NVIDIA A100 性能测试指导手册

修订记录

Date	Version	Authors	Description
2021.12.31	V1.5	Leon Wang	单节点测试
2022.02.25	V2.0	Leon Wang	多节点测试

1. 概述

本文档的目标，是协助NPN用户，进行A100产品的基本性能测试。测试总体分为单节点测试和多节点测试两个部分,每个部分，都会提供相应的测试用例及测试结果说明。

测试基础环境准备，包括带有NVIDIA A100 GPU的服务器和上层软件环境，关于GPU驱动和上层软件环境安装指导，请参考《NVIDIA 驱动和基础软件环境安装指导手册》。

2. 单节点测试

本测试基于NVIDIA DGX A100服务器，基础硬件环境配置如下：

1	**GPU**	**8 个 NVIDIA A100 40GB GPU**
3	处理器	两个 AMD Rome 7742、共 128 个核心、2.25 GHz（基准频率）、3.4 GHz（最大加速频率）
4	内存	1TB
5	存储	操作系统：两个 1.92TB M.2 NVMe 驱动器 内部存储：15TB （4 个 3.84TB） U.2 NVMe 驱动器
6	网络	8 个单端口 Mellanox ConnectX-6 VPI 200Gb/s HDR InfiniBand ； 1 个双端口 Mellanox ConnectX-6 VPI 10/25/50/100/200Gb/s 以太网

基础软件环境配置如下：

1	**操作系统**	**DGX OS 5 （Based on Ubuntu 20.04）**
2	GPU driver	470.57.02
3	docker	20.10.7

2.1 基础性能测试

2.1.1 P2P

本测试，使用CUDA Samples中提供的p2pBandwidthLatencyTest工具，测试多卡间带宽。测试结果中，包含了单向带宽p2p，双向带宽p2p，以及Latency的性能参数。

测试步骤：

1. 从NGC拉取测试用镜像，本示例中使用TensorFlow镜像

```
docker pull nvcr.io/nvidia/tensorflow:21.07-tf1-py3
```

2. 将CUDA Samples复制到本地用户目录

```
git clone https://github.com/NVIDIA/cuda-samples
```

3. 启动容器，挂载本地CUDA Samples的目录

```
nvidia-docker run -it --rm -v /your_local_dir/cuda-samples/:/workspace
nvcr.io/nvidia/tensorflow:21.07-tf1-py3
```

4. 在容器中，进入CUDA Samples的目录

```
cd cuda-samples/Samples/p2pBandwidthLatencyTest
```

