest algorithms, frameworks and libraries. Putting these new items together, and insuring compatibility among all variables, is difficult and can easily turn data scientists into system administrators which is not the right use of their skills.

The second challenge is around configuration.

Deciding which GPU to use will in turn depend on which workload is used. For example, computer vision's requirements may differ from Natural Language Processing (NLP) tasks' requirements, which itself is different from classical machine learning's requirements. Different dataset types will also require divergent computational platform requirements. All this makes it tough to choose which platform and which GPU that will be best suited for different use cases.

Our benchmarking processes

This paper examines three use cases: deep learning – across two instances – and machine learning. The first deep learning examples looks at computer vision image classification and the second augments this with BERT Fine Tuning, while the machine learning example uses XGBoost (eXtreme Gradient Boosting) in a classical model.

In all cases, several Dell Precision Data Science Workstations with various GPU configurations are considered, benchmarked and presented, to provide some guidance on the expected performance speedups for model development. The details of all this are covered in the relevant sections below.

New technologies to counter modern challenges

The Dell Precision Data Science Workstation (DSW) is a purpose-built product line to address the challenge of integration. It curates a set of latest hardware powered by NVIDIA GPUs as well as the NVIDIA Data Science Software stack. The latter delivers the latest GPU-accelerated frameworks and libraries that have been validated at the Dell factory.

This means data scientists can now embark on an AI journey without having to worry about spending countless hours, or days, figuring out which hardware and software works together. However, it does still leave the challenge of configuration in the hands of IT departments, data scientists and their companies. Dell has put this benchmark together to help solve this problem.

Factors to consider when configuring a Dell Precision Data Science Workstation

To correctly size and configure a Dell Data Science Workstation based on a data scientist's needs takes three general steps. These are (1) determining the dataset size and best AI model, (2) matching the GPU and GPU memory to the dataset, and (3) deciding the right CPU and CPU memory configuration.

STEP ONE

Is sizing the dataset and choosing the AI model development approach. One size does not fit all in this industry, with approaches and models varying widely.

The dilemma is typically either:

- The system's memory is too small, so results are distorted or compromised.
- The system's memory is too large and expensive, or cannot physically fit in one system.

It is possible to break the dataset into smaller pieces, called minibatches, to fit into the system's available memory.

STEP TWO

Revolves around matching the best GPU and GPU memory size for that dataset. When GPU utilization approaches 100% (as viewed via the "nvidia-smi" command line utility), it is an indication that a larger memory GPU or a dual/triple GPU configuration is required.

STEP THREE

Requires deciding on the optimal CPU and CPU memory configuration for the CPU side to keep the GPU side of the system fed. And finally, selecting the storage configuration and sizing.

Let's use a BERT (Bidirectional Encoder Representations) benchmark as an example. We have noticed during our BERT benchmarks that the RTX 6000 configurations outperform the RTX 8000 configurations when using lower sequence lengths and batch sizes.

However, once the sequence length and the batch size increase to a certain threshold, the RTX 6000 card does not have enough memory to perform the task.
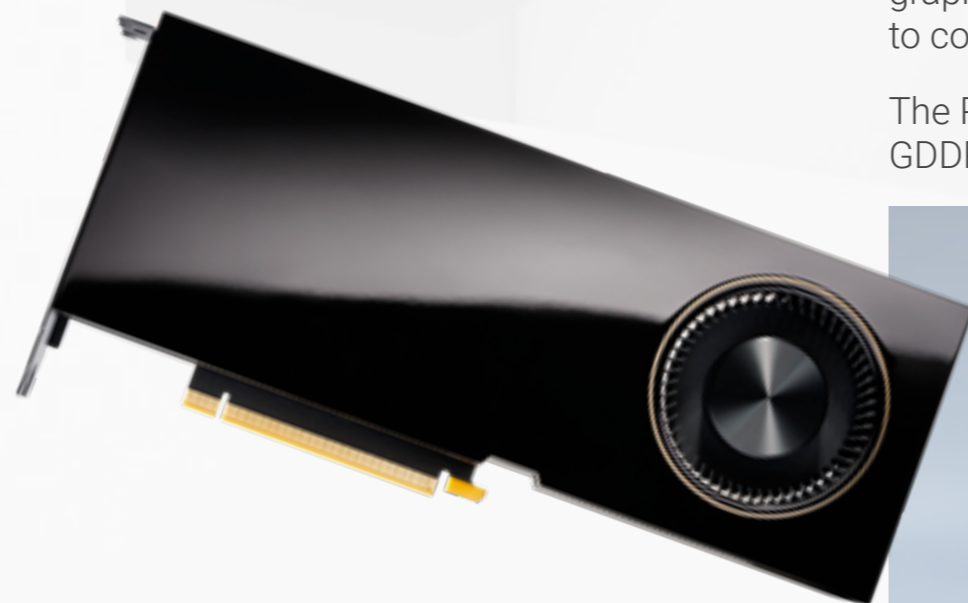
The data set

Depending on the size requirements of the dataset, it may fit entirely in the memory being used while training the model. In a CPU-only system, the dataset is placed in the CPU DDR memory and the model is trained using the CPU(s).

In a system like the DSW containing GPUs, the model is trained using the GPU resources, and the dataset resides in the GPU memory instead. Many models and datasets can fit in the memories of modern GPU cards.

In some cases when the dataset size exceeds single GPU memory, tricks like using distributed multi-GPU libraries (e.g., DASK) can be used to load the dataset across the GPUs.

## Select and size the GPU that best accelerates AI modelling

Note: refer to 'Table 2: GPUs used for benchmarking' for the memory buffer in each of the NVIDIA GPUs available at the time of publication.

The RTX A6000 has 48GB of GDDR6 memory capable of 768 GB/s, 10752 Ampere-class CUDA cores, 84 Ampere-class RT Cores, and 336 Ampere-class Tensor Cores. Combining two RTX A6000 GPUs together using NVLink doubles the number of cores and memory, combining the two cards GPU memory into 96GB of unified memory available for the dataset.

The RTX 8000 has 48GB of GDDR6 memory capable of 672 GB/s throughput, 4608 CUDA cores, 72 RT Cores, and 576 Tensor Cores. Combining two RTX 8000 GPUs together using NVLink doubles the number of cores and memory, combining the two cards GPU memory into 96GB of unified memory available for the dataset.

The RTX 6000 GPU has the same number of cores as the RTX 8000, 24GB of graphics GDDR6 memory capable of 672 GB/s throughput, and can also use NVLink to connect to a second RTX 6000 for a combined 48GB of unified memory.

The RTX 5000 has 3072 CUDA cores, 48 RT Cores, 384 Tensor Cores and 16GB of GDDR6 memory capable of 448 GB/s throughput.
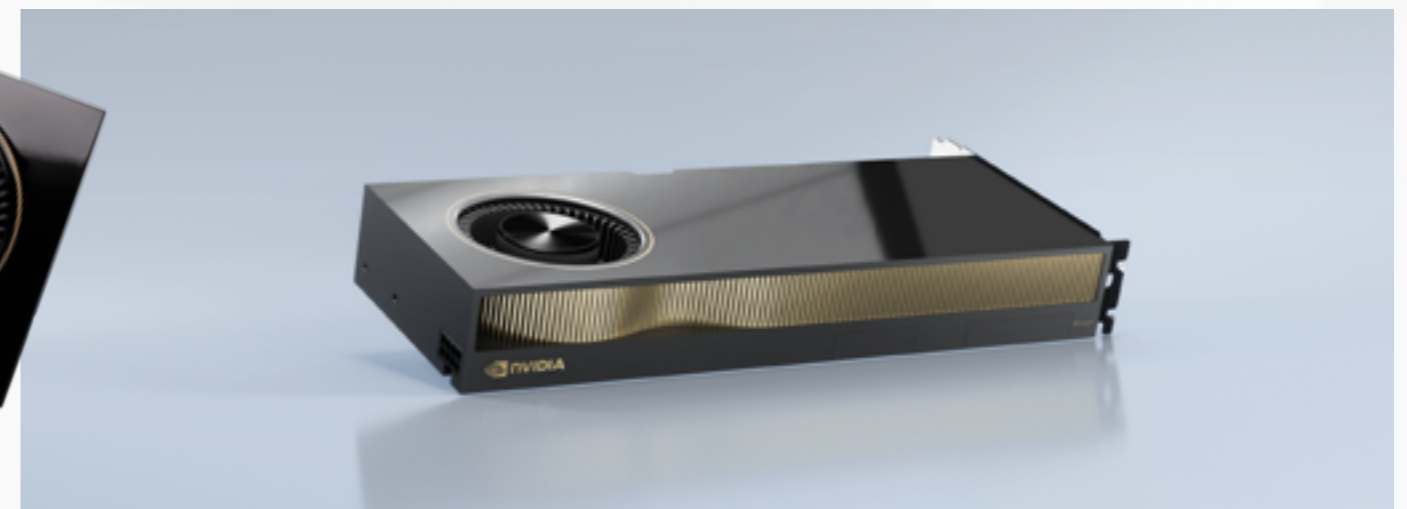
Why were the different setups and workloads used for this benchmark?

