S8688 : INSIDE DGX-2

Glenn Dearth, Vyas Venkataraman

Mar 28, 2018

NVIDIA

Agenda

Why was DGX-2 created

DGX-2 internal architecture

Software programming model

Simple application

Results

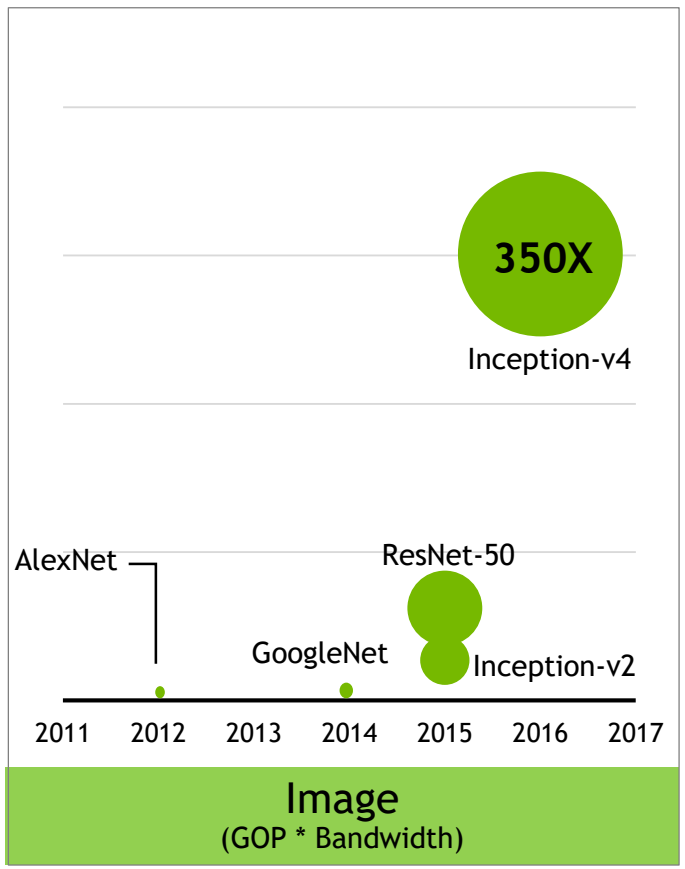
DEEP LEARNING TRENDS

Application properties

Explosive DL growth:

- Increasing data, computation & complexity demands
- Exceeds memory capacity of single GPU
- Exceeds compute performance of a single GPU

Driving scale-out across GPUs



DGX-1

8 V100 GPUs

- 6 NVLinks per GPU
- Each link is 50GB/s (bidirectional)
- 300GB/s bidirectional BW from GPU

DGX-1 uses Hybrid Cube Mesh topology

Internal bisection bandwidth 300GB/s

Optimized data parallel training with NCCL

DESIRED SCALE-OUT BEHAVIOR

Scale up to 16 GPUs

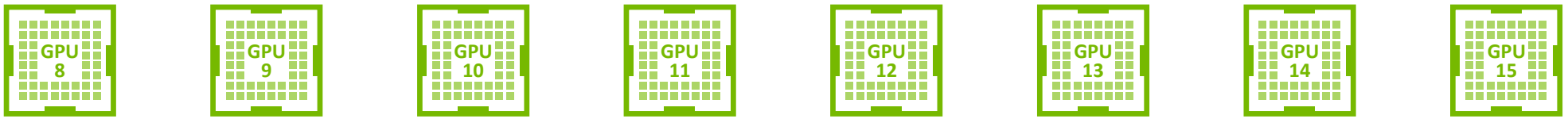
Direct peer GPU memory access

Full non-blocking bandwidth

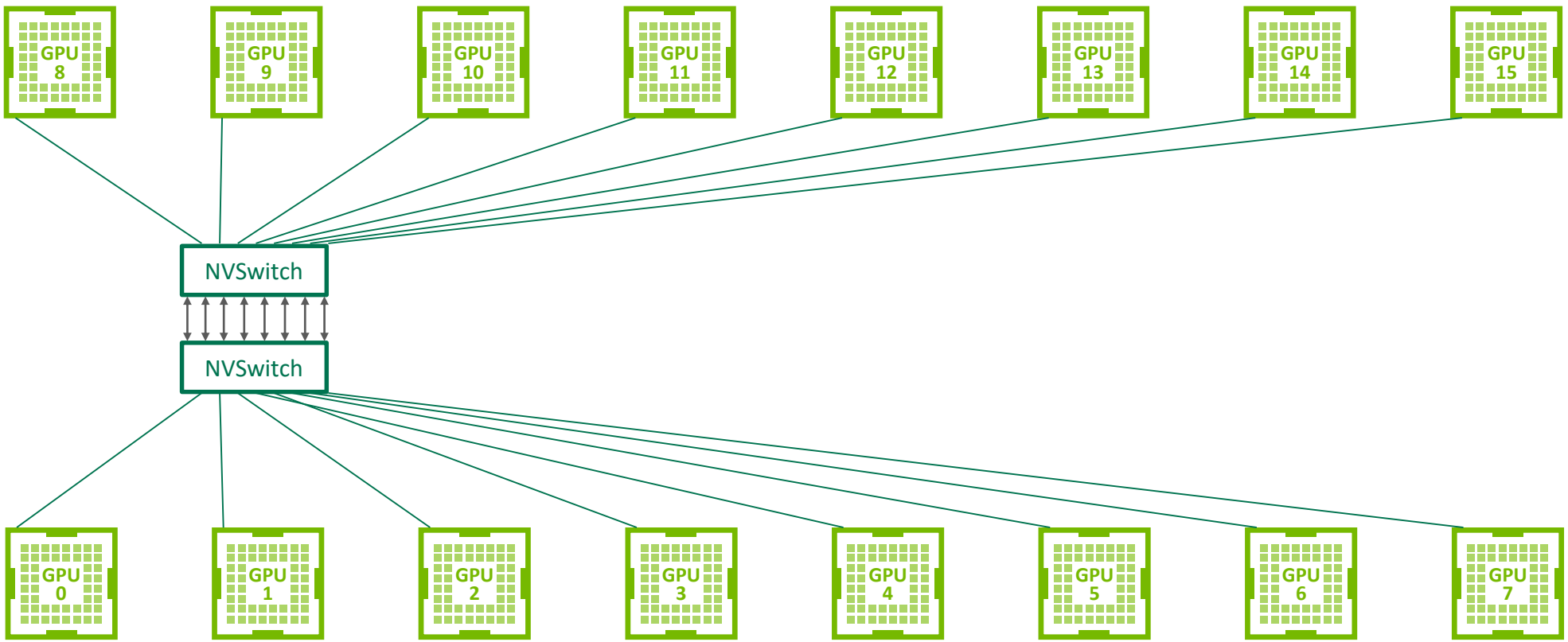
Utilize all GPU links when accessing memory

