White Paper

NVIDIA DGX-1 System Architecture

The Fastest Platform for Deep Learning

TABLE OF CONTENTS

Abstract

The NVIDIA® DGX-1™ (Figure 1) is an integrated system for deep learning. DGX-1 features 8 NVIDIA® Tesla® P100 GPU accelerators connected through NVIDIA® NVLink™, the NVIDIA high-performance GPU interconnect, in a hybrid cube-mesh network. Together with dual socket Intel Xeon CPUs and four 100 Gb InfiniBand network interface cards, DGX-1 provides unprecedented performance for deep learning training. Moreover, the DGX-1 system software and powerful libraries are tuned for scaling deep learning on its network of Tesla P100 GPUs to provide a flexible and scalable platform for the application of deep learning in both production and research settings.

Figure 1 NVIDIA DGX-1

1 INTRODUCTION

Deep learning is quickly changing the field of computer science and having a large impact on the economics of large enterprises such as Google [Metz 2015], Facebook [Statt 2016], and Amazon [Finley 2016]. Even as this new discipline and technology becomes mainstream, it is evolving rapidly. To serve increasingly sophisticated applications and deliver higher quality results, neural networks are becoming exponentially more complex, and deeper. At the same time, neural networks deployed in mainstream interactive applications are being driven to infer and predict results faster.

The demand for deep learning performance is rapidly growing. Facebook CTO Mike Schroepfer recently noted that:

- Facebook's deployed neural networks process more than 6 million predictions per second,
- 25% of Facebook engineers are now using AI and machine learning APIs and infrastructure, and
- Facebook has deployed more than 40 PFLOP/s of GPU capability in house to support deep learning across their organization [Schroepfer 2016:6:54].

In an April 2016 interview, Baidu Chief Scientist Andrew Ng stated that Baidu has adopted an HPC (High Performance Computing) point of view to machine learning:

> *The faster we train our networks, the more iteration we can make on our datasets and models, and the more iterations we make, the more we advance our machine learning*. [Ng 2016a].

According to Ng, training one of Baidu's speech models requires 10 exaflops of computation [Ng 2016b:6:50].

As neural networks get deeper and more complex they provide a dramatic increase in accuracy (for example, Microsoft Deep Residual Networks [He et al. 2015]), but training these higher accuracy networks requires much higher computation time, and their complexity increases prediction latency.

Aimed at satisfying this insatiable need for performance, the NVIDIA DGX-1 (shown in Figure 1) is a new artificial intelligence (AI) supercomputer for deep learning training. NVIDIA's goal with DGX-1 was to create the world's fastest platform for training deep neural networks that can be deployed quickly and simply for plug-and-play use by deep learning researchers and data scientists. The architecture of DGX-1 draws on NVIDIA's experience in the field of high-performance computing and knowledge gained from optimizing deep learning frameworks on NVIDIA GPUs through work with every major cloud service provider and multiple Fortune 1000 companies.

2 NVIDIA DGX-1 SYSTEM ARCHITECTURE

The NVIDIA® DGX-1™ is a deep learning system architected for high throughput and high interconnect bandwidth to maximize neural network training performance. The core of the system is a complex of eight Tesla P100 GPUs connected in a hybrid cube-mesh NVLink network topology, described in Section 3. In addition to the eight GPUs, DGX-1 includes two CPUs for boot, storage management, and deep learning framework coordination. DGX-1 is built into a three-rack-unit (3U) enclosure that provides power, cooling, network, multi-system interconnect, and SSD file system cache, balanced to optimize throughput and deep learning training time.

NVLink is an energy-efficient, high-bandwidth interconnect that enables NVIDIA® Pascal™ architecture GPUs to connect to peer GPUs or other devices within a node at an aggregate bidirectional bandwidth of 160 GB/s per GPU: roughly five times that of current PCIe Gen3 x16 interconnections. The NVLink interconnect and the DGX-1 architecture's hybrid cube-mesh GPU network topology enable the highest bandwidth data interchange between a group of eight Tesla P100 GPUs.

Tesla P100's Page Migration Engine allows high bandwidth, low overhead sharing of data between the GPUs and bulk host memory [NVIDIA Corporation 2016]. For scaling to many-node high-performance clusters, DGX-1 provides high system-to-system bandwidth through InfiniBand (IB) networking (see Section 3.3).

Figure 2 shows a diagram of DGX-1 system components.

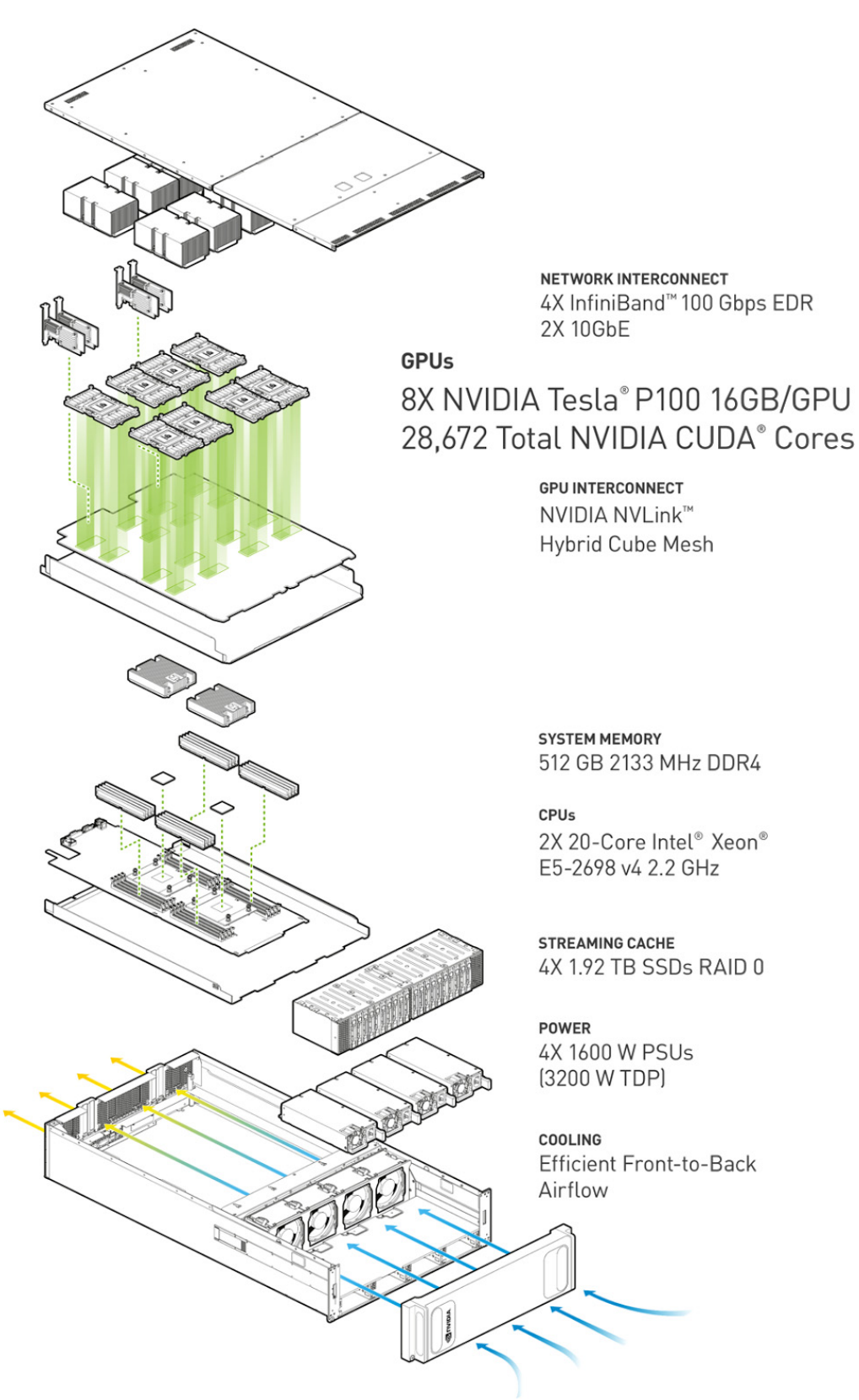

Figure 2 DGX-1 system components

2.1 DGX-1 System Technologies

Tesla P100 (Figure 3) is the latest NVIDIA accelerator, designed for high performance computing and deep learning applications [NVIDIA Corporation 2016]. P100 features the GP100 GPU, which incorporates 56 streaming multiprocessors (SMs), each with:

- 64 single-precision (FP32) NVIDIA® CUDA® cores,
- 32 double-precision (FP64) CUDA cores,
- 256KB register file (RF), and
- 64KB of shared memory.