5. 使用make编译测试命令

```
sudo make
```

6. 开始测试

```
./p2pBandwidthLatencyTest
```

7. 测试结果

```
root@6f4a66d8238b:/workspace/Samples/p2pBandwidthLatencyTest#
./p2pBandwidthLatencyTest
[P2P (Peer-to-Peer) GPU Bandwidth Latency Test]
Device: 0, NVIDIA A100-SXM4-40GB, pciBusID: 7, pciDeviceID: 0, pciDomainID:0
Device: 1, NVIDIA A100-SXM4-40GB, pciBusID: f, pciDeviceID: 0, pciDomainID:0
Device: 2, NVIDIA A100-SXM4-40GB, pciBusID: 47, pciDeviceID: 0, pciDomainID:0
Device: 3, NVIDIA A100-SXM4-40GB, pciBusID: 4e, pciDeviceID: 0, pciDomainID:0
Device: 4, NVIDIA A100-SXM4-40GB, pciBusID: 87, pciDeviceID: 0, pciDomainID:0
Device: 5, NVIDIA A100-SXM4-40GB, pciBusID: 90, pciDeviceID: 0, pciDomainID:0
Device: 6, NVIDIA A100-SXM4-40GB, pciBusID: b7, pciDeviceID: 0, pciDomainID:0
Device: 7, NVIDIA A100-SXM4-40GB, pciBusID: bd, pciDeviceID: 0, pciDomainID:0
Device=0 CAN Access Peer Device=1
Device=0 CAN Access Peer Device=2
Device=0 CAN Access Peer Device=3
Device=0 CAN Access Peer Device=4
Device=0 CAN Access Peer Device=5
Device=0 CAN Access Peer Device=6
Device=0 CAN Access Peer Device=7
Device=1 CAN Access Peer Device=0
```

```
Device=1 CAN Access Peer Device=2
Device=1 CAN Access Peer Device=3
Device=1 CAN Access Peer Device=4
Device=1 CAN Access Peer Device=5
Device=1 CAN Access Peer Device=6
Device=1 CAN Access Peer Device=7
Device=2 CAN Access Peer Device=0
Device=2 CAN Access Peer Device=1
Device=2 CAN Access Peer Device=3
Device=2 CAN Access Peer Device=4
Device=2 CAN Access Peer Device=5
Device=2 CAN Access Peer Device=6
Device=2 CAN Access Peer Device=7
Device=3 CAN Access Peer Device=0
Device=3 CAN Access Peer Device=1
Device=3 CAN Access Peer Device=2
Device=3 CAN Access Peer Device=4
Device=3 CAN Access Peer Device=5
Device=3 CAN Access Peer Device=6
Device=3 CAN Access Peer Device=7
Device=4 CAN Access Peer Device=0
Device=4 CAN Access Peer Device=1
Device=4 CAN Access Peer Device=2
Device=4 CAN Access Peer Device=3
Device=4 CAN Access Peer Device=5
Device=4 CAN Access Peer Device=6
Device=4 CAN Access Peer Device=7
Device=5 CAN Access Peer Device=0
Device=5 CAN Access Peer Device=1
Device=5 CAN Access Peer Device=2
Device=5 CAN Access Peer Device=3
Device=5 CAN Access Peer Device=4
Device=5 CAN Access Peer Device=6
Device=5 CAN Access Peer Device=7
Device=6 CAN Access Peer Device=0
Device=6 CAN Access Peer Device=1
Device=6 CAN Access Peer Device=2
Device=6 CAN Access Peer Device=3
Device=6 CAN Access Peer Device=4
Device=6 CAN Access Peer Device=5
Device=6 CAN Access Peer Device=7
Device=7 CAN Access Peer Device=0
Device=7 CAN Access Peer Device=1
Device=7 CAN Access Peer Device=2
Device=7 CAN Access Peer Device=3
Device=7 CAN Access Peer Device=4
Device=7 CAN Access Peer Device=5
Device=7 CAN Access Peer Device=6

***NOTE: In case a device doesn't have P2P access to other one, it falls back to
normal memcopy procedure.
So you can see lesser Bandwidth (GB/s) and unstable Latency (us) in those cases.

P2P Connectivity Matrix
     D\D     0     1     2     3     4     5     6     7
     0       1     1     1     1     1     1     1     1
     1       1     1     1     1     1     1     1     1
     2       1     1     1     1     1     1     1     1
```

```
     3       1     1     1     1     1     1     1     1
     4       1     1     1     1     1     1     1     1
     5       1     1     1     1     1     1     1     1
     6       1     1     1     1     1     1     1     1
     7       1     1     1     1     1     1     1     1
Unidirectional P2P=Disabled Bandwidth Matrix (GB/s)
   D\D     0      1      2      3      4      5      6      7
     0 1291.92  15.88  17.64  18.28  17.71  16.94  16.95  17.94
     1  15.53 1290.26  18.52  17.92  17.76  16.93  16.94  17.92
     2  17.35  18.41 1290.26  15.71  17.72  16.94  16.92  17.96
     3  17.21  18.35  15.84 1290.26  17.71  16.92  16.93  17.93
     4  18.07  18.60  18.12  18.37 1290.26  14.56  16.86  17.92
     5  18.20  18.61  18.37  18.39  14.74 1290.26  17.02  17.73
     6  18.35  18.46  18.54  18.20  18.03  16.76 1288.13  14.53
     7  18.11  18.62  17.96  18.33  17.85  16.91  14.70 1290.26
Unidirectional P2P=Enabled Bandwidth (P2P Writes) Matrix (GB/s)
   D\D     0      1      2      3      4      5      6      7
     0 1292.66 273.52 274.66 275.32 274.47 275.68 274.30 273.94
     1 274.22 1305.35 275.53 272.32 274.56 274.89 275.61 273.95
     2 273.75 275.90 1301.00 273.25 275.75 275.16 275.63 273.87
     3 274.31 275.80 275.81 1302.08 274.68 273.41 272.99 275.62
     4 274.07 274.82 275.59 275.32 1303.17 275.44 275.27 274.66
     5 273.60 274.26 274.90 274.74 275.89 1303.17 275.27 274.59
     6 272.44 275.38 273.38 275.36 275.45 274.96 1299.92 275.06
     7 275.66 275.45 275.63 275.49 275.31 275.37 275.50 1304.26
Bidirectional P2P=Disabled Bandwidth Matrix (GB/s)
   D\D     0      1      2      3      4      5      6      7
     0 1300.83  17.98  20.02  19.41  19.35  23.08  23.10  19.64
     1  18.35 1301.54  19.54  19.99  17.87  20.59  19.99  19.70
     2  19.99  19.64 1304.80  18.35  19.36  23.06  23.06  19.71
     3  19.93  19.65  18.47 1302.63  19.33  23.07  23.04  19.56
     4  20.22  20.34  20.12  20.17 1303.71  15.89  23.37  19.57
     5  25.79  25.90  26.09  25.89  15.91 1305.35  18.53  23.34
     6  25.97  26.18  26.01  26.07  23.48  18.95 1303.17  15.77
     7  20.08  20.06  20.05  20.03  19.43  23.37  15.80 1304.80
Bidirectional P2P=Enabled Bandwidth Matrix (GB/s)
   D\D     0      1      2      3      4      5      6      7
     0 1301.83 516.43 515.97 516.11 516.83 515.96 516.68 517.30
     1 519.37 1302.63 519.17 516.26 518.32 517.79 517.63 516.94
     2 519.96 517.97 1301.54 517.51 517.80 520.00 517.28 517.80
     3 517.45 517.94 515.92 1301.00 519.83 518.65 516.76 518.48
     4 518.46 518.47 518.47 517.10 1303.71 517.78 517.72 516.58
     5 516.92 518.30 518.76 516.92 517.27 1301.00 517.78 518.12
     6 518.12 516.24 517.61 517.44 516.92 519.08 1302.08 518.24
     7 516.58 515.56 517.44 517.27 517.10 516.92 517.78 1304.26
P2P=Disabled Latency Matrix (us)
   GPU     0      1      2      3      4      5      6      7
     0   2.35  24.58  24.59  24.66  24.59  24.59  24.51  24.49
     1  24.59   2.42  24.59  24.60  24.35  24.54  24.58  24.27
     2  24.62  24.49   2.47  24.60  24.59  24.60  24.58  24.53
     3  24.58  24.59  24.60   2.51  24.57  24.58  24.46  24.54
     4  24.61  24.59  24.58  24.59   2.44  24.60  24.59  24.60
     5  24.61  24.59  24.59  24.59  24.66   2.45  24.59  24.60
     6  24.60  24.58  24.60  24.60  24.65  24.59   2.63  24.60
     7  24.62  24.59  24.65  24.59  24.59  24.59  24.58   2.34

   CPU     0      1      2      3      4      5      6      7
     0   3.23   9.76   9.38   9.40  10.46  10.53  10.41  10.41
```

```
     1   9.45   3.13   9.23   9.19  10.25  10.35  10.41  10.34
     2   9.25   9.23   3.13   9.12  10.14  10.23  10.29  10.20
     3   9.24   9.21   9.02   3.12  10.16  10.26  10.30  10.15
     4   9.90  10.00   9.76   9.71   3.44  11.05  11.04  10.88
     5   9.98  10.00   9.81   9.81  10.96   3.49  10.96  10.97
     6   9.99  10.05   9.84   9.88  11.00  11.02   3.50  10.97
     7  10.11  10.08   9.88   9.73  10.86  10.92  10.94   3.45
P2P=Enabled Latency (P2P Writes) Matrix (us)
   GPU     0      1      2      3      4      5      6      7
     0   2.33   3.11   3.02   3.02   3.02   3.02   3.09   3.08
     1   3.07   2.41   3.02   3.02   3.03   3.03   3.09   3.03
     2   3.03   3.03   2.58   3.09   3.04   3.15   3.11   3.02
     3   3.11   3.03   3.09   2.49   3.02   3.03   3.10   3.08
     4   2.94   2.95   3.00   2.99   2.56   3.00   2.93   2.94
     5   2.95   2.96   3.07   2.99   2.94   2.44   2.93   2.97
     6   2.97   2.95   2.93   2.93   2.93   2.96   2.62   3.06
     7   3.00   2.95   2.93   2.97   2.97   2.95   2.99   2.31

   CPU     0      1      2      3      4      5      6      7
     0   3.21   2.50   2.54   2.52   2.55   2.54   2.58   2.57
     1   2.64   3.22   2.56   2.57   2.55   2.57   2.54   2.55
     2   2.59   2.52   3.27   2.51   2.51   2.52   2.50   3.73
     3   2.58   2.53   2.54   3.15   2.56   2.55   2.51   2.45
     4   2.95   2.89   3.01   2.92   3.52   2.91   2.79   2.88
     5   3.04   2.94   2.95   2.98   2.99   3.59   2.99   2.95
     6   3.00   2.96   2.90   2.94   2.96   3.02   3.57   2.93
     7   3.08   2.97   2.99   2.95   2.97   2.95   2.97   3.58

NOTE: The CUDA Samples are not meant for performance measurements. Results may
vary when GPU Boost is enabled.
```

2.1.2 Bandwidth

本测试，使用CUDA Samples中提供的bandwidthTest工具，可以测试主机到GPU（Host to device）复制带宽和GPU到GPU(device to device)复制带宽。

测试步骤：

1. 从NGC拉取测试用镜像，本示例中使用TensorFlow镜像

```
docker pull nvcr.io/nvidia/tensorflow:21.07-tf1-py3
```

2. 将CUDA Samples复制到本地用户目录

```
git clone https://github.com/NVIDIA/cuda-samples
```

3. 启动容器，挂载本地CUDA Samples的目录

```
nvidia-docker run -it --rm -v /your_local_dir/cuda-samples/:/workspace
nvcr.io/nvidia/tensorflow:21.07-tf1-py3
```

4. 在容器中，进入CUDA Samples的目录

```
cd cuda-samples/Samples/bandwidthTest
```

5. 使用make编译测试命令

```
sudo make
```

6. 开始测试

```
./bandwidthTest
```

7. 测试结果

```
root@6f4a66d8238b:/workspace/Samples/bandwidthTest# ./bandwidthTest
[CUDA Bandwidth Test] - Starting...
Running on...

 Device 0: NVIDIA A100-SXM4-40GB
 Quick Mode

 Host to Device Bandwidth, 1 Device(s)
 PINNED Memory Transfers
   Transfer Size (Bytes)        Bandwidth(GB/s)
   32000000                     24.8

 Device to Host Bandwidth, 1 Device(s)
 PINNED Memory Transfers
   Transfer Size (Bytes)        Bandwidth(GB/s)
   32000000                     25.9

 Device to Device Bandwidth, 1 Device(s)
 PINNED Memory Transfers
   Transfer Size (Bytes)        Bandwidth(GB/s)
   32000000                     1159.7

Result = PASS

NOTE: The CUDA Samples are not meant for performance measurements. Results
may vary when GPU Boost is enabled.
```

