REFERENCE ARCHITECTURE VERSION 2.0

AIRI™

SCALE-OUT AI-READY INFRASTRUCTURE ARCHITECTED BY PURE STORAGE AND NVIDIA

WITH CISCO NEXUS 9300 SWITCH

PURESTORAGE®

Powered by NVIDIA.

TABLE OF CONTENTS

INTRODUCTION

Will rising infrastructure complexity delay your AI initiatives? Learn how to deploy a fully integrated stack from NVIDIA® and Pure Storage® to improve time-to-insight and drive success in these crucial, investment-heavy projects.

Advances in NVIDIA GPU computing have enabled a new wave of applications for artificial intelligence (AI). Powerful new tools and techniques have enabled breakthroughs in fields as diverse as self-driving cars, natural-language translation, and predictive health care. As a result, investment in AI initiatives has skyrocketed, with companies worldwide investing between $26B and $39B in 2016 alone, according to a recent McKinsey report.[1]

Deep neural networks comprise millions of trainable parameters, connected through a configurable network of layers. In addition to the large set of parameters, each network has countless variations of topologies, connection strategies, learning rates, etc. – collectively referred to as *hyper-parameters*. Identifying an optimal set of parameters and hyper-parameters amounts to an enormous search problem that demands broad access to massive computational power.

In addition to large-scale computational requirements, recent research[2] shows that training accuracy increases logarithmically with the volume of training data. Thus, small improvements in accuracy could require a 10x or greater increase in dataset size. Along with large-scale computing power, creating state-of-the-art models requires larger, more diverse data sets with high-performance access.

Designing, configuring, and maintaining infrastructure to satisfy the challenges of large-scale deep learning requires a significant investment of time and resources to avoid unforeseen delays, bottlenecks, or downtime. Engineers at NVIDIA and Pure Storage have worked to deliver a fully integrated platform that offers scale-out deep learning *out of the box*, with time-to-insight in hours rather than days or weeks.

Accelerating Computation: NVIDIA® DGX-1™

Originally designed for computer graphics, NVIDIA engineers tapped into the massively parallel architecture of the modern graphics processing unit (GPU) and optimized it in hardware and software to deliver the performance demanded by the data-parallel, computationally-intensive algorithms that enable deep learning today. Coupled with NVIDIA-engineered, optimized software frameworks to harness the underlying computational capability, NVIDIA GPUs have become the de facto computational platform of deep learning.

While the advances in individual GPU performance have been impressive, state-of-the-art results require a scalable architecture to ensure sufficient compute. NVIDIA developed the Tesla series of GPU accelerators specifically for data-center scale systems, architecting them with high-bandwidth, low latency interconnects between GPUs in a server and enabling the GPUs to communicate directly between servers over RDMA-capable network fabrics.

[1] J. Bughin, E. Hazan, S. Ramaswamy, M. Chui, T. Allas, P. Dahlström, N. Henke, M. Trench, Artificial Intelligence: The Next Digital Frontier, McKinsey Global Institute, 2017.
[2] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, Revisiting Unreasonable Effectiveness of Data in Deep Learning Era, ICCV, 2017.

However, building a platform maximized for GPU-accelerated performance demands the optimal integration of hardware and software, one that's purpose-built for deep learning. To simplify the process of selecting and integrating compute hardware with complex deep learning software, NVIDIA created the DGX-1™ platform. Each DGX-1 packs the AI computational power of 800 CPUs, packaged in a 3U appliance, pre-configured with eight state-of-the-art Tesla V100 GPUs, high-performance interconnects, and NVIDIA-optimized software frameworks and applications that ensure maximized deep learning training performance.

Accelerating Data Access: Pure Storage FlashBlade™

Deep learning requires more than fast compute and high-bandwidth interconnects. When designing a full-stack platform for large-scale deep learning, the system architect's goal is to *provision as many GPUs as possible, while ensuring linearity of performance as the environment is scaled, all the while keeping the GPUs fed with data*.