Dell chose and ran benchmarks to report on the performance differences across multiple Precision DSW platforms and configurations. The platforms span from the Dell mobile DSWs, the 15" 7550 and 17" 7750 mobile, through the Dell tower DSWs, the Dell 5820 and 7920 Towers, up to the Dell 7920 Rack DSW.

Within these platforms, we selected various typical CPU memory and GPU configurations, reflecting a spectrum of compute resources suitable for AI model training workloads. The NVIDIA GPUs benchmarked were the Quadro RTX 5000 mobile GPU in the mobile DSWs, and the Quadro RTX 6000 and RTX 8000 GPU in the tower and rack DSWs.

All of these platforms and configurations were loaded with identical Ubuntu Linux OS and NVIDIA Data Science Software stack, the software bundle that is validated and shipped as part of every DSW from Dell.

The NVIDIA Data Science Software stack includes the factory-optimised Tensorflow, Python, XGBoost, and Jupyter Notebook software that goes to every user, and being benchmarked along with the underlying hardware. Comparison of the different platforms, configurations and GPU details are shown in the tables on this page:

Model	Precision 7550 Mobile	Precision 7750 Mobile	Precision 5820 Tower	Precision 5820 Tower	Precision 7920 Tower	Precision 7920 Rack
Processor	Intel® Xeon W-10885M (8 Core, 16MB Cache, 2.40 GHz to 5.30 GHz, 45W, vPro)	Intel® Xeon W-10885M (8 Core, 16MB Cache, 2.40 GHz to 5.30 GHz, 45W, vPro)	Intel® Xeon Processor W-2175 (14Core, 19.25M, 2.5 GHz to 4.3 GHz, 140W, vPro)	Intel® Xeon Processor W-2245 (8 Core, 11MB, 3.9 GHz to 4.7GHz, 155W, vPro)	Dual Intel® Xeon Processor Gold 6134 (8 Core, 24.75MB, 3.2GHz to 3.7, 130W, vPro)	Dual Intel® Xeon Gold 6244 (8 Core, 24.75MB, 3.6GHz to 4.4GHz, 150W, vPro)
Graphics Card	NVIDIA® Quadro RTX 5000 w/16GB GDDR6	NVIDIA® Quadro RTX 5000 w/16GB GDDR6	Single/Dual NVIDIA® Quadro RTX 6000 24GB GDDR6	Single NVIDIA® Quadro RTX 8000 48GB GDDR6	Dual/Triple NVIDIA® Quadro RTX 6000 24GB GDDR6	Dual/Triple NVIDIA® Quadro RTX 6000 24GB GDDR6
Memory	64GB, (4X16GB) DDR4 2933Mhz Non- ECC	64GB, (4X16GB) DDR4 2933Mhz Non- ECC	128GB (8x16GB) DDR4 2666MHz RDIMM ECC	256GB (4x64GB) DDR4 2933MHz RDIMM ECC	128GB (8x16GB) DDR4 2666MHz RDIMM ECC	128GB (8x16GB) DDR4 2666MHz RDIMM ECC
Storage	2x 1TB M.2 NVMe PCIe SSD Class 40	2x 2TB M.2 NVMe PCIe SSD Class 50	1x 1TB M.2 NVMe PCIe SSD Class 40	1x 1TB M.2 NVMe PCIe SSD Class 50	1x 1TB M.2 NVMe PCIe SSD Class 40	1x 2.5" 1TB SATA Class 20 SSD + 1x 2.5" 512GB SATA Class 20 SSD
Operating System	Ubuntu Linux 18.04.5 LTS	Ubuntu Linux 18.04.5 LTS	Ubuntu Linux 18.04.5 LTS	Ubuntu Linux 18.04.5 LTS	Ubuntu Linux 18.04.5 LTS	Ubuntu Linux 18.04.5 LTS
TensorFlow	1.14	1.14	1.14	1.14	1.14	1.14
Python	3.7.6	3.7.6	3.7.6	3.7.6	3.7.6	3.7.6
XGBoost	1.1.0	1.1.0	1.1.0	1.1.0	1.1.0	1.1.0
Jupyter Notebook	6.0.3	6.0.3	6.0.3	6.0.3	6.0.3	6.0.3

Table 1: DSW configurations tested

	Desktop GPUs			Mobile GPUs
	RTX A6000	RTX 8000	RTX 6000	RTX 5000
CUDA Cores	10752 Ampere	4608	4608	3072
RT Cores	84 Ampere	72	72	48
Tensor Cores	336 Ampere*	576	576	384
Memory	48GB GDDR6 768 GB/s	48GB GDDR6 Up to 672GB/s	24GB GDDR6 Up to 672GB/s	16GB GDDR6 Up to 448GB/s

Table 2: GPUs used for benchmarking

See *Appendix* for all RTX A6000 GPU benchmark findings

Deep Learning: Image Classification

The first benchmarking was done using tf_cnn_benchmark from TensorFlow. It is based on Convolutional Neural Network (CNN) which is considered the foundation of computer vision tasks like image classifications, object detection, image segmentations, Generative Adversarial Network (GAN) and others. Residual networks (ResNets), one of many CNN topologies, is comprised of residual blocks.

Each block has two branches, one feeding input directly to the output and the other performing two to three convolutions. The two branches are added and fed to the next block. This approach yields high performance CNN as we stack layers together while avoiding the issue of vanishing gradients. As such, ResNet50 (comprised of 50 layers) is considered as the standard for image classification benchmarking.

We used tf_cnn_benchmark with ResNet50 as the topology for CNN. The tf_cnn_benchmark repository contains scripts that run training and inferencing of standard image classification models on synthetic images as well as other public datasets such as ImageNet.

We used the default dataset in this case, which is a synthetic dataset. This benchmark is also capable of running on single or multiple GPUs within one workstation node or across multiple workstation nodes. For more details on tf_cnn_benchmark, please refer to GitHub.

Another common practice of deep learning practitioners is the use of Automatic Mixed Precision (AMP). This entails using both single precision (FP32) and half precision (FP16) math to speed up the neural network training without the loss of accuracy.

The latest NVIDIA Quadro GPUs, starting with the Volta architecture and continuing in the newer GPUs, include specialised Tensor Cores to leverage AMP. In our case, we used FP16 when we ran this benchmark.

The last variable we used is the batch size (BS). Arguably, batch size and learning rate are the two most important hyper parameters when conducting different experiments once the neural network model is determined. Additionally, these two parameters are often tightly coupled. The flexibility of being able to run different batch sizes enables data scientists to explore more features and, in some cases, reduces the training time.

The tf_cnn_benchmark used here is for model training and it was run for a set number of iterations while the average speed is measured in images/sec. This example script is for a dual GPU, with a batch size of 32 and FP16:

```
python tf_cnn_benchmarks.py --num_gpus=2 --batch_size=32 --model=resnet50 --use_fp16=true
```

With this in mind, we experimented with different batch sizes across different GPUs, from the Quadro Mobile RTX 5000 16GB to single, dual and triple Quadro RTX 6000 24GB installed in Precision Performance 5820 and 7920 Towers. The last two are Precision performance towers boasting Intel Xeon Cascade Lake single and dual socket processor platforms respectively.
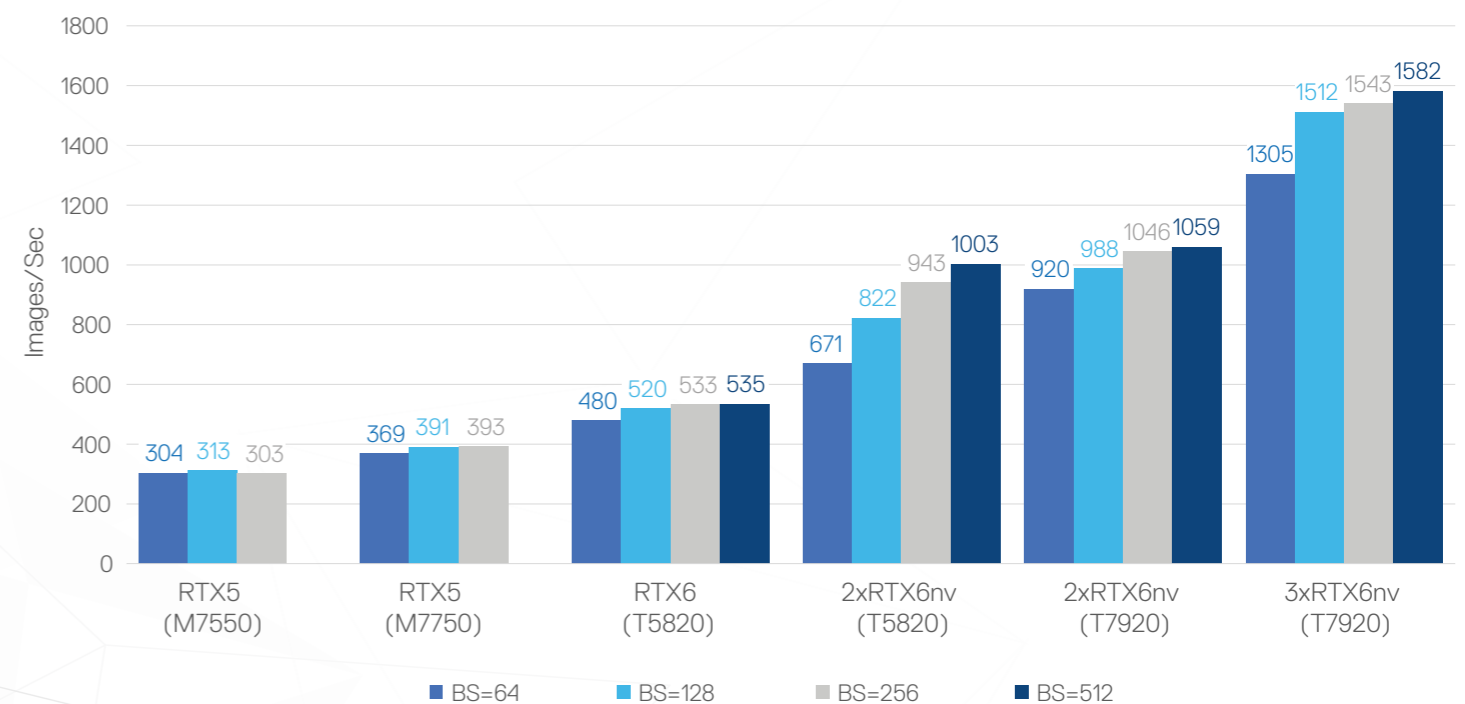
tf_cnn_benchmark, ResNet50 FP16