Simplify multi-GPU programming

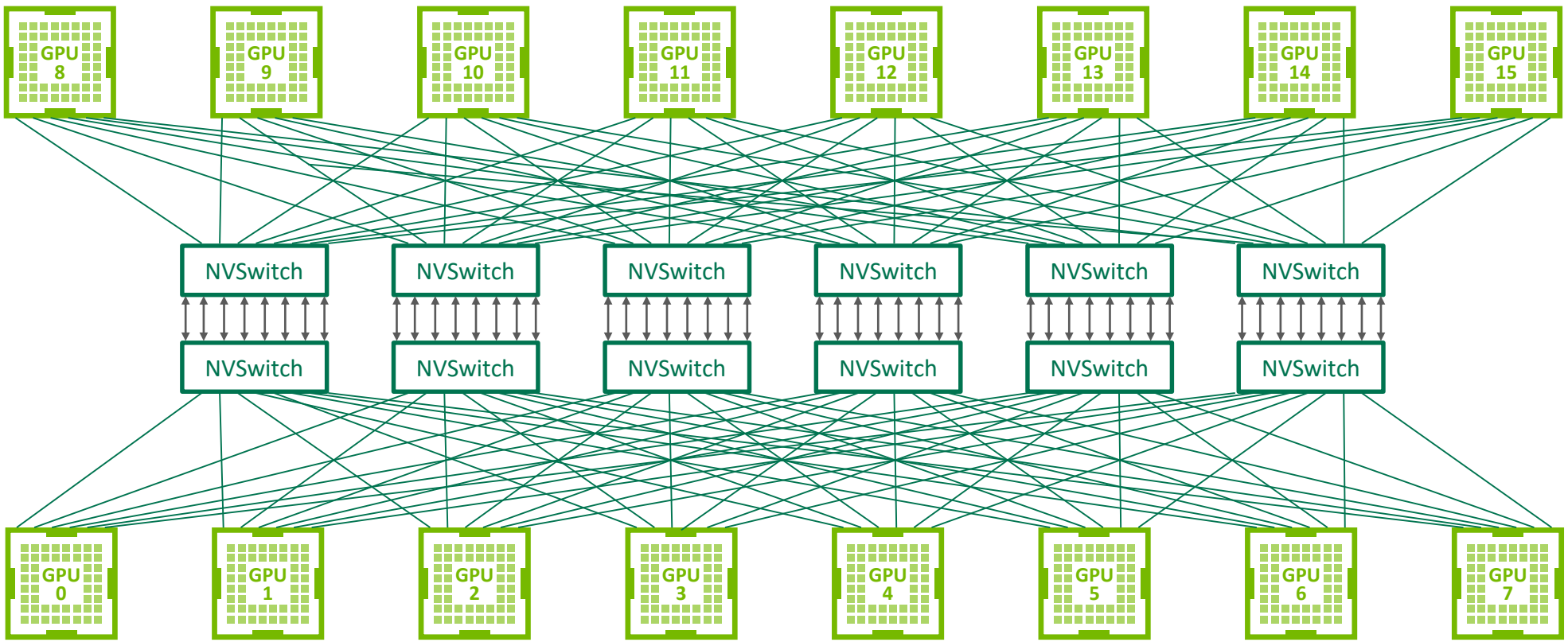
SCALE UP TO 16 GPUS



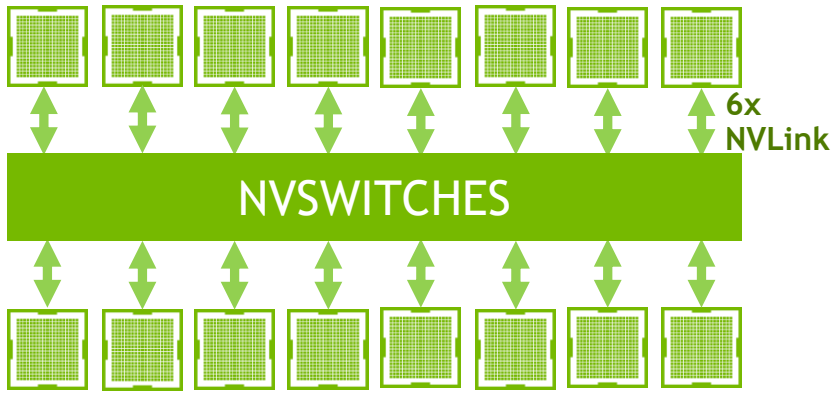
DIRECT PEER MEMORY ACCESS



FULL NON-BLOCKING BANDWIDTH



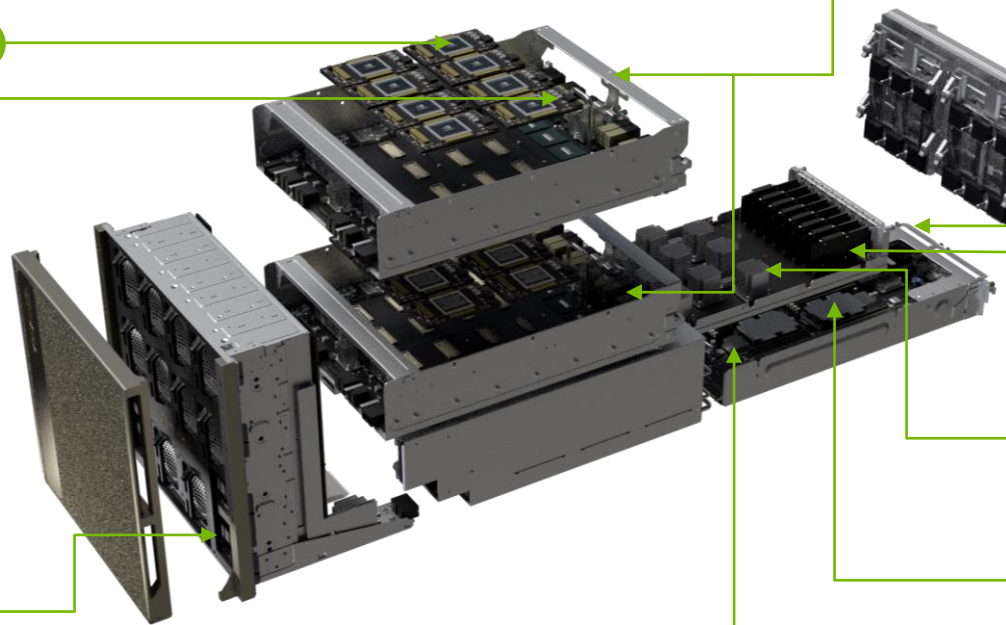
DGX-2 AT A GLANCE



	DGX-2
GPU Density	16
1GPU to 1GPU	Always 6 NVLink
Connectivity	Fully Connected
Topology	Symmetric
Bisection Bandwidth	2.4 TB/s

DESIGNED TO TRAIN THE PREVIOUSLY IMPOSSIBLE

Introducing NVIDIA DGX-2



1. **NVIDIA Tesla V100 32GB**
2. **Two GPU Boards**
   8 V100 32GB GPUs per board
   6 NVSwitches per board
   512GB Total HBM2 Memory
   interconnected by
   Plane Card
3. **Twelve NVSwitches**
   2.4 TB/sec bi-section bandwidth
4. **Eight EDR Infiniband/100 GigE**
   1600 Gb/sec Total
   Bi-directional Bandwidth
5. **PCIe Switch Complex**
6. **Two Intel Xeon Platinum CPUs**
7. **1.5 TB System Memory**
8. **30 TB NVME SSDs**
   Internal Storage
9. **Dual 10/25 GigE**

NVSWITCH

Features:

- 18 NVLink ports
  - @ 50GB/s per port
  - 900 GBs total
