Technical Report

NetApp ONTAP AI Reference Architecture for Autonomous Driving Workloads

Solution Design

David Arnette and Sung-Han Lin, NetApp
January 2020 | TR-4799-DESIGN

In partnership with

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Executive Summary

The NVIDIA DGX™ family of systems is the world's first integrated artificial intelligence (AI) platform that is purpose-built for enterprise AI. NetApp® AFF storage systems deliver extreme performance and industry-leading hybrid cloud data-management capabilities. NetApp and NVIDIA have partnered to create the NetApp ONTAP® AI reference architecture to offer customers a turnkey solution for supporting AI and machine learning (ML) workloads with enterprise-class performance, reliability, and support.

This reference architecture offers directional guidance to customers who are building an AI infrastructure using DGX systems and NetApp AFF storage for autonomous vehicle (AV) development. It includes information about the high-level workflows used in the development of deep learning (DL) models for AVs and sizing recommendations for customer deployments.

The target audience for the solution includes the following groups:

- Infrastructure architects who design solutions for the development of AI models and software for automotive use cases, including AV.
- Data scientists who are looking for efficient ways to achieve DL development goals.
- Executive decision makers who are interested in achieving the fastest time to market from AI initiatives.

2 Solution Overview

2.1 Autonomous Driving Use Case Summary

Autonomous vehicles have incredible potential to improve roadway safety as well as efficiency. Deep neural networks (DNNs) and convolutional neural networks (CNNs) enable this autonomy, including computer vision models to interpret image and sensor data, prediction models to estimate object positions and movements, and recommendation models to suggest or execute specific actions. The National Highway Traffic Safety Administration rates levels of autonomy on a scale of 0 to 5:

- **Level 0.** No automation. Zero autonomy; the driver performs all driving tasks.
- **Level 1.** Driver assistance. The vehicle is controlled by the driver, but some driving-assist features might be included in the vehicle design.
- **Level 2.** Partial automation. The vehicle has combined automated functions, such as acceleration and steering, but the driver must always remain engaged with the driving task and monitor the environment.
- **Level 3.** Conditional automation. The driver is necessary but is not required to monitor the environment. The driver must always be ready to take control of the vehicle with notice.
- **Level 4.** High automation. The vehicle can perform all driving functions under certain conditions. The driver can have the option to control the vehicle.
- **Level 5.** Full automation. The vehicle can perform all driving functions under all conditions. The driver can have the option to control the vehicle.

Each level of autonomy, starting from L2 and above, needs development of multiple AI models, each having complex neural network architectures as well as a need for a very large training dataset. To quantify these numbers and provide a directional estimation, consider the following:

- One survey car driving 8 hours per day, 250 days per year, will create 2,000 hours of video data per year.
- With five cameras at 30 frames per second, this equates to one billion images per year.
- Using 2-megapixel camera resolution (approximately 2MB per image), each car generates about 1TB of data per hour, or 2PBs per year.

- Roughly one third of this data is cleansed and labeled to become useful for model training.
- Best practices require a few million images for the training of simple networks and five to eight million images to train highly complex networks.

During neural network development, the initial exploration phase requires fewer images, yet still needs a quick turnaround time (TAT). Throughout the model selection phase of development, the neural networks need to be trained on much larger datasets. On average, 24 hours is a good TAT target for this phase. Developers need to launch an experiment in one day, get results the next day, and develop or explore ideas continuously. Based on observed experience, on average it takes 24 hours to run training with 300K images on one NVIDIA DGX-1™ system. The total number of DGX systems can be summarized as follows:

> 10 AI models x 10 parallel experiments x 1 DGX/Model/Exp = 100 DGX-1 systems minimum needed to support the AV development with a total labeled dataset of ~3M (300K per AI model).

The key take-away is that the number of DGX systems needed is driven by labeled dataset size, how many explorations need to be run in parallel, as well as how short the TAT needs to be.

Multi-GPU training with DGX-1 systems reduces turnaround time by increasing available computing power, while features like NVIDIA® Automatic Mixed Precision (AMP) for TensorFlow leverage mixed precision on NVIDIA Tesla® V100 GPUs to provide up to eight times more throughput compared to single-precision on the same GPU. The result is that developers spend more time working and less time waiting for experiments to finish, and the resulting neural networks achieve higher prediction accuracy.

Implementing complex neural networks in development on an integrated infrastructure powered by DGX-1 systems and NetApp AFF A800 storage systems offers high performance and integrated data management. The solution eliminates bottlenecks during data preparation and model training, streamlines data movement between locations and/or storage tiers, and keeps costs to a minimum as the data footprint grows.