Tesla P100 peak computational throughput is:

- 5.3 TFLOP/s for FP64 computation,
- 10.6 TFLOP/s for FP32, and
- 21.2 TFLOP/s for FP16 (half-precision) [NVIDIA Corporation 2016]

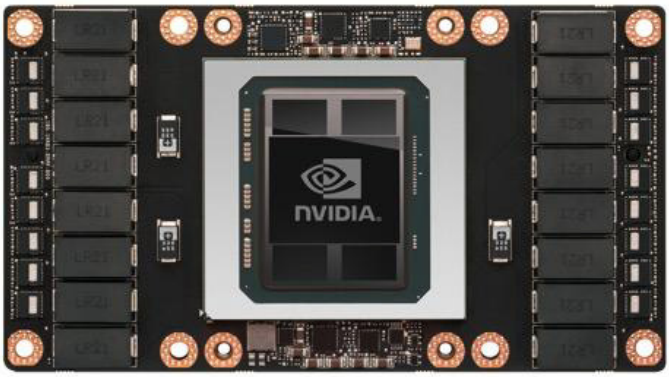

Figure 3 The Tesla P100 Accelerator

To support this high computational throughput, Tesla P100 is the first GPU architecture to incorporate HBM2 (High Bandwidth Memory version 2). P100 includes 16 GB of HBM2 stacked memory with 732 GB/s of bandwidth, significantly higher than the bandwidth of GDDR5 RAM used on previous GPUs. Because HBM2 memory is stacked memory located on the same physical package as the GPU, it provides considerable space savings compared to traditional GDDR5. This space savings enables high-density GPU servers like DGX-1.

Each Tesla P100 has 4 NVLink connections for an aggregate 160 GB/s bidirectional bandwidth. NVLink and the DGX-1 interconnect topology and its implications are discussed in detail in Section 3.

The PCIe links between the GPUs and CPUs enable access to the CPUs' bulk DRAM memory to enable working set and dataset streaming to and from the GPUs. The CPU memory capacity is configured with four times the GPU memory capacity, to enable simplified buffer management and balance for deep learning workloads. While twice the GPU memory footprint would normally be sufficient to manage background data moves and back buffering, four times gives greater flexibility for managing in-memory working sets and streaming data movement. In addition to the 512 GB of system DRAM, each Tesla P100 GPU includes 16 GB of HBM2 co-packaged memory, yielding a total of 128 GB HBM2 memory with net GPU memory bandwidth of 8 × 732 GB/s = 5.86 TB/s.

DGX-1 is provisioned with both ethernet and InfiniBand (IB) network interfaces. Two 10-gigabit Ethernet interfaces provide remote user access into the system and also access to remote storage. To connect multiple DGX-1 systems, each system has four high-bandwidth, low-latency EDR IB (Extended Data Rate

InfiniBand) ports for a total of 400 Gb/s of communication. With 8 P100 GPUs and 4 EDR IB ports, each DGX-1 system has one IB port for every two GPUs. In addition, EDR IB supports NVIDIA GPUDirect Remote Direct Memory Access (RDMA) protocol providing the ability to transfer data directly from the GPU memory in one system to GPU memory in another without involving the CPU or system memory.

Efficient, high-bandwidth streaming of training data is critical to the performance of DGX-1 as a deep learning system, as is reliable, low failure rate storage. Each system comes configured with a single 480 GB boot OS SSD, and four 1.92 TB SAS SSDs (7.6 TB total) configured as a RAID 0 striped volume for performance. For working sets larger than 7 TB, data can be staged and cached through the SSDs in the background during a training run.

In addition to high computational throughput and memory bandwidth, training deep neural networks requires high streaming data rates from disk storage. DGX-1 is designed to support large training datasets that are hundreds of gigabytes to terabytes in size. The four SAS SSDs together provide an aggregate streaming read bandwidth of 2 GB/s, which is sufficient to avoid disk streaming bottlenecks during training.

The thermal design power (TDP)[1] of DGX-1 is 3.2KW, but actual power consumption is dynamic based on workload. For ease of deployment, DGX-1 is designed to be air cooled in most datacenter environments with inlet air from 5°C – 35°C.

3 MULTI-GPU AND MULTI-SYSTEM SCALING WITH NVLINK AND INFINIBAND

Servers with two or more GPUs per CPU are becoming common as developers increasingly expose and leverage the available parallelism in their applications. While dense GPU systems provide a great vehicle for scaling single-node performance, multi-GPU application efficiency can be constrained by the performance of the PCIe (Peripheral Component Interconnect Express) bus connections between GPUs. Similarly, data center applications are growing outside the box, requiring efficient scaling across multiple interconnected systems. To address both of these needs, DGX-1 incorporates the new NVLink high-speed GPU interconnect for multi-GPU scalability within a system, and multiple EDR InfiniBand ports to provide high bandwidth between many connected DGX-1 systems.

Given that communication is an expensive operation, developers must overlap data transfers with computation or carefully orchestrate GPU accesses over PCIe interconnect to maximize performance. As GPUs get faster and GPU-to-CPU ratios climb, a higher-performance GPU interconnect is warranted.

1. TDP specifies the maximum system power used while running real-world applications, and is used to correctly size power and cooling requirements in data centers.

This challenge motivated the creation of the NVLink high-speed interconnect, which enables NVIDIA Pascal GPUs to connect to peer GPUs and/or to NVLink-enabled CPUs or other devices within a node. NVLink supports the GPU ISA, which means that programs running on NVLink-connected GPUs can execute directly on data in the memory of another GPU as well as on local memory. GPUs can also perform atomic memory operations on remote GPU memory addresses, enabling much tighter data sharing and improved application scaling.

NVLink uses NVIDIA's new High-Speed Signaling interconnect (NVHS). NVHS transmits data over a differential pair running at up to 20 Gb/s. Eight of these differential connections form a "Sub-Link" that sends data in one direction, and two sub-links—one for each direction—form a "Link" that connects two processors (GPU-to-GPU or GPU-to-CPU). A single link supports up to 40 GB/s of bidirectional bandwidth between the endpoints. Multiple links can be ganged together for even higher bandwidth between processors. The NVLink implementation in Tesla P100 supports up to four Links, allowing for an aggregate maximum theoretical bandwidth of 160 GB/s bidirectional bandwidth.

3.1 DGX-1 NVLink Network Topology for Efficient Application Scaling

High-performance applications typically scale their computations in one of two ways, known as *strong scaling* and *weak scaling*[2]. Strong scaling measures the improvement in time to solution when increasing the number of parallel processors applied to a fixed total problem size. With perfect strong scaling, the speedup achieved would be equal to the number of processors used.

Weak scaling, on the other hand, measures the improvement in time to solution when increasing the number of parallel processors applied to a problem of fixed size *per processor.* In other words, the problem size is increased along with the number of processors. Here the execution time tends to remain fairly constant as the problem size (and the number of processors) increases. Perfect weak scaling, then, implies that the time to solution did not increase by scaling up the problem linearly with the number of processors.

As individual processors and clusters of processors get ever wider (having ever more parallel processing elements), the benefit of weak scaling for some problems diminishes—eventually these problems may run out of parallelism. It is at this point that these problems are forced into the realm of strong scaling. But in reality, while most applications do exhibit some degree of strong scaling, it is usually not perfectly linear.

A key reason for this is the cost of communication. Strong-scaling a problem onto an increasing number of processors gives each processor progressively less work to do, and increases the relative cost of

2. Note that "weak" is not an inferior form of scaling to "strong" scaling. Both are important metrics in practice.

communicating among those processors. In the strong-scaling regime, fast interconnects and communication primitives tuned for those interconnects are essential.

To provide the highest possible computational density, DGX-1 includes eight NVIDIA Tesla P100 accelerators. Application scaling on this many highly parallel GPUs is hampered by today's PCIe interconnect. NVLink provides the communications performance needed to achieve good (weak and strong) scaling on deep learning and other applications. Each Tesla P100 GPU has four NVLink connection points, each providing a point-to-point connection to another GPU at a peak bandwidth of 20 GB/s. Multiple NVLink connections can be bonded together, multiplying the available interconnection bandwidth between a given pair of GPUs. The result is that NVLink provides a flexible interconnect that can be used to build a variety of network topologies among multiple GPUs. Pascal also supports 16 lanes of PCIe 3.0. In DGX-1, these are used for connecting between the CPUs and GPUs. PCIe is also used for high-speed networking interface cards.

The design of the NVLink network topology for DGX-1 aims to optimize a number of factors, including the bandwidth achievable for a variety of point-to-point and collective communications primitives, the flexibility of the topology, and its performance with a subset of the GPUs. During the design, NVIDIA engineers modeled projected strong and weak scaling of a variety of applications, such as deep learning, sorting, Fast Fourier Transforms (FFT), molecular dynamics, graph analytics, computational fluid dynamics, seismic imaging, ray tracing, and others. This paper focuses on the analysis of scaling of deep learning training.

The hybrid cube-mesh topology (Figure 4) can be thought of as a cube with GPUs at its corners and with all twelve edges connected through NVLink, and with two of the six faces having their diagonals connected as well. It can also be thought of as two interwoven rings of single NVLink connections.