更多CUDA Spamples测试，请查阅[cuda-samples]

2.1.3 GEMM

矩阵乘法是高性能计算中最常用到一类计算模型。无论在HPC领域，例如做FFT、卷积、相关、滤波等，还是在 Deep Learning 领域，例如卷积层，全连接层等，其核心算法都直接或者可以转换为矩阵乘法。

cuBLAS 是标准线性代数库 (standard basic linear algebra subroutines (BLAS)) 的 GPU 加速实现，它支持 Level 1 (向量与向量运算)，Level 2 (向量与矩阵运算)，Level 3 (矩阵与矩阵运算) 级别的标准矩阵运算。

GEMM（General matrix multiplication）是NVIDIA提供的二进制测试工具，利用cuBLAS库，通过随机数进行矩阵乘运算，测试GPU的Peak TFLOPs。通过设定参数，GEMM可以测试不同数据类型。该工具在 NVIDIA A100 Benchmark Guide中作为附件提供。请查阅PDF文件的附件。

测试步骤：

1. 将'NVIDIA A100 Benchmark Guide'附件中的cublasMatmulBench.nvzip下载到本地，解压缩后，再将cublasMatmulBench二进制文件删除到要测试的GPU服务器。
2. 为二进制文件赋予执行权限

```
sudo chmod -R 777 cublasMatmulBench
```

3. 执行测试

   注意：该测试命令运行在单GPU上，默认调用GPU 0，如需测试全部GPU，可以使用Docker镜像挂载不同的GPU来测试。

   使用docker挂载单个GPU示例

```
nvidia-docker run -it --gpus '"device=1"' --rm -v
/your_cublasMatmulBench_file_dir/:/workspace
nvcr.io/nvidia/tensorflow:21.07-tf1-py3
```

   进入容器后，通过nvidia-smi查看，仅挂载了1个GPU

```


root@1c49a3fcece6:/workspace# nvidia-smi
Thu Dec 30 08:02:32 2021
+-----------------------------------------------------------------------------
--+
| NVIDIA-SMI 470.57.02    Driver Version: 470.57.02    CUDA Version: 11.4
  |
|-------------------------------+----------------------+----------------------
--+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr.
ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute
M. |
|                               |                      |               MIG
M. |
|===============================+======================+======================
==|
|   0  NVIDIA A100-SXM...  On   | 00000000:0F:00.0 Off |
0 |
| N/A   30C    P0    51W / 400W |      3MiB / 40536MiB |      0%
Default |
|                               |                      |
Disabled |
+-------------------------------+----------------------+----------------------
--+

+-----------------------------------------------------------------------------
--+
| Processes:
|
|  GPU   GI   CI        PID   Type   Process name                  GPU
Memory |
|        ID   ID                                                   Usage
|
|=============================================================================
==|


```

```
|  No running processes found
  |
+-----------------------------------------------------------------------------
--+
```

GEMM测试命令如下：

Int8:

```
./cublasMatmulBench -P=bisb_imma -m=8192 -n=3456 -k=16384 -T=1000 -ta=1 -B=0
```

FP16:

```
./cublasMatmulBench -P=hsh -m=12288 -n=9216 -k=32768 -T=1000 -tb=1 -B=0
```

TF32:

```
./cublasMatmulBench -P=sss_fast_tf32 -m=8192 -n=3456 -k=16384 -T=1000 -ta=1 -B=0
```

FP64:

```
./cublasMatmulBench -P=ddd -m=3456 -n=2048 -k=16384 -T=1000 -tb=1 -B=0
```

FP32:

```
./cublasMatmulBench -P=sss -m=3456 -n=2048 -k=16384 -T=1000 -tb=1 -B=0
```

4. 测试结果

测试输出结果示例

```
root@1c49a3fcece6:/workspace# ./cublasMatmulBench -P=bisb_imma -m=8192 -
n=3456 -k=16384 -T=1000 -ta=1 -B=0
testing cublasLt
#### args: ta=N tb=T m=8192 n=3456 k=16384  alpha = (0x3f800000, 1) beta=
(0x00000000, 0)
#### args: lda=262144 ldb=110592 ldc=262144 loop=1000
^^^^ CUDA : elapsed = 1.83777 sec,  Gflops = 504803.984
testing cublasLt pass
root@1c49a3fcece6:/workspace# ./cublasMatmulBench -P=hsh -m=12288 -n=9216 -
k=32768 -T=1000 -tb=1 -B=0
testing cublasLt
#### args: ta=N tb=T m=12288 n=9216 k=32768  alpha = (0x3f800000, 1) beta=
(0x00000000, 0)
#### args: lda=12288 ldb=9216 ldc=12288 loop=1000
^^^^ CUDA : elapsed = 29.3814 sec,  Gflops = 252598.764
testing cublasLt pass
root@1c49a3fcece6:/workspace# ./cublasMatmulBench -P=sss_fast_tf32 -m=8192 -
n=3456 -k=16384 -T=1000 -ta=1 -B=0
testing cublasLt
#### args: ta=T tb=N m=8192 n=3456 k=16384  alpha = (0x3f800000, 1) beta=
(0x00000000, 0)
#### args: lda=16384 ldb=16384 ldc=8192 loop=1000
^^^^ CUDA : elapsed = 8.01993 sec,  Gflops = 115675.948
testing cublasLt pass
```

```
root@1c49a3fcece6:/workspace# ./cublasMatmulBench -P=ddd -m=3456 -n=2048 -
k=16384 -T=1000 -tb=1 -B=0
testing cublasLt
#### args: ta=N tb=T m=3456 n=2048 k=16384  alpha = (0x3ff0000000000000, 1)
beta= (0x0000000000000000, 0)
#### args: lda=3456 ldb=2048 ldc=3456 loop=1000
^^^^ CUDA : elapsed = 12.702 sec,  Gflops = 18259.165
testing cublasLt pass
root@1c49a3fcece6:/workspace# ./cublasMatmulBench -P=sss -m=3456 -n=2048 -
k=16384 -T=1000 -tb=1 -B=0
testing cublasLt
#### args: ta=N tb=T m=3456 n=2048 k=16384  alpha = (0x3f800000, 1) beta=
(0x00000000, 0)
#### args: lda=3456 ldb=2048 ldc=3456 loop=1000
^^^^ CUDA : elapsed = 12.0867 sec,  Gflops = 19188.728
testing cublasLt pass
```

综合多次测试的DGX A100整机输出结果示例

GPU ID	FP64	FP32	TF32	FP16	INT8
GPU 0	18352.932	19274.983	115660.124	251452.141	495411.247
GPU 1	18322.574	19315.177	115626.624	251970.452	491357.464
GPU 2	18226.888	19332.412	115005.810	251339.840	489362.036
GPU 3	18215.569	19255.078	114791.677	250146.030	492475.635
GPU 4	18451.290	19332.187	116599.681	256276.925	495438.391
GPU 5	18325.219	19302.091	115503.909	251562.375	494460.517
GPU 6	18150.548	19309.257	114613.384	248635.652	487978.464
GPU 7	18151.756	19226.536	114145.384	250610.918	494158.458

2.1.4 Stream

NVIDIA 为 STREAM 基准测试提供优化的 CUDA 实现，用于测量单个 GPU 上的内存带宽。 该工具在NVIDIA A100 Benchmark Guide中作为附件提供。请查阅PDF文件的附件。

测试步骤：

1. 将‘NVIDIA A100 Benchmark Guide’附件中的stream_test.nvzip下载到本地，解压缩后，再将stream_test二进制文件删除到要测试的GPU服务器。
2. 为二进制文件赋予执行权限

```
sudo chmod -R 777 stream_test
```