- Fully connected crossbar
- x4 PCIe Gen2 Management Port
- GPIO
- I2C

Transistor count:

- 2 billion

Package:

- 47.5 x 47.5mm
- 1937 Ball @ 1mm pitch

SWITCH FUNCTIONS

NVLink Performs physical, datalink & transaction layer functions

Forwarding Determines packet routing

Crossbar (non-blocking) Schedules traffic flows to outputs

Management Configuration, errors, monitors

NVSWITCH RELIABILITY FEATURES

Link CRC and retry

ECC on routing structures and data path

Secondary checks:

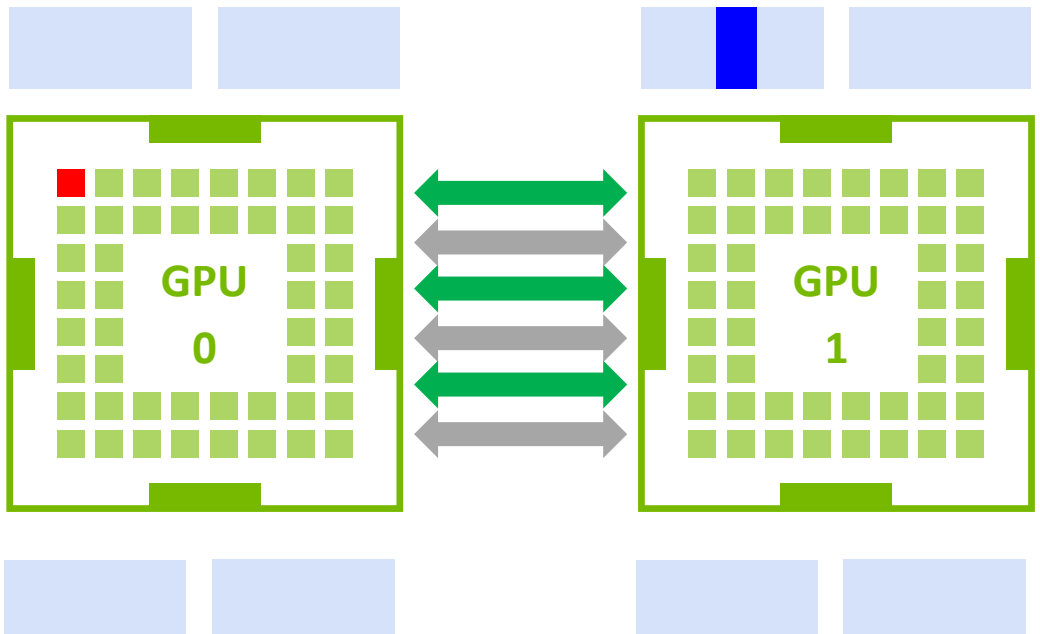
- Routing checks
- Data path overflow/underflow checks

Access control checks

NVIDIA.

Programming Model

MULTI GPU PROGRAMMING IN CUDA



EXECUTION CONTROL

Asynchronous CUDA calls execute in a CUDA stream

- Default to null stream
- Can specify stream explicitly

CUDA runtime API calls have implicit current device selected

Current device can be changed using `cudaSetDevice()` call

Cooperative groups have a multi device launch `cudaLaunchCooperativeKernelMultiDevice()`