Figure 1

Deep Learning: Image Classification (Continued)

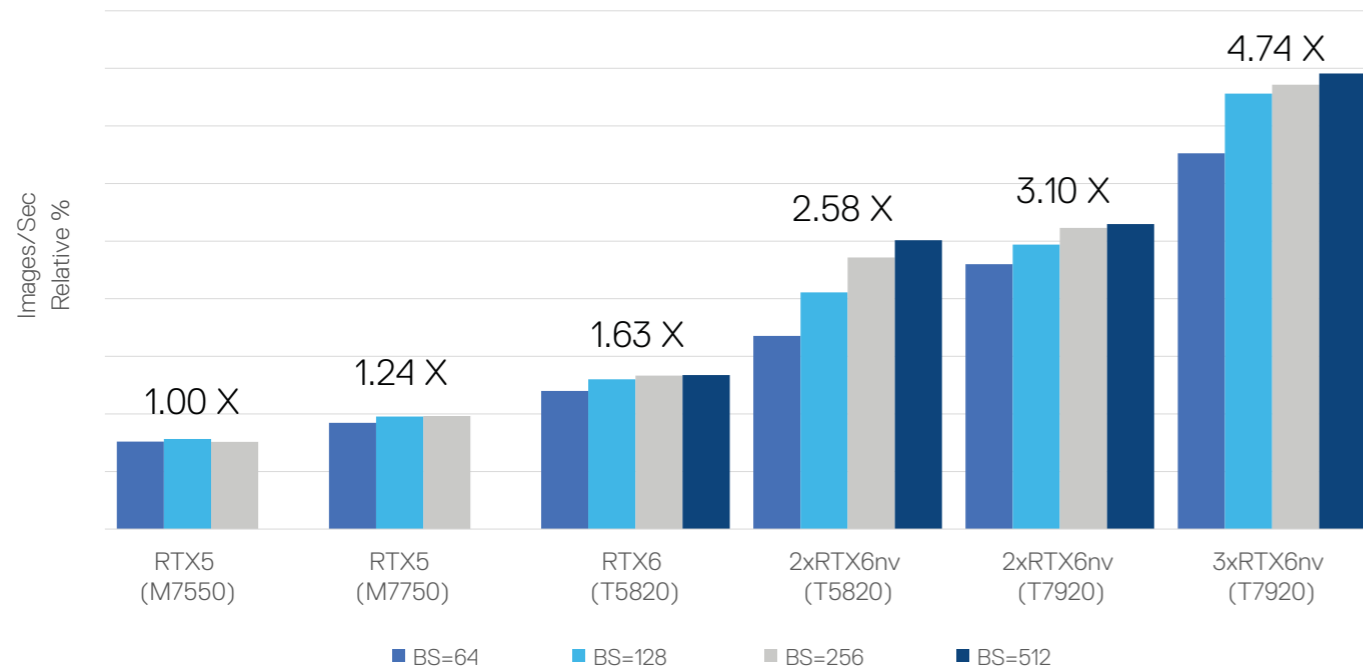**tf_cnn_benchmark, ResNet50 FP16**

Figure 2

The chart shows that both Precision 7550 and 7750 mobile workstations are more than capable of performing these tasks with respectable images/sec. It's worth noting that moving from the 15" 7550 mobile workstation to the 17" 7750 mobile workstation shows a performance improvement of around 24% even though both platforms use the same Quadro RTX 5000.

This is attributed to the larger chassis in the 7750 with better thermal and cooling solution that allows the RTX 5000 to run at higher clock speeds (i.e., higher utilization).

Moving to the 5820 Tower with a single RTX 6000, we saw a 63% improvement in throughput which is due to more CUDA cores, Tensor Cores and video memory.
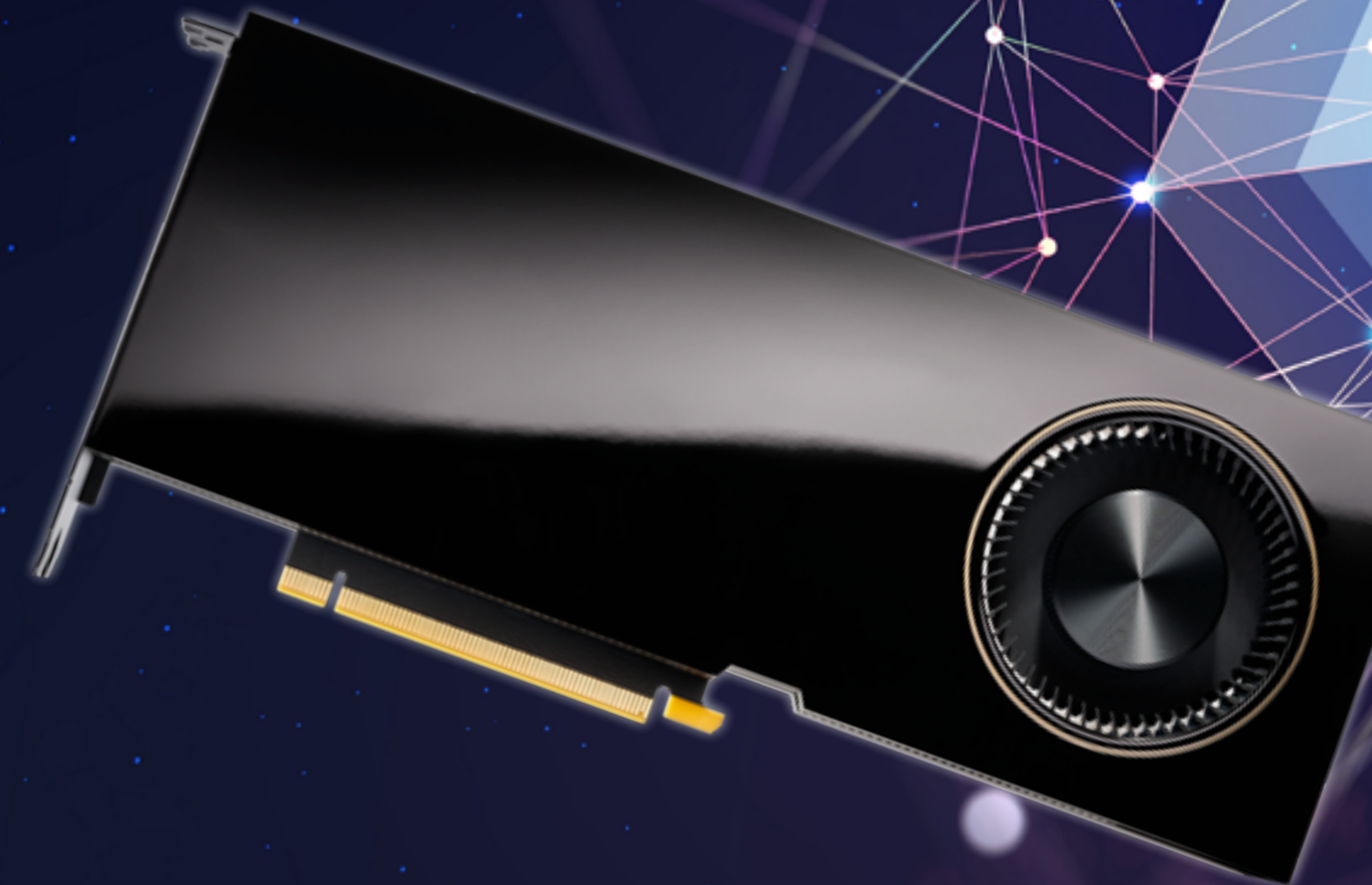
Dual RTX 6000 with NVLink almost doubles the throughput of Single RTX 6000, especially in the 7920 Tower platform (3.1X vs. 1.63X).

In addition to having the second GPU, the 7920 Tower also adds a second CPU multicore microprocessor, providing more cores and more memory channels which are necessary to prevent starving these two high performance GPUs.

Introducing the third GPU provides an extra boost of almost linear scaling. Using the nvidia-smi command, we were able to confirm that when running these benchmarks and others in areas of NLP, all GPUs (dual and triple) are being kept fully utilized. In other words, the GPUs were kept fully fed with data by the CPU, CPU-memory, and storage systems. No performance or throughput is being left on the table unused. No system architectural bottlenecks.

