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

<table>
<thead>
<tr>
<th></th>
<th>DGX-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU Density</td>
<td>16</td>
</tr>
<tr>
<td>1GPU to 1GPU</td>
<td>Always 6 NVLink</td>
</tr>
<tr>
<td>Connectivity</td>
<td>Fully Connected</td>
</tr>
<tr>
<td>Topology</td>
<td>Symmetric</td>
</tr>
<tr>
<td>Bisection Bandwidth</td>
<td>2.4 TB/s</td>
</tr>
</tbody>
</table>
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

Total bi-directional bandwidth: 1600 Gb/sec
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
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 MANAGEMENT

NVLINK PROVIDES

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

Full inter-GPU memory interconnect (incl. Atomics)
// 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

```c
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 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

```c
int* ptr;
// Allocate memory
cudaMallocManaged((void**)&ptr, size * numDevices);
```
UNIFIED MEMORY
Allocating across multiple GPUs

```c
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

Time = 0

Time = 1
BROADCAST ON DGX-1

Ring Scatter

Time = 0

Time = 1

Time = 2
BROADCAST ON DGX-1

Ring Scatter

Time = 0

Time = 7
BROADCAST ON DGX-2

Direct Broadcast (DGX-2)

Time = 0
IMPLEMENTATION COMPARISON

Ring Scatter (DGX-1)

```c
__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)

```c
__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

Source: Performance measured on pre production NVSwitch hardware
Performance is measured. NVSwitch uses early bring-up software
FFT is measured with cufftbench
2X HIGHER PERFORMANCE WITH NVSWITCH

- **Physics (MILC benchmark)** 4D Grid: 2x DGX-1 (Volta) vs DGX-2 with NVSwitch, 2X FASTER
- **Weather (ECMWF benchmark)** All-to-all: 2.4X FASTER
- **Recommender (Sparse Embedding)** Reduce & Broadcast: 2X FASTER
- **Language Model (Transformer with MoE)** All-to-all: 2.7X FASTER

Note: 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

<table>
<thead>
<tr>
<th>Session Code</th>
<th>Date/Time</th>
<th>Title</th>
<th>Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>S8670</td>
<td>Wed 3/28</td>
<td>Multi-GPU Programming Techniques in CUDA</td>
<td>Stephen Jones (SA)</td>
</tr>
<tr>
<td>S8474</td>
<td>Thur 3/29</td>
<td>GPUDirect: Life in the Fast Lane</td>
<td>Davide Rosetti (SA)</td>
</tr>
</tbody>
</table>