2.2 Iterative Workflow Process

Development of AI software for autonomous driving is an iterative workflow process that involves several steps. Here is a summary of those steps:

- **Data collection** is achieved by a fleet of survey cars collecting real-life data using onboard cameras and sensors such as radar and lidar. Typically, each car in the fleet collects about 1TB per hour. This amounts to more than 2PBs of data from one billion or more total images collected per car per year. This raw data must be moved from the car to the data factory before processing it into useable training data. Public or private cloud-based object storage can provide a cost-effective and scalable way to store the massive quantity of initial data and provides the ability to input data in one location and retrieve it for processing at another. Object storage also allows the association of rich metadata, enabling detailed data management policies for retention and protection throughout the data and model's lifecycle.
- **Data factory processing** includes data ingestion and transcoding, metadata management and processing, and active learning. It also involves human labelers using domain-specific guidelines to get useful cleansed data to feed model training. In this step data is moved from the bulk object store into faster local storage for preprocessing before entering the training step. Processed data may then be passed directly to the training cluster for immediate use or returned to the object store for later use.
- **Model training** uses labeled datasets to train neural networks for a specific task. The optimization and prediction accuracy that can be achieved for the model depend on the quality of immutable datasets and applied techniques for multi-GPU training and mixed precision. This step requires massive compute capacity and high-performance storage that can be accessed by the entire compute cluster. Object storage is typically not fast enough for this process. Data can be staged in shared storage local to the cluster before training or cached directly on the server if there is enough available space.

- **Validation and verification** involve both component level testing on real data as well as the simulation of driving scenarios and conditions, using simulation to validate model behavior under certain conditions. AI-powered autonomous vehicles must be able to respond accurately to diverse driving, traffic, and road situations, such as emergency vehicles, pedestrians, animals, and a virtually infinite number of other variations, obstacles, and environmental factors. Many of these conditions are either too dangerous to capture and test in the real world or are so uncommon that examples are not captured quickly enough for effective model training. Simulation and testing are critical to validate the full hardware and software stack before deploying the AV platform on the road in a test vehicle. The NVIDIA DRIVE Constellation™ platform can be used to replay real data in faster than real-time component level testing as well as simulate driving conditions and perform real-time processing of synthetic data using the complete hardware and software stack to be deployed in the autonomous vehicle.

2.3 Data Challenges

The data and accelerated computing requirements for achieving higher levels of autonomy are monumental. Data grows exponentially over the life of a project, and time is a critical factor in this process for businesses to make effective use of developer resources and to deliver new capabilities to market in a competitive timeframe. The infrastructure used for this development process must be able to reduce the time involved in each step and scale as the data and the organization grow, without impacting the effectiveness of individual developers.

For model training purposes, the data types in AV development vary widely, including images and video from cameras, sensor data from radar and lidar sensor sweeps, and text data from logging. All this data contributes to different aspects of the autonomous driving process and building robust and resilient AV software, including object detection, understanding of urban street scenes in various conditions and locations, comprehension of the environment around the vehicle, obstacles, and more. Model training requirements vary for distinct data types, and the achievable performance on compute and storage also vary. The goal is always to achieve high GPU utilization and provide the highest throughput at the lowest latency from the storage side.

This reference architecture focuses on addressing the challenges in the AI training phase for delivering best-in-class performance to reduce TAT and time-to-insight. In this report, model training was validated using publicly available datasets to represent and test AV relevant workloads. See Section 4 for more details.

3 Solution Technology

The NetApp ONTAP AI architecture, powered by DGX systems and NetApp cloud-connected storage systems, was developed and verified by NetApp and NVIDIA. This reference architecture gives IT organizations the following advantages:

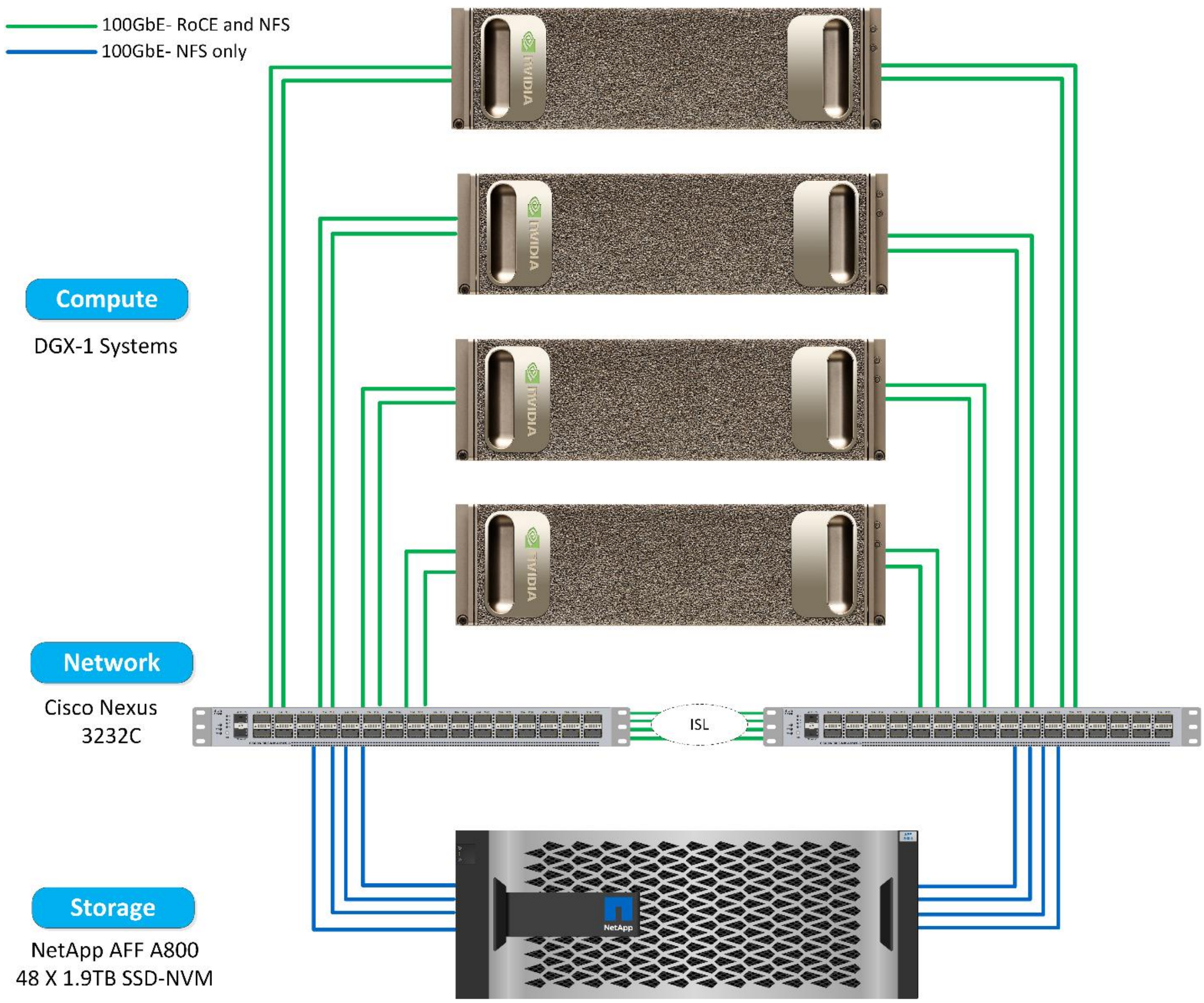
- Eliminates design complexities
- Allows independent scaling of compute and storage
- Enables customers to start small and scale seamlessly
- Offers a range of storage options for various performance and cost points