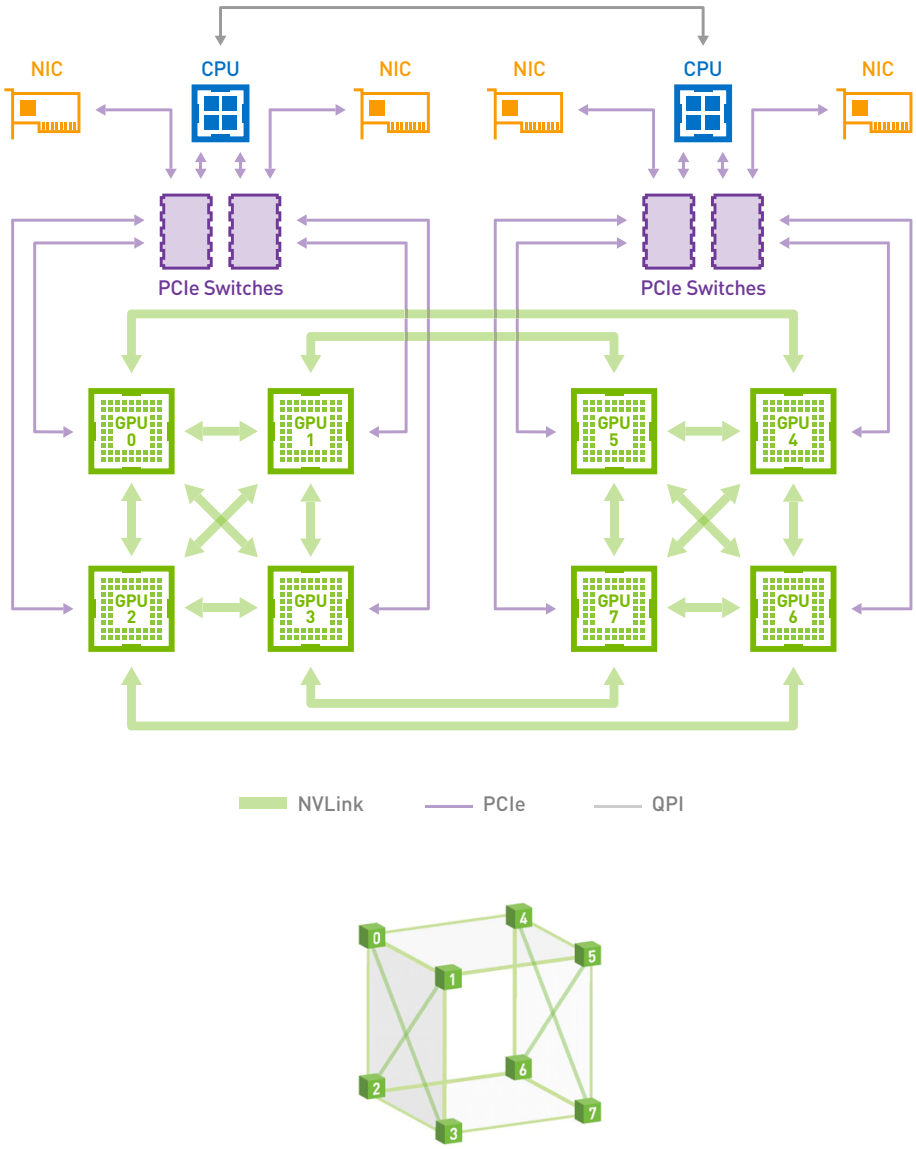

Figure 4 DGX-1 uses an 8-GPU hybrid cube-mesh interconnection network topology. The corners of the mesh-connected faces of the cube are connected to the PCIe tree network, which also connects to the CPUs and NICs.

The cube-mesh topology provides the highest bandwidth of any 8-GPU NVLink topology for multiple collective communications primitives, including broadcast, gather, and all-gather, which are important to deep learning. Using NVLink connections to span the gap between the two clusters of four GPUs relieves pressure on the PCIe bus and on the inter-CPU SMP link, and avoids staging transfers through system memory when transferring across the two clusters.

Furthermore, this 8-GPU cube-mesh topology is easily subdivided into fully connected halves, or into NVLink-connected GPU pairs. This flexibility is important when multiple applications share the server.

3.2 Scaling Deep Learning Training on NVLink

Deep neural networks learn many levels of abstraction, ranging from simple to complex concepts. The strength of deep models is that they are not only powerful but learnable. A deep neural network is trained by feeding it input and letting it compute layer-by-layer to generate output for comparison with a known correct answer. After computing the error at the output, this error flows backward through the net by back-propagation. At each step backward the model parameters are tuned in a direction that tries to reduce the error using a numerical optimization method known as stochastic gradient descent (SGD). This process sweeps over the data improving the model as it goes.

Training deep neural networks in parallel across multiple GPUs and/or multiple nodes requires distributing either:

- the input data ("data parallel"): In data-parallel approaches, separate parallel tasks must periodically resynchronize the gradients with respect to the model that are calculated during back-propagation such that the model parameters are kept in sync across parallel tasks. This amounts to an all-reduce operation.
- the model being trained ("model parallel"): Model-parallel approaches may either elect one parallel task at a time to broadcast its gradients with respect to the input data, or they may use an all-gather of (successive subsets of) the data gradients so that all GPUs' outbound bandwidths are utilized concurrently.
- a hybrid of the two [Wu et al. 2015] [Krizhevsky 2014].

In weak-scaling the training of a deep neural network, the global effective SGD minibatch size increases as the number of GPUs increases. Perhaps unsurprisingly, weak-scaled approaches have high parallel efficiency, even with relatively slow interconnections among GPUs. However, the minibatch size can only be scaled to a certain point before the convergence of the SGD optimization is negatively impacted. The relative tolerance of various networks to increased amounts of weak scaling varies with the network.

To demonstrate scaling of training performance on DGX-1, the bars in Figure 5 and Figure 6 represent training performance in images per second for the ResNet-50 deep neural network architecture using the Microsoft Cognitive Toolkit (CNTK), and the lines represent the parallel speedup of 2, 4, or 8 P100 GPUs versus a single GPU. Figure 5 shows weak scaling performance with a minibatch size of 64 images per GPU, while Figure 6 shows strong scaling performance with a total minibatch size of 64 images (divided among all active GPUs).

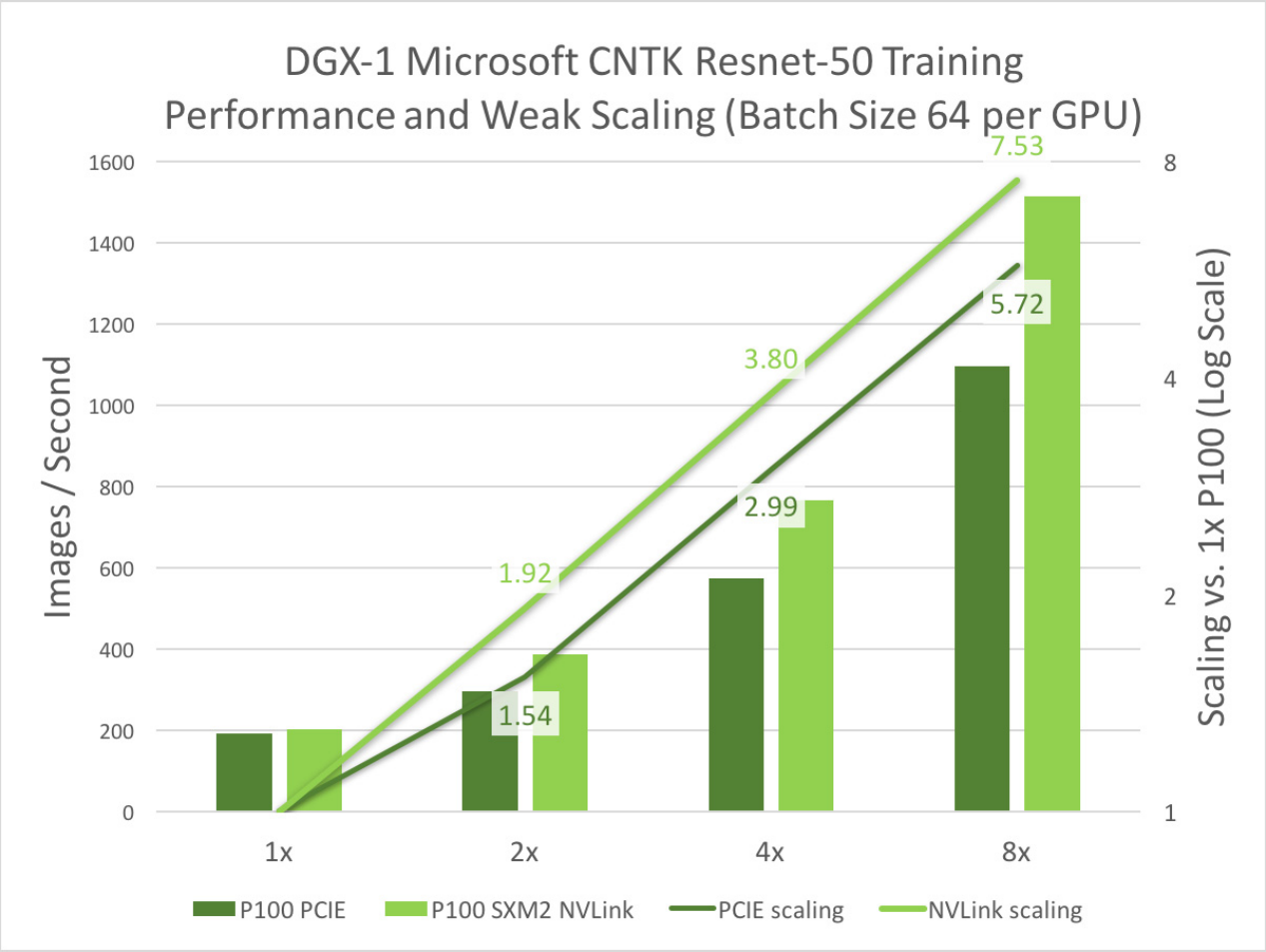

Figure 5 DGX-1 weak scaling results and performance for training the ResNet-50 neural network architecture using the Microsoft Cognitive Toolkit (CNTK) with a batch size of 64 per GPU. The bars present performance on one, two, four, and eight Tesla P100 GPUs in DGX-1 using NVLink for inter-GPU communication (light green) compared to an off-the shelf system with eight Tesla P100 GPUs using PCIe for communication (dark green). The lines present the speedup compared to a single GPU. On eight GPUs, NVLink provides about 1.4x (1513 images/s vs. 1096 images/s) higher training performance than PCIe. Tests used NVIDIA DGX containers version 16.12, processing real data with cuDNN 6.0.5, NCCL 1.6.1, gradbits=32.