It was also clear that not only do the towers' Quadro RTX GPUs outperform their mobile counterparts, they also allowed for larger batch sizes; e.g., BS=512. This is directly attributed not only to the larger video memory footprint but also to more CUDA cores and Tensor Cores in the RTX 6000 as shown earlier.

Dual GPU with NVLink also provided higher throughput performance, due to direct GPU-GPU communication, avoiding the comparatively slow PCIe bus.

Deep Learning: Natural Language Processing

The second benchmarking was done using BERT Fine Tuning NLP benchmark from TensorFlow. Natural Language Processing (NLP) leverages AI to create the ability for a computer to understand, analyse, manipulate and generate human language.

BERT (Bidirectional Encoder Representations from Transformers), is a powerful tool developed by Google in late 2018 that allows computers to process, analyze and 'understand' human language. It has become a standard in various NLP applications such as question answering, named entity recognition, natural language inference and text classification. Previously, all language models (i.e., Skip-gram and Continuous Bag of Words) were uni-directional. They can traverse over the word's context window from only left to right or right to left. BERT uses bi-directional language modeling to understand a word's context; i.e., the model learns the context of a word based on all its surrounding.

BERT Pre-Training

The pre-training procedure largely follows the existing literature on language model pre-training. For the pre-training corpus, BERT uses the BooksCorpus (800M words) and English Wikipedia (2,500M words). As you can imagine this would take several weeks using workstations, so we focused on BERT Fine Tuning benchmarks.

BERT Fine-Tuning

The pre-trained model is used as a base (Transfer Learning) with the same weights while adding few layers specific to the NLP task at hand. In our case the task was question and answering (Q&A). This is a commonly used approach for creating new NLP specific tasks and reducing the complexity of fine-tuning significantly.

Benchmark scripts that we used were finetune_train_benchmar.sh from the NVIDIA NGC Repository BERT for Tensorflow. Using the script, we were able to test on the SQuAD v1.1 dataset using fp16 or fp32.

Sequence lengths of 128, 384, and batch size 1, 2, 4, 8, 16, 32, and 64 were run during this script. In order to run all the batch sizes, we edited the file finetune_train_benchmark.sh on line 81 and added batch sizes 8, 16, 32, and 64 to the following line.

for batch_size in 1, 2, 4, 8, 16, 32, 64: do

Setup of BERT Fine Tuning Training provides options of Base or Large. BERT LARGE (L=24, H=1024, A=16, Total Parameters=340M) and BERT BASE (L=12, H=768, A=12, Total Parameters=110M). We chose BERT Large for our benchmarks and used the following command:
BERT# scripts/finetune_train_benchmark large true <num_gpus> squad.

Precision Workstations configurations

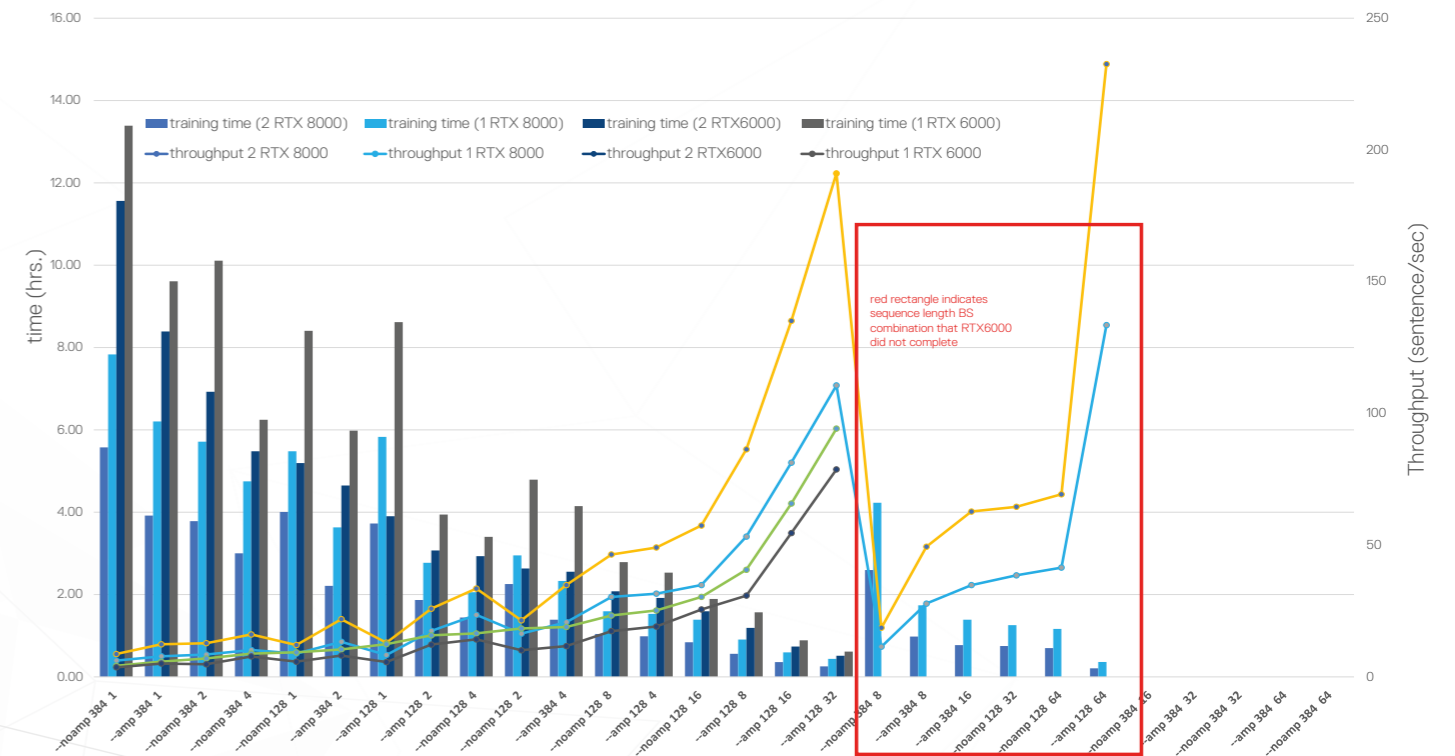| System	Precision Model	Processor	Frequency	Cores	Memory	GPU
Config 1	T5820	W-2245	3.9-4.7GHz	8 Core	256GB Mem	1-2 x RTX 8000
Config 2	T5820	W-2245	3.9-4.7GHz	8 Core	256GB Mem	1-2 x RTX 6000
Config 3	T7920	5217	3.0- 3.7GHz	8 Core	196GB Mem	3 x RTX 8000
Config 4	T7920	5217	3.0- 3.7GHz	8 Core	196GB Mem	3 x RTX 6000
Config 5	T5820	W-2175	2.5- 3.4GHz	14 Core	64GB Mem	2 x GV100

Table 3

Figure 3

Deep Learning: Natural Language Processing (Continued)

Different runs are displayed on the horizontal axis where:

--amp seq_len BS means FP16, Sequence Length & BS= Batch Size and

--nonamp seq_len BS means FP32, Sequence Length & BS= Batch Size

In Figure 3, ordered by throughput, the (2) Dual RTX 6000 (GREEN) outperformed the Single RTX 6000 (GREY) in training time and throughput (sent/sec). This was amplified as sequence length and batch size increase.

Example 1:

FP 16, Sequence Length 128, Batch Size 1.

1 x RTX 6000   Sentences per Second 8.82

2 x RTX 6000   Sentences per Seconds 13.04

Result: 2 RTX 6000 had 47% performance increase

Example 2:

FP 16, Sequence Length 384, Batch Size 2.

1 x RTX 6000   Sentences per Second 13.77

 2 x RTX 6000   Sentences per Seconds 22.08

Result: 2 RTX 6000's had 60% performance increase.

Notice Examples 1 and 2 below. We also noticed that RTX 6000 runs FP 32 128 32, but FP 16/32 64 did not run to completion. It was also observed that the RTX 6000 runs FP16 384 8 or higher batch size were incomplete as well. These are shown in red rectangle in the above chart. The RTX 8000's could complete all sequence lengths and batch sizes up to FP 32 384 8 (Example 3).

Example 3:

FP 16, Sequence Length 384, Batch Size 8.

 2 x RTX 6000   Sentence per Second - Incomplete

 2 x RTX 8000   Sentence per Second 49.38

Result: 2x RTX 6000 - Incomplete

Deep Learning: Natural Language Processing (Continued)

In table 4, there are several incomplete areas that are annotated by the blue shaded cells. The majority of these are from the RTX 6000 systems. Triple RTX 8000/RTX 6000s on the 7920s performed slower than Dual RTX 8000/RTX 6000 on some of the runs.

When the graphic cards are the same, the systems with better processor base frequency and more memory outperformed the systems with lower processor base frequency, and less memory.