NetApp ONTAP AI tightly integrates DGX systems and NetApp AFF A800 storage systems with state-of-the-art networking. NetApp ONTAP AI with DGX systems simplifies artificial intelligence deployments by eliminating design complexity and guesswork. Customers can start small and grow their systems in an uninterrupted manner while intelligently managing data from the edge to the core to the cloud and back.

The AFF A800 storage system has been verified with nine DGX-1 systems and three NVIDIA DGX-2™ systems. Furthermore, by adding more network switches and storage controller pairs to the ONTAP cluster, the architecture can scale to multiple racks to deliver extremely high throughput, accelerating training and inferencing. With this flexible approach, the ratio of compute to storage can be altered independently based on the size of the data lake, the models that are used, and the required performance metrics. For detailed information about ONTAP AI with DGX-1 systems, see NetApp Verified Architectures [NVA-1121](#) and [NVA-1138.](#) For information about ONTAP AI with DGX-2 systems, see [NVA-1135](#).

This solution was validated with one NetApp AFF A800, four DGX-1 systems, and two Cisco Nexus 3232C 100Gb Ethernet (100GbE) switches. As illustrated in Figure 1, each DGX-1 system is connected to the Nexus switches with four 100GbE connections that are used for inter-GPU communications by using remote direct memory access (RDMA) over Converged Ethernet (RoCE). Traditional IP communications for NFS storage access also occur on these links. Each storage controller is connected to the network switches by using four 100GbE links.

Figure 1) ONTAP AI AV solution topology.



3.1 Hardware Requirements

This solution was validated using four DGX-1 systems and one AFF A800 storage system. This configuration is consistent with the 18kW DGX-1 rack design described in the [NVIDIA DGX POD Data Center Reference Design](#).

Table 1 lists the hardware components that are required to implement the solution as tested. The hardware components used in a specific customer implementation should be based on the sizing guidance in Section 5.

Table 1) Hardware requirements.

Hardware	Quantity
DGX-1 systems	4
NetApp AFF A800 system	1 high-availability (HA) pair, includes 2 controllers and 48 NVMe SSDs (3.8TB or above)
Cisco Nexus 3232C network switches	2

3.2 Software Requirements

Table 2 lists the software components that are required to implement the solution as tested.

Table 2) Software requirements.

Software	Version or Other Information
NetApp ONTAP data management software	9.5
Cisco NX-OS switch firmware	7.0(3)I6(1)
NVIDIA DGX OS	4.0.4 - Ubuntu 18.04 LTS
Docker container platform	18.06.1-ce [e68fc7a]
Container version	netapp_tf_19.10 based on nvcr.io/nvidia/tensorflow:19.10-py3
Machine learning framework	TensorFlow 1.14.1
OpenMPI	3.1.4
Object Detection and Segmentation Model	mask-rcnn-tf-nvidia-release-19.11

4 Computer Vision Model Training with Berkeley BDD100K Dataset

The model training performance of this solution was validated using the Berkeley BDD100K dataset. The following subsections contain full information about the dataset and testing results.

Note: The datasets included in this report are for research purposes only. The datasets cannot be shared without written permission of the license holders.