The benefits of NVLink show clearly on production DGX-1 hardware when comparing deep learning training using 1, 2, 4 and 8 GPUs on PCIe (tree topology) to the 8-GPU hybrid cube-mesh NVLink interconnect of DGX-1, as Figure 5 shows. NVLink really shines in the 4x and 8x cases, where DGX-1 aggregates multiple NVLink connections in a way that cannot be done with PCIe, achieving nearly 1.4x total speedup vs. PCIe. Not only does the DGX-1 architecture's NVLink interconnect achieve better weak scaling than PCIe, the NVLink hybrid cube-mesh network topology provides the best overall weak scaling for deep learning, compared to alternative NVLink network configurations such as a ring topology.

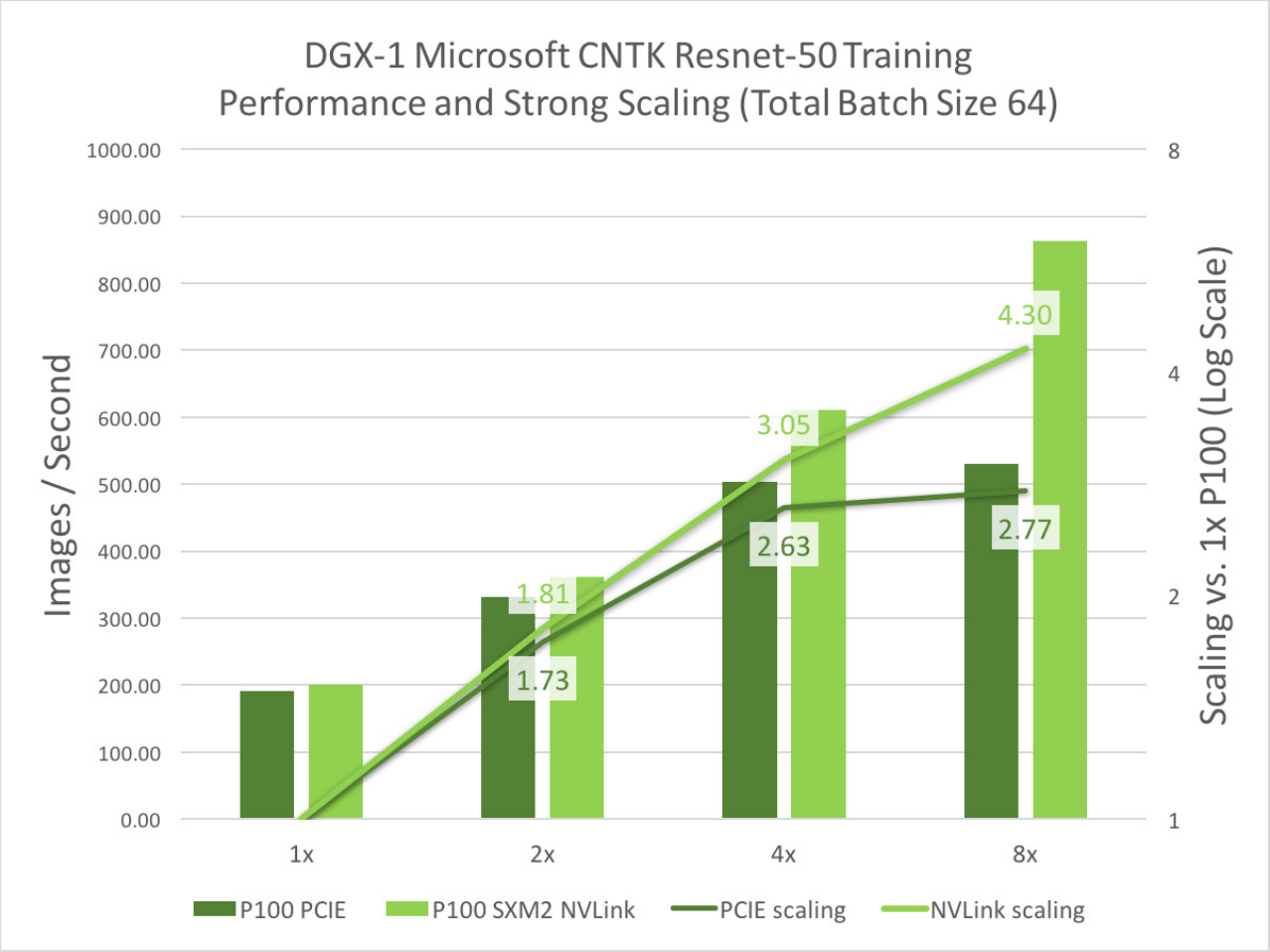

Figure 6 ResNet-50 training performance and strong-scaling speedup on 1, 2, 4, and 8 Tesla P100 GPUs in DGX-1 with NVLink interconnect, compared to scaling on the same number of P100 GPUs on an off-the-shelf system with PCIe interconnect. Training was performed using the Microsoft Cognitive Toolkit (CNTK) with DGX-1 optimizations, with total batch size of 64 images. The bars represent training throughput in images per second, while the lines represent strong-scaling speedup vs. a single P100 GPU. NVLink provides much higher scaling efficiency, resulting in more than 1.6x (863 images/s vs. 531 images/s) higher training performance on DGX-1 vs. a system with 8 PCIe connected P100s. Tests used NVIDIA DGX containers version 16.12, processing real data with cuDNN 6.0.5, NCCL 1.6.1, gradbits=32.

In strong scaling, the global effective minibatch size is fixed regardless of the number of GPUs. The parallel efficiency when strong scaling is much more sensitive to communication bandwidths. In this case, scaling training of ResNet-50 on CNTK with a total batch size of 64 to eight GPUs through PCIe alone suffers significantly, with an 8-GPU-PCIe system achieving a parallel efficiency of less than 35%. In contrast, the DGX-1 architecture's eight-GPU NVLink hybrid cube-mesh topology achieves about 54% efficiency on the same test, as Figure 6 shows.

The DGX-1 architecture's NVLink cube-mesh interconnect topology provides much better strong scaling than PCIe interconnect. With 8 Tesla P100 GPUs, the DGX-1 achieves more than **1.6x higher** ResNet-50 training performance than a system with 8 P100 GPUs connected via PCIe (with tree topology).

3.3 InfiniBand for Multi-System Scaling of DGX-1 Systems

Multi-system scaling of the latest computational workloads, especially deep learning, requires strong communications between GPUs both inside the system and between systems to match the significant GPU performance of each system. In addition to NVLink for high speed communication internally between GPUs, DGX-1 also uses Mellanox ConnectX-4 EDR InfiniBand ports to provide significant bandwidth between systems and to reduce bottlenecks. The latest InfiniBand standard, EDR IB, configured in DGX-1 provides:

- For each port, 8 data lanes operating at 25 Gb/s or 200 Gb/s total (4 lanes in (100 Gb/s) and 4 lanes out (100 Gb/s) simultaneously);
- Low-latency communication and built-in primitives and collectives to accelerate large computations across multiple systems;
- High performance network topology support to enable data transfer between multiple systems simultaneously with minimal contention;
- NVIDIA GPUDirect RDMA[3] across InfiniBand for direct transfers between GPUs in multiple systems.

DGX-1 comes configured with four EDR IB ports providing 800 Gb/s (400 Gb/s in and 400 Gb/s out of the system simultaneously) that can be used to build a high-speed cluster of DGX-1 systems. Four EDR IB ports balance intra- and inter-node bandwidth, and in certain use cases can be fully consumed by inter-node communication. When compared to typical networking technologies such as Ethernet, InfiniBand provides twenty times the bandwidth and four times lower latency even across a large multi-system cluster (see Table 1).

Table 1 Multi-system DGX cluster with InfiniBand provides 20x network performance.

	Typical multi-system cluster	**DGX-1 multi-system cluster**
Number of systems	124	124
Technology	Dual 10 Gb Ethernet	Quad EDR IB

3. NVIDIA GPUDirect Remote Direct Memory Access (RDMA) protocol provides the ability to transfer data directly between GPUs in two different systems across the InfiniBand network without involving the CPU or system memory. This reduces latency and increases performance of multi-systems significantly.

Table 1 Multi-system DGX cluster with InfiniBand provides 20x network performance. (Continued)

	Typical multi-system cluster	DGX-1 multi-system cluster
Single system peak network Bandwidth	40 Gb/s	800 Gb/s
Full Cluster peak Bisection Bandwidth	310 GB/s	6,012 GB/s[a]
System wide latency	5 us	1.28 us[b]

a. InfiniBand full cluster peak bandwidth of 6,012 GB/s computed based on 124 systems with 4 EDR IB ports per system at 200 Gb/s (100 Gb/s in and 100 Gb/s out simultaneously) * 64/66 bit encoding divided by 8 bits/byte to convert to bytes times 0.5 to calculate bi-section bandwidth.
b. Infiniband latency of 1.28 us based on system to system latency of 1.01 us plus 3 switch hops at 0.09 us each.