3. 执行测试

   注意：该测试命令运行在单GPU上，通过调整-d参数后面的数字（0-7），就可以选择不同的GPU执行。

```
./stream_test -d0 -n113246208 -r0
```

4. 测试结果

   测试输出结果示例

```
dgxnpn@dgx-a100:~/Leonwang$ ./stream_test -d0 -n113246208 -r0
usage: stream_test -n<elements> -d<device> -r<random>

There are 8 devices supporting CUDA
Device 0: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 1: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 2: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 3: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 4: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 5: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 6: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 7: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON

Device Selected 0: "NVIDIA A100-SXM4-40GB"
 STREAM Benchmark implementation in CUDA
 Array size (double)=113246208*8*8 (6912 MB)
 Running 100 loops

Optimizing...
 [|||||||||||||||||||||||||||||||||||||||||||||||||||||||/] 100.0%

Optimization Complete...

--- Best Copy  ---  lt: 0 lp: 3  block: 64       stride: 4       shmem: 5790
    occ: 90.6%      GB/s: 1397.5   eff: 89.9%
--- Best Scale ---  lt: 0 lp: 3  block: 64       stride: 4       shmem: 5597
    occ: 93.8%      GB/s: 1396.8   eff: 89.8%
--- Best Add   ---  lt: 0 lp: 2  block: 64       stride: 32      shmem:
15266    occ: 34.4%      GB/s: 1380.3   eff: 88.8%
--- Best Triad ---  lt: 1 lp: 2  block: 64       stride: 32      shmem:
15266    occ: 34.4%      GB/s: 1380.7   eff: 88.8%
--- Best Read  ---  lt: 0 lp: 2  block: 128      stride: 1       shmem:
20992    occ: 50.0%      GB/s: 1505.1   eff: 96.8%
--- Best Write ---  lt: 0 lp: 1  block: 512      stride: 4       shmem:
49152    occ: 85.4%      GB/s: 1538.1   eff: 98.9%

starting test...

test done ...

Function      Rate (MB/s)   Avg time     Min time     Max time
Copy:      1393079.3101       0.0013       0.0013       0.0013
Scale:     1394108.3932       0.0013       0.0013       0.0013
```

```
Add:       1375549.0929       0.0020       0.0020       0.0020
Triad:     1379413.9916       0.0020       0.0020       0.0022
Read:      1474559.9300       0.0006       0.0006       0.0006
Write:     1544041.8915       0.0006       0.0006       0.0006

Checking result for Copy:      : all values equal 2.66667, passed!
Checking result for Scale:     : all values equal 0.8, passed!
Checking result for Add:       : all values equal 6, passed!
Checking result for Triad:     : all values equal 4.66667, passed!
Checking result for Read:      : all values equal 0.5, passed!
Checking result for Write:     : all values equal 0.1, passed!
dgxnpn@dgx-a100:~/Leonwang$ ./stream_test -d1 -n113246208 -r0
usage: stream_test -n<elements> -d<device> -r<random>

There are 8 devices supporting CUDA
Device 0: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 1: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 2: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 3: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 4: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 5: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 6: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 7: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON

Device Selected 1: "NVIDIA A100-SXM4-40GB"
 STREAM Benchmark implementation in CUDA
 Array size (double)=113246208*8*8 (6912 MB)
 Running 100 loops

Optimizing...
 [|||||||||||||||||||||||||||||||||||||||||||||||||/] 100.0%

Optimization Complete...

--- Best Copy  ---  lt: 0 lp: 2  block: 128      stride: 4       shmem:
15266    occ: 68.8%      GB/s: 1397.7   eff: 89.9%
--- Best Scale ---  lt: 1 lp: 2  block: 128      stride: 1       shmem:
15266    occ: 68.8%      GB/s: 1398.8   eff: 89.9%
--- Best Add   ---  lt: 0 lp: 2  block: 64       stride: 32      shmem:
15266    occ: 34.4%      GB/s: 1379.5   eff: 88.7%
--- Best Triad ---  lt: 0 lp: 2  block: 64       stride: 32      shmem:
13994    occ: 37.5%      GB/s: 1382.4   eff: 88.9%
--- Best Read  ---  lt: 0 lp: 2  block: 128      stride: 1       shmem:
20992    occ: 50.0%      GB/s: 1504.7   eff: 96.7%
--- Best Write ---  lt: 0 lp: 2  block: 128      stride: 64      shmem:
12918    occ: 81.3%      GB/s: 1538.7   eff: 98.9%

starting test...

test done ...
```

```
Function      Rate (MB/s)   Avg time     Min time     Max time
Copy:      1394382.9784       0.0013       0.0013       0.0013
Scale:     1393285.0307       0.0013       0.0013       0.0013
Add:       1375237.2860       0.0020       0.0020       0.0021
Triad:     1381680.4042       0.0020       0.0020       0.0020
Read:      1472106.5433       0.0006       0.0006       0.0006
Write:     1544041.8915       0.0006       0.0006       0.0006

Checking result for Copy:      : all values equal 2.66667, passed!
Checking result for Scale:     : all values equal 0.8, passed!
Checking result for Add:       : all values equal 6, passed!
Checking result for Triad:     : all values equal 4.66667, passed!
Checking result for Read:      : all values equal 0.5, passed!
Checking result for Write:     : all values equal 0.1, passed!
dgxnpn@dgx-a100:~/Leonwang$ ./stream_test -d2 -n113246208 -r0
usage: stream_test -n<elements> -d<device> -r<random>

There are 8 devices supporting CUDA
Device 0: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 1: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 2: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 3: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 4: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 5: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 6: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 7: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON

Device Selected 2: "NVIDIA A100-SXM4-40GB"
 STREAM Benchmark implementation in CUDA
 Array size (double)=113246208*8*8 (6912 MB)
 Running 100 loops

Optimizing...
 [|||||||||||||||||||||||||||||||||||||||||||||||||/] 100.0%

Optimization Complete...

--- Best Copy  ---  lt: 0 lp: 3  block: 64       stride: 4       shmem: 6219
    occ: 84.4%      GB/s: 1398.8   eff: 89.9%
--- Best Scale ---  lt: 0 lp: 2  block: 128      stride: 1       shmem:
15266    occ: 68.8%      GB/s: 1398.8   eff: 89.9%
--- Best Add   ---  lt: 0 lp: 2  block: 64       stride: 32      shmem:
15266    occ: 34.4%      GB/s: 1379.5   eff: 88.7%
--- Best Triad ---  lt: 2 lp: 2  block: 64       stride: 32      shmem:
15266    occ: 34.4%      GB/s: 1380.2   eff: 88.8%
--- Best Read  ---  lt: 0 lp: 2  block: 256      stride: 4       shmem:
33587    occ: 62.5%      GB/s: 1509.8   eff: 97.1%
--- Best Write ---  lt: 0 lp: 2  block: 128      stride: 4       shmem:
16793    occ: 62.5%      GB/s: 1538.7   eff: 98.9%
```

```
starting test...

test done ...

Function      Rate (MB/s)   Avg time     Min time     Max time
Copy:      1393285.0307       0.0013       0.0013       0.0013
Scale:     1394382.9784       0.0013       0.0013       0.0013
Add:       1375950.2190       0.0020       0.0020       0.0020
Triad:     1379526.0005       0.0020       0.0020       0.0020
Read:      1467223.8973       0.0006       0.0006       0.0006
Write:     1549450.1356       0.0006       0.0006       0.0006

Checking result for Copy:      : all values equal 2.66667, passed!
Checking result for Scale:     : all values equal 0.8, passed!
Checking result for Add:       : all values equal 6, passed!
Checking result for Triad:     : all values equal 4.66667, passed!
Checking result for Read:      : all values equal 0.5, passed!
Checking result for Write:     : all values equal 0.1, passed!
dgxnpn@dgx-a100:~/Leonwang$ ./stream_test -d3 -n113246208 -r0
usage: stream_test -n<elements> -d<device> -r<random>

There are 8 devices supporting CUDA
Device 0: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 1: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 2: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 3: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 4: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 5: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 6: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON
Device 7: "NVIDIA A100-SXM4-40GB"        108 SMs(8.0)   Memory: 1215MHz x
5120-bit = 1555.2 GB/s PEAK    ECC is ON

Device Selected 3: "NVIDIA A100-SXM4-40GB"
 STREAM Benchmark implementation in CUDA
 Array size (double)=113246208*8*8 (6912 MB)
 Running 100 loops

Optimizing...
 [|||||||||||||||||||||||||||||||||||||||||||||||||||/] 100.0%

Optimization Complete...

--- Best Copy  ---  lt: 0 lp: 2  block: 256      stride: 1       shmem:
23990    occ: 87.5%      GB/s: 1397.7   eff: 89.9%
--- Best Scale ---  lt: 0 lp: 3  block: 64       stride: 2       shmem: 5248
    occ: 100.0%     GB/s: 1397.7   eff: 89.9%
--- Best Add   ---  lt: 0 lp: 2  block: 64       stride: 128     shmem:
13994    occ: 37.5%      GB/s: 1381.0   eff: 88.8%
--- Best Triad ---  lt: 1 lp: 2  block: 64       stride: 32      shmem:
15266    occ: 34.4%      GB/s: 1381.0   eff: 88.8%
```

```
--- Best Read  ---  lt: 0 lp: 2  block: 256      stride: 4       shmem:
33587    occ: 62.5%      GB/s: 1512.4   eff: 97.2%
--- Best Write ---  lt: 0 lp: 2  block: 128      stride: 4       shmem:
10496    occ: 100.0%     GB/s: 1538.7   eff: 98.9%

starting test...

test done ...

Function      Rate (MB/s)   Avg time     Min time     Max time
Copy:      1391094.3143       0.0013       0.0013       0.0013
Scale:     1395482.6579       0.0013       0.0013       0.0013
Add:       1375950.2190       0.0020       0.0020       0.0020
Triad:     1378809.3581       0.0020       0.0020       0.0020
Read:      1467223.8973       0.0006       0.0006       0.0006
Write:     1549450.1356       0.0006       0.0006       0.0006

Checking result for Copy:      : all values equal 2.66667, passed!
Checking result for Scale:     : all values equal 0.8, passed!
Checking result for Add:       : all values equal 6, passed!
Checking result for Triad:     : all values equal 4.66667, passed!
Checking result for Read:      : all values equal 0.5, passed!
Checking result for Write:     : all values equal 0.1, passed!
```