Keeping the GPUs fed requires a high-performance data pipeline all the way from storage to GPUs. When defining storage for deep-learning systems, architects must consider three requirements:

- **DIVERSE PERFORMANCE** Deep learning often requires multi-gigabytes-per-second I/O rates but isn't restricted to a single data type or I/O size. Training deep neural network models for applications as diverse as machine vision, natural-language processing, and anomaly detection requires different data types and dataset sizes. Storage systems that fail to deliver the performance required during neural network training will starve the GPU tier for data, and prolong the length of the run, inhibiting developer productivity and efficiency. Providing consistency of performance at various IO sizes and profiles at a capacity scale will ensure success.
- **SCALABLE CAPACITY** Successful machine learning projects often have ongoing data acquisition and continuous training requirements, resulting in a continued growth of data over time. Furthermore, enterprises that succeed with one AI project find ways to apply these powerful techniques to new application areas, resulting in further data expansion to support multiple use cases. Storage platforms with inflexible capacity limits result in challenging administration overheads to federate disparate pools.
- **STRONG RESILIENCY** As the value of AI grows within an organization, so does the value of the infrastructure supporting its delivery. Storage systems that result in excessive downtime or require extensive administrative outages can cause costly project delays or service disruptions.

> Existing storage systems sacrifice one or more of these dimensions, or force architects and administrators to suffer through excessive deployment and management complexity.

Initial deployments for deep learning often start with direct-attached storage (DAS), resulting in hard capacity limits and challenges in sharing data sets across multiple compute units. Collecting multiple DAS servers into a shared file system with the Hadoop Distributed Filesystem (HDFS) can alleviate the capacity concerns, but comes at a stark performance cost for small, random I/O patterns that are common in many deep learning use cases. Furthermore, burdening the CPUs in a GPU server with storage management can lead to bottlenecks in the overall pipeline and poor resource utilization.

Parallel file systems such as Lustre, designed specifically for the needs of high-performance computing (HPC), can b e tuned by expert-level administrators to meet the requirements of a particular workload. However, a new data set or training paradigm inevitably requires a new configuration and tuning process, resulting in project delays and potential stranded capacity.

Traditional NAS offerings can provide strong resilience and scalable capacity but often fail to deliver the performance required across a range of I/O patterns and at large-scale compute clusters.

Pure Storage FlashBlade™, with its scale-out, all-flash architecture and a distributed file system purpose-built for massive concurrency across all data types, is the only storage system to deliver on all of these dimensions, while keeping required configuration and management complexity to a bare minimum.

Accelerating Time-to-Value in AI: AIRI™

AIRI is a converged infrastructure stack, purpose built for large-scale deep learning environments. Engineers at NVIDIA and Pure Storage have worked to deliver a fully integrated platform that offers scale-out deep learning *out of the box*, with time-to-insight in hours rather than days or weeks. AIRI's flexible, rack-scale architecture enables enterprises to add DGXs or blades independently, based on their growing AI initiatives: from a compact AIRI "Mini" to rack-scale with no downtime or data migration. The entire stack is configured and tested as a complete solution, avoiding the intricate configuration and tuning required otherwise.

AIRI brings together all the benefits of the NVIDIA® DGX-1™ and Pure Storage FlashBlade, wrapping them in a high-bandwidth, low-latency network fabric that unifies storage and compute interconnects with RDMA-capable 100Gb/s Ethernet.

AIRI enables seamless scaling for both GPU servers and storage systems. As compute demands grow, additional DGX-1 servers can be provisioned in the high-performance fabric and instantly access all available datasets. Similarly, as storage capacity or performance demands grow, additional blades can be added to the FlashBlade system with zero downtime or re-configuration.

FIGURE 1. "AIRI Mini" and AIRI with 100GbE Cisco Nexus 9000 Switches

How to Use This Guide

This reference architecture describes the design for AIRI Mini and AIRI, containing 2x and 4x NVIDIA DGX-1s respectively and a Pure Storage FlashBlade. To optimize DGX-1 node-to-node communication, the configuration uses an RDMA over Converged Ethernet (RoCE) fabric. The same high-performance fabric carries storage traffic from the FlashBlade to the DGX-1 servers, simplifying the system configuration and deployment.