Training performance	Seq Length	Batch Size	"Test 1 RTX8000 (ECC)"		"Test 2 RTX6000 (ECC)"		"Test 3 RTX6000"		"Test 4 RTX8000 (ECC)"			"Test 5 RTX6000 (ECC)"		"Test 6 GV100 (ECC)"	
			T5820/W-2245/256GB		T5820/W-2245/256GB		T5820/W-2245/256GB		T7920/5217/196GB					T5820/W-2175/64G	
			2 GPU	1 GPU	2 GPU	1 GPU	2 GPU	1 GPU	3 GPU	2 GPU	1 GPU	2 GPU	1GPU	2 GPU	1 GPU
FP16	128	1	13.07	8.66	13.04	8.82	13.11	9.64	9.01	12.97	8.29	12.39	5.61	13.25	8.14
FP32	128	1	12.41	10.27	12.52	10.9	12.5	11.35	5.88	12.07	8.82	9.31	5.75	13.44	10.15
FP16	128	2	26.26	17.05	26.08	17.11	26.18	18.89	18.03	25.89	17.45	15.74	12.26	26.63	16.44
FP32	128	2	21.98	17.08	22.13	16.96	22.19	18.5	11.26	21.43	16.38	18.36	10.09	22.46	16.44
FP16	128	4	50.15	31.59	49.82	31.77	50.12	34.89	35.39	49.08	31.58	25.19	19.1	49.94	30.15
FP32	128	4	34.86	24.34	35.31	24.59	35.39	26.03	20.74	33.51	23.51	16.48	14.21	35.21	23.19
FP16	128	8	88.23	54.53	88.01	54.43	88.32	59.75	66.76	86.37	53.26	40.64	30.82	88.46	50.43
FP32	128	8	49.44	31.27	50.7	31.76	50.66	33.2	35.51	46.42	30.33	23.27	17.35	48.52	30.26
FP16	128	16	138.19	83.45	138.88	84.25	139.25	91.14	116.87	135.11	81.31	65.77	54.57	136.72	80.44
FP32	128	16	61.23	35.95	64.18	36.74	63.09	37.82	54.7	57.44	34.84	30.33	25.56	58.26	34.5
FP16	128	32	196.17	112.5	198.96	114.98	198.81	124.1	188.34	191.11	110.61	94.22	78.72	194.62	111.38
FP32	128	32	71.09	39.01					73.84	64.54	38.53			67.3	39.91
FP16	128	64	246.55	136.15					265.85	232.6	133.5				
FP32	128	64	78.21	41.73					91.84	69.29	41.47				
FP16	384	1	12.53	7.98	12.52	7.89	12.54	8.66	8.96	12.34	7.79	5.76	5.03	12.49	7.53
FP32	384	1	9.12	6.47	9.2113	6.5	9.21	6.91	5.31	8.67	6.17	4.18	3.61	9.77	6.35
FP16	384	2	22.14	13.66	22.08	13.77	22.21	15.2	16.84	21.87	13.31	10.4	8.08	22.41	12.86
FP32	384	2	13.61	8.81	13.89	8.92	13.9	9.36	9.39	12.78	8.46	6.98	4.78	13.56	8.55
FP16	384	4	35.59	21.14	35.64	21.33	35.77	23.32	29.95	34.83	20.74	18.92	11.65	36.22	20.64
FP32	384	4	17.55	10.41	18.08	10.57	18.02	11.04	14.83	16.1	10.18	8.82	7.74	17.13	10.31
FP16	384	8	51.25	29.66					48.83	49.38	27.8			51.17	30.09
FP32	384	8	20.79	11.56					21.07	18.63	11.43			20.06	11.66
FP16	384	16	65.8	36.12					71.79	62.74	34.81				
FP32	384	16													
FP16	384	32													
FP32	384	32													
FP16	384	64													
FP32	384	64													

Table 4

Summary

Deciding on GPU for NLP tasks should be determined based on the importance of floating point, sequence length and batch sizes. The RTX 8000s will complete more floating point, sequence length and batch sizes. Although one RTX 8000 will complete many of the sequence lengths and batch sizes during Finetuning, using two RTX 8000 appears to be the performance solution.

Machine Learning: Classification

The third benchmarking was done using the XGBoost library. XGBoost (eXtreme Gradient Boosting) is the latest evolution of decision tree-based algorithms. It's built upon an ensemble of tree methods that apply the principle of boosting weak learners. For tabular and structured data, it's considered among the best-in-class techniques.

For this exercise, we used a synthetic numerical dataset of 6 Million rows X 501 Columns (one being the output), all are full precision (FP32). The actual Python Jupyter notebook was used from the demo notebook found in Rapids.ai. RAPIDS is NVIDIA's open source suite of GPU-accelerated machine learning libraries, which includes XGBoost.

Dell Data Science Workstation comes preconfigured with the most common data science libraries, including RAPIDS and XGBoost. So, no additional software installation is needed. Additionally, XGBoost is now GPU-accelerated.

The test was done comparing CPU-only vs. different GPUs as before, but this time including RTX 5000 Mobile, and RTX 6000, RTX 8000 and RTX GV100 Tower GPUs.

The test only considers the time to train the above-mentioned dataset using XGBoost with the same parameters.

We increased the num_round to 100, set the training-to-validation ratio to 90%-10%, and left the rest of the parameters at the default settings of the demo notebook.

The chart below shows that training time is reduced dramatically when using GPU vs. CPU-only. Simply by moving the training from a Xeon W-10885M CPU to the RTX 5000 GPU, it took 50% more time to run on the CPU compared to GPU.

Using higher 32GB HBM2 (RTX GV100 T5820), the speed increase was more substantial (6.4x).

This translates to reducing week-long training runs to perhaps one day. The RTX 6000, RTX 8000 and RTX GV100 benefit greatly not only from their large video memory allowing the entire dataset to fit in, but also from their larger amount of CUDA and Tensor Cores which directly impact the processing and convergence time.
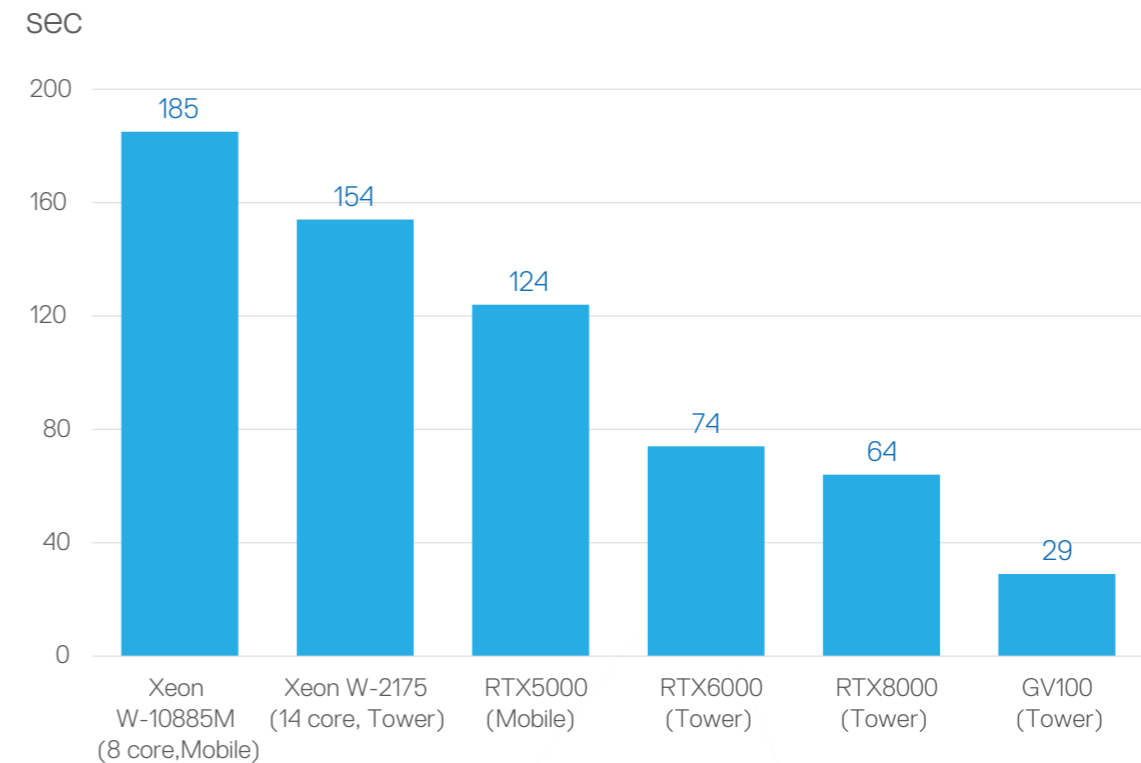
Time to train for 100 Epochs. (lower is better)