测试结果说明，请查看结果输出中，各个Function的输出速率，单位是MB/s。

综合多次测试的DGX A100整机输出结果示例

GPU ID	Copy	Scale	Add	Triad	Read	Write
GPU-0	1393696.526	1395964.297	1384924.572	1378607.931	1474867.267	1543200.2227
GPU-1	1392633.981	1391675.622	1374747.624	1379413.991	1473562.251	1543705.0511
GPU-2	1393216.4c50	1394623.377	1375838.790	1387231.848	1476020.640	1546994.7398
GPU-3	1393353.617	1393147.876	1375170.512	1380916.606	1472489.315	1539675.4920
GPU-4	1392668.178	1395654.615	1375571.417	1378518.408	1473485.541	1545643.5784
GPU-5	1393868.044	1394692.096	1387934.547	1379033.197	1476636.519	1548263.8867
GPU-6	1391983.541	1395998.657	1384450.519	1378630.272	1472872.286	1548856.7840
GPU-7	1393422.211	1394245.608	1375571.417	1378697.465	1472642.537	1542191.5558

2.1.5 HPL

Linpack 已成为全球最流行的测试高性能计算机系统浮点性能的基准。 用高性能计算机通过高斯消元法求解n元一阶稠密线性代数方程来评价高性能计算机的浮点性能。

Linpack测试包括三类：Linpack 100、Linpack 1000和HPL。

NVIDIA NGC为 的 HPL benchmark测试提供了软件包，可在配备 NVIDIA GPU 的分布式内存计算机上，基于 netlib HPL 基准测试，使用 Tensor Cores 以双精度（64 位）算法求解密集线性系统。

HPL-AI benchmark测试包含在 HPL benchmark测试中。 HPL-AI benchmark测试提供软件包以使用 Tensor Cores 在混合精度算法中解决（随机）密集线性系统。

测试步骤：

1. 从NGC拉取测试用镜像，本示例中使用TensorFlow镜像

```
docker pull nvcr.io/nvidia/hpc-benchmark:20.10-hpl
```

2. 获取测试用data

   本测试需要使用HPL测试data，文件名为HPL-dgx-a100-1N.dat 。请查阅PDF附件。

3. 启动容器，挂载本地data的目录

```
nvidia-docker run --privileged -it --rm -v $(pwd):/my-dat-files
nvcr.io/nvidia/hpc-benchmarks:20.10-hpl
```

4. 在容器中，设置环境变量

```
export UCX_TLS=all
```

   注意：NOTE: **UCX_TLS=***rc_x** are set in the container, so for a single node without IB or down-state IB, there will be errors.

5. 开始测试

```
mpirun --bind-to none -np 8 hpl.sh --config dgx-a100 --dat /my-dat-
files/HPL-dgx-a100-1N.dat
```

6. 测试结果

测试输出结果

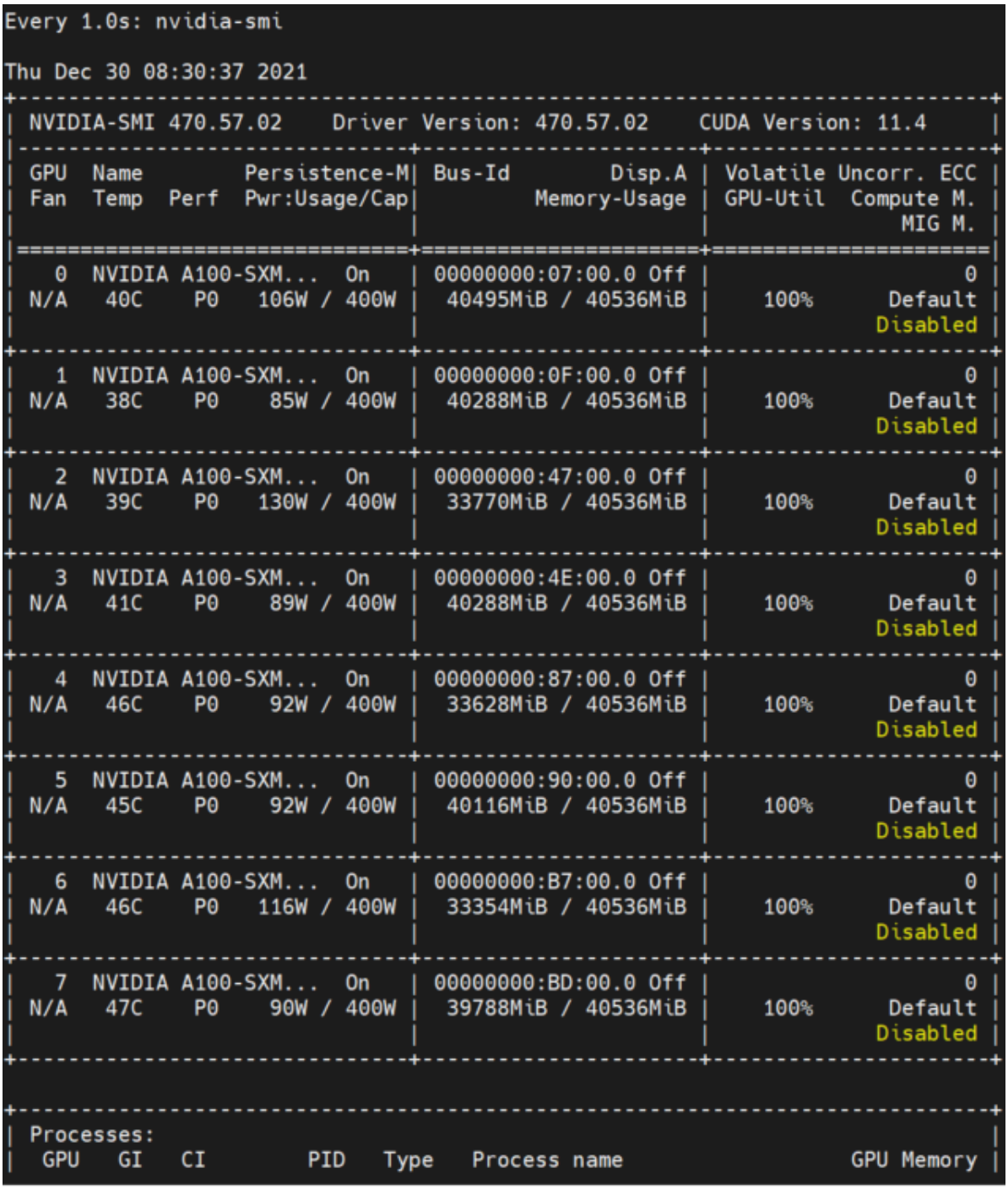
```
2021-12-30 08:33:26.138
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR03L2L2      235520   768     4     2             204.00              4.269e+04
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0000512 ...... PASSED
================================================================================

Finished      1 tests with the following results:
              1 tests completed and passed residual checks,
              0 tests completed and failed residual checks,
              0 tests skipped because of illegal input values.
--------------------------------------------------------------------------------

End of Tests.
================================================================================
```

