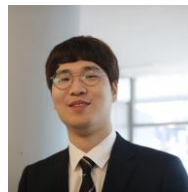


Exercise 3: Comparing Systems



Will Won

Ph.D. Student, School of Computer Science
Georgia Institute of Technology
william.won@gatech.edu

Acknowledgments: Srinivas Sridharan (Facebook), Sudarshan Srinivasan (Intel)

Agenda

Time (CET)	Time (ET)	Topic	Presenter
15:00 – 16:00	9:00 – 10:00	Introduction to Distributed Deep Learning Training Platforms	Tushar Krishna
16:00 – 17:00	10:00 – 11:00	ASTRA-sim	Saeed Rashidi
17:00 – 17:10	11:00 – 11:10	Break	
17:10 – 17:50	11:10 – 11:50	Demo and Exercises	William Won and Taekyung Heo
17:50 – 18:00	11:50 – 12:00	Extensions and Future Development	Tushar Krishna and Saeed Rashidi

Tutorial Website

includes agenda, slides, ASTRA-sim installation instructions (via source + docker image)

<https://astra-sim.github.io/tutorials/asplos-2022>

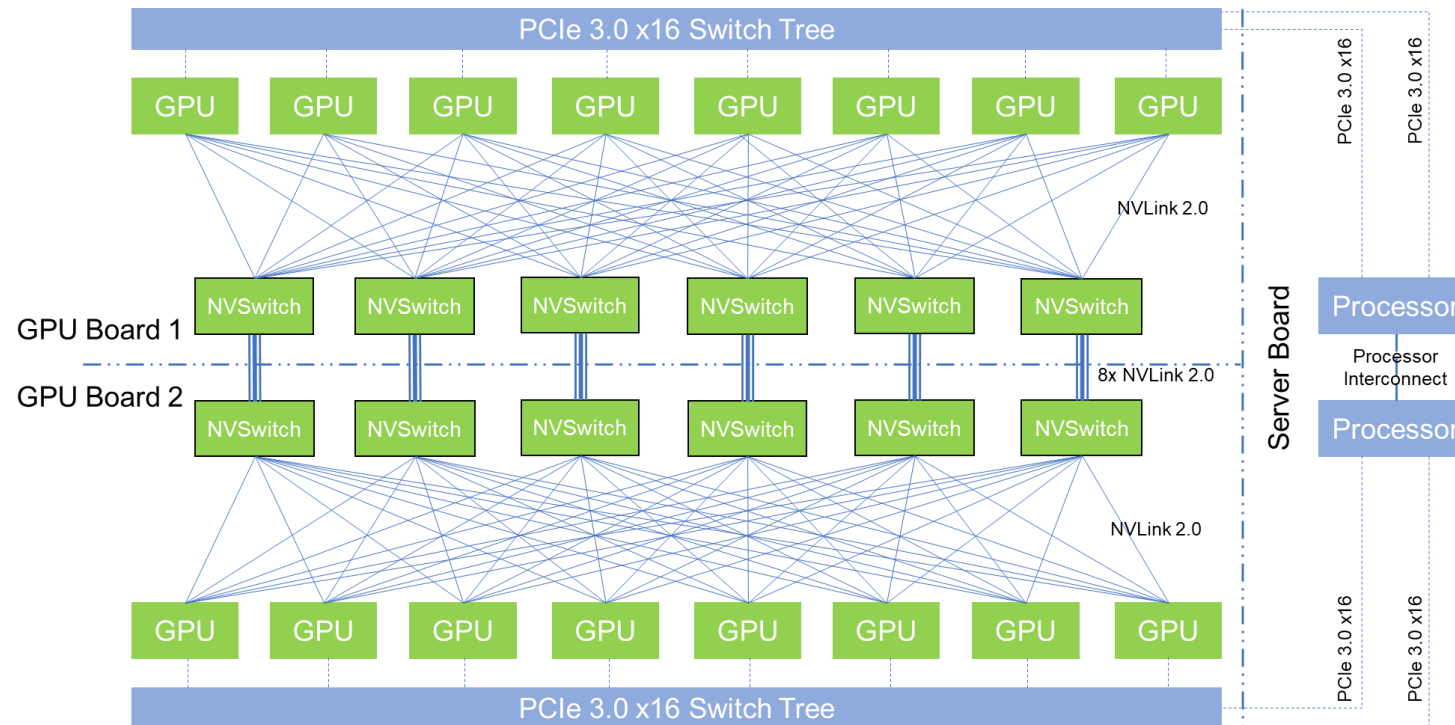
Attention: Tutorial is being recorded

Objective

- Representing real systems using ASTRA-sim
 - NVIDIA DGX-2 Pods
 - Google Cloud TPU
- Running real DL workload benchmarks
 - Vision model (VGG-16)
 - Language model (GPT-3)
- Comparing ASTRA-sim results

NVIDIA DGX-2 Architecture

- 16 V100 GPUs