The latest DGX-1 multi-system clusters use a network based on a fat tree topology providing well-routed, predictable, contention-free communication from each system to every other system (see Figure 7). A fat tree is a tree-structured network topology with systems at the leaves that connect up through multiple switch levels to a central top-level switch. Each level in a fat tree has the same number of links providing equal bandwidth. The fat tree topology ensures the highest communication bisection[4] bandwidth and lowest latency for all-to-all or all-gather type collectives that are common in computational and deep learning applications.

In addition, the internal cube-mesh network is connected to the external InfiniBand network in a way that provides the best performance. The EDR IB ports in a DGX-1 system are configured at the corners of the NVLink hybrid cube-mesh network and allow direct GPU-to-GPU RDMA communication across InfiniBand. Since each 100 Gb IB link has similar bandwidth to a x16 PCIe gen 3 interface, system-to-system interconnect is well-balanced. From an application point of view, GPUDirect RDMA and the unique network design provide the ability for any GPU kernel to directly access any other GPU's memory in the network with minimal overhead, latency and contention.

4. Bisection bandwidth is the total bandwidth available between two halves of a networked cluster. It is determined by splitting the system network down the center and adding the bandwidth of all the links that were split.

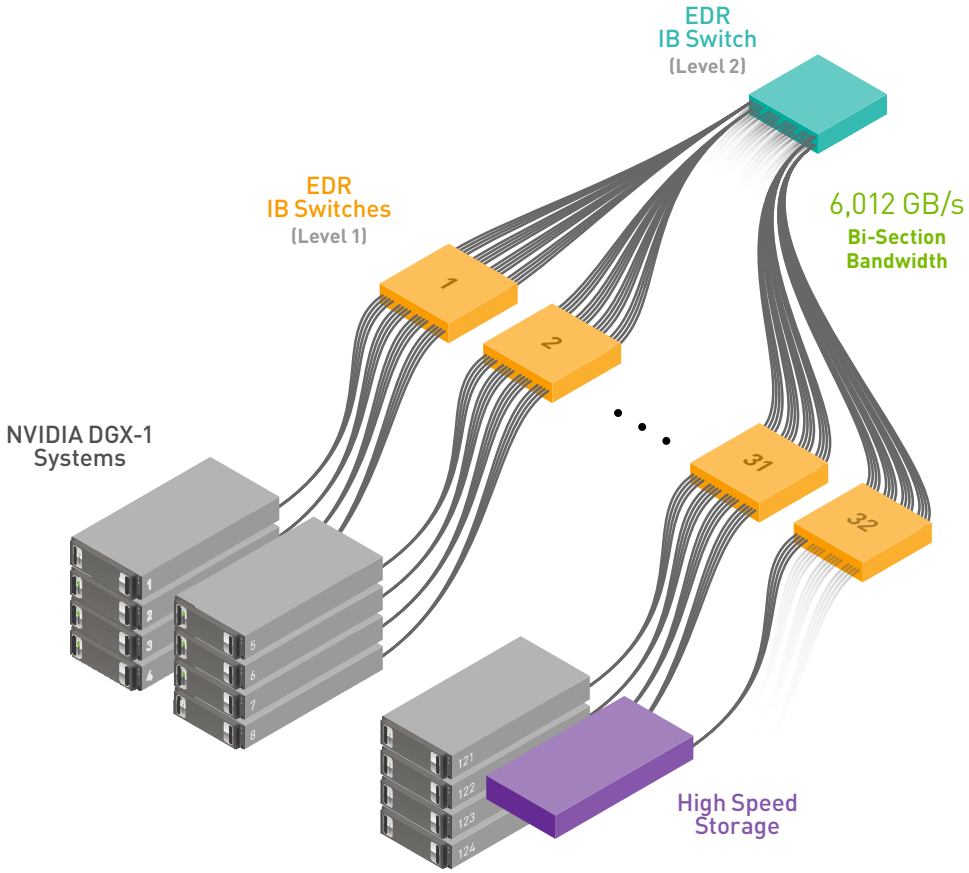

Figure 7 Example Multi-System Cluster of 124 DGX-1 Systems Tuned for Deep Learning

While the reference system architecture shown in Figure 7 is based on 124 systems, the fat tree topology used is extensible to much larger configurations with more switches while still maintaining the high performance characteristics of InfiniBand. With the design in Figure 7, where each first-level switch supports four DGX-1 systems, up to 32 first-level and 16 second-level switches can be configured to support up to a maximum of 128 systems. To grow larger, a third switch level would be added.

4 DGX-1 SOFTWARE

The DGX-1 software has been built to run deep learning at scale. A key goal is to enable practitioners to deploy deep learning frameworks and applications on DGX-1 with minimal setup effort. The design of the platform software is centered around a minimal OS and driver install on the server, and provisioning of all application and SDK software in NVIDIA Docker (see section 4.2) containers through the DGX Container Registry[5], maintained by NVIDIA. Containers available for DGX-1 include multiple optimized deep learning frameworks, the NVIDIA DIGITS deep learning training application, third party accelerated solutions, and the NVIDIA CUDA Toolkit. Figure 8 shows the DGX-1 deep learning software stack.

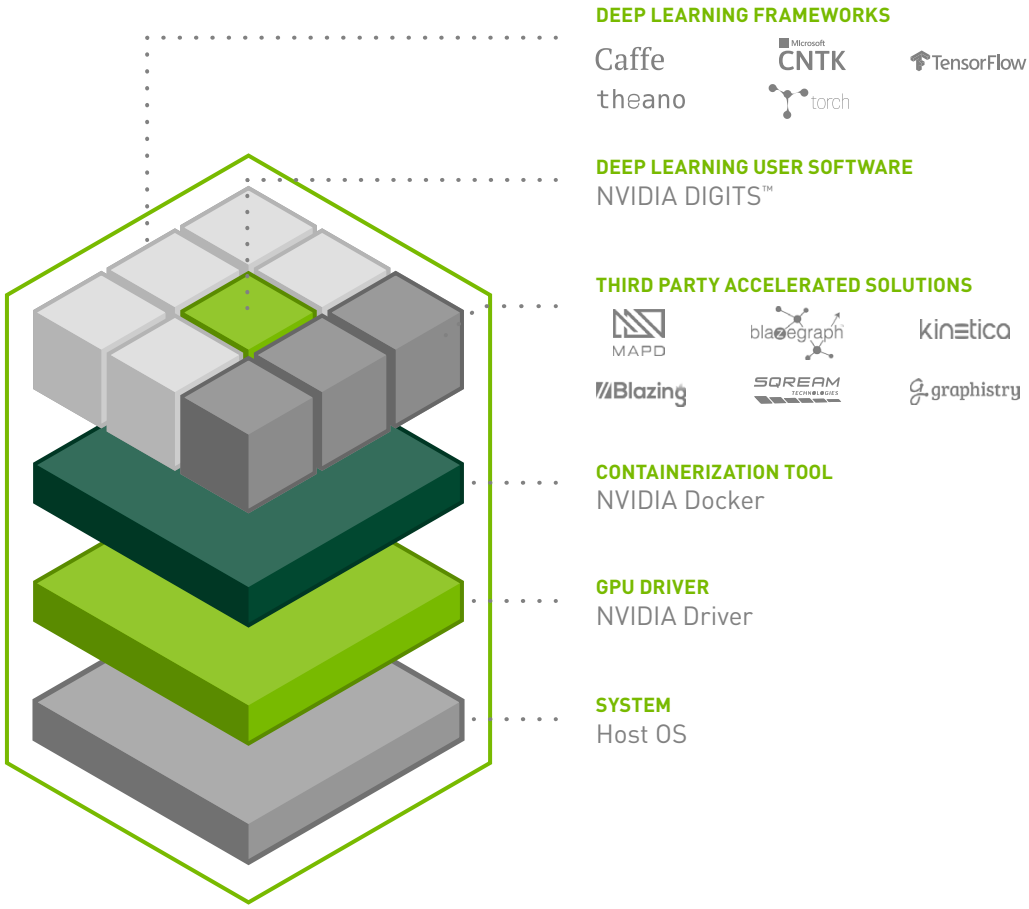

Figure 8 The DGX-1 Deep Learning Software Stack.

5. Docker registry service provided by NVIDIA. See http://docs.nvidia.com/dgx/dgx-registry-guide/

This software architecture has many advantages:

- Since each deep learning framework is in a separate container, each framework can use different versions of libraries like libc, cuDNN, and others, and not interfere with each other.
- As deep learning frameworks are improved for performance or bug fixes, new versions of the containers are made available in the DGX Container Registry.
- The system is easy to maintain, and the OS image stays clean, since applications are not installed directly on the OS.
- Security updates, driver updates and OS patches can be delivered seamlessly.

The deep learning frameworks and the CUDA Toolkit include libraries that have been custom-tuned to provide high multi-GPU performance on DGX-1.

The remainder of this section covers the key components of the DGX-1 software stack (above the GPU Compute Software Driver) in detail. Section 5 provides details of optimizations to deep learning frameworks for DGX-1.