The Berkeley BDD100K Dataset is released by the Berkeley DeepDrive (BDD) Driving Project, which focuses on advancing computer vision capabilities for automotive applications. This dataset includes more than 100,000 HD video sequences across many different times of the day, weather conditions, and driving scenarios. In this test, 70,000 annotated images were used, each of which has a resolution of 1280x 760 pixels. In order to meet the requirements of many cameras and sensors used for autonomous driving, the images were scaled up to 1920x1080 to reach 2-megapixel resolution and an average image size of 609KB. To further show the benefits of NetApp storage systems, the scaled-up images were duplicated multiple times to generate a 1.4TB dataset.

4.1 Preprocessing Phase

Figure 2 shows the CPU and GPU utilization during the preprocessing phase, in which the images were scaled up and duplicated to produce the larger dataset. This process is very CPU intensive, while placing a much lower load on the GPUs. This finding indicates that this process could be more effectively performed in the data factory by using CPUs before the data is presented to DGX-1 systems for training.

Figure 2) DGX-1 system GPU and CPU utilization for preprocessing the BDD100K dataset.

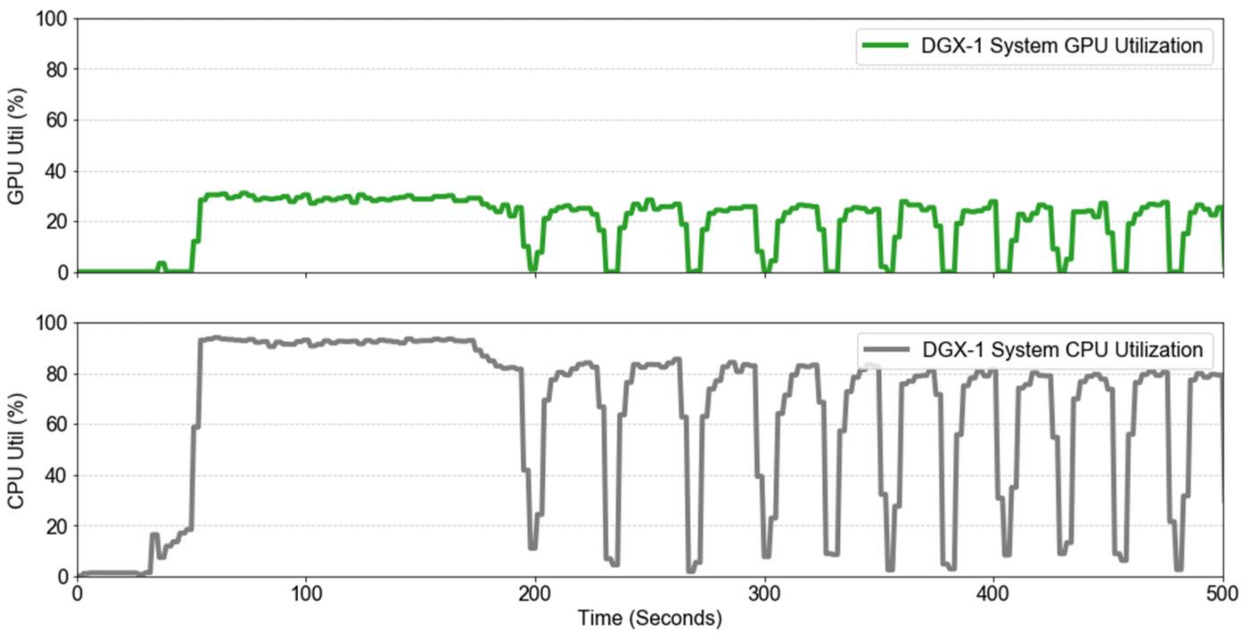
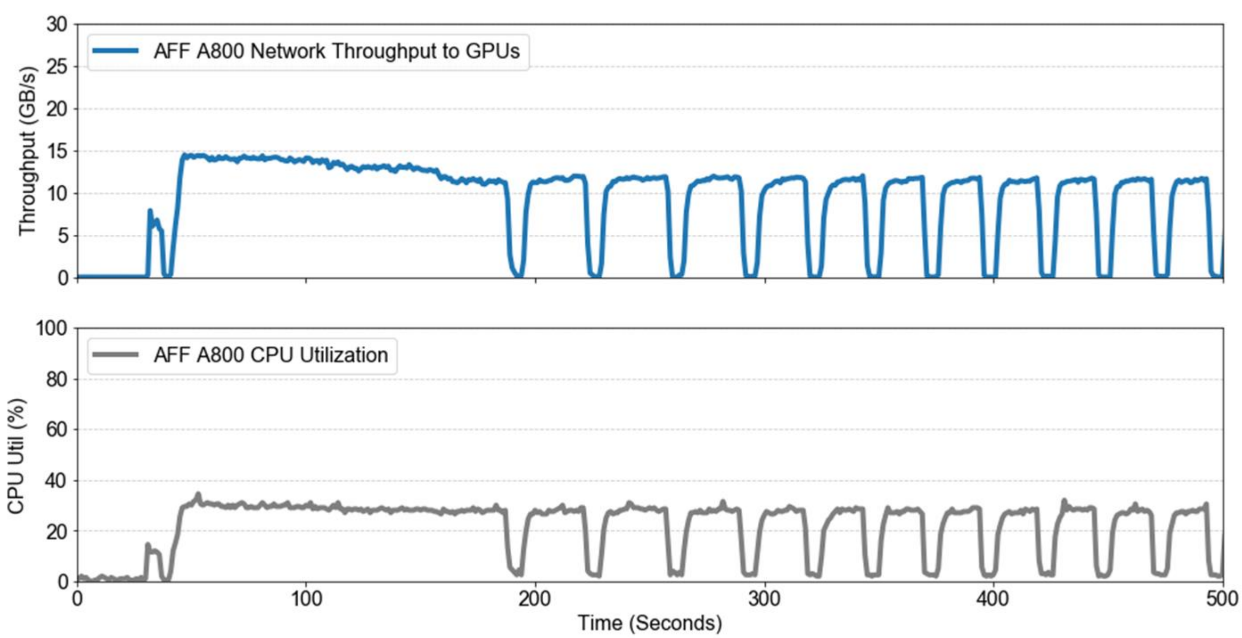