- Connected Using NVSwitch / NVLink
- 100 GbE InfiniBand Scale-out per 2 GPUs (i.e., effectively 50 GbE per GPU)









- NVSwitch:
 - 25 GB/s per NVLink
 - 6 NVLinks per GPU
- InfiniBand Switch:
 - 6.25 GB/s

<https://docs.it4i.cz/dgx2/introduction/>

Representing DGX-2

- 16 DGX-2 connected (**total 256 GPUs**)

`inputs/dgx2.json`

```
{  
  "dimensions-count": 2,  2D network  
  "topologies-per-dim": ["Switch", "Switch"],  Switch_Switch Topology  
  "units-count": [16, 16],  16x16 GPUs (total 256 GPUs)  
  "links-count": [6, 1],  [6, 1] links per NPU, dim  
  "link-latency": [500, 500],  link latency  
  "link-bandwidth": [25, 6.25]  link bandwidth  
}
```

Will (TODO): latency value?

Representing DGX-2

- 16 DGX-2 connected (total 256 GPUs)

inputs/dgx2.txt

scheduling-policy: LIFO

endpoint-delay: 10

active-chunks-per-dimension: 1

preferred-dataset-splits: 4

boost-mode: 1

all-reduce-implementation: halvingDoubling_halvingDoubling  Hierarchical All-Reduce Algorithm

all-gather-implementation: halvingDoubling_halvingDoubling

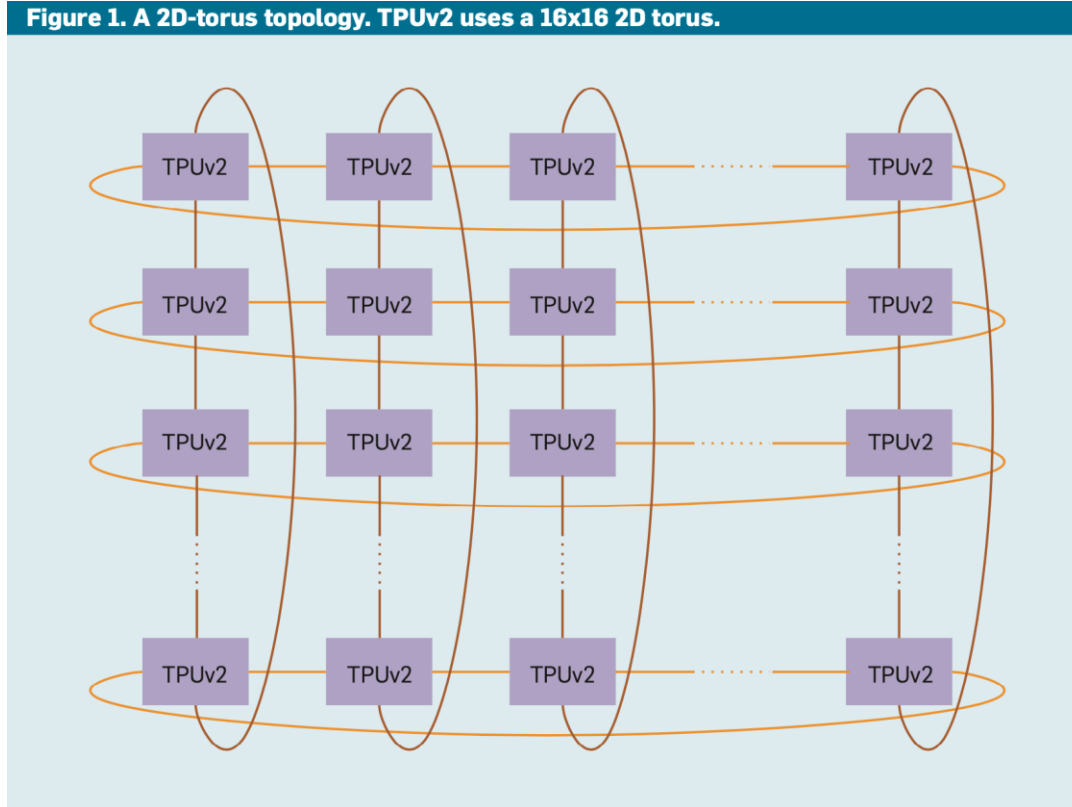
reduce-scatter-implementation: halvingDoubling_halvingDoubling