4.1 NVIDIA CUDA Toolkit

CUDA is a parallel computing platform and programming model created by NVIDIA to give application developers access to the massive parallel processing capability of GPUs. CUDA is the foundation for GPU acceleration of deep learning as well as a wide range of other computation- and memory-intensive applications ranging from astronomy, to molecular dynamics simulation, to computational finance. Today there are over 400 GPU-accelerated applications that leverage the CUDA parallel computing platform [NVIDIA 2017b]. DGX-1 is not only the fastest platform for deep learning, but the most advanced CUDA platform for a wide variety of GPU-accelerated applications.

The NVIDIA CUDA Toolkit provides a comprehensive environment for C and C++ developers building GPU-accelerated applications. The CUDA Toolkit includes NVCC, the CUDA C++ compiler for NVIDIA GPUs, a suite of libraries of GPU-accelerated algorithms, debugging and profiling tools, examples, and comprehensive programming guides and documentation. While the CUDA Toolkit can be directly installed on DGX-1, it is also provided as an NVIDIA Docker container image which can be used as the base layer for any containerized CUDA application (as Figure 9 shows). In addition, the full CUDA Toolkit is embedded in every Deep Learning Framework container image.

4.2 NVIDIA Docker

Over the last few years there has been a dramatic rise in the use of software containers for simplifying deployment of data center applications at scale. Containers encapsulate an application's dependencies to provide reproducible and reliable execution of applications and services without the overhead of a full virtual machine.

A Docker container is a mechanism for bundling a Linux application with all of its libraries, configuration files, and environment variables so that the execution environment is always the same, on whatever Linux system it runs and between instances on the same host (see Figure 9). Docker containers are user-mode only, so all kernel calls from the container are handled by the host system kernel. DGX-1 uses Docker containers as the mechanism for deploying deep learning frameworks.

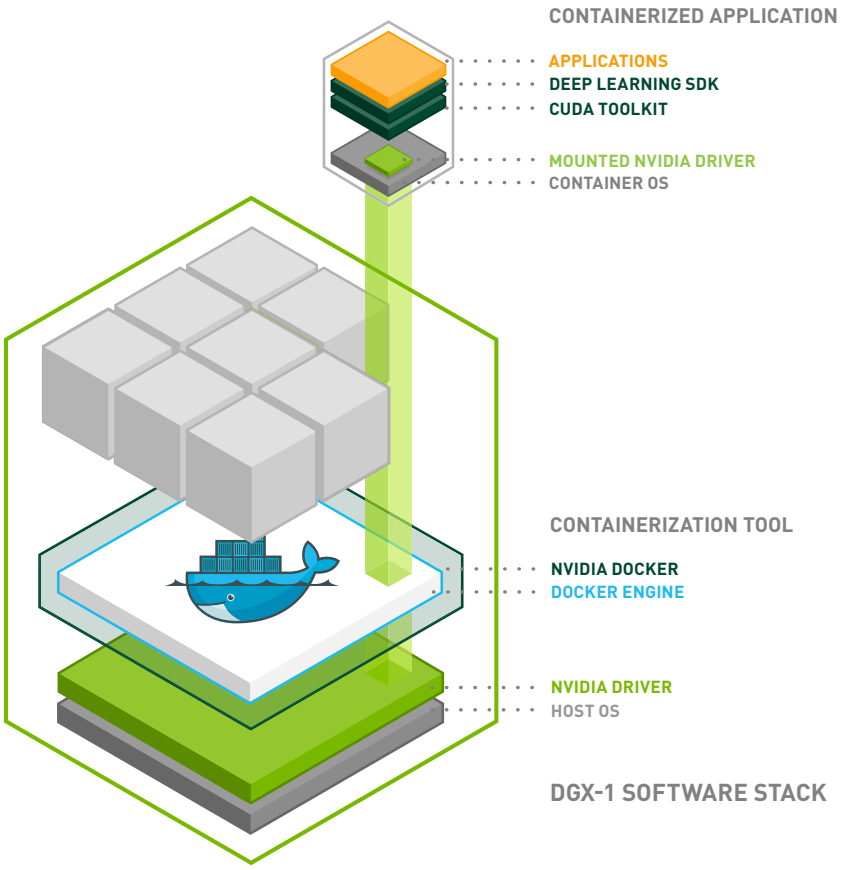

Figure 9 Docker containers encapsulate application dependencies to provide reproducible and reliable execution. NVIDIA Docker mounts the user-mode components of the NVIDIA driver and the GPUs into the Docker container at launch.

Docker containers are platform- and hardware-agnostic, and achieve this with separation of user mode code (in the container) from kernel mode code (accessed on the host). This separation presents a problem when using specialized hardware such as NVIDIA GPUs, since GPU drivers consist of a matched set of user mode and kernel mode modules. An early work-around to this problem was to fully install the NVIDIA drivers inside the container and map in the character devices corresponding to the NVIDIA GPUs (for example, /dev/nvidia0) on launch. This solution is brittle because the version of the host driver must exactly match the version of the driver installed in the container. This requirement drastically reduced the portability of these early containers, undermining one of Docker's more important features.

To enable portability in Docker images that leverage GPUs, NVIDIA developed NVIDIA Docker [NVIDIA Corporation 2015], an open-source project that provides a command-line tool to mount the user mode components of the NVIDIA driver and the GPUs into the Docker container at launch, as Figure 9 shows. For this to work, it is essential that the developer does not install an NVIDIA driver into the Docker image at docker build time.

`nvidia-docker` is essentially a wrapper around Docker that transparently provisions a container with the necessary components to execute code on the GPU.

4.3 NVIDIA Deep Learning SDK

NVIDIA provides a complete suite of GPU-accelerated libraries built on top of the CUDA parallel computing platform. The following two libraries provide GPU-accelerated primitives for deep neural networks:

- the CUDA Basic Linear Algebra Subroutines library (cuBLAS): cuBLAS is a GPU-accelerated version of the complete standard BLAS library that delivers significant speedup running on GPUs. The cuBLAS generalized matrix-matrix multiplication (GEMM) routine is a key computation used in deep neural networks, for example in computing fully connected layers.
- the CUDA Deep Neural Network library (cuDNN): cuDNN provides highly tuned implementations for standard routines such as forward and backward convolution, pooling, normalization, and activation layers.

When deployed using the NVIDIA Docker containers for DGX-1, deep learning frameworks are automatically configured to use parallel routines optimized for the Tesla P100 GPU architecture in DGX-1.

4.4 NCCL

The NVIDIA Collective Communication Library (NCCL, pronounced “Nickel”) is a library of multi-GPU collective communication primitives that are topology-aware and can be easily integrated into applications. Initially developed as an open-source research project[6], NCCL is designed to be light-weight, depending only on common C++ and CUDA libraries. NCCL can be deployed in single-process or multi-process applications, handling required inter-process communication transparently. The NCCL API is designed to be familiar to anyone with experience using MPI collectives such as broadcast, reduce, gather, scatter, all-gather, all-reduce, or all-to-all.

NVIDIA Docker containers for DGX-1 include a version of NCCL that optimizes these collectives for the DGX-1 architecture’s eight-GPU hybrid cube-mesh NVLink network. When deployed using these containers, deep learning frameworks such as NVIDIA Caffe, Torch, CNTK (Microsoft Cognitive Toolkit), and TensorFlow automatically use this version of NCCL when run on multiple GPUs.

There are numerous approaches to implementing collectives efficiently. However, it is critical that our implementation takes the topology of interconnects between processors into account. To illustrate this, consider a broadcast of data from GPU0 to all other GPUs in the PCIe tree topology pictured in Figure 10.

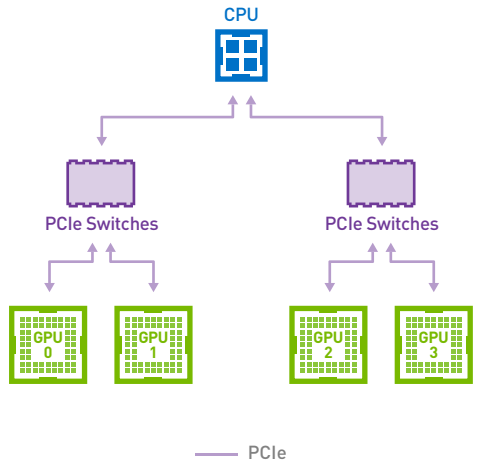

Figure 10 A common PCIe topology for 4 GPUs attached to a single CPU. Purple arrows represent PCIe x16 connections.

A two-step tree algorithm is one approach: in the first step the data is sent from GPU0 to a second GPU, and in the second step both of these send data to the remaining processors. However, there is a choice. Either send data from GPU0 to GPU1 in the first step and then GPU0 to GPU2 and GPU1 to GPU3 in the second, or perform the initial copy from GPU0 to GPU2 and then GPU0 to GPU1 and GPU2 to GPU3 in the second step. Examining the topology, it is clear that the second option is preferred, since sending data simultaneously from GPU0 to GPU2 and GPU1 to GPU3 would cause contention on the upper PCIe links, halving the effective bandwidth for this step. In general, achieving good performance for collectives requires careful attention to the interconnect topology.