We demonstrate performance and scalability using TensorFlow benchmarks for ImageNet, a popular method for measuring system performance of deep learning environments. Our testing results show:

- A RoCE fabric provides excellent scalability for convolutional neural networks
- FlashBlade delivers a high-performance data platform for deep learning, with performance on par with DRAM-resident datasets

SYSTEM ARCHITECTURE

The architecture for AIRI Mini and AIRI looks as follows:

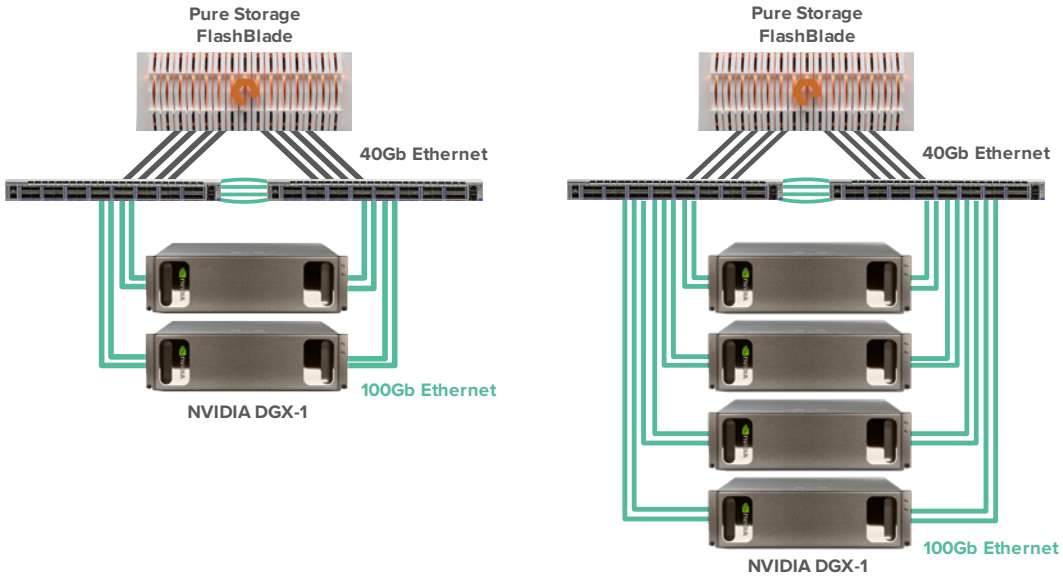

FIGURE 2. AIRI Mini and AIRI system architectures

The AIRI architecture is designed for scale-out deep learning workloads and is not restricted to these sizes. As datasets and workload requirements scale, additional DGX-1 servers can be provisioned and instantly access all available data. Similarly, as storage capacity or performance demands grow, additional blades can be added to the FlashBlade system with zero downtime or re-configuration.

Each of the core components in the architecture is described in further detail below.

- **COMPUTE** Each NVIDIA DGX-1 server with Tesla V100 comprises
  - 8x Tesla V100 GPUs (SXM2 form factor)
  - 2x Intel E5-2698 v4 @ 2.20GHz
  - 4x Mellanox MT27700 100Gb/s VPI adapters
  - 512GB DDR4-2400
- **STORAGE** Pure Storage FlashBlade contains:
  - for AIRI Mini: 7x 17TB blades (119T usable total, before data reduction)
  - for AIRI: 15x 17TB blades (179T usable total, before data reduction)
  - 8x 40Gb/s uplinks
- **NETWORKING**
  - 2x Cisco Nexus 9336C-FX2 Ethernet switch containing:
    - 36x 40/100Gbps QSFP28 ports
  - 1x Cisco Nexus 9348GC-FXP Ethernet switch containing:
    - 48x 100M/1G BASE-T ports
    - 4x 1/10/25Gbps SFP28 ports
    - 2x 40/100Gbps QSFP28 ports

NETWORK ARCHITECTURE

The diagram below shows the overall network topology of AIRI, excluding a separate management network. In the diagram, FM-1/2 are the two internal fabric modules of FlashBlade, SW-1/2 are dual 100G switches, and DGX-1 (a), DGX-1 (b), DGX-1 (c), and DGX-1 (d) are the DGX-1 servers. Each of the numbers on SW-1/2 indicate the Ethernet port number, and the numbers on the DGX label which of the ethernet devices (enp5s0, enp12s0, etc.). The network topology for AIRI Mini is the same except that there are only two DGX-1 servers (DGX-1 (a) & DGX-1 (b)).