sec

Figure 4

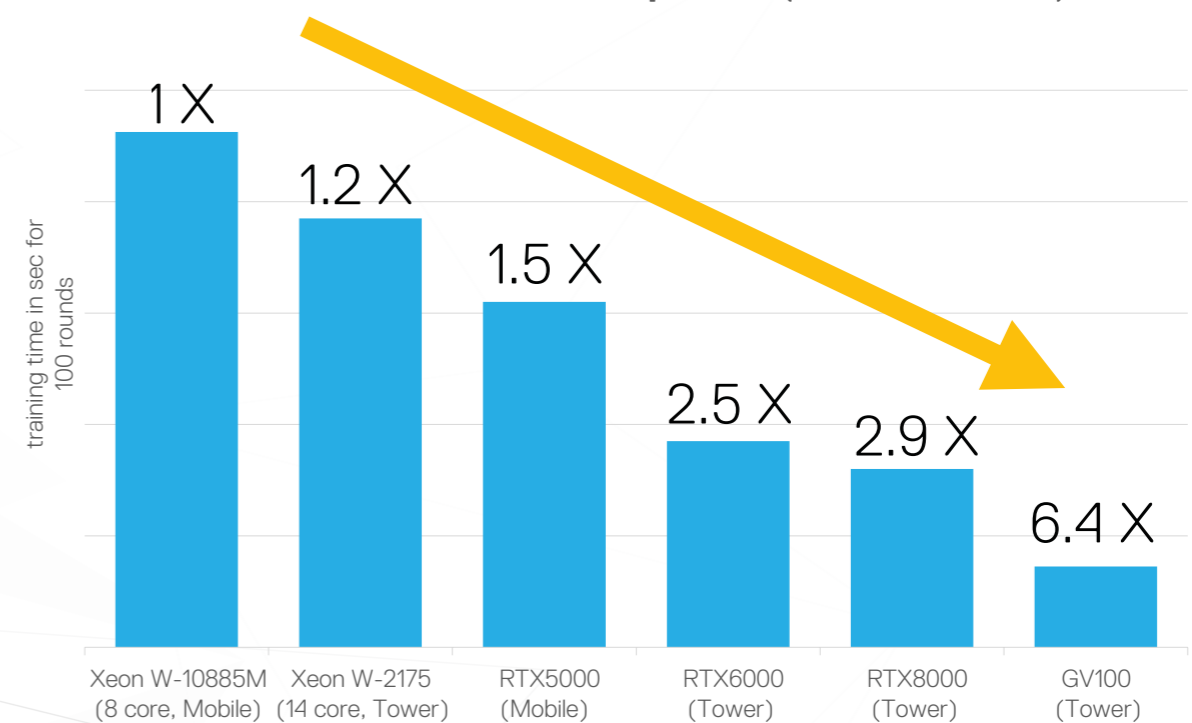**Time to train for 100 Epochs. (lower is better)**

Figure 5

How to right-size a DSW

There are three general steps to correctly size and configure GPU-accelerated workstations for use in AI model development:

1. Determine the type and memory size of the AI model to be used and assessing the size of the dataset that model will be operating on.

2. Size the GPU and GPU memory.

3. Size the CPU, CPU memory and mass storage.

STEP ONE

Determining the dataset size and AI model development approach

One size does not fit all in the AI and data science industry, as the datasets and models vary widely. Classic AI modelling approaches operate with relatively smaller sized data sets. However, many of the problems machine learning and deep learning models are selected to solve operate on very large and unstructured datasets.

They can be massive if the data being analyzed is image data or video. Regardless of whether the modeling approach to be used is classic AI, machine learning, or deep learning, the amount of memory ideally needed to house the complete data set and model far exceeds the amount of RAM

that can physically fit in the platform. Therefore, many situations require slicing the data set into smaller batch sizes. Most data scientists will choose to maximize batch size according to available physical memory.

In a GPU-accelerated AI workstation, the GPU and GPU memory system are housing and performing the model training, not the CPU and CPU memory system. Therefore after determining the modeling approach and assessing the size of the data set being used, the next step is sizing the GPU and GPU memory.

STEP TWO

Matching the best GPU and GPU memory size.

GPU DDR memory used in GPU video buffers is higher performing, higher speed DDR than is used in CPU memory. This is because the GPU DDR memory's job is to keep up with the voracious appetite of the parallel processing computational cores in the GPU.

- **Dell Precision 7550 and 7750 Mobile Data Science Workstations**
They use NVIDIA's RTX 5000 GPU that contains 3072 CUDA cores, 48 RT Cores, 384 Tensor Cores and 16GB of GDDR6 memory capable of 448GB/sec transfer rate. This GPU memory buffer is suitable for data set or batch sizes up to 16GBytes.

- **Dell Precision 5820 and 7920 Tower Data Science Workstations**
They use NVIDIA's RTX A6000, RTX 6000, and RTX 8000 GPUs. The RTX 6000 GPU contains 4608 CUDA cores, 72 RT cores, 576 Tensor cores and GDDR6 memory capable of 672GB/sec transfer rate.
The RTX 6000 and RTX 8000 differ in their respective GPU memory buffer size only. The RTX 6000 has 24GBytes, whereas the RTX 8000 has 48GBytes. This makes each RTX 6000 suitable for data set or batch sizes up to 24GBytes, and each RTX 8000 suitable for data set or batch sizes up to 48GBytes.

The 5820 DSW is a tower form factor platform that can be configured with single or dual RTX A6000, RTX 8000, or RTX 6000 GPUs. If the two GPU cards are connected via NVLink, the GPU video buffer size is doubled. This allows the 5820 DSW to support data set or batch sizes up to 96GBytes.

The 7920 DSW is available as a tower or rack form factor platform that can be configured with single, dual, or triple RTX A6000, RTX 8000, or RTX 6000 GPUs, providing as much as 144GBytes of GPU memory from three RTX A6000 or RTX 8000 GPUs.

On the 7920 Tower DSW, two of those three GPU cards can be connected via NVLink, yielding one GPU video buffer that is doubled.

This allows the 7920 DSW to support data set or batch sizes up to 96GBytes, plus up to 48GBytes of additional GPU memory from the third GPU.

STEP THREE

Deciding on the optimal CPU memory and CPU processor configuration for the CPU side of the system to keep up with the computational and data demands of the GPU side of the system.

The objective is to achieve near 100% GPU utilization during model training runs. And finally, the storage configuration and sizing.

Our experience in configuring and measuring performance on many GPU-accelerated workloads show a best practice in sizing the CPU memory for keeping the GPU side of the system fully utilized is to have double the amount of CPU DDR memory as in the GPU memory.

For example, in a 5820 Tower DSW that has two RTX A6000 GPUs, configure the CPU memory to be 192 GB. For a 7550 Mobile DSW with an RTX 5000 GPU, configure the CPU memory to be at least 32 GB.

From what we've gathered from data scientists and our own experience training various machine learning and especially deep learning models when configuring any GPU-accelerated AI workstations, it takes more than one multi-core processor to keep dual or triple GPU configured systems fully utilized. For this reason, a best practice we recommend is the following:

- Plan on having one processing core per 8-16GB of CPU DDR memory

- For when one processor does not have enough cores, divide the number of cores between the two CPU processors

- It may also be more economically viable to split the number of cores between two processors

For example, if you have dual RTX A6000 GPUs in a 7920 Tower DSW, then configure the 7920 with 192GBytes of CPU DDR memory, and plan to use two Xeon processors that each contain ~6-12 cores.

How to configure your CPU DDR memory

It is important to take full advantage of all memory channels in your processor's memory controller. This maximizes the CPU memory bandwidth available from the CPU side of the system to keep the GPU side of the system fully utilized. If you don't, you will leave sizable system performance unutilized:

- For the 7550 and 7750 Mobile DSW, this means populating the DDR DIMM sockets in multiples of four

- For the 5820 Tower DSW, this means populating the DDR DIMM sockets in multiples of four

- For the 7920 Tower and 7920 Rack DSW, this means populating each processor's DDR DIMM sockets in multiples of six

And finally, size the system's mass storage. The best practices we have gathered from data scientists call for separating the system's "warm data" data mass storage from the mass storage used as the boot drive. For AI workloads, both the boot drive and the data drive(s) should be Solid State Drives (SSD). Class 50 speed SSDs are recommended, but dropping back to Class 40 SSDs can be a good trade-off if cost effectiveness is a consideration or if doing so frees up more budget for higher core count CPUs.

With the size and complexity of data increasing dramatically in AI workloads, the size of the data drives is increasing. As of this document's publishing date, 1 to 2 TB SSDs contain most workstation-class AI model development data requirements.

Some data scientists choose to include "cold data" storage, to support the warm data SSDs. These can certainly be SSD drives, but more cost-effective SATA drives can also be used. We recommend this should be the only storage where SATA-class performance storage drives are used.

Conclusion

The insights gained from performing the benchmarks in this white paper indicate that selecting the right GPU and platform is workload dependent.

Mobile workstations provide capabilities to perform tasks in computer vision and NLP from anywhere. For more demanding workloads, such as high-resolution medical image classification or NLP with large model using large sequence length and batch size, higher-performance GPUs like the RTX 6000/8000/A6000 found in Dell tower workstations will be the right choice.

In the state-of-the-art NLP models such as BERT LARGE with over 340M parameters, experimenting with higher sequence length and batch size combination was only possible with RTX A6000 or RTX 8000 thanks to their 48GB GDDR6 memory footprint and high CUDA cores count.

In the case of machine learning tasks, such as working on tabular and structured datasets, high end GPUs such as RTX 6000/8000/A6000 can speed up loading the dataset into the GPU memory and accelerate training when using GPU-accelerated libraries such as XGBoost.

In the last section of this paper, a general guideline and best practices on how to properly size Dell Precision Workstation for Data Science was presented.

More information and related topics

Here you can find information and links referenced in this paper, plus more information and easy-to-use links to preconfigured Dell Precision Data Science Workstations ready for purchase and customization. Also, check this landing page for the most current information not available at the time of this publication.