6. http://github.com/NVIDIA/nccl

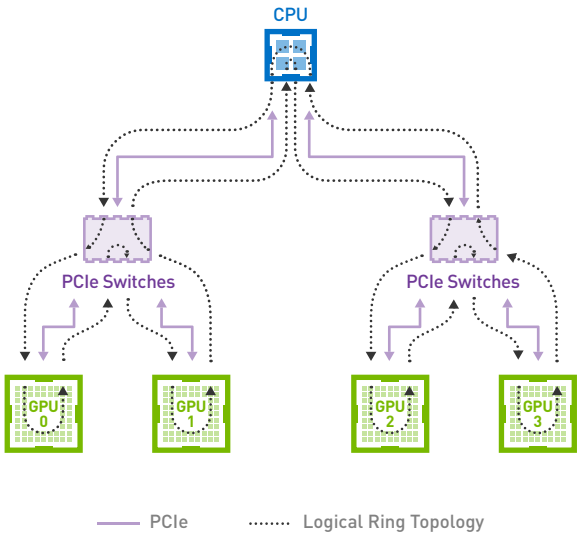

Figure 11 Ring order of GPUs in PCIe tree.

To optimize Broadcast bandwidth, an even better approach is to treat the PCIe tree topology as a ring, as Figure 11 shows. The broadcast is then performed by relaying small chunks of the input around the ring from GPU0 to GPU3. Interestingly, ring algorithms provide near optimal bandwidth for nearly all of the standard collective operations, even when applied to tree-like PCIe topologies. However, note that selecting the correct ring order remains important.

In order to provide maximum bandwidth, NCCL implements ring-style collectives. NCCL implicitly indexes the GPUs into the optimal ring order under the hood. This provides great performance for applications while freeing developers from having to worry about specific hardware configurations.

The 8-GPU hybrid cube-mesh network can be thought of as two interwoven bidirectional rings of single NVLink connections, as Figure 12 shows. Treating the topology this way ensures that the performance of collectives other than all-to-all is largely equivalent.

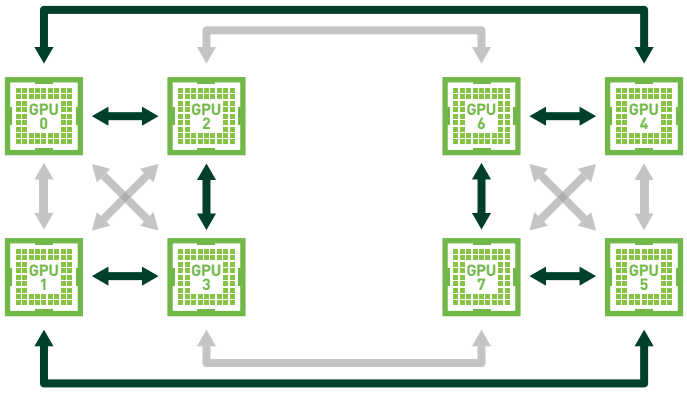

Figure 12 The DGX-1 NVLink hybrid cube-mesh topology can be treated as two interwoven bidirectional rings of single NVLink connections, shown here in grey and dark green.

Inter-GPU transfers for deep learning are performed using these two distinct bidirectional rings. Both rings connect all eight GPUs and together they use all four links of each Pascal GPU in both directions.

With this approach, reduction and broadcast operations can be performed at a speed of more than 60 GB/s, compared to 10 GB/s using PCIe on previous hardware generations. This performance is essential to achieving high scaling for deep learning training.

5 DEEP LEARNING FRAMEWORKS FOR DGX-1

The NVIDIA Deep Learning SDK accelerates widely-used deep learning frameworks such as Caffe, CNTK, MXNet, TensorFlow, Theano, and Torch. The following sections describe each of these frameworks as delivered on DGX-1.

The DGX-1 software stack provides containerized versions of these frameworks optimized for the system. These frameworks, including all necessary dependencies, are pre-built, tested, and ready to run. For users who need more flexibility to build custom deep learning solutions, each framework container image also includes the framework source code to enable custom modifications and enhancements, along with the complete software development stack described in Section 4.

5.1 NVIDIA Caffe

Caffe[7] is a deep learning framework made with flexibility, speed, and modularity in mind. It was originally developed by the Berkeley Vision and Learning Center (BVLC) and by community contributors.

NVIDIA Caffe [NVIDIA Corporation 2017a] is an NVIDIA-maintained fork of BVLC Caffe tuned for NVIDIA GPUs, particularly in multi-GPU configurations. It includes multi-precision support as well as other NVIDIA-enhanced features and offers performance specially tuned for the NVIDIA DGX-1.

The following list summarizes the NVIDIA Caffe optimizations and changes.

- Use of the latest cuDNN release.
- Integration of the latest version of NCCL with NVLink support for improved multi-GPU scaling. NCCL with NVLink boosts the training performance of ResNet-50 by 2x when using data parallel SGD.
- A parallelized parser for improved I/O performance.
- Performance fine-tuning.
- Support for 16-bit floating point (FP16) computation and storage to reduce memory and storage requirements by up to a factor of two.
- Automatic selection of the best convolution algorithm.

7. http://caffe.berkeleyvision.org/

5.2 Microsoft Cognitive Toolkit

The Microsoft Cognitive Toolkit[8] (CNTK), is a unified deep-learning toolkit that allows users to easily realize and combine popular model types such as feed-forward deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs). CNTK implements stochastic gradient descent (SGD) learning with automatic differentiation and parallelization across multiple GPUs and servers [Microsoft Corporation 2017]. CNTK can be called as a library from Python or C++ applications, or executed as a standalone tool using the BrainScript model description language.

NVIDIA and Microsoft worked closely to accelerate the Cognitive Toolkit on GPU-based systems such as DGX-1 and Azure N-Series virtual machines. This combination offers startups and major enterprises alike tremendous ease of use and scalability since a single framework can be used to first train models on premises with the DGX-1 and later deploy those models at scale in the Microsoft Azure cloud[9].

The following list summarizes the DGX-1 CNTK optimizations and changes.

- Use of the latest cuDNN release.
- Integration of the latest version of NCCL with NVLink support for improved multi-GPU scaling. NCCL with NVLink boosts the training performance of ResNet-50 by 2x when using data parallel SGD.
- Image reader pipeline improvements allow AlexNet [Krizhevsky et al. 2012] to train at over 12,000 images/second.
- Reduced GPU memory overhead for multi-GPU training by up to 2 GB per GPU.
- Dilated convolution support.

5.3 MXNet

MXNet[10] is a deep learning framework designed for both efficiency and flexibility, which allows you to mix the symbolic and imperative programming to maximize efficiency and productivity. At the core of MXNet is a dynamic dependency scheduler that automatically parallelizes both symbolic and imperative operations on the fly. A graph optimization layer on top of the scheduler makes symbolic execution fast and memory efficient. MXNet is portable and lightweight, and scales to multiple GPUs and multiple machines.

The following list summarizes the DGX-1 MXNet optimizations and changes.

8. https://www.microsoft.com/en-us/research/product/cognitive-toolkit/
9. For information on using CNTK in Azure see https://github.com/Microsoft/CNTK/wiki/CNTK-on-Azure
10. http://mxnet.io

- Use of the latest cuDNN release.
- Improved input pipeline for image processing.
- Optimized embedding layer CUDA kernels.
- Optimized tensor broadcast and reduction CUDA kernels.

5.4 TensorFlow

TensorFlow[11] is an open-source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) that flow between them. This flexible architecture lets you deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device without rewriting code.

TensorFlow was originally developed by researchers and engineers working on the Google Brain team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research. The system is general enough to be applicable in a wide variety of other domains, as well.

The following list summarizes the DGX-1 TensorFlow optimizations and changes.

- Use of the latest cuDNN release.
- Replacement of libjpeg with libjpeg-turbo.
- Integration of the latest version of NCCL with NVLink support for improved multi-GPU scaling. NCCL with NVLink boosts the training performance of ResNet-50 by 2x when using data parallel SGD.
- Support for the ImageNet preprocessing script.

11. https://www.tensorflow.org

5.5 Theano

Theano[12] is a Python library that allows you to efficiently define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays. Theano has been powering large-scale computationally intensive scientific investigations since 2007.

The following list summarizes the DGX-1 Theano optimizations and changes:

- Use of the latest cuDNN release.
- Runtime code generation: evaluate expressions faster.
- Extensive unit-testing and self-verification: detect and diagnose many types of errors.

5.6 Torch

Torch[13] is a scientific computing framework with wide support for deep learning algorithms. Torch is easy to use and efficient, thanks to an easy and fast scripting language, Lua, and an underlying C/CUDA implementation. Torch offers popular neural network and optimization libraries that are easy to use yet provide maximum flexibility to build complex neural network topologies.

The following list summarizes the DGX-1 Torch optimizations and changes.

- Use of the latest cuDNN release.
- Integration on the latest version of NCCL with NVLink support for improved multi-GPU scaling. NCCL with NVLink boosts the training performance of ResNet-50 by 2x when using data parallel SGD.
- Buffering of parameters to be communicated by NCCL to reduce latency overhead.
- cuDNN bindings for recurrents networks (RNN, GRU, LSTM), including persistent versions, which greatly improving the performance of small batch training.
- Dilated convolution support.
- Support for 16- and 32-bit floating point (FP16 and FP32) data input to cuDNN routines.
- Support for operations on FP16 tensors (using FP32 arithmetic).