Figure 3 illustrates storage system utilization during preprocessing. It shows throughput requirements between 10 and 15GBps for all four DGX-1 systems.

Figure 3) AFF A800 network throughput and CPU utilization for preprocessing the BDD100K dataset.

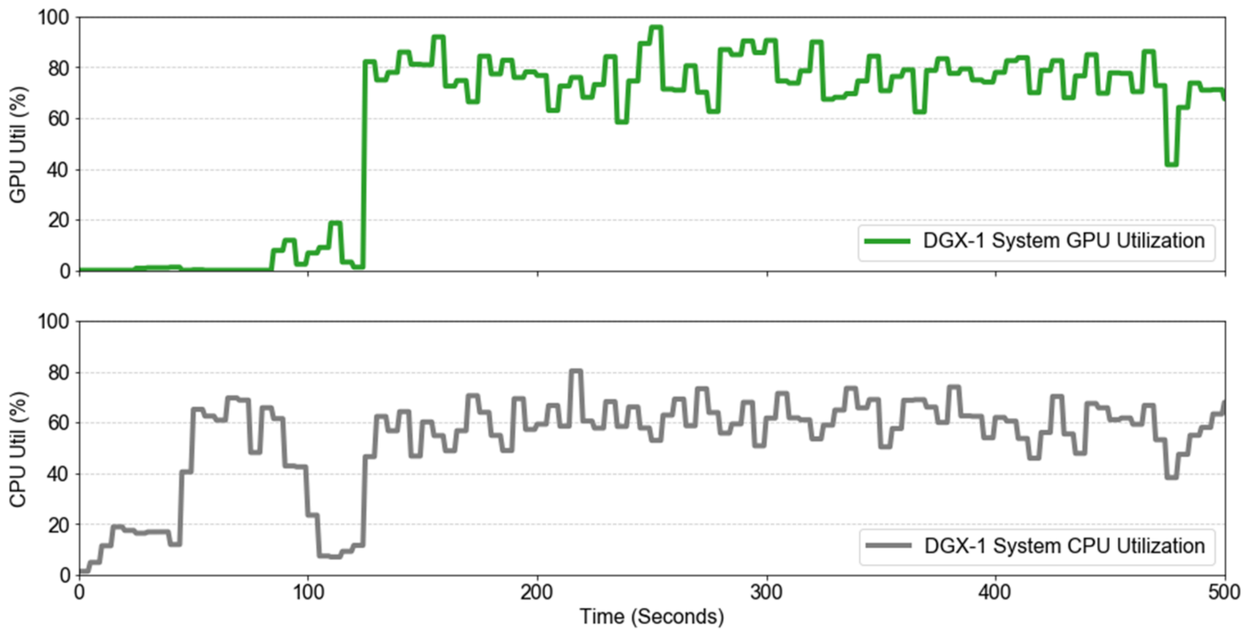


4.2 Training Phase

This test focuses on the training of an instance segmentation model, which detects the exact shape of objects on the road. Mask RCNN was used as the deep neural network, which can separate different objects in an image by defining object bounding boxes, classes, and masks. Multiple configuration options were tested to show the impacts of those options on performance and resource utilization. Any or all of the configurations tested here can be found in current AV software development operations.

The base configuration starts with ResNet-50 as the backbone, and trains with full precision (FP32) and one example per batch. Figure 4 shows the GPU and CPU utilization of a single DGX-1 system with this configuration, which results in roughly 80% GPU utilization and 60% CPU utilization. The following experiments show how other configurations affect system resource utilization.

Figure 4) DGX-1 system GPU and CPU utilization using ResNet-50 with FP32 and one example per batch.