CUDA ON DGX-2

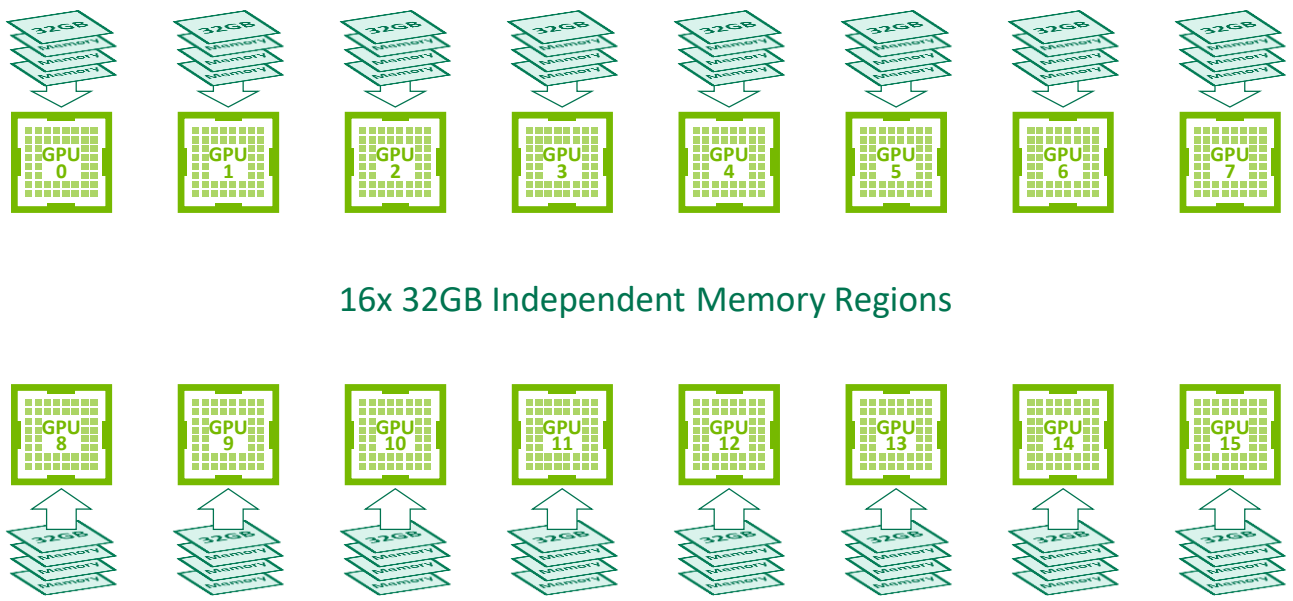
DGX-2 enables up to 16 peer GPUs

DGX-2 enables full NVLink bandwidth to peer GPUs

GPU memory model extended to all GPUs

Unified Memory and CUDA aware MPI use NVLink for transfers

MEMORY MANAGMENT



NVLINK PROVIDES

All-to-all high-bandwidth peer mapping between GPUs

Full inter-GPU memory interconnect (incl. Atomics)

PINNED MEMORY ALLOCATION

Enable peer memory access

```
// Enable Peer accesses between all pairs of GPUs
for (int i = 0; i < numDevices; ++i)
  for (int j = 0; j < numDevices; ++j)
    if (i != j) {
       cudaEnablePeerAccess(i, j);
     }
```

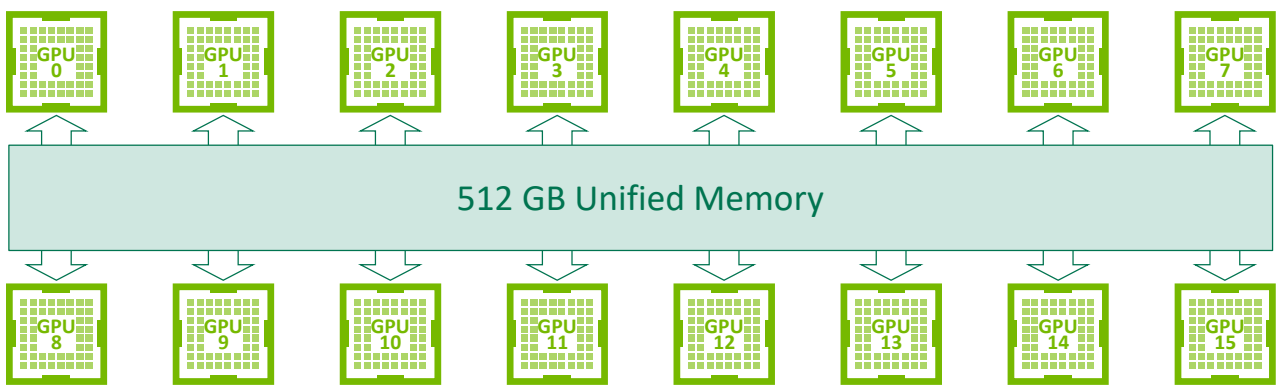
PINNED MEMORY ALLOCATION

cudaMalloc with CUDA P2P

```
int* ptr[MAX_DEVICES];

for (int i = 0; i<numDevices; ++i) {
  // Set a device
  cudaSetDevice(i);
  // Allocate memory on the device
  cudaMalloc((void**)&ptr[i], size);
}
```

UNIFIED MEMORY + DGX-2



UNIFIED MEMORY PROVIDES

Single memory view shared by all GPUs

Automatic migration of data between GPUs

User control of data locality

UNIFIED MEMORY

Allocating across multiple GPUs

```
int* ptr;
// Allocate memory
cudaMallocManaged((void**)&ptr, size * numDevices);
```

UNIFIED MEMORY

Allocating across multiple GPUs

```
int* ptr;
// Allocate memory
cudaMallocManaged((void**)&ptr, size * numDevices);

for (int i = 0; i < numDevices; ++i) {
// Mark the memory as preferring a specific GPU
  cudaMemAdvise(ptr + i*size, size, cudaMemAdviseSetPreferredHome, i);
  // Mark this memory accessed by all devices
  for (int j = 0; j < numDevices; ++j)  {
    cudaMemAdvise(ptr + i*size, size, cudaMemAdviseSetAccessedBy, j);
  }
}
```