测试结果说明，表中显示的4.269e+04 Gflops表示 4.269*10^4 Gflops ， 也就是42.69TFlops。

nvida-smi 监控GPU使用率



2.1.6 NCCL

NCCL 是一种多 GPU、多节点通信原语，针对 NVIDIA GPU 通信进行了优化。 NCCL allreduce 是一种非常有效的方式来验证通过各种网络技术（例如 IB、RoCE、TCP/IP）的节点间 GPU 通信。

测试步骤：

1. 从NGC拉取测试用镜像，本示例中使用Pytorch镜像

```
docker pull nvcr.io/nvidia/pytorch:21.06-py3
```

2. 启动容器，挂载本地用户的目录

```
nvidia-docker run -it -v /home/your_user_name:/nccl
nvcr.io/nvidia/pytorch:21.06-py3 bash
```

3. 在容器中，下载nccl-test并编译

```
cd /nccl
git clone https://github.com/nvidia/nccl-tests
cd nccl-tests
make MPI=1 MPI_HOME=/usr/local/mpi
```

4. 开始测试

```
./build/all_reduce_perf -b 8 -e 128M -f 2 -g 8
```

测试命令说明，以上测试命令，是在8个GPU上（-g 8）运行NCCL Test，测试数据包大小从8 Bytes 到128MBytes。

5. 测试结果

测试输出结果

```
root@8e202ffdb108:/nccl# ./build/all_reduce_perf -b 8 -e 128M -f 2 -g 8
# nThread 1 nGpus 8 minBytes 8 maxBytes 134217728 step: 2(factor) warmup iters: 5
iters: 20 validation: 1
#
# Using devices
#   Rank  0 Pid   1913 on 8e202ffdb108 device  0 [0x07] NVIDIA A100-SXM4-40GB
#   Rank  1 Pid   1913 on 8e202ffdb108 device  1 [0x0f] NVIDIA A100-SXM4-40GB
#   Rank  2 Pid   1913 on 8e202ffdb108 device  2 [0x47] NVIDIA A100-SXM4-40GB
#   Rank  3 Pid   1913 on 8e202ffdb108 device  3 [0x4e] NVIDIA A100-SXM4-40GB
#   Rank  4 Pid   1913 on 8e202ffdb108 device  4 [0x87] NVIDIA A100-SXM4-40GB
#   Rank  5 Pid   1913 on 8e202ffdb108 device  5 [0x90] NVIDIA A100-SXM4-40GB
#   Rank  6 Pid   1913 on 8e202ffdb108 device  6 [0xb7] NVIDIA A100-SXM4-40GB
#   Rank  7 Pid   1913 on 8e202ffdb108 device  7 [0xbd] NVIDIA A100-SXM4-40GB
#
#                                                     out-of-place
          in-place
#       size         count      type   redop     time   algbw   busbw  error
time   algbw   busbw  error
#        (B)    (elements)                       (us)  (GB/s)  (GB/s)
(us)  (GB/s)  (GB/s)
           8             2     float     sum    37.47    0.00    0.00  1e-07
32.26    0.00    0.00  1e-07
          16             4     float     sum    32.47    0.00    0.00  1e-07
32.42    0.00    0.00  1e-07
          32             8     float     sum    32.12    0.00    0.00  6e-08
31.92    0.00    0.00  6e-08
          64            16     float     sum    32.23    0.00    0.00  6e-08
32.31    0.00    0.00  6e-08
         128            32     float     sum    32.25    0.00    0.01  6e-08
32.37    0.00    0.01  6e-08
         256            64     float     sum    32.46    0.01    0.01  3e-08
32.08    0.01    0.01  3e-08
         512           128     float     sum    32.13    0.02    0.03  3e-08
32.16    0.02    0.03  3e-08
        1024           256     float     sum    32.44    0.03    0.06  1e-07
32.21    0.03    0.06  1e-07
        2048           512     float     sum    32.27    0.06    0.11  2e-07
32.69    0.06    0.11  2e-07
```

```
        4096           1024     float     sum    32.94    0.12    0.22  2e-07
32.66    0.13    0.22  2e-07
        8192           2048     float     sum    33.26    0.25    0.43  2e-07
32.79    0.25    0.44  2e-07
       16384           4096     float     sum    32.69    0.50    0.88  2e-07
33.05    0.50    0.87  2e-07
       32768           8192     float     sum    32.43    1.01    1.77  2e-07
32.31    1.01    1.77  2e-07
       65536          16384     float     sum    34.47    1.90    3.33  2e-07
34.18    1.92    3.36  2e-07
      131072          32768     float     sum    37.75    3.47    6.08  2e-07
37.91    3.46    6.05  2e-07
      262144          65536     float     sum    43.71    6.00   10.50  2e-07
44.08    5.95   10.41  2e-07
      524288         131072     float     sum    57.06    9.19   16.08  2e-07
56.16    9.34   16.34  2e-07
     1048576         262144     float     sum    70.02   14.97   26.21  2e-07
71.09   14.75   25.81  2e-07
     2097152         524288     float     sum    94.51   22.19   38.83  2e-07
92.27   22.73   39.77  2e-07
     4194304        1048576     float     sum    106.4   39.42   68.98  2e-07
110.2   38.07   66.62  2e-07
     8388608        2097152     float     sum    152.2   55.10   96.43  2e-07
150.5   55.74   97.54  2e-07
    16777216        4194304     float     sum    240.3   69.82  122.18  2e-07
240.2   69.86  122.26  2e-07
    33554432        8388608     float     sum    353.2   95.01  166.26  2e-07
355.7   94.33  165.07  2e-07
    67108864       16777216     float     sum    565.9  118.58  207.52  2e-07
565.6  118.64  207.62  2e-07
   134217728       33554432     float     sum   1115.9  120.28  210.49  2e-07
1115.0  120.37  210.65  2e-07
# Out of bounds values : 0 OK
# Avg bus bandwidth    : 39.028
#
```