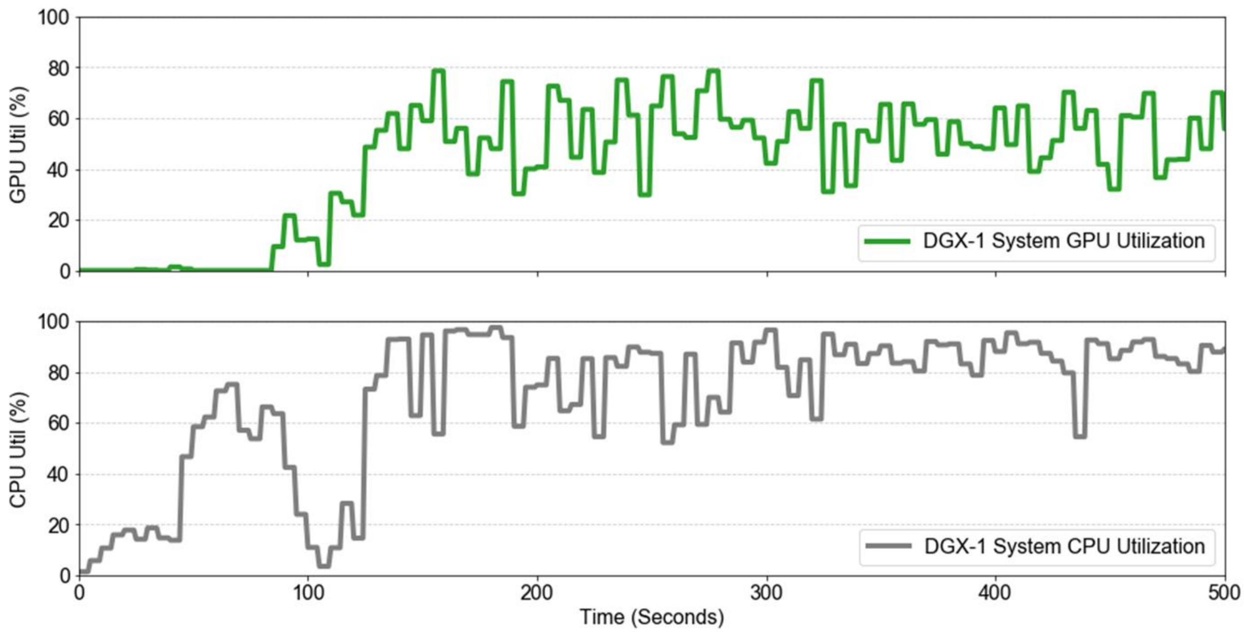


Batch Size

As shown in Figure 4, the GPUs and CPUs in the DGX-1 systems were not fully utilized. In order to reach maximum performance, the first optimization is to add more examples to each batch.

Figure 5 shows system utilization with two examples per batch, and Figure 6 shows system utilization with four examples per batch. As shown in Figure 5, using two examples per batch drives CPU utilization much higher, but GPU utilization is lower than shown in Figure 4. This is because the CPU is saturated with more preprocessing per batch and can't supply the GPUs with data fast enough to keep them busy. In this case, driving GPU utilization to maximum levels would require nearly twice the CPU performance.

Figure 5) DGX-1 system GPU and CPU utilization using ResNet-50 with FP32 and two examples per batch.



When using four images per batch, CPU utilization is also higher, as shown in Figure 6, but GPU utilization also rises higher than with two images per batch. This is because the GPUs are now processing four images at a time, which allows more time for the CPUs to prepare the next batch. Using four images per batch leads to higher overall training throughput, measured in images per second.

Figure 6) DGX-1 system GPU and CPU utilization using ResNet-50 with FP32 and four examples per batch.