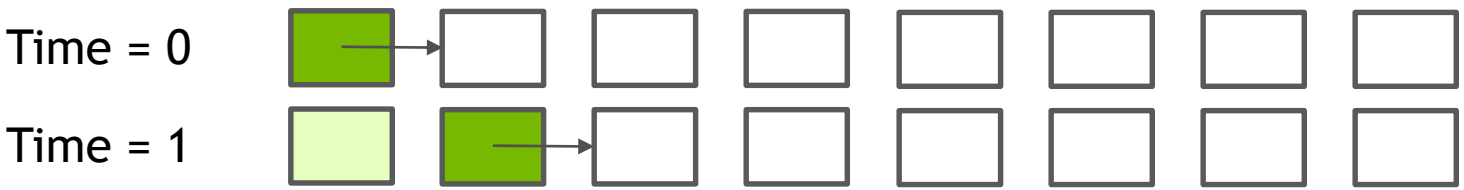
BROADCAST ON DGX-1

Ring Scatter

Time = 0

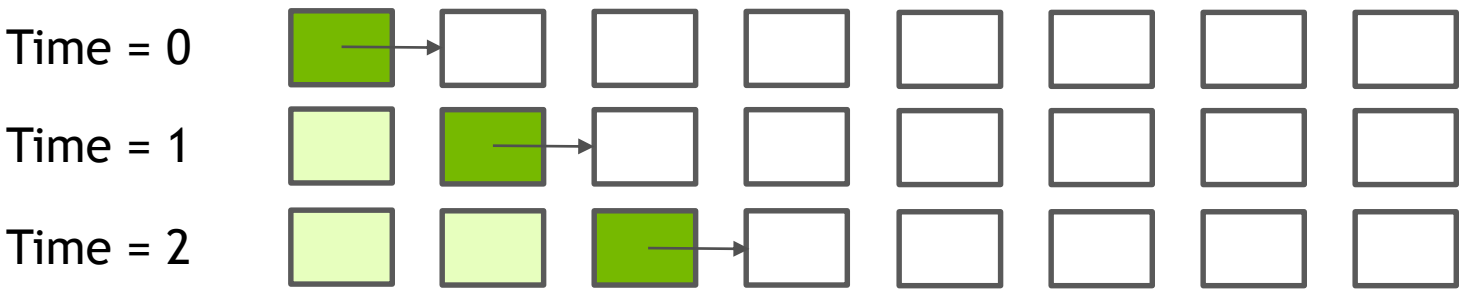
BROADCAST ON DGX-1

Ring Scatter



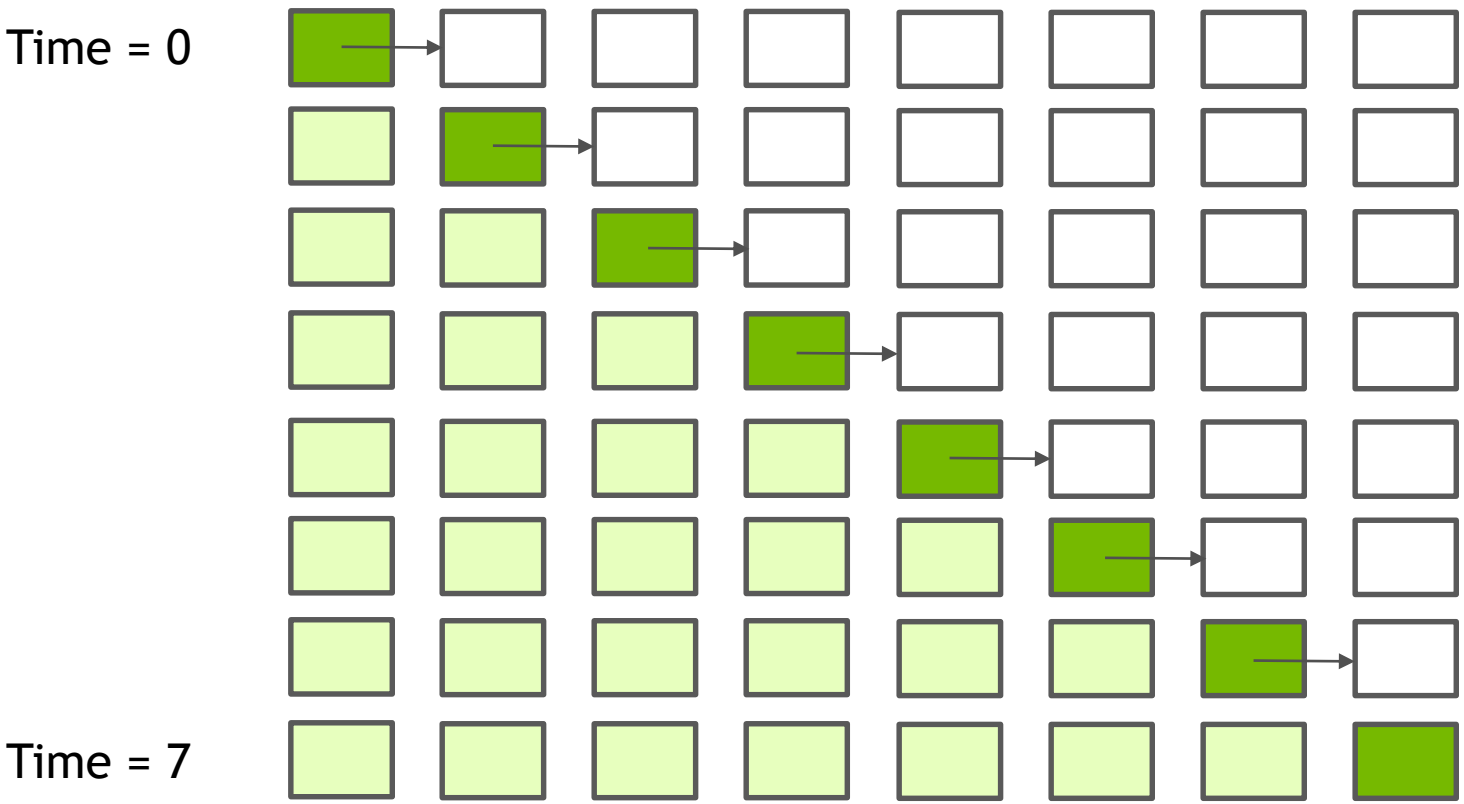
BROADCAST ON DGX-1

Ring Scatter



BROADCAST ON DGX-1

Ring Scatter



BROADCAST ON DGX-2

Direct Broadcast (DGX-2)

Time = 0

IMPLEMENTATION COMPARISON

Ring Scatter (DGX-1)

```
__global__ void broadcast_ring(int *src, int *dst)
{
       int index  = blockIdx.x*gridDim.x + threadIdx.x;
       dst[index] = src[index];
}
// CPU Code
cudaEvent_t ev[MAX_DEVICES];
For (int i = 0; i < numDevices - 1; i++) {
  cudaEventCreate(&ev[i]);
  cudaSetDevice(i);
  if (i > 0)
     cudaStreamEventWait(NULL, ev[i-1], 0);
  broadcast_ring<<<blocks, threads >>>(ptr[i], ptr[i+1]);
  cudaEventRecord(ev[i])
}
cudaSetDevice(0);
cudaStreamWaitEvent(NULL, ev[numDevices – 2], 0);
cudaDeviceSynchronize();
```