Dell links:

Dell DSW and Isilon H400 NAS solution performance white paper

AI QRG

DSW Installation Guide

AI/DSW Industry Brief

DSW Ingredients Brief

NVIDIA links:

NVIDIA Data Science Stack

NVIDIA GPU-accelerated AI education & training

Canonical links:

https://certification.ubuntu.com

https://ubuntu.com/dell

https://ubuntu.com/contact-us

RTX A6000 GPU Performance Results

This addendum adds results from running performance analysis benchmarks on NVIDIA's RTX A6000 GPU. We followed the same benchmarking methodology as described in the body of this white paper as applied to the RTX 5000, RTX 6000, RTX 8000, and GV100 GPUs. In the performance analysis benchmarks run this time, the Linux OS was updated from Ubuntu 18.04 to Ubuntu 20.04, and the NV Data Science Software Stack updated from 2.4.0 to 2.8.0. Therefore, the results reflect not only the performance improvements from the RTX A6000 GPU hardware, but also from the optimizations in Ubuntu 20.04 and the libraries and device drivers in NVIDIA Data Science Software Stack 2.8.0.

The Dell DSW hardware platforms and configurations

We re-used and benchmarked Precision 5820 Tower DSWs and Precision 7920 Tower DSWs, and installed single, dual, and triple RTX A6000 GPUs, with and without NVLinking pairs of RTX A6000s. We also reran benchmarks on single, dual, and triple RTX 6000 and RTX 8000 GPUs since there had been updates in Ubuntu Linux OS and the NVIDIA Data Science Software Stack since we had run the original benchmarks described in the main body of this document.

The Benchmarks

We ran current versions of the benchmarks run previously, to capture Dell DSW performance on the three different workloads:

- Machine Learning - XGBoost on epidemiology data (structured data)

- Deep Learning for Image Classification - tf_cnn on ResNet50 imaging (unstructured data)

- Deep Learning for Natural Language Processing (NLP) - BERT Large on SQuAD v1.1 dataset (unstructured data)

A6000 Findings – Machine Learning using XGBoost on epdidemiology data (structured data).

XGBoost (eXtreme Gradient Boosting) is the latest evolution of decision tree based algorithms. It's built upon an ensemble of tree methods that apply the principle of boosting weak learners. It is considered best-in-class technique for tabular and structured data.

We used a simulated dataset for UK population, synthesized from official UK census data, to predict the probability of infecting a person with a simulated virus. The dataset was used in RAPIDS dli training. We used a demo notebook from Rapids.ai to run the test. RAPIDS and XGBoost are included in NVIDIA's Data Science Software Stack. We used the default RAPIDS 0.18 part of the data science stack 2.8.0

The dataset we used was a 6 Million rows X 501 Columns (the 501st column being the output representing if the person is infected or not), all in FP32 math. The model training was done on RTX A6000s and RTX 6000 to compare the duration it took to finish 500 iterations.

Figure 6 shows our results. The RTX 6000 took 23.7 seconds to complete 100 epochs, while the RTX A6000 took 14.5 seconds, 38% faster. The GV100 completed fastest, taking only 9.8 seconds.
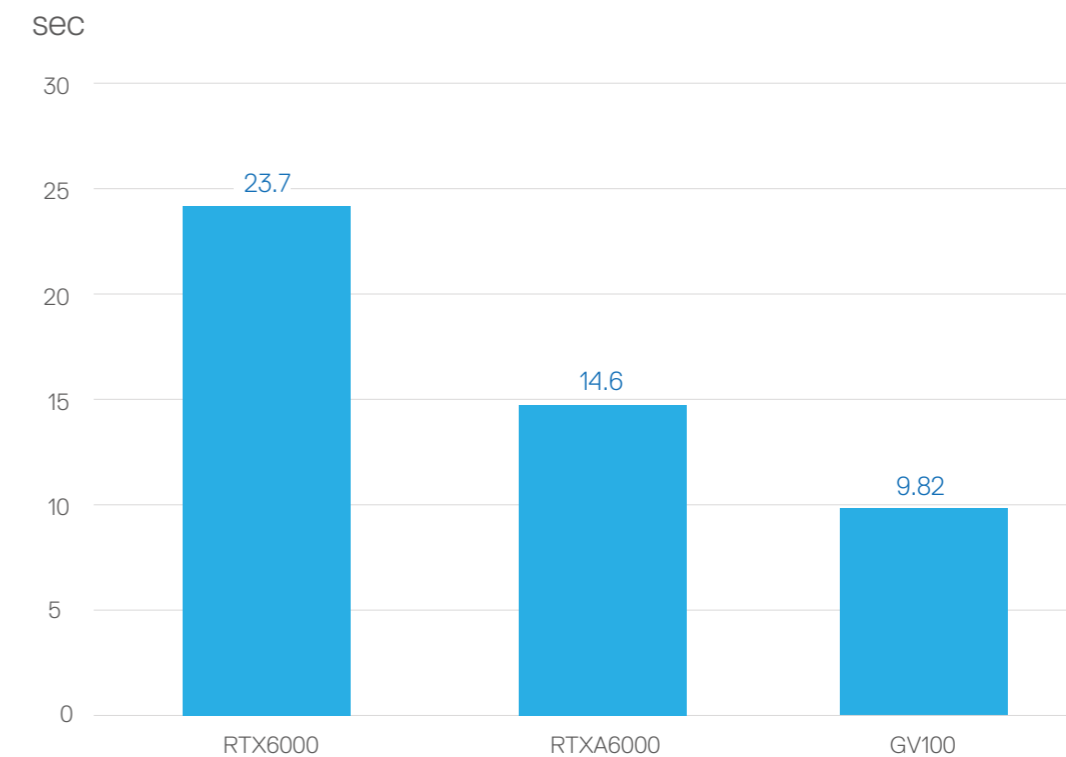
Time to train for 100 Epochs. (lower is better)

Figure 6

A6000 Findings – Deep Learning for Image Classification – tf_cnn on ResNet50 imaging (unstructured data)

The test was done using the official tf_cnn_benchmarks from TensorFlow. The repository contains scripts that run a training of standard image classification models on the synthetic images as well as ImageNet datasets. Training is run for a set number of iterations while the average speed is measured in images/second. To fully test the workstation's capabilities, we ran the benchmark with various batch sizes (BS) and number of GPUs. We used tf_cnn to train ResNet50 synthetic image dataset. We used half precision (FP16) math.

Figure 7 shows tf_cnn on Resnet50 Image Classification results. The figure shows that Dell DSWs configured with the Ampere RTX A6000 significantly outperformed both the RTX 6000 and RTX 8000 GPUs, and even the GV100 GPU. It also shows that large memory footprint was needed to complete large batch size runs (batch size=1024) the RTX 6000 simply could not handle. Dual NVLinked GPUs did not provide substantial improvement in the case of image classification using tf_cnn benchmark. It was deduced that PCIe3 bus is simply sufficient to handle the traffic between the two cards. Dual RTX A6000 performance in most cases was double the performance of the single RTX A6000. Adding a third RTX A6000 provides linear scaling which allows for training on even more samples and in less training time. This translates to a reduction in the time it takes to converge to the final deep learning model.
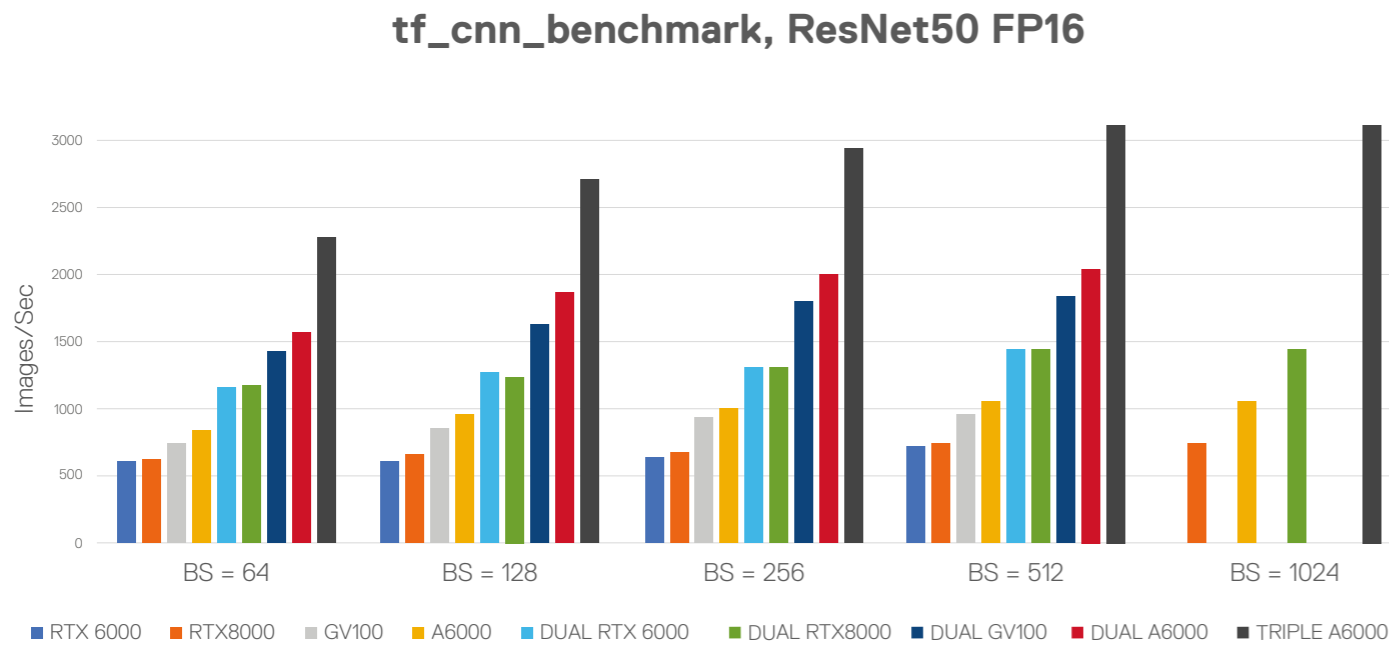
A6000 Findings – for Natural Language Processing (NLP) - BERT Large on SQuAD v1.1 dataset (unstructured data)

Natural Language Processing (NLP) is used to understand and generate human language. For NLP performance, we used BERT (Bidirectional Encoder Representations from Transformers) Large.