all-to-all-implementation: direct_direct

collective-optimization: localBWAware

Google Cloud TPU Architecture

Figure 1. A 2D-torus topology. TPUv2 uses a 16x16 2D torus.









- 16×16 TPUv2 (Total 256 TPUs)
- 2D Torus Topology
- Inter-core Interconnect (ICI)
 - 496 Gbps (= 62 GB/s)

N. Jouppi *et al.*, "A Domain-Specific Supercomputer for Training Deep Neural Networks," Communications of the ACM, 63, 7, 67-78.

Representing Cloud TPU

- 16x16 TPUv2 (Total 256 TPUs)

inputs/tpu.json

```
{  
  "dimensions-count": 2,  2D network  
  "topologies-per-dim": ["Ring", "Ring"],  Ring_Ring Topology (2D Torus)  
  "units-count": [16, 16],  16x16 TPUs (total 256 TPUs)  
  "links-count": [2, 2],  [2, 2] links per NPU, dim  
  "link-latency": [500, 500],  link latency  
  "link-bandwidth": [62, 62]  62GB/s link bandwidth  
}
```


Representing Cloud TPU

- 16×16 TPUv2 (Total 256 TPUs)

inputs/tpu.txt

scheduling-policy: LIFO

endpoint-delay: 10

active-chunks-per-dimension: 1

preferred-dataset-splits: 4

boost-mode: 1

all-reduce-implementation: ring_ring

← Hierarchical All-Reduce Algorithm

all-gather-implementation: ring_ring

reduce-scatter-implementation: ring_ring

all-to-all-implementation: direct_direct

collective-optimization: localBWAware

Representing Workload

Metadata		Forward			Input grad			Weight grad			Layer
Layer Name	(rsvd.)	Compute Time	Comm. Type	Comm. size	Compute Time	Comm. Type	Comm. Size	Compute Time	Comm. Type	Comm. Size	Delay
allreduce	-1	1	NONE	0	1	NONE	0	1	ALLREDUCE	1048576	1

- Compute Time
- Communication Type
- Communication Size

Representing Workload

	V100	TPUv2
Peak Tensor Performance (TFLOPS)	112	46

- V100 is **2.43x** faster than TPUv2
- *i.e.*, V100 **compute time** is **0.41x** of TPUv2