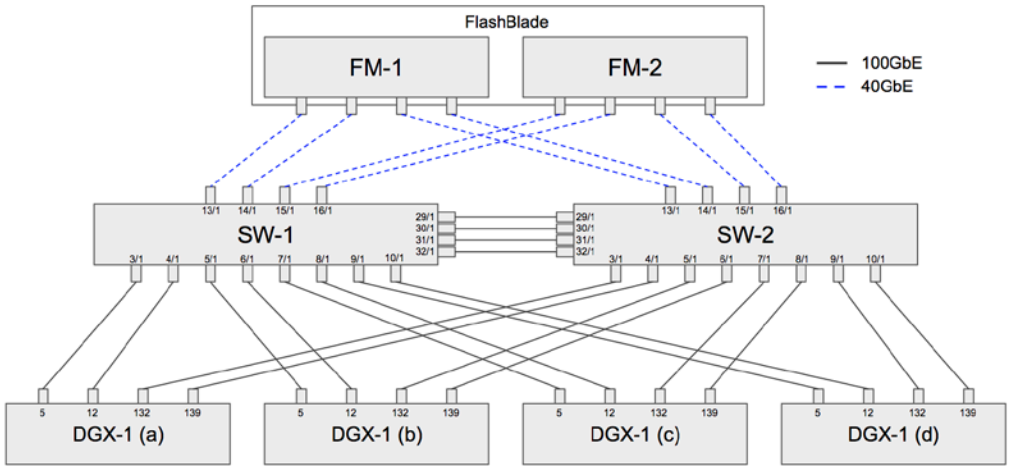

FIGURE 3. Network topology

Our reference configuration used Cisco 100G, 36-port Ethernet switches (Nexus 9336c-FX2, running NXOS 7.0(3)I7(3)), but other switch models may be used in the future. We configure the network to support two classes of traffic over the unified fabric: a VLAN for TCP flows, which includes NFS access from FlashBlade; and the RDMA-over-Converged-Ethernet (RoCE) traffic used between DGX-1 servers during distributed deep learning. For more details on the network architecture and configuration, please refer to the AIRI Configuration Guide at https://github.com/PureStorage-OpenConnect/AIRI.

BENCHMARKING DEEP NEURAL NETWORK TRAINING

This section presents benchmark results from training a variety of deep neural networks. Though the test setup is the same for both AIRI Mini and AIRI, for this reference architecture, we specifically focus on the AIRI system. We measure training performance for a variety of convolutional networks with the popular ImageNet dataset. We use these tests to answer two important questions:

1. What is the benefit of RDMA-over-Ethernet vs TCP/IP-based messaging between DGX-1 nodes?
2. What is the end-to-end system training performance when sourcing all data from FlashBlade over the system's network fabric?

Test Setup

The tests used the Imagenet 2012 dataset – one of the most influential in deep learning research – as the input for training. The dataset consists of 1.28 million images, 143 GB in total size. Input data was labeled jpeg images packed into larger files (~135MB each). To stress the data-movement and core GPU processing, all tests ran with distortions – image pre-processing steps – disabled.

We followed common best practices for TensorFlow performance optimization and trained each model until reaching a steady state of images/sec processed, and then recorded performance measurements for 100 training iterations. Each benchmark test was run three times and the overall metric reflects the median of these runs.

Our tests use the Horovod library[3] to scale training across GPUs and across DGX-1s. Although TensorFlow has internal capabilities to distribute training operations, our experiments have found higher performance and simpler overall operation using the combination of TensorFlow and Horovod. We used the 18.10-py2 release of the NVIDIA-supplied container for TensorFlow, which comes pre-packaged with CUDA 10.0.130, CuDNN 7.4.0.11, NCCL 2.3.6, and other essential libraries. Building off this base image, we added Horovod v0.13.10 and OpenMPI 3.1.2 into the containers used in the following tests.

We used the Tensorflow CNN benchmarks for Horovod, a derivative of the TensorFlow benchmark suite, which holds results curated by the TensorFlow team. We ran **git SHA 220659196** from the TensorFlow benchmark suite.[4]

[3] https://github.com/uber/horovod
[4] https://github.com/tensorflow/benchmarks

We compared FP32 precision performance across a range of standard neural network models. These models make various tradeoffs between computational complexity, number of parameters, and predictive accuracy.