BERT has become a standard measure of NLP. BERT uses bi-directional language modeling to understand a word's context (i.e., the model learns the context of a word based on all its surrounding). For our measurements, we chose BERT Large training on the SQuAD v1.1 dataset.

Using Dell DSWs, we swept across the following BERT parameters...

- math: FP16 and FP32

- sequence length: 128, 384

- batch size: 1, 2, 4, 8, 16, 32, and 64

Figure 8 shows results, expressed as 'training per second'. Significant linear scaling occurs from single to dual to triple RTX A6000s. The RTX A6000 was able to handle larger combination of sequence length and batch sizes. As you move to longer data sequences, larger batch sizes and higher accuracy (i.e., FP32 math), the scaling from single to dual to triple becomes even more pronounced. In such cases, the RTX A6000 (in single or multiple configurations) could not run due to its 24GB memory limitation compared to the RTX A6000's 48GB.
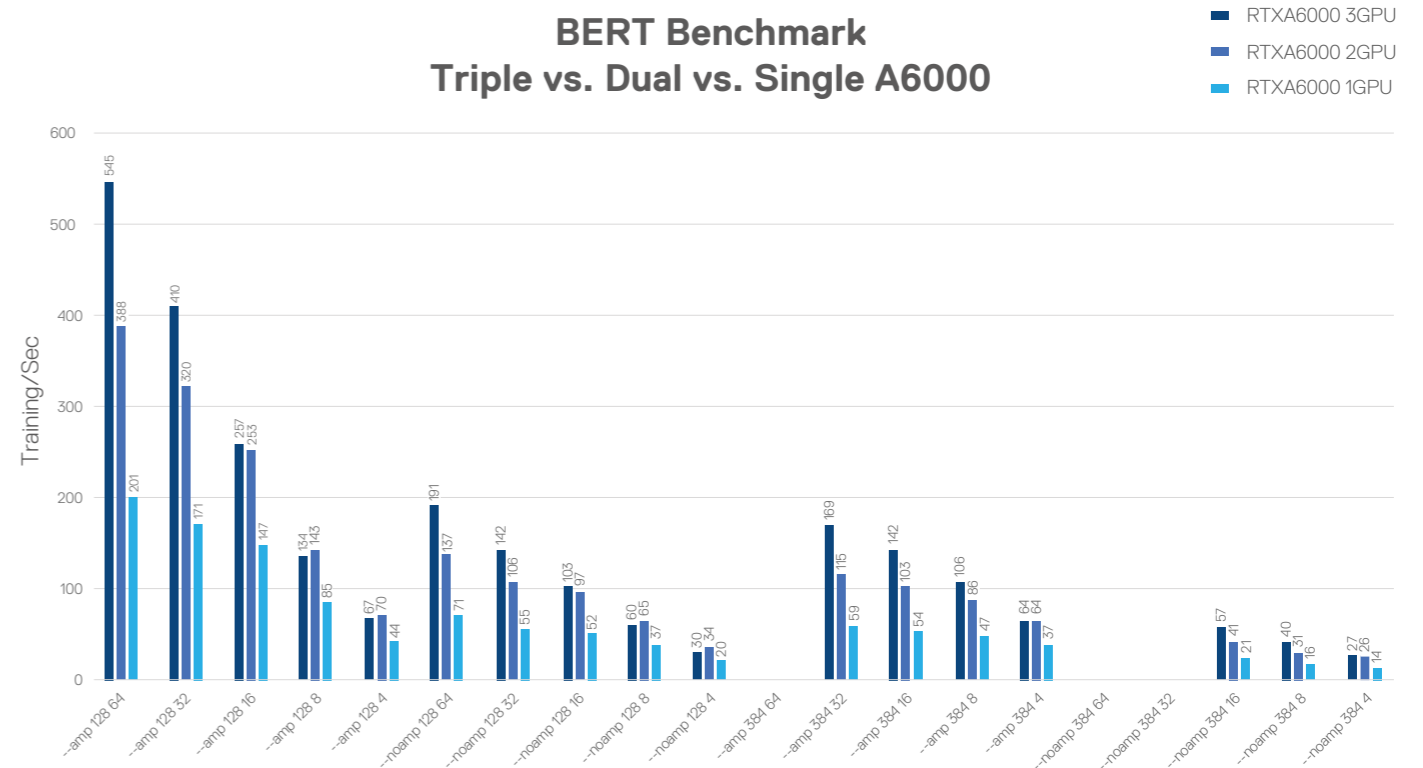

tf_cnn_benchmark, ResNet50 FP16

Figure 7

**BERT Benchmark
Triple vs. Dual vs. Single A6000**

Figure 8

A6000 Findings – The benefit of NVLinking two RTX A6000s into a pair

Figure 9 shows results of the BERT Large benchmarking comparing a pair of RTX A6000 GPUs with and without being NVLinked together. An NVLink connector connected between a pair of RTX GPUs allows each of the GPU cards' memories to be seen as one unified memory, doubling the amount of memory and doubling the amount of Tensor cores operating across that total memory for the model training. Equally important, data and communication between the two GPU cards can take place over the NVLink instead of the much slower PCIe bus. This can lead to much higher performance on large Deep Learning or Machine Learning models being trained using very large data sets. Such is the case with BERT on NLP training runs like the one we used. The chart shows that a pair of NVLinked RTX A6000 GPUs provides much higher performance than if the pair of RTX A6000 GPUs are not connected via NVLink. The performance gain is anywhere from 2.5x to 9x faster than if the two GPUS are only connected by PCIe3.2 bus.
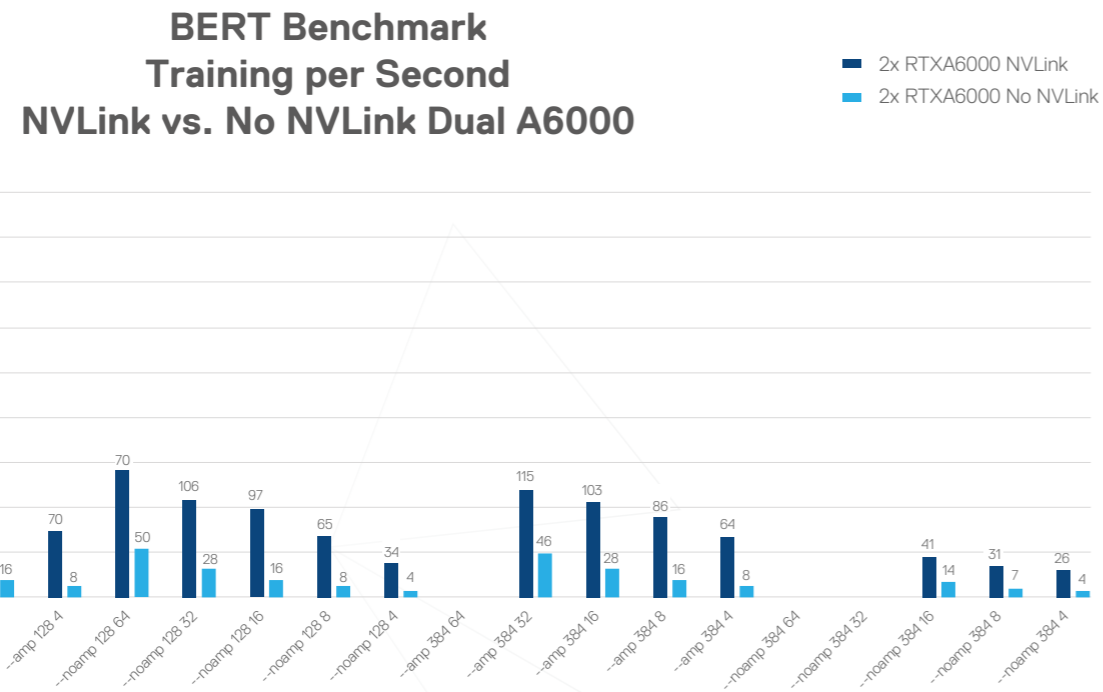
**BERT Benchmark
Training per Second
NVLink vs. No NVLink Dual A6000**

- 2x RTXA6000 NVLink
- 2x RTXA6000 No NVLink

Figure 9

DELLTechnologies

NVIDIA.
RTX