测试数据说明，查看128MBytes包大小时，bus bandwidth的值，在此表中，是210.65。

更多详细内容，请查阅nccl-test。

2.2 DL性能测试

2.2.1 RestNet50

在计算机视觉领域，图像分类是非常重要的基本问题，是图像目标检测、图像分割、图像检索、视频理解、物体跟踪、行为分析等其他高层视觉任务的基础，在实际场景中，有着广泛应用。

ResNet是近几年非常流行的卷积神经网络结构，其创造性提出的残差结构，一举在ILSVRC2015比赛中取得冠军，并且获得计算机视觉顶级会议CVPR 2016的最佳论文。其中50层的网络结构(ResNet50)的效果优化，备受学术界和工业界关注。

本测试基于TensorFlow框架，使用随机生成的合成数据集（您无需准备ImageNet数据集），即可快速实现DL模型的Benchmark测试。

测试步骤：

1. 从NGC拉取测试用镜像，本示例中使用TensorFlow镜像

```
docker pull nvcr.io/nvidia/tensorflow:21.07-tf1-py3
```

2. 启动容器

```
nvidia-docker run -it --rm -v $(pwd):/work  nvcr.io/nvidia/tensorflow:21.07-tf1-py3
```

3. 在容器中，进入cnn的目录

```
cd nvidia-examples/cnn/
```

4. 开始测试

   参数说明：

   8节点 -np 8

   Batch-size大小 --batch_size 256

   数据精度 --precision fp16

   以下命令使用8节点，256 Batch-size，fp16测试

```
mpiexec --allow-run-as-root -np 8 --bind-to socket python -u ./resnet.py --batch_size 256 --num_iter 1000 --precision fp16 --iter_unit batch --layers 50
```

   以下命令使用8节点，256 Batch-size，fp32测试

```
mpiexec --allow-run-as-root -np 8 --bind-to socket python -u ./resnet.py --batch_size 256 --num_iter 1000 --precision fp32 --iter_unit batch --layers 50
```

5. 测试结果

   FP16

```
   Step Epoch Img/sec   Loss  LR
    1   1.0    62.9  7.615  8.586 2.00000
    10  10.0   864.7  5.813  6.784 1.62000
    20  20.0 18397.2  2.276  3.252 1.24469
    30  30.0 18408.4  0.012  0.993 0.91877
    40  40.0 18400.0  0.001  0.978 0.64222
    50  50.0 18388.5  0.000  0.970 0.41506
    60  60.0 18417.4  0.000  0.964 0.23728
    70  70.0 18381.2  0.000  0.961 0.10889
    80  80.0 18274.4  0.000  0.960 0.02988
    90  90.0 18367.9  0.000  0.959 0.00025
   100 100.0 18416.5  0.000  0.959 0.00000
   110 110.0 18331.9  0.000  0.959 0.00000
   120 120.0 18430.1  0.000  0.959 0.00000
   130 130.0 18424.1  0.000  0.959 0.00000
   140 140.0 18380.6  0.000  0.959 0.00000
   150 150.0 18396.3  0.000  0.959 0.00000
   160 160.0 18402.8  0.000  0.959 0.00000
```

```
170 170.0 18331.2  0.000  0.959 0.00000
180 180.0 18421.1  0.000  0.959 0.00000
190 190.0 18380.0  0.000  0.959 0.00000
200 200.0 18411.1  0.000  0.959 0.00000
210 210.0 18396.1  0.000  0.959 0.00000
220 220.0 18359.1  0.000  0.959 0.00000
230 230.0 18341.2  0.000  0.959 0.00000
240 240.0 18402.8  0.000  0.959 0.00000
250 250.0 18367.9  0.000  0.959 0.00000
260 260.0 18404.8  0.000  0.959 0.00000
270 270.0 18393.8  0.000  0.959 0.00000
280 280.0 18447.7  0.000  0.959 0.00000
290 290.0 18304.8  0.000  0.959 0.00000
300 300.0 17969.9  0.000  0.959 0.00000
310 310.0 18216.0  0.000  0.959 0.00000
320 320.0 18436.9  0.000  0.959 0.00000
330 330.0 18395.0  0.000  0.959 0.00000
340 340.0 18424.6  0.000  0.959 0.00000
350 350.0 18296.1  0.000  0.959 0.00000
360 360.0 18390.7  0.000  0.959 0.00000
370 370.0 18445.5  0.000  0.959 0.00000
380 380.0 18486.2  0.000  0.959 0.00000
390 390.0 18441.3  0.000  0.959 0.00000
400 400.0 18467.1  0.000  0.959 0.00000
410 410.0 18351.1  0.000  0.959 0.00000
420 420.0 18405.7  0.000  0.959 0.00000
430 430.0 18426.8  0.000  0.959 0.00000
440 440.0 18422.0  0.000  0.959 0.00000
450 450.0 18348.7  0.000  0.959 0.00000
460 460.0 18407.1  0.000  0.959 0.00000
470 470.0 18027.6  0.000  0.959 0.00000
480 480.0 18430.6  0.000  0.959 0.00000
490 490.0 18431.9  0.000  0.959 0.00000
500 500.0 18401.3  0.000  0.959 0.00000
510 510.0 18420.1  0.000  0.959 0.00000
520 520.0 18194.9  0.000  0.959 0.00000
530 530.0 18364.5  0.000  0.959 0.00000
540 540.0 18364.1  0.000  0.959 0.00000
550 550.0 18422.1  0.000  0.959 0.00000
560 560.0 18326.9  0.000  0.959 0.00000
570 570.0 18342.0  0.000  0.959 0.00000
580 580.0 18440.1  0.000  0.959 0.00000
590 590.0 18423.2  0.000  0.959 0.00000
600 600.0 17530.9  0.000  0.959 0.00000
610 610.0 18356.0  0.000  0.959 0.00000
620 620.0 18361.1  0.000  0.959 0.00000
630 630.0 18347.6  0.000  0.959 0.00000
640 640.0 18330.1  0.000  0.959 0.00000
650 650.0 18358.6  0.000  0.959 0.00000
660 660.0 18377.8  0.000  0.959 0.00000
670 670.0 18390.0  0.000  0.959 0.00000
680 680.0 18410.2  0.000  0.959 0.00000
690 690.0 18372.1  0.000  0.959 0.00000
700 700.0 18402.0  0.000  0.959 0.00000
710 710.0 18314.1  0.000  0.959 0.00000
720 720.0 18329.8  0.000  0.959 0.00000
730 730.0 18391.1  0.000  0.959 0.00000
740 740.0 18410.2  0.000  0.959 0.00000
```

```
  750 750.0 18370.0  0.000  0.959 0.00000
  760 760.0 18361.7  0.000  0.959 0.00000
  770 770.0 18386.2  0.000  0.959 0.00000
  780 780.0 18268.2  0.000  0.959 0.00000
  790 790.0 18344.2  0.000  0.959 0.00000
  800 800.0 18322.8  0.000  0.959 0.00000
  810 810.0 18288.0  0.000  0.959 0.00000
  820 820.0 18375.2  0.000  0.959 0.00000
  830 830.0 18351.1  0.000  0.959 0.00000
  840 840.0 18388.9  0.000  0.959 0.00000
  850 850.0 18358.9  0.000  0.959 0.00000
  860 860.0 18316.3  0.000  0.959 0.00000
  870 870.0 18387.7  0.000  0.959 0.00000
  880 880.0 18372.7  0.000  0.959 0.00000
  890 890.0 18406.7  0.000  0.959 0.00000
  900 900.0 18417.4  0.000  0.959 0.00000
  910 910.0 18343.5  0.000  0.959 0.00000
  920 920.0 18425.7  0.000  0.959 0.00000
  930 930.0 18143.7  0.000  0.959 0.00000
  940 940.0 18155.4  0.000  0.959 0.00000
  950 950.0 18392.0  0.000  0.959 0.00000
  960 960.0 18339.7  0.000  0.959 0.00000
  970 970.0 18419.4  0.000  0.959 0.00000
  980 980.0 18418.6  0.000  0.959 0.00000
  990 990.0 18371.1  0.000  0.959 0.00000
 1000 1000.0 10106.4  0.000  0.959 0.00000
```