For each neural-network model, we varied a number of TensorFlow parameters to find the settings for each model that maximized training performance, measured as images per second. Although there are many configurable options for TensorFlow, we found the ones listed in the following table to affect performance the most.

	RESNET152	RESNET50	INCEPTION3	VGG16
BATCH SIZE (BATCH_SIZE)	64	64	64	64
PREFETCH SCALE (BATCH_GROUP_SIZE)	24	20	24	28
THREAD-POOL SIZE (NUM_INTER_THREADS)	10	5	20	40

TABLE 1. Configurable options in TensorFlow

Distributed Training with RDMA over Converged Ethernet

The Horovod library relies on an MPI substrate to orchestrate the distributed execution, but all data exchanges – gradient updates from neural network training – are performed by the NCCL library provided by NVIDIA. The NCCL library includes significant optimizations for communication over NVLink within each DGX-1, as well as communication over RDMA between DGX-1s.

By default, NCCL uses the RDMA fabric present in the system. By setting the environment flags **NCCL_IB_DISABLE=1** and **NCCL_SOCKET_IFNAME=bond0**, we can force NCCL to use the 100Gb TCP VLAN.

The figure below shows training performance, measured in overall images-per-second, when run over the RoCE-enabled fabric vs TCP/IP-based messaging. In these initial results, we are not performing external I/O of any kind. That is, all data is synthetic and uses random values initialized in GPU memory. This places maximum stress on the network fabric between the DGX-1 nodes and highlights any potential bottlenecks in GPUs and inter-GPU exchange.

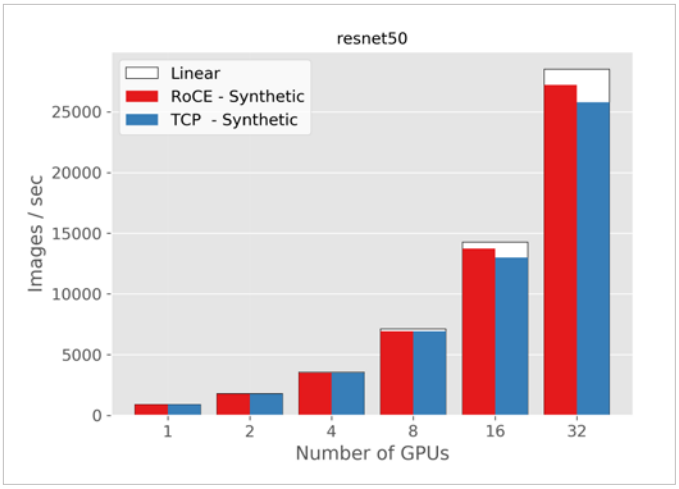

FIGURE 4. Training throughput for resnet50 with varying numbers of GPUs, contrasting performance of RDMA-over-Converged-Ethernet (RoCE) and conventional TCP/IP fabrics. All measurements use synthetic data, isolating the performance effects of GPUs and network interconnects. Results are contrasted with a linear-scaling result corresponding to (# GPUs) * throughput with 1 GPU.

We use resnet50 to show the performance scaling by GPU count. With GPU counts of 8 or less, a single DGX-1 was used. 16-GPU experiments used two DGX-1s, and 32-GPU experiments used four DGX-1s. The "Linear" bars represent linear scaling, taking the throughput result with one GPU and multiplying that by the number of GPUs in the cluster.

When using the RoCE fabric, resnet50 exhibits excellent scaling – over 91% efficiency relative to linear – to four DGX-1 nodes. This scaling efficiency is the result of overlapping computation and communication in Horovod and efficient communication collectives provided by NCCL.

By disabling RDMA and reverting to TCP/IP for inter-node messaging, resnet50 scales less efficiently.

Although the peak network demand does not continuously saturate the four 100GbE links on a DGX-1, the overall performance benefit of RoCE comes from its combination of high throughput with minimal CPU utilization. The system further benefits from NVIDIA's GPUDirect feature, enabling GPUs to transfer data from GPU memory on one node into a remote node with no CPU involvement.

Distributed Training with Flashblade™

Where the prior experiments established the importance of RDMA messaging by focusing on a synthetic dataset, we now show the performance results when sourcing real data into the training pipeline.