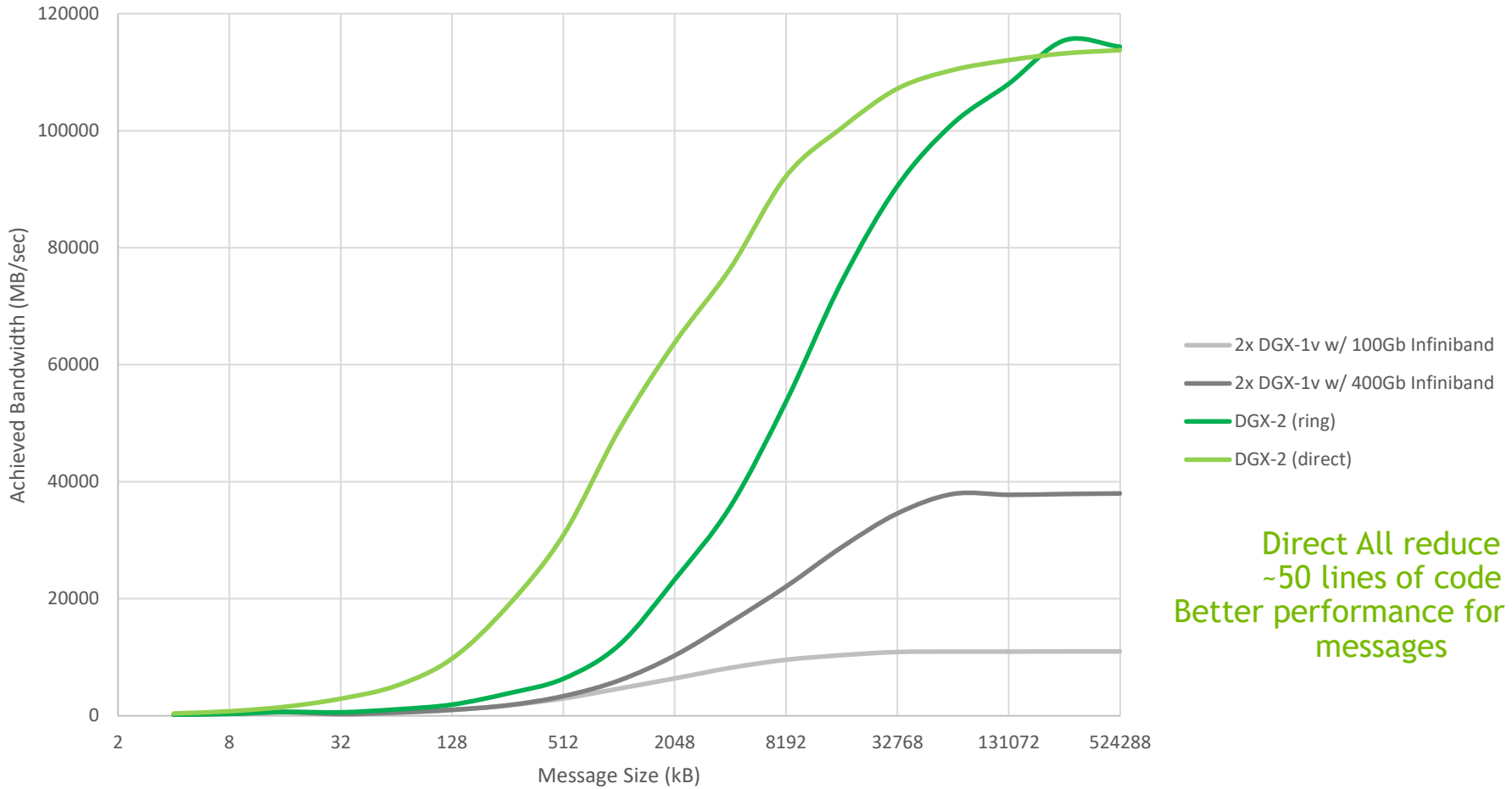
Direct broadcast (DGX-2)

```
__global__ void broadcast_direct(int *src, int **pDst, int
numDevices)
{
       int index  = blockIdx.x*gridDim.x + threadIdx.x;
       for (int i = 0; i < numDevices; ++i) {
              int *dst = pDst[i];
              dst[index] = src[index];
       }
}



// CPU code
cudaSetDevice(0);
broadcast_direct<<<blocks, threads>>>(ptr[0], dPtr);
cudaDeviceSynchronize();
```