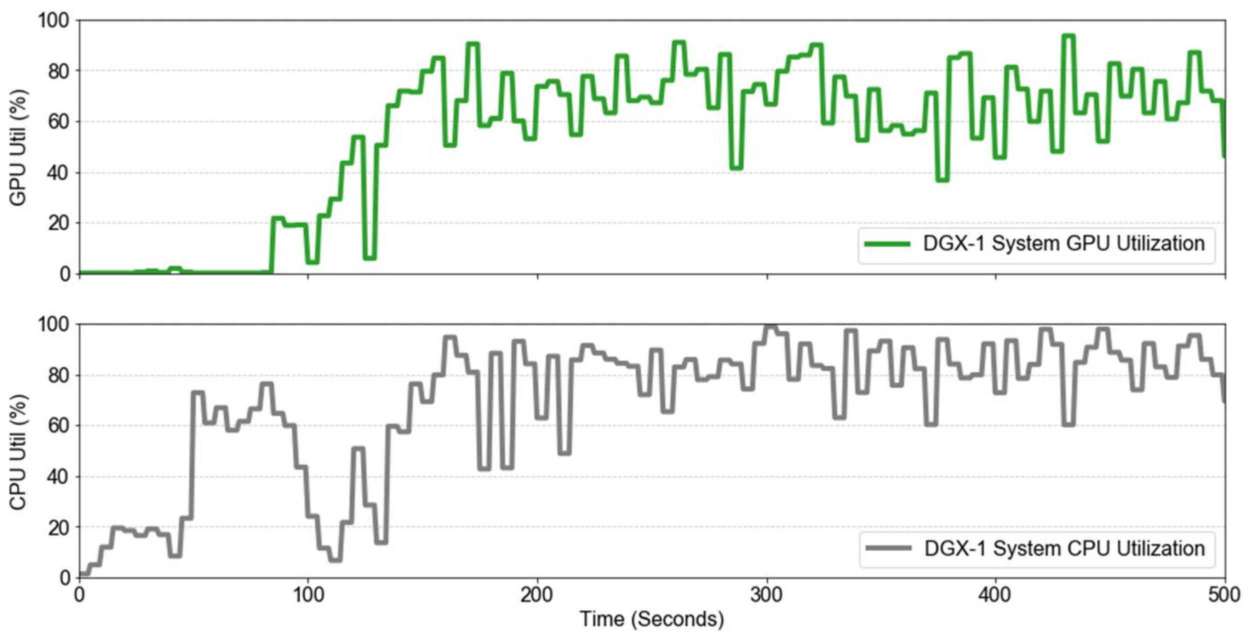


Table 3 shows the training throughput for each of these batch sizes and indicates that higher GPU utilization does not necessarily lead to higher training throughput. Factors such as batch size can increase overall throughput without reaching maximum resource utilization.

Table 3) Throughput for different batch sizes with ResNet-50 and FP32.

Examples/Batch	1	2	4
Throughput (examples/sec)	81.94	81.47	90.74

Level of Precision

Most opportunities for optimization come from using lower precision, such as FP16, especially for multiple operations, by leveraging Tensor cores available on Volta and Turing GPUs. Using lower precision can deliver higher throughput than full precision (FP32), but some operations in neural networks still require FP32. Automatic mixed precision (AMP) automatically applies FP16 when possible to improve performance, while using FP32 when necessary.

Table 4 shows the throughput results for each batch from the experiments with the same configurations as in Table 3 but using AMP instead of FP32. This table also shows the throughput of using eight examples in one batch, which cannot be executed using FP32. This benefit results from the fact that AMP reduces memory allocation for certain operations without losing accuracy, allowing more examples to be processed simultaneously. However, using a larger batch offers only a small improvement in throughput, because the pipeline cannot provide more examples to GPUs. As shown in the previous figures, CPU utilization has only a small amount of room left to leverage. Thus, even though demand from the GPU can increase, the CPUs do not have sufficient resources to preprocess more examples fast enough to meet the demand.

Generally, using AMP can help reduce the overall training time by training more examples simultaneously. However, the amount of improvement is affected by the efficiency of the data pipeline.

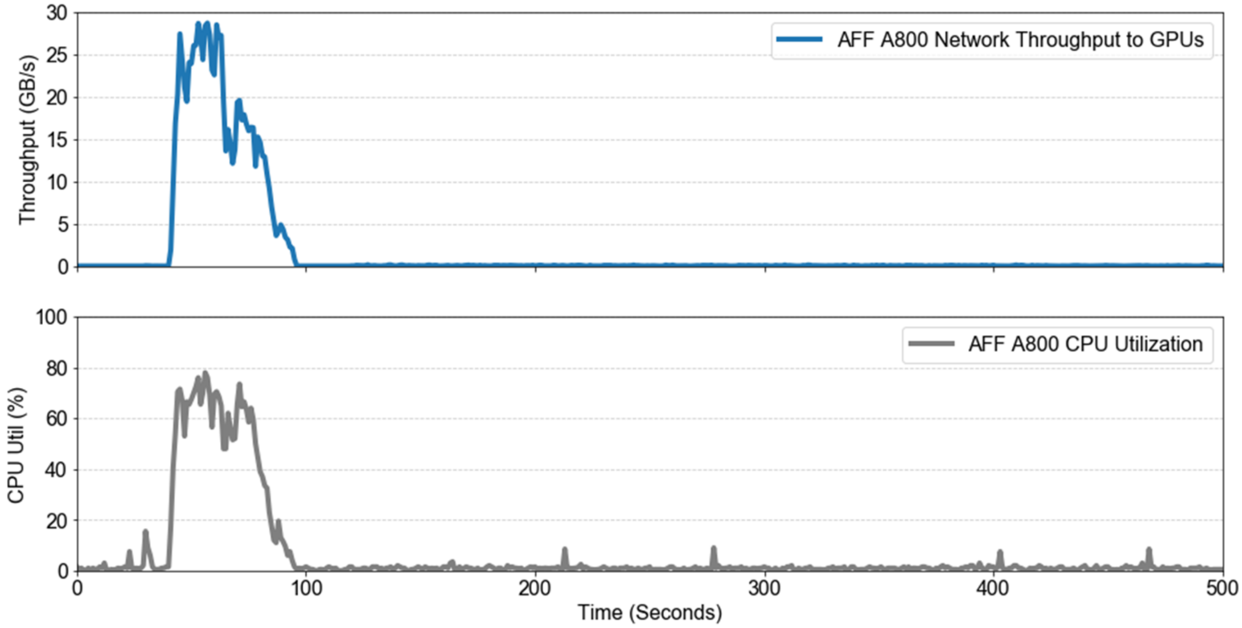
Table 4) Throughput for different batch sizes with ResNet-50 and AMP.

Examples/Batch	1	2	4	8
Throughput (examples/sec)	83.02	85.433	94.3	105.2

Storage Throughput

Figure 7 shows the network throughput from A800 to four DGX-1 systems for training and the corresponding CPU utilization of A800. As seen in this graph, the training phase produces a fairly high initial throughput spike, which is caused by buffering data into the pipeline. However, because the training process is much more compute intensive, the demand of pulling traffic from the storage to the computation is small, resulting in very low sustained throughput. This behavior is common for the workload discussed earlier.