Representing Workload

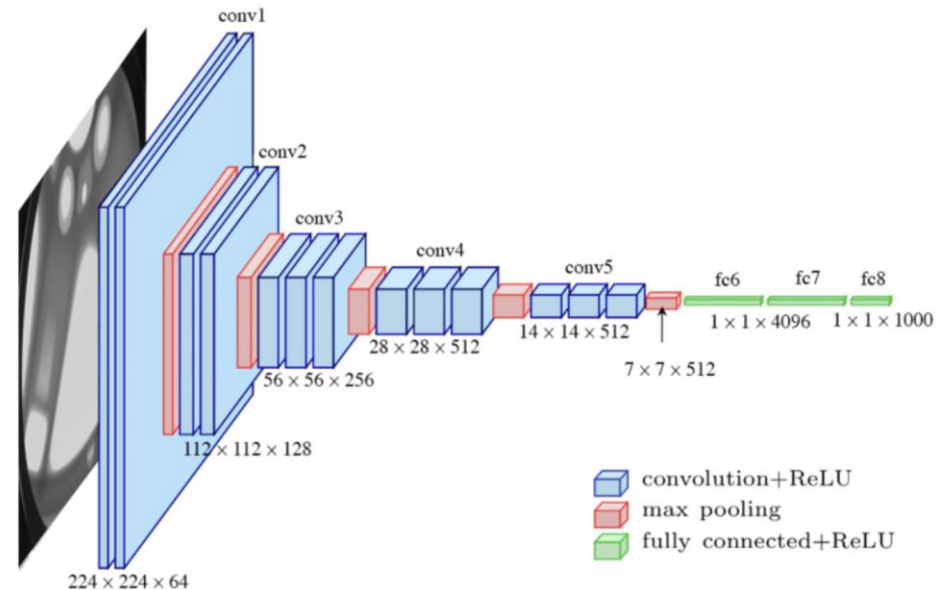
- VGG-16 first layer: $(50,176 \times 27) \times (27 \times 64)$
 - Total 173,408,256 operations
 - TPUv2: 46 TFLOPS (46×2^{40} op/s)
 - **3429 ns**
- Can leverage **Workload Generator** or other performance estimations

- Communication: Filter

- Size: $((3 \times 3) \times 3 + 1) \times 64 = 1,792$

filter input channel bias output channel

- $1,792 \times 2B = 3,584$ (=3.5 KB)



<https://medium.com/mllearning-ai/an-overview-of-vgg16-and-nin-models-96e4bf398484>

Representing Workload

inputs/vgg16.txt

DATA ← Data Parallel

16 ← #layers

block1_conv1 -1 3429 NONE 0 3429 NONE 0 3429 ALLREDUCE 3584 1 ← 1st layer

Metadata		Forward			Input grad			Weight grad			Layer
Layer Name	(rsvd.)	Compute Time	Comm. Type	Comm. size	Compute Time	Comm. Type	Comm. Size	Compute Time	Comm. Type	Comm. Size	Delay
block1_conv1	-1	3429	NONE	0	3429	NONE	0	3429	ALLREDUCE	3584	1

↑
Estimated
Compute Time

↑
Data-Parallel

↑
3.5 KB

Running Experiment

- Objective:
 - Run VGG-16
 - On DGX-2 Pod and Cloud TPU
- V100 compute time is **0.41x** of TPUv2

```
"${BINARY}" \
```

```
--run-name="DGX2-VGG16" \ ← DGX-2 Configuration
```

```
--network-configuration="${NETWORK}" \
```

```
--system-configuration="${SYSTEM}" \
```

```
--workload-configuration="${WORKLOAD}" \
```

```
--compute-scale="0.41" \ ← V100 compute time is 0.41x of TPUv2
```

```
--path="${RESULT_DIR}"/
```

Running Experiment

- Objective:
 - Run **VGG-16**
 - On **DGX-2** Pod and **Cloud TPU**

```
$ cd exercise_3/  
$ ./build.sh  
$ ./exercise_3_1.sh
```

Understanding Results

result_1/tutorial_result.csv

Name	Total Time (us)	Compute Time (us)	Exposed Communication Time (us)	Total Message Size (MB)
DGX2-VGG16	5632.316	751.019	4881.297	525.729748
TPU-VGG16	5489.225	1831.809	3657.416	525.730019

TODO: Add graph here

cf., VGG16: Pure DP, so all 256 nodes should sync

DGX2: IB comm. becomes the bottleneck, thus slow

Running Experiment

- Objective:
 - Run **GPT-3** (First 3 Transformer layers)
 - On DGX-2 Pod and Cloud TPU

```
$ ./build.sh
```

```
$ ./exercise_3_2.sh
```

Understanding Results

result_2/tutorial_result.csv

Name	Total Time (us)	Compute Time (us)	Exposed Communication Time (us)	Total Message Size (MB)
DGX2-GPT3	787767.34	575821.446	211945.894	32943.252
TPU-GPT3	1655238.43	1404442.610	250795.814	32943.252

TODO: Add graph here

cf., GPT3: MP-DP (16-16), Comm. size : MP >> DP

DGX-2 offers higher BW for inter-node than TPU, thus outperforming TPUv2