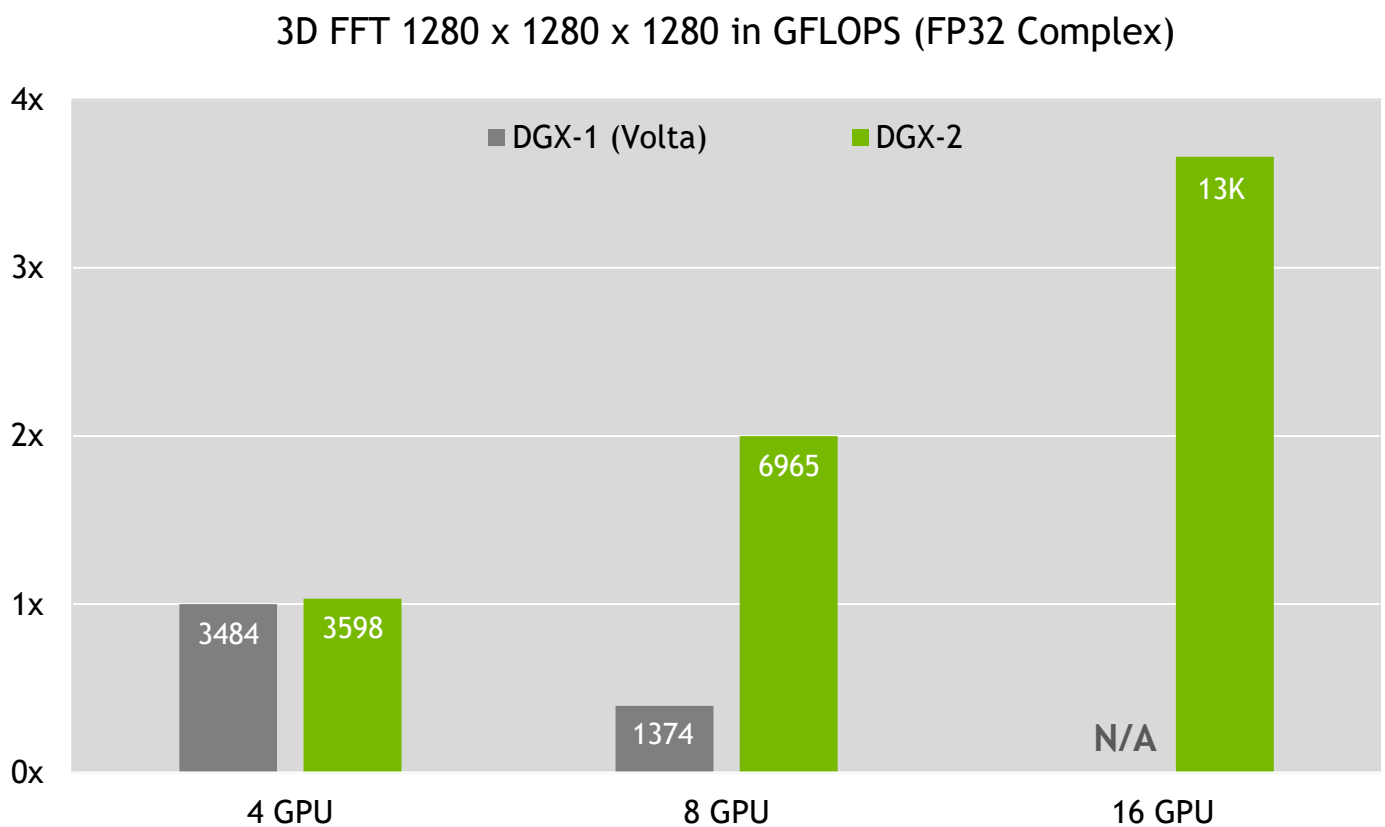
ALL REDUCE BENCHMARK



Direct All reduce
~50 lines of code
Better performance for small messages

Source: Performance measured on pre production NVSwitch hardware

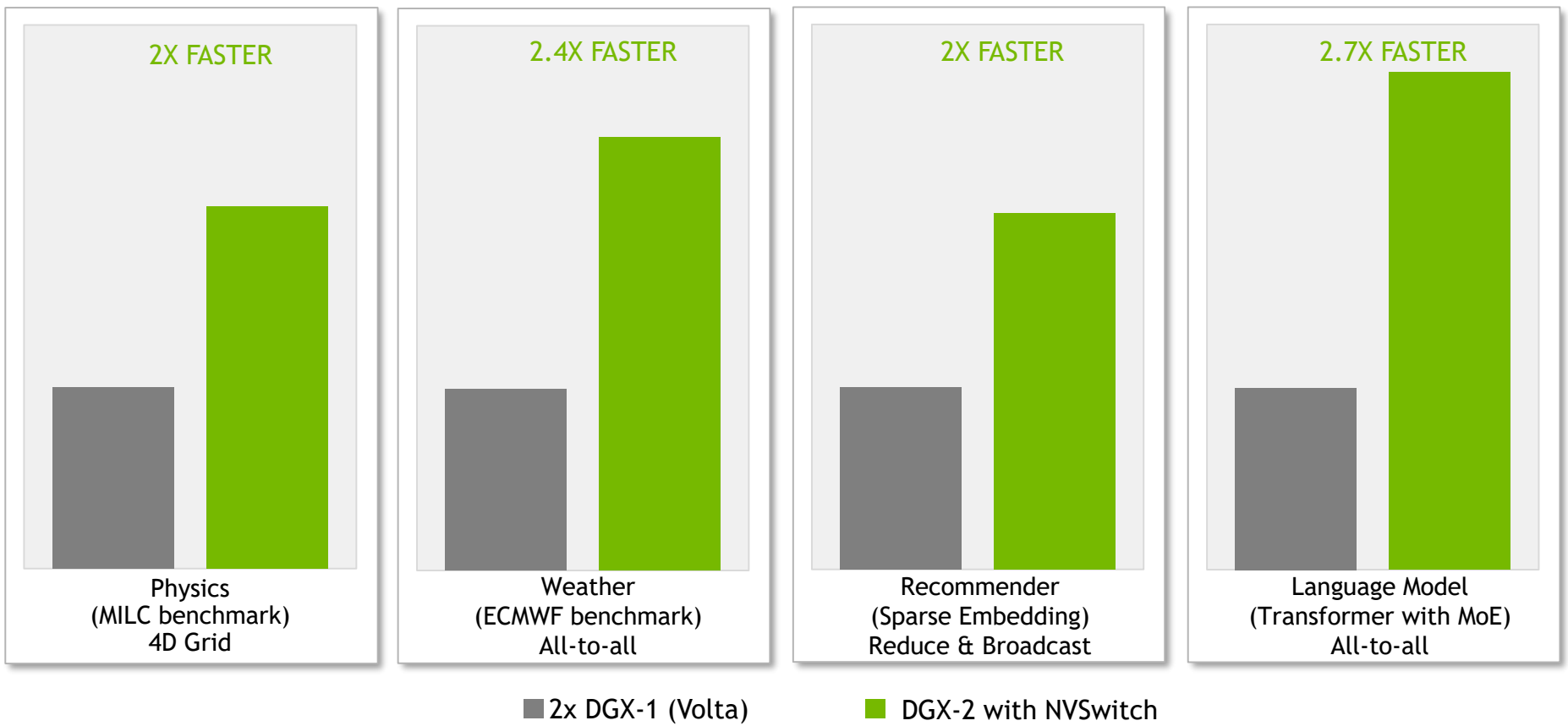
NVSWITCH FFT BENCHMARK



Performance is measured. NVSwitch uses early bring-up software
FFT is measured with cufftbench

NVIDIA.