Figure 7) AFF A800 Network throughput and CPU utilization for training the BDD100K dataset.



5 Solution Sizing Guidance

As validated in this solution as well as in [NVA-1121](), most AI training workloads require storage read throughput of roughly 1.5GBps per DGX-1 system. Based on the test results above for training with the BDD100K dataset using four DGX-1 systems, the AFF A800 storage system could easily support nine DGX-1 systems with the same workload characteristics. Synthetic performance benchmarks performed in NVA-1121 show that the AFF A800 storage system delivers up to 25GBps of read performance allowing significant headroom for data preparation or other workflow tasks.

The image data used in most AV model training is not conducive to storage efficiency techniques such as deduplication and compression, so customers can typically expect very little savings from those features. NetApp AFF A800 systems support a variety of SSD capacity options ranging from 100TB to 800TB per A800 storage system. Customers can choose the SSD size that meets their capacity requirements without any effect on performance.

For organizations that require larger numbers of DGX-1 systems, NetApp ONTAP supports storage clusters of up to 24 nodes enabling linear scaling of capacity and performance as DGX-1 systems are added to the environment. Please consult with a NetApp technical representative about detailed sizing for specific workload requirements.

6 Conclusion

The [ONTAP AI]() reference architecture is an optimized platform for the development of machine learning models for autonomous vehicles and many other use cases. With the accelerated compute power of DGX-1 systems and the performance and data management of NetApp ONTAP, ONTAP AI enables a full range of data pipelines that spans the edge, the core, and the cloud for successful AV projects.

The models and datasets tested in this solution show that ONTAP AI can easily support the workload requirements for model training with AV datasets and feed the DGX-1 GPUs with 90% utilization, while the AFF A800 storage system delivered all the required storage performance with headroom to spare.

Acknowledgements

The authors gratefully acknowledge the contributions that were made to this technical report by our esteemed colleagues from NVIDIA: Patricia de Boer, Bill Veenhuis, Jacci Cenci, Pradeep Gupta, Jarrod Puyot, Darrin Johnson, and Robert Sohigian. The authors would also like to acknowledge the contributions of key NetApp team members Santosh Rao, Erik Mulder, and Robert Nagy.

Our sincere appreciation and thanks go to all these individuals who provided insight and expertise that greatly assisted in the creation of this paper.

Where to Find Additional Information

To learn more about the information that is described in this document, review the following resources:

- NVIDIA DGX-1 systems
  - NVIDIA DGX-1 systems
    https://www.nvidia.com/en-us/data-center/dgx-1/
  - NVIDIA Tesla V100 Tensor core GPU
    https://www.nvidia.com/en-us/data-center/tesla-v100/
  - NVIDIA GPU Cloud
    https://www.nvidia.com/en-us/gpu-cloud/
- NetApp AFF systems
  - AFF datasheet
    https://www.netapp.com/us/media/ds-3582.pdf
  - NetApp Flash Advantage for AFF
    https://www.netapp.com/us/media/ds-3733.pdf
  - ONTAP 9.x documentation
    https://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286
  - NetApp FlexGroup technical report
    https://www.netapp.com/us/media/tr-4557.pdf
- NetApp ONTAP AI
  - ONTAP AI with DGX-1 and Cisco Networking Design Guide
    https://www.netapp.com/us/media/nva-1121-design.pdf
  - ONTAP AI with DGX-1 and Cisco Networking Deployment Guide
    https://www.netapp.com/us/media/nva-1121-deploy.pdf
  - ONTAP AI with DGX-1 and Mellanox Networking Design Guide
    https://www.netapp.com/us/media/nva-1138-design.pdf
  - ONTAP AI with DGX-2 Design Guide
    https://www.netapp.com/us/media/nva-1135-design.pdf
- ONTAP AI networking
  - Cisco Nexus 3232C series switches
    https://www.cisco.com/c/en/us/products/switches/nexus-3232c-switch/index.html
  - Mellanox Spectrum 2000-series switches
    https://www.mellanox.com/page/products_dyn?product_family=251&mtag=sn2000
- Machine learning frameworks and tools
  - TensorFlow: An Open-Source Machine Learning Framework for Everyone
    https://www.tensorflow.org/
  - Enabling GPUs in the Container Runtime Ecosystem
    https://devblogs.nvidia.com/gpu-containers-runtime/

- MaskR CNN – Convolutional Neural Network for object detection and segmentation
  https://arxiv.org/abs/1703.06870
- Dataset and benchmarks
  - ImageNet
    https://www.image-net.org/
  - Berkeley
    https://bdd-data.berkeley.edu/
  - TensorFlow benchmarks
    https://github.com/tensorflow/benchmarks

Version History

Version	Date	Document Version History
Version 1.0	September 2019	Initial release
Version 1.1	January 2020	Berkeley BDD100K dataset added

Refer to the [Interoperability Matrix Tool (IMT)] on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

TR-4799-0120

NetApp