FP32

```
  Step Epoch Img/sec   Loss  LR

    1   1.0    64.7  7.601  8.572 2.00000
   10  10.0   766.0  5.768  6.739 1.62000
   20  20.0  8215.8  2.142  3.118 1.24469
   30  30.0  8180.2  0.012  0.993 0.91877
   40  40.0  8192.2  0.014  0.992 0.64222
   50  50.0  8198.0  0.005  0.978 0.41506
   60  60.0  8175.7  0.006  0.974 0.23728
   70  70.0  8197.7  0.001  0.966 0.10889
   80  80.0  8123.0  0.000  0.965 0.02988
   90  90.0  8205.5  0.000  0.964 0.00025
  100 100.0  8199.6  0.000  0.964 0.00000
  110 110.0  8181.5  0.000  0.964 0.00000
  120 120.0  8214.5  0.000  0.964 0.00000
  130 130.0  8214.4  0.000  0.964 0.00000
  140 140.0  8214.1  0.000  0.964 0.00000
  150 150.0  8205.2  0.000  0.964 0.00000
  160 160.0  8206.1  0.000  0.964 0.00000
  170 170.0  8211.8  0.000  0.964 0.00000
  180 180.0  8210.8  0.000  0.964 0.00000
  190 190.0  8205.5  0.000  0.964 0.00000
  200 200.0  8189.3  0.000  0.964 0.00000
  210 210.0  8198.8  0.000  0.964 0.00000
  220 220.0  8142.8  0.000  0.964 0.00000
  230 230.0  8210.2  0.000  0.964 0.00000
```

```
240 240.0  8218.1  0.000  0.964 0.00000
250 250.0  8209.2  0.000  0.964 0.00000
260 260.0  8223.6  0.000  0.964 0.00000
270 270.0  8229.4  0.000  0.964 0.00000
280 280.0  8229.0  0.000  0.964 0.00000
290 290.0  8228.6  0.000  0.964 0.00000
300 300.0  8222.0  0.000  0.964 0.00000
310 310.0  8177.5  0.000  0.964 0.00000
320 320.0  8218.4  0.000  0.964 0.00000
330 330.0  8226.9  0.000  0.964 0.00000
340 340.0  8221.6  0.000  0.964 0.00000
350 350.0  8217.6  0.000  0.964 0.00000
360 360.0  8164.7  0.000  0.964 0.00000
370 370.0  8217.7  0.000  0.964 0.00000
380 380.0  8228.3  0.000  0.964 0.00000
390 390.0  8221.1  0.000  0.964 0.00000
400 400.0  8211.2  0.000  0.964 0.00000
410 410.0  8195.0  0.000  0.964 0.00000
420 420.0  8220.9  0.000  0.964 0.00000
430 430.0  8179.9  0.000  0.964 0.00000
440 440.0  8205.1  0.000  0.964 0.00000
450 450.0  8190.1  0.000  0.964 0.00000
460 460.0  8211.8  0.000  0.964 0.00000
470 470.0  8210.7  0.000  0.964 0.00000
480 480.0  8215.3  0.000  0.964 0.00000
490 490.0  8209.4  0.000  0.964 0.00000
500 500.0  8204.0  0.000  0.964 0.00000
510 510.0  8203.3  0.000  0.964 0.00000
520 520.0  8196.7  0.000  0.964 0.00000
530 530.0  8209.8  0.000  0.964 0.00000
540 540.0  8206.7  0.000  0.964 0.00000
550 550.0  8200.4  0.000  0.964 0.00000
560 560.0  8205.1  0.000  0.964 0.00000
570 570.0  8213.2  0.000  0.964 0.00000
580 580.0  8184.8  0.000  0.964 0.00000
590 590.0  8226.5  0.000  0.964 0.00000
600 600.0  8222.3  0.000  0.964 0.00000
610 610.0  8208.4  0.000  0.964 0.00000
620 620.0  8223.7  0.000  0.964 0.00000
630 630.0  8205.7  0.000  0.964 0.00000
640 640.0  8166.8  0.000  0.964 0.00000
650 650.0  8211.4  0.000  0.964 0.00000
660 660.0  8225.2  0.000  0.964 0.00000
670 670.0  8225.7  0.000  0.964 0.00000
680 680.0  8223.6  0.000  0.964 0.00000
690 690.0  8222.3  0.000  0.964 0.00000
700 700.0  8221.6  0.000  0.964 0.00000
710 710.0  8216.9  0.000  0.964 0.00000
720 720.0  8230.1  0.000  0.964 0.00000
730 730.0  8235.4  0.000  0.964 0.00000
740 740.0  8233.4  0.000  0.964 0.00000
750 750.0  8236.4  0.000  0.964 0.00000
760 760.0  8245.7  0.000  0.964 0.00000
770 770.0  8222.7  0.000  0.964 0.00000
780 780.0  8239.7  0.000  0.964 0.00000
790 790.0  8226.6  0.000  0.964 0.00000
800 800.0  8224.0  0.000  0.964 0.00000
810 810.0  8210.0  0.000  0.964 0.00000
```

```
 820 820.0  8205.6  0.000  0.964 0.00000
 830 830.0  8229.4  0.000  0.964 0.00000
 840 840.0  8225.9  0.000  0.964 0.00000
 850 850.0  8223.6  0.000  0.964 0.00000
 860 860.0  8230.3  0.000  0.964 0.00000
 870 870.0  8220.6  0.000  0.964 0.00000
 880 880.0  8232.5  0.000  0.964 0.00000
 890 890.0  8218.1  0.000  0.964 0.00000
 900 900.0  8227.2  0.000  0.964 0.00000
 910 910.0  8153.3  0.000  0.964 0.00000
 920 920.0  8233.7  0.000  0.964 0.00000
 930 930.0  8241.7  0.000  0.964 0.00000
 940 940.0  8179.4  0.000  0.964 0.00000
 950 950.0  8228.2  0.000  0.964 0.00000
 960 960.0  8225.7  0.000  0.964 0.00000
 970 970.0  8228.8  0.000  0.964 0.00000
 980 980.0  8233.8  0.000  0.964 0.00000
 990 990.0  8241.3  0.000  0.964 0.00000
1000 1000.0  6167.5  0.000  0.964 0.00000
```

3. 多节点测试

本测试基于2台NVIDIA DGX-1服务器，基础硬件环境配置如下：

1	GPU	8 个 NVIDIA V100 16GB GPU
3	处理器	两个 Intel Xeon E5-2698 V 4、共 40 个核心、2.2 GHz（基准频率）
4	内存	512 GB 2133MHz DDR4
5	存储	4X 1.92 TB SSD RAID 0
6	网络	Dual 10 GbE, 4 IB EDR

基础软件环境配置如下：

1	操作系统	Ubuntu 20.04.3 LTS
2	GPU driver	470.57.02 / 270.82.01
3	docker	20.10.12

两台DGX-1服务器，通过100G 以太网互连，通过RoCE进行跨设备多卡训练。

3.1 基础性能测试

3.1.1 HPL

待完善。

3.1.2 NCCL

请查阅《基于Horovod的分布式训练测试指导手册》，2.4.2章节。

3.2 DL性能测试

请查阅《基于Horovod的分布式训练测试指导手册》，2.4.2章节。