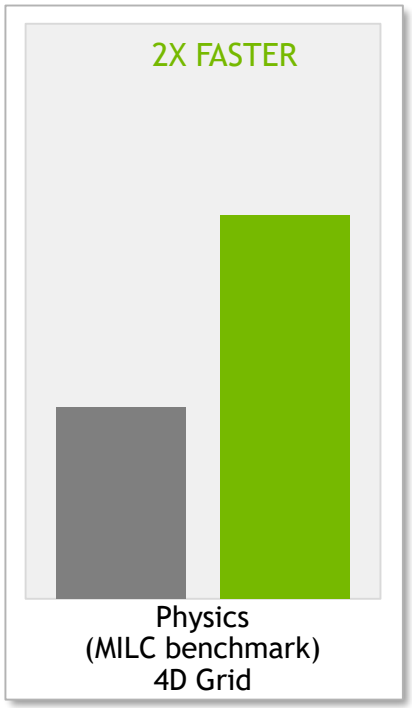
2X HIGHER PERFORMANCE WITH NVSWITCH



2 DGX-1V servers have dual socket Xeon E5 2698v4 Processor. 8 x V100 GPUs. Servers connected via 4X 100Gb IB ports | DGX-2 server has dual-socket Xeon Platinum 8168 Processor. 16 V100 GPUs

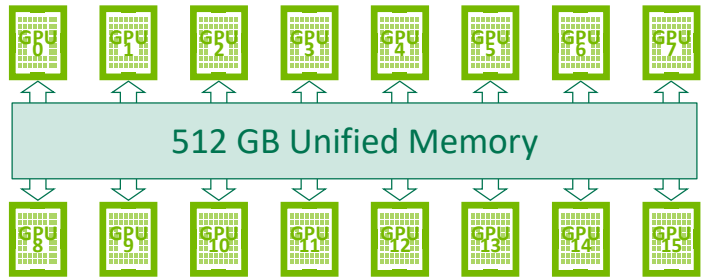
SUMMARY

DGX-2 Advantages



Faster Solutions

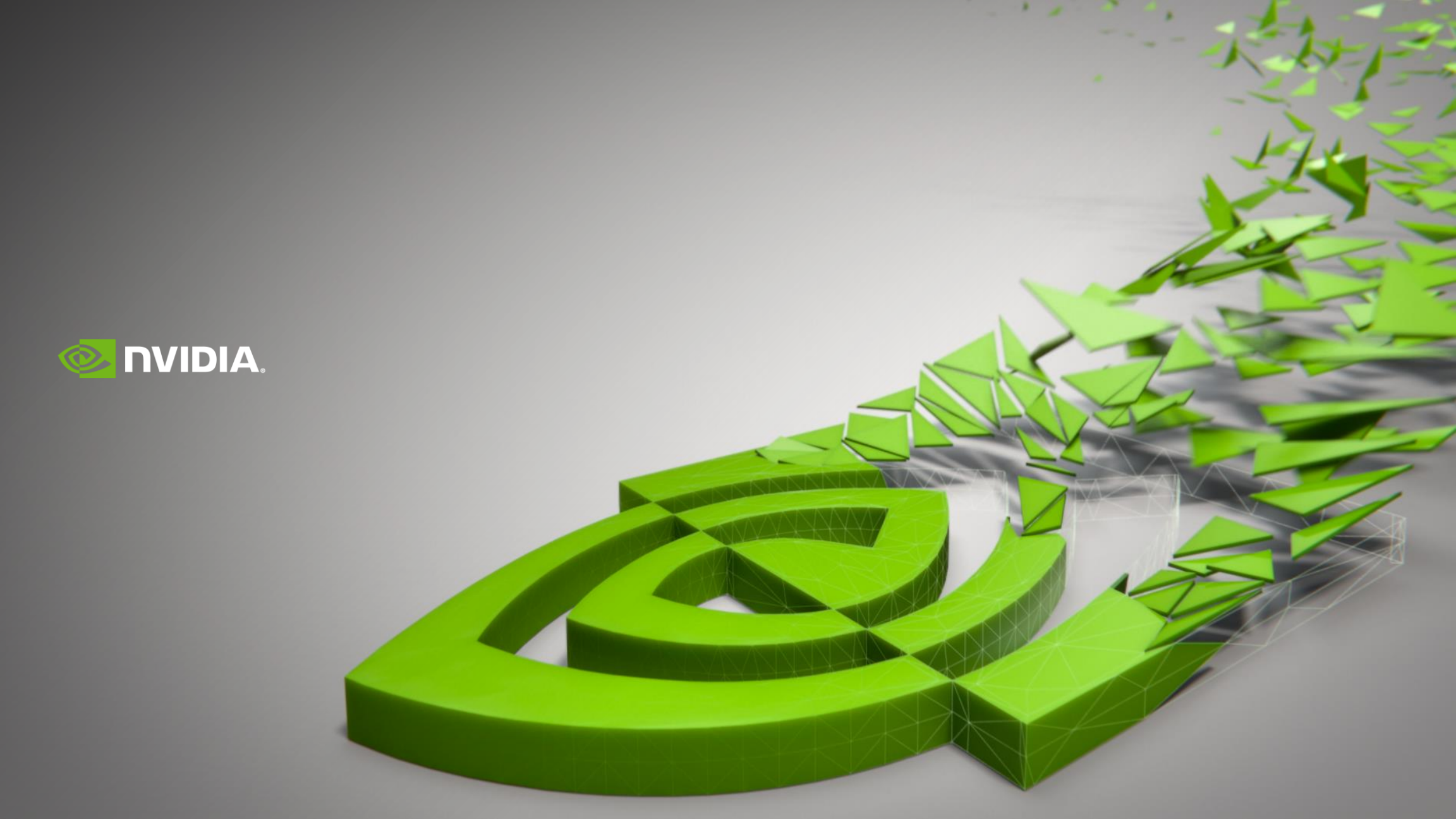
Faster Development



Gigantic Problems

OTHER NVIDIA SESSIONS TO ATTEND

ADDITIONAL NVIDIA LED SESSIONS	
S8670 Wed 3/28 Time?	**Multi-GPU Programming Techniques in CUDA** - Stephen Jones (Software Architect)
S8474 Thur 3/29 10:00 am	**GPUDirect: Life in the Fast Lane** - Davide Rosetti (Software Architect)


NVIDIA