In the figure below, we contrast three configurations for resnet50. First is the synthetic test from the previous section as a baseline, representing the highest performance we can expect to achieve when sourcing real data. Second, because the overall dataset for this benchmark is small enough, we can cache the entirety of it in DRAM. Real-world datasets are often orders-of-magnitude larger than this, but as a performance test this allows us to create a storage layer with ideal performance characteristics. Finally, we run with real datasets accessed over NFS from FlashBlade.

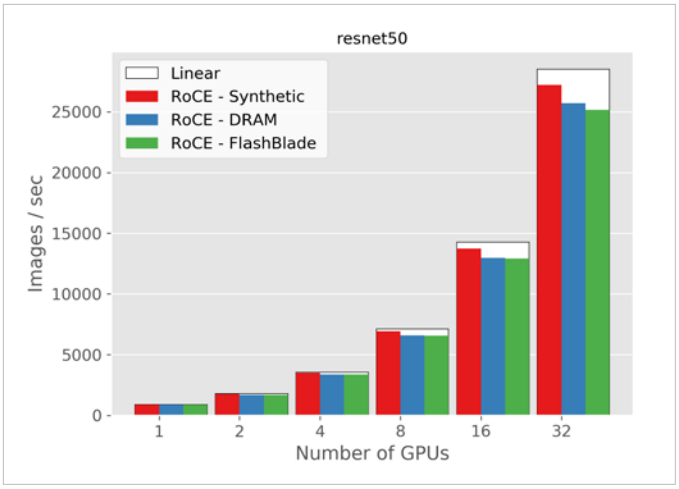

FIGURE 5. Training throughput with varying numbers of GPUs, contrasting performance of synthetic data with real input pipelines fed either from DRAM or over NFS from FlashBlade. All tests were run with FP32 precision and batch size 256.

FlashBlade throughput is within 5% of the results obtained with synthetic data, closely matching the performance of data hosted entirely from DRAM. These results were obtained after tuning the input pipeline in TensorFlow. Notably, the **batch_group_size** option in the benchmark kit controls the prefetch depth in the input pipeline. A larger prefetch depth helps hide storage latency at a small increase in memory consumption.

While prefetch depth can hide small latency variations, the storage system needs sufficient steady-state throughput to keep up with the overall pipeline. As shown in the figure below, taken from the FlashBlade UI, the aggregate bandwidth from 4 DGX-1 servers running ResNet-50 was nearly 3 GB/s. Other neural network models tested had similar bandwidth demands. We can see the effect of the prefetch operations, as the storage activity quiesces when the prefetch queue is filled. The high, sustained transfer rate highlights the importance of the storage system with this many high-performance GPUs.

FIGURE 6. Screenshot taken from the FlashBlade UI while training a ResNet-50 model. The storage bandwidth, shown in the bottom graph, indicates an aggregate demand of nearly 3.5 GB/s. The brief dip in bandwidth indicates the input pipeline is filled with images waiting on GPUs to process.

CONCLUSION

Artificial intelligence, fueled by rapid innovation in deep learning ecosystems, is becoming prevalent in a wide range of industries. Experts now believe new industry leaders will arise, led by enterprises who invest in AI and turn their data into intelligence. While many enterprises want to jumpstart their AI initiatives, challenges in building an AI-optimized infrastructure often hold them back.

AIRI aims to solve infrastructure complexities, providing enterprises a simple solution to hit the ground running. Engineers at NVIDIA and Pure Storage partnered to architect an affordable, simple, yet powerful infrastructure that delivers maximum performance out-of-the-box. It is a building block that is capable of scaling to multiple racks, supporting large enterprises as their AI needs grow.

AIRI eliminates the difficulties of building an AI data center – while neatly and compactly delivering all the necessary components in a small form-factor. Now every enterprise can finally start to explore what AI can do with their most important asset, data.

ADDITIONAL RESOURCES

- AIRI Github Site: https://github.com/PureStorage-OpenConnect/AIRI
- AIRI Configuration Guide: See github site above
- AIRI Product Page: www.purestorage.com/airi

ps_wp12p_airi-reference-architecture-cisco_ltr_02

PURESTORAGE®

Powered by NVIDIA.