12. http://deeplearning.net/software/theano/
13. http://torch.ch

5.7 DIGITS

The NVIDIA Deep Learning GPU Training System (DIGITS)[14] puts the power of deep learning into the hands of engineers and data scientists.

DIGITS can be used to rapidly train highly accurate deep neural network (DNNs) for image classification, segmentation and object detection tasks. DIGITS simplifies common deep learning tasks such as managing data, designing and training neural networks on multi-GPU systems, monitoring performance in real time with advanced visualizations, and selecting the best performing model from the results browser for deployment. DIGITS is completely interactive so that data scientists can focus on designing and training networks rather than programming and debugging.

The following list summarizes the DGX-1 DIGITS optimizations and changes.

- DIGITS runs on top of Caffe and Torch frameworks which are optimized for DGX-1.

14. https://developer.nvidia.com/digits

6 RESULTS: DGX-1 FOR HIGHEST DEEP LEARNING PERFORMANCE

The performance of DGX-1 for training popular deep neural networks speaks volumes about the value of an integrated system for deep learning. The graph in Figure 13 shows the training speedup of DGX-1 compared to an off-the-shelf system with the same GPUs for the ResNet-50 and ResNet-152 deep neural networks using the Microsoft Cognitive Toolkit, TensorFlow, and Torch. This graph demonstrates two clear benefits:

1. The P100 GPUs in DGX-1 achieve much higher throughput than the previous-generation NVIDIA Tesla M40 GPUs for deep learning training.
2. DGX-1 achieves significantly higher performance than a comparable system with eight Tesla P100 GPUs interconnected using PCIe.

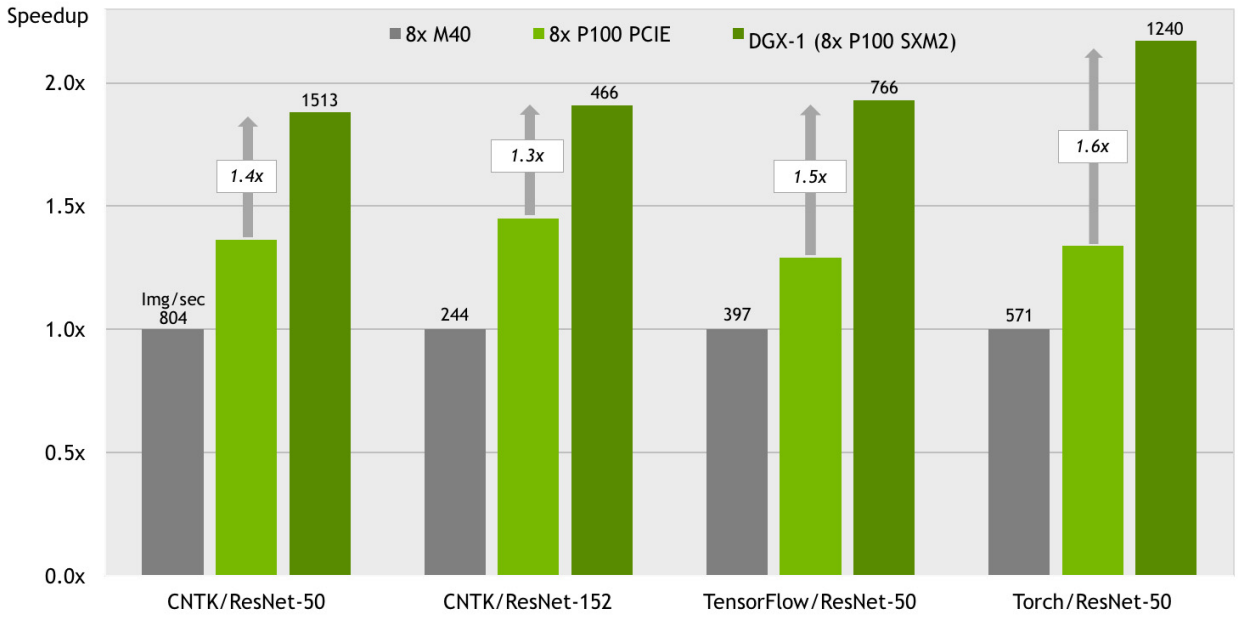

Figure 13 DGX-1 deep learning training speedup using all 8 Tesla P100s of DGX-1 vs. 8-GPU Tesla M40 and Tesla P100 systems using PCI-e interconnect for the ResNet-50 and Resnet-152 deep neural network architecture on the popular CNTK (2.0 Beta5), TensorFlow (0.12-dev), and Torch (11-08-16) deep learning frameworks. Training used 32-bit floating point arithmetic and total batch size 512 for ResNet-50 and 128 for ResNet-152. Other software: NCCL 1.6.1, CUDA 8.0.54, cuDNN 6.0.5, Ubuntu 14.04. NVIDIA Linux display driver 375.30. The 8x M40 and 8x P100 PCIe server is an SMC 4028GR with dual Intel Xeon E5-2698v4 CPUs and 256GB DDR4-2133 RAM (DGX-1 has 512GB DDR4-2133). Tests used NVIDIA DGX containers version 16.12.

The high performance of DGX-1 is due in part to the NVLink hybrid cube-mesh interconnect between its eight Tesla P100 GPUs, but that is not the whole story. Much of the performance benefit of DGX-1 comes from the fact that it is an integrated system, with a complete software platform aimed at deep learning, as described in Sections 4 and 5. This includes the deep learning framework optimizations such as those

in NVIDIA Caffe, cuBLAS, cuDNN, and other GPU-accelerated libraries, and NVLink-tuned collective communications through NCCL. This integrated software platform, combined with Tesla P100 and NVLink, ensures that DGX-1 outperforms similar off-the-shelf systems.

To learn more about NVIDIA DGX-1, visit http://www.nvidia.com/dgx1.

References

Finley, K. 2016. Amazon's Giving Away the AI Behind Its Product Recommendations http://www.wired.com/2016/05/amazons-giving-away-ai-behind-product-recommendations/

He, K., Zhang, X., Ren, S., and Sun, J. 2015. Deep Residual Learning for Image Recognition. *arXiv [cs.CV]*. http://arxiv.org/abs/1512.03385.

Krizhevsky, A. 2014. One weird trick for parallelizing convolutional neural networks. *arXiv [cs.NE]*. http://arxiv.org/abs/1404.5997.

Krizhevsky, A., Sutskever, I., and Hinton, G.E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In: F. Pereira, C.J.C. Burges, L. Bottou and K.Q. Weinberger, eds., *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Metz, C. 2015. TensorFlow, Google's Open Source AI, Signals Big Changes in Hardware Too. http://www.wired.com/2015/11/googles-open-source-ai-tensorflow-signals-fast-changing-hardware-world/

Microsoft Corporation. 2017. The Microsoft Cognitive Toolkit (CNTK). *Github*. https://github.com/Microsoft/CNTK#what-is-the-microsoft-cognitive-toolkit.

Ng, A. 2016a. Baidu's Chief Scientist on Intersection of Supercomputing, Machine Learning. http://www.nextplatform.com/2016/04/01/baidus-chief-scientist-intersection-supercomputing-machine-learning/.

Ng, A. 2016b. AI: The New Electricity. https://www.youtube.com/watch?v=4eJhcxfYR4I.

NVIDIA Corporation. 2015. nvidia-docker software. *Github*. https://github.com/NVIDIA/nvidia-docker.

NVIDIA Corporation. 2016. NVIDIA® Tesla® P100 – The Most Advanced Data Center Accelerator Ever Built. Featuring Pascal P100, the World's Fastest GPU. http://www.nvidia.com/object/pascal-architecture-whitepaper.html

NVIDIA Corporation. 2017a. *NVIDIA Caffe branch.* https://github.com/NVIDIA/caffe

NVIDIA Corporation. 2017b. *GPU-Accelerated Applications*. http://www.nvidia.com/content/gpu-applications/PDF/gpu-applications-catalog.pdf

Schroepfer, M. 2016. F8 2016 Day 1 Keynote. https://developers.facebook.com/videos/f8-2016/keynote/.

Statt, N. 2016. Exploring Facebook's massive, picture-painting AI brain. http://www.theverge.com/2016/7/13/12172904/facebook-ai-big-sur-machine-learning-prineville-data-center

Wu, R., Yan, S., Shan, Y., Dang, Q., and Sun, G. 2015. Deep Image: Scaling up Image Recognition. *arXiv [cs.CV]*. http://arxiv.org/abs/1501.02876.

Notice

ALL INFORMATION PROVIDED IN THIS WHITE PAPER, INCLUDING COMMENTARY, OPINION, NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and other changes to this specification, at any time and/or to discontinue any product or service without notice. Customer should obtain the latest relevant specification before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer. NVIDIA hereby expressly objects to applying any customer general terms and conditions with regard to the purchase of the NVIDIA product referenced in this specification.

NVIDIA products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on these specifications will be suitable for any specified use without further testing or modification. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to ensure the product is suitable and fit for the application planned by customer and to do the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this specification. NVIDIA does not accept any liability related to any default, damage, costs or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this specification, or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this specification. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA. Reproduction of information in this specification is permissible only if reproduction is approved by NVIDIA in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the NVIDIA terms and conditions of sale for the product

Trademarks

NVIDIA, the NVIDIA logo, CUDA, Pascal, Tesla, NVLink, and DGX-1 are trademarks or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright