<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TensorFlow Overview</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Pulling A Container</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Running TensorFlow</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>TensorFlow Release 19.04</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>TensorFlow Release 19.03</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>TensorFlow Release 19.02</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>TensorFlow Release 19.01</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>TensorFlow Release 18.12</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>TensorFlow Release 18.11</td>
<td>23</td>
</tr>
<tr>
<td>10</td>
<td>TensorFlow Release 18.10</td>
<td>26</td>
</tr>
<tr>
<td>11</td>
<td>TensorFlow Release 18.09</td>
<td>28</td>
</tr>
<tr>
<td>12</td>
<td>TensorFlow Release 18.08</td>
<td>31</td>
</tr>
<tr>
<td>13</td>
<td>TensorFlow Release 18.07</td>
<td>33</td>
</tr>
<tr>
<td>14</td>
<td>TensorFlow Release 18.06</td>
<td>35</td>
</tr>
<tr>
<td>15</td>
<td>TensorFlow Release 18.05</td>
<td>37</td>
</tr>
<tr>
<td>16</td>
<td>TensorFlow Release 18.04</td>
<td>39</td>
</tr>
<tr>
<td>17</td>
<td>TensorFlow Release 18.03</td>
<td>41</td>
</tr>
<tr>
<td>18</td>
<td>TensorFlow Release 18.02</td>
<td>43</td>
</tr>
<tr>
<td>19</td>
<td>TensorFlow Release 18.01</td>
<td>45</td>
</tr>
<tr>
<td>20</td>
<td>TensorFlow Release 17.12</td>
<td>47</td>
</tr>
<tr>
<td>21</td>
<td>TensorFlow Release 17.11</td>
<td>49</td>
</tr>
<tr>
<td>22</td>
<td>TensorFlow Release 17.10</td>
<td>51</td>
</tr>
<tr>
<td>23</td>
<td>TensorFlow Release 17.09</td>
<td>53</td>
</tr>
<tr>
<td>24</td>
<td>TensorFlow Release 17.07</td>
<td>55</td>
</tr>
<tr>
<td>25</td>
<td>TensorFlow Release 17.06</td>
<td>56</td>
</tr>
<tr>
<td>26</td>
<td>TensorFlow Release 17.05</td>
<td>57</td>
</tr>
<tr>
<td>27</td>
<td>TensorFlow Release 17.04</td>
<td>59</td>
</tr>
<tr>
<td>28</td>
<td>TensorFlow Release 17.03</td>
<td>61</td>
</tr>
<tr>
<td>29</td>
<td>TensorFlow Release 17.02</td>
<td>62</td>
</tr>
<tr>
<td>30</td>
<td>TensorFlow Release 17.01</td>
<td>63</td>
</tr>
<tr>
<td>31</td>
<td>TensorFlow Release 16.12</td>
<td>64</td>
</tr>
</tbody>
</table>
Chapter 1.
TENSORFLOW OVERVIEW

The NVIDIA Deep Learning SDK accelerates widely-used deep learning frameworks such as TensorFlow™.

TensorFlow 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.

This document describes the key features, software enhancements and improvements, any known issues, and how to run this container.
Chapter 2.
PULLING A CONTAINER

Before you can pull a container from the NGC container registry, you must have Docker and nvidia-docker installed. For DGX users, this is explained in Preparing to use NVIDIA Containers Getting Started Guide.

For users other than DGX, follow the NVIDIA® GPU Cloud™ (NGC) container registry nvidia-docker installation documentation based on your platform.

You must also have access and be logged into the NGC container registry as explained in the NGC Getting Started Guide.

There are four repositories where you can find the NGC docker containers.

nvcr.io/nvidia
   The deep learning framework containers are stored in the nvcr.io/nvidia repository.

nvcr.io/hpc
   The HPC containers are stored in the nvcr.io/hpc repository.

nvcr.io/nvidia-hpcvis
   The HPC visualization containers are stored in the nvcr.io/nvidia-hpcvis repository.

nvcr.io/partner
   The partner containers are stored in the nvcr.io/partner repository. Currently the partner containers are focused on Deep Learning or Machine Learning, but that doesn’t mean they are limited to those types of containers.
Chapter 3. 
RUNNING TENSORFLOW

Before running the container, use the docker pull command to ensure an up-to-date image is installed. Once the pull is complete, you can run the container image. This is because nvidia-docker ensures that drivers that match the host are used and configured for the container. Without nvidia-docker, you are likely to get an error when trying to run the container.

1. Issue the command for the applicable release of the container that you want. The following command assumes you want to pull the latest container.

   ```bash
   docker pull nvcr.io/nvidia/tensorflow:19.04-py<x>
   ```

2. Open a command prompt and paste the pull command. The pulling of the container image begins. Ensure the pull completes successfully before proceeding to the next step.

3. Run the container image. A typical command to launch the container is:

   ```bash
   nvidia-docker run -it --rm -v local_dir:container_dir
   nvcr.io/nvidia/tensorflow:<xx.xx>-py<x>
   ```

   Where:
   - `-it` means interactive
   - `--rm` means delete the image when finished
   - `-v` means mount directory
   - `local_dir` is the directory or file from your host system (absolute path) that you want to access from inside your container. For example, the `local_dir` in the following path is `/home/jsmith/data/mnist`.

   ```bash
   -v /home/jsmith/data/mnist:/data/mnist
   ```

   If you are inside the container, for example, `ls /data/mnist`, you will see the same files as if you issued the `ls /home/jsmith/data/mnist` command from outside the container.

   `container_dir` is the target directory when you are inside your container. For example, `/data/mnist` is the target directory in the example:
-v /home/jsmith/data/mnist:/data/mnist

- `<xx.xx>` is the container version. For example, `19.01`.
- `<x>` is the Python version. For example, `py3`.

TensorFlow is run by importing it as a Python module:

```
$ python
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> sess.run(hello)
Hello, TensorFlow!
>>> a = tf.constant(10)
>>> b = tf.constant(32)
>>> sess.run(a+b)
42
```

You might want to pull in data and model descriptions from locations outside the container for use by TensorFlow or save results to locations outside the container. To accomplish this, the easiest method is to mount one or more host directories as Docker® data volumes.

In order to share data between ranks, NVIDIA® Collective Communications Library™ (NCCL) may require shared system memory for IPC and pinned (page-locked) system memory resources. The operating system’s limits on these resources may need to be increased accordingly. Refer to your system’s documentation for details.

In particular, Docker containers default to limited shared and pinned memory resources. When using NCCL inside a container, it is recommended that you increase these resources by issuing:

```
--shm-size=1g --ulimit memlock=-1
```
in the command line to `nvidia-docker run`

4. See `/workspace/README.md` inside the container for information on customizing your TensorFlow image.

For more information about TensorFlow, including tutorials, documentation, and examples, see:

- TensorFlow tutorials
- TensorFlow API
The NVIDIA container image of TensorFlow, release 19.04, is available on NGC.

**Contents of the TensorFlow container**

This container image contains the complete source of the version of NVIDIA TensorFlow in `/opt/tensorflow`. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04

  Container image 19.04-py2 contains Python 2.7; 19.04-py3 contains Python 3.5.

- NVIDIA CUDA 10.1.105 including cuBLAS 10.1.0.105
- NVIDIA cuDNN 7.5.0
- NVIDIA NCCL 2.4.6 (optimized for NVLink™)
- Horovod 0.16.1
- OpenMPI 3.1.3
- TensorFlow 1.13.1
- MLNX_OFED 3.4
- OpenSeq2Seq at commit 6e8835f
- TensorRT 5.1.2
- DALI 0.8.1 Beta
- Tensor core optimized example:
  - SSD320 v1.2
  - Neural Collaborative Filtering (NCF)
Bert
U-Net Industrial Defect Segmentation
GNMT v2
ResNet-50 v1.5

Jupyter and JupyterLab:

Jupyter Client 5.2.4
Jupyter Core 4.4.0
JupyterLab 0.35.4
JupyterLab Server 0.2.0

Driver Requirements

Release 19.04 is based on CUDA 10.1, which requires NVIDIA Driver release 418.xx.x+. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384.111+ or 410. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.04 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- Added the GNMT v2, U-Net Industrial Defect Segmentation, Bert, Neural Collaborative Filtering (NCF), and SSD320 v1.2 tensor core examples
- Latest version of NVIDIA NCCL 2.4.6
- Latest version of cuBLAS 10.1.0.105
- Latest version of DALI 0.8.1 Beta
- Latest version of Horovod 0.16.1
- Improved stability for auto-tuning of fastest convolutional algorithms.
- Ubuntu 16.04 with March 2019 updates
Accelerating Inference In TensorFlow With TensorRT (TF-TRT)

For step-by-step instructions on how to use TF-TRT, see Accelerating Inference In TensorFlow With TensorRT User Guide. To view the key features, software enhancements and improvements, and known issues, see the Release Notes.

The TF-TRT examples were relocated from /workspace/nvidia-examples/tftrt/scripts/ to workspace/nvidia-examples/inference.

Automatic Mixed Precision (AMP)

Automatic mixed precision converts certain float32 operations to operate in float16 which can run much faster on tensor cores. Automatic mixed precision is built on two components:

- a loss scaling optimizer
- graph rewriter

For models already using a `tf.Optimizer()` for both `compute_gradients()` and `apply_gradients()` operations, automatic mixed precision can be enabled by defining the following environment variable before calling the usual float32 training script:

```
export TF_ENABLE_AUTO_MIXED_PRECISION=1
```

Models implementing their own optimizers can use the graph rewriter on its own (while implementing loss scaling manually) with the following environment variable:

```
export TF_ENABLE_AUTO_MIXED_PRECISION_GRAPH_REWRITE=1
```

For more information about how to access and enable Automatic mixed precision for TensorFlow, see Automatic Mixed Precision Training In TensorFlow from the TensorFlow User Guide, along with Training With Mixed Precision.

Tensor Core Examples

These examples focus on achieving the best performance and convergence from NVIDIA Volta tensor cores by using the latest deep learning example networks for training.

Each example model trains with mixed precision tensor cores on Volta, therefore you can get results much faster than training without tensor cores. This model is tested against each NGC monthly container release to ensure consistent accuracy and performance over time. This container includes the following tensor core examples.

- An implementation of the SSD320 v1.2 model. The SSD320 v1.2 model is based on the SSD: Single Shot MultiBox Detector paper, which describes SSD as “a method for detecting objects in images using a single deep neural network”. Our implementation is based on the existing model from the TensorFlow models repository.
An implementation of the Neural Collaborative Filtering (NCF) model. The NCF model is a neural network that provides collaborative filtering based on implicit feedback, specifically, it provides product recommendations based on user and item interactions. The training data for this model should contain a sequence of user ID, item ID pairs indicating that the specified user has interacted with, for example, was given a rating to or clicked on, the specified item.

An implementation of the Bert model. BERT, or Bidirectional Encoder Representations from Transformers, is a new method of pre-training language representations which obtains state-of-the-art results on a wide array of Natural Language Processing (NLP) tasks. This model is based on BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding paper. NVIDIA’s BERT is an optimized version of Google’s official implementation, leveraging mixed precision arithmetic and tensor cores on V100 GPUs for faster training times while maintaining target accuracy.

An implementation of the U-Net Industrial Defect Segmentation model. This U-Net model is adapted from the original version of the U-Net model which is a convolutional auto-encoder for 2D image segmentation. U-Net was first introduced by Olaf Ronneberger, Philip Fischer, and Thomas Brox in the paper: U-Net: Convolutional Networks for Biomedical Image Segmentation. This work proposes a modified version of U-Net, called TinyUNet which performs efficiently and with very high accuracy on the industrial anomaly dataset DAGM2007.

An implementation of the GNMT v2 model. The GNMT v2 model is similar to the one discussed in the Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation paper. The most important difference between the two models is in the attention mechanism. In our model, the output from the first LSTM layer of the decoder goes into the attention module, then the re-weighted context is concatenated with inputs to all subsequent LSTM layers in the decoder at the current timestep.

An implementation of the ResNet-50 v1.5 model. The ResNet-50 v1.5 model is a modified version of the original ResNet-50 v1 model. The difference between v1 and v1.5 is in the bottleneck blocks which requires downsampling, for example, v1 has stride = 2 in the first 1x1 convolution, whereas v1.5 has stride = 2 in the 3x3 convolution. The following features were implemented in this model; data-parallel multi-GPU training with Horovod, tensor cores (mixed precision) training, and static loss scaling for tensor cores (mixed precision) training.

Known Issues

- There is a known performance regression with TensorFlow 1.13.1 for some networks when run with small batch sizes. As a workaround, increase the batch size.
- The AMP preview implementation is not compatible with Distributed Strategies. We recommend using Horovod for parallel training with AMP.
A known issue in TensorFlow results in the error **Cannot take the length of Shape with unknown rank** when training variable sized images with the Keras `model.fit` API. Details are provided [here](#) and a fix will be available in a future release.
Chapter 5.
TENSORFLOW RELEASE 19.03

The NVIDIA container image of TensorFlow, release 19.03, is available on NGC.

Contents of the TensorFlow container

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 10.1.105 including cuBLAS 10.1.105
- NVIDIA cuDNN 7.5.0
- NVIDIA NCCL 2.4.3 (optimized for NVLink™)
- Horovod 0.16.0
- OpenMPI 3.1.3
- TensorBoard 1.13.1
- MLNX_OFED 3.4
- OpenSeq2Seq at commit 6e8835f
- TensorRT 5.1.2
- DALI 0.7 Beta
- Tensor core optimized example:
  - ResNet-50 v1.5
- Jupyter and JupyterLab:
Driver Requirements

Release 19.03 is based on CUDA 10.1, which requires NVIDIA Driver release 418.xx+. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384.111+ or 410. The CUDA driver’s compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.03 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- TensorFlow container image version 19.03 is based on TensorFlow 1.13.1.
- Latest version of NVIDIA CUDA 10.1.105 including cuBLAS 10.1.105
- Latest version of NVIDIA cuDNN 7.5.0
- Latest version of NVIDIA NCCL 2.4.3
- Latest version of DALI 0.7 Beta
- Latest version of TensorRT 5.1.2
- Latest version of Horovod 0.16.0
- Latest version of TensorBoard 1.13.1
- Added the ResNet-50 v1.5 tensor core example
- Added support for TensorFlow Automatic Mixed Precision (TF-AMP); see below for more information.
- Ubuntu 16.04 with February 2019 updates
Accelerating Inference In TensorFlow With TensorRT (TF-TRT)

For step-by-step instructions on how to use TF-TRT, see Accelerating Inference In TensorFlow With TensorRT User Guide. To view the key features, software enhancements and improvements, and known issues, see the Release Notes.

The TF-TRT examples were relocated from /workspace/nvidia-examples/tftrt/scripts/ to workspace/nvidia-examples/inference.

Announcements

TensorRT 3.x is not longer supported, therefore, models that were accelerated using TensorRT 3.x will no longer run. If you have a production model that was accelerated with TensorRT 3.x, you will need to convert your model with TensorRT 5.x or later again.

For more information, see the Note in Serializing A Model In C++ or Serializing A Model In Python.

Automatic Mixed Precision (AMP)

Automatic mixed precision converts certain float32 operations to operate in float16 which can run much faster on tensor cores. Automatic mixed precision is built on two components:

- a loss scaling optimizer
- graph rewriter

For models already using a `tf.Optimizer()` for both `compute_gradients()` and `apply_gradients()` operations, automatic mixed precision can be enabled by defining the following environment variable before calling the usual float32 training script:

```bash
export TF_ENABLE_AUTO_MIXED_PRECISION=1
```

Models implementing their own optimizers can use the graph rewriter on its own (while implementing loss scaling manually) with the following environment variable:

```bash
export TF_ENABLE_AUTO_MIXED_PRECISION_GRAPH_REWRITE=1
```

For more information about how to access and enable Automatic mixed precision for TensorFlow, see Automatic Mixed Precision Training In TensorFlow from the TensorFlow User Guide, along with Training With Mixed Precision.

Tensor Core Examples

These examples focus on achieving the best performance and convergence from NVIDIA Volta tensor cores by using the latest deep learning example networks for training.
Each example model trains with mixed precision tensor cores on Volta, therefore you can get results much faster than training without tensor cores. This model is tested against each NGC monthly container release to ensure consistent accuracy and performance over time. This container includes the following tensor core examples.

- An implementation of the ResNet-50 v1.5 model. The ResNet-50 v1.5 model is a modified version of the original ResNet-50 v1 model. The difference between v1 and v1.5 is in the bottleneck blocks which requires downsampling, for example, v1 has stride = 2 in the first 1x1 convolution, whereas v1.5 has stride = 2 in the 3x3 convolution. The following features were implemented in this model; data-parallel multi-GPU training with Horovod, tensor cores (mixed precision) training, and static loss scaling for tensor cores (mixed precision) training.

**Known Issues**

- There is a known performance regression with TensorFlow 1.13.1 for some networks when run with small batch sizes. As a workaround, increase the batch size.
- The AMP preview implementation is not compatible with Distributed Strategies. We recommend using Horovod for parallel training with AMP.
- If using or upgrading to a 3-part-version driver, for example, a driver that takes the format of `xxx.yy.zz`, you will receive a **Failed to detect NVIDIA driver version** message. This is due to a known bug in the entry point script's parsing of 3-part driver versions. This message is non-fatal and can be ignored. This will be fixed in the 19.04 release.
The NVIDIA container image of TensorFlow, release 19.02, is available.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in `/opt/tensorflow`. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.4.2
- NVIDIA Collective Communications Library (NCCL) 2.3.7 (optimized for NVLink™)
- Horovod 0.15.1
- OpenMPI 3.1.3
- TensorBoard 1.12.2
- MLNX_OFED 3.4
- OpenSeq2Seq v18.12 at commit 59c70e7
- TensorRT 5.0.2
- DALI 0.6.1 Beta
- Jupyter and JupyterLab:
Driver Requirements

Release 19.02 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.02 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- TensorFlow container image version 19.02 is based on TensorFlow 1.13.0-rc0.
- Latest version of DALI 0.6.1 Beta
- Latest version of TensorBoard 1.12.2
- Added Jupyter and JupyterLab software in our packaged container.
- Latest version of jupyter_client 5.2.4
- Latest version of jupyter_core 4.4.0
- Ubuntu 16.04 with January 2019 updates

Accelerating Inference In TensorFlow With TensorRT

For step-by-step instructions on how to use TF-TRT, see Accelerating Inference In TensorFlow With TensorRT User Guide. To view the key features, software enhancements and improvements, and known issues, see the Release Notes.

Announcements

TensorRT 3.x is not longer supported, therefore, models that were accelerated using TensorRT 3.x will no longer run. If you have a production model that was accelerated
with TensorRT 3.x, you will need to convert your model with TensorRT 5.x or later again.

For more information, see the Note in Serializing A Model In C++ or Serializing A Model In Python.

Known Issues

- Horovod and XLA cannot be used together due to a known issue in upstream TensorFlow. We expect this to be resolved in an upcoming release.
- There is a known performance regression with TensorFlow 1.13.0-rc0 for some networks when run with small batch sizes. As a workaround, increase the batch size.
- If using or upgrading to a 3-part-version driver, for example, a driver that takes the format of \texttt{xxx.yy.zz}, you will receive a \texttt{Failed to detect NVIDIA driver version}. message. This is due to a known bug in the entry point script's parsing of 3-part driver versions. This message is non-fatal and can be ignored. This will be fixed in the 19.04 release.
Chapter 7.
TENSORFLOW RELEASE 19.01

The NVIDIA container image of TensorFlow, release 19.01, is available.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.4.2
- NCCL 2.3.7 (optimized for NVLink™)
- Horovod 0.15.1
- OpenMPI 3.1.3
- TensorBoard 1.12.0
- MLNX_OFED 3.4
- OpenSeq2Seq v18.12 at commit 59c70e7
- TensorRT 5.0.2
- DALI 0.6 Beta
Driver Requirements

Release 19.01 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.01 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

‣ TensorFlow container image version 19.01 is based on TensorFlow 1.12.0.
‣ Latest version of DALI 0.6 Beta
‣ Latest version of NVIDIA cuDNN 7.4.2
‣ Latest version of OpenMPI 3.1.3
‣ Ubuntu 16.04 with December 2018 updates

Accelerating Inference In TensorFlow With TensorRT

For step-by-step instructions on how to use TF-TRT, see Accelerating Inference In TensorFlow With TensorRT User Guide. To view the key features, software enhancements and improvements, and known issues, see the Release Notes.

The TF-TRT examples were relocated from /workspace/nvidia-examples/tftrt/scripts/ to workspace/nvidia-examples/inference.

Announcements

Support for accelerating TensorFlow with TensorRT 3.x will be removed in a future release (likely TensorFlow 1.13). The generated plan files are not portable across platforms or TensorRT versions. Plans are specific to the exact GPU model they were built on (in addition to platforms and the TensorRT version) and must be retargeted to the specific GPU in case you want to run them on a different GPU. Therefore, models that were accelerated using TensorRT 3.x will no longer run. If you have a production model that was accelerated with TensorRT 3.x, you will need to convert your model with TensorRT 4.x or later again.

For more information, see the Note in Serializing A Model In C++ or Serializing A Model In Python.
Known Issues

- Horovod and XLA cannot be used together due to a known issue in upstream TensorFlow. We expect this to be resolved in an upcoming release.

- If using or upgrading to a 3-part-version driver, for example, a driver that takes the format of xxx.yy.zz, you will receive a Failed to detect NVIDIA driver version message. This is due to a known bug in the entry point script's parsing of 3-part driver versions. This message is non-fatal and can be ignored. This will be fixed in the 19.04 release.
Chapter 8.
TENSORFLOW RELEASE 18.12

The NVIDIA container image of TensorFlow, release 18.12, is available.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.4.1
- NCCL 2.3.7 (optimized for NVLink™)
- Horovod 0.15.1
- OpenMPI 3.1.2
- TensorFlow 1.12.0
- MLNX_OFED 3.4
- OpenSeq2Seq v18.12 at commit 59c70e7
- TensorRT 5.0.2
- DALI 0.5.0 Beta
Driver Requirements

Release 18.12 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 18.12 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- TensorFlow container image version 18.12 is based on TensorFlow 1.12.0.
- Latest version of DALI 0.5.0 Beta.
- OpenSeq2Seq’s custom CTC decoder is now pre-built in the container.
- The tensorflow.contrib.nccl module has been moved into core as tensorflow.python.ops.nccl_ops. User scripts may need to be updated accordingly. No changes are required for Horovod users. For an example of using Horovod, refer to the nvidia-examples/cnn/ directory.
- Inference image classification examples have been removed from the container and are now available at: GitHub: TensorFlow/TensorRT Integration.
- Ubuntu 16.04 with November 2018 updates

Accelerating Inference In TensorFlow With TensorRT

For step-by-step instructions on how to use TF-TRT, see Accelerating Inference In TensorFlow With TensorRT User Guide. To view the key features, software enhancements and improvements, and known issues, see the Release Notes.

The TF-TRT examples were relocated from /workspace/nvidia-examples/tftrt/scripts/ to workspace/nvidia-examples/inference.

Announcements

Support for accelerating TensorFlow with TensorRT 3.x will be removed in a future release (likely TensorFlow 1.13). The generated plan files are not portable across platforms or TensorRT versions. Plans are specific to the exact GPU model they were built on (in addition to platforms and the TensorRT version) and must be retargeted to the specific GPU in case you want to run them on a different GPU. Therefore, models
that were accelerated using TensorRT 3.x will no longer run. If you have a production model that was accelerated with TensorRT 3.x, you will need to convert your model with TensorRT 4.x or later again.

For more information, see the Note in Serializing A Model In C++ or Serializing A Model In Python.

Known Issues

- OpenSeq2Seq is only supported in the Python 3 container.
- Horovod and XLA cannot be used together due to a known issue in upstream TensorFlow. We expect this to be resolved in an upcoming release.
Chapter 9.
TENSORFLOW RELEASE 18.11

The NVIDIA container image of TensorFlow, release 18.11, is available.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04

Container image 18.11-py2 contains Python 2.7; 18.11-py3 contains Python 3.5.

- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.4.1
- NCCL 2.3.7 (optimized for NVLink™)
- Horovod 0.15.1
- OpenMPI 3.1.2
- TensorFlow 1.12.0
- MLNX_OFED 3.4
- OpenSeq2Seq v18.11 at commit 4b95346
- TensorRT 5.0.2
- DALI 0.4.1 Beta
Driver Requirements

Release 18.11 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- TensorFlow container image version 18.11 is based on TensorFlow 1.12.0-rc2.
- Latest version of Horovod 0.15.1.
- Latest version of NCCL 2.3.7.
- Latest version of NVIDIA cuDNN 7.4.1.
- Latest version of TensorFlow 5.0.2
- Latest version of DALI 0.4.1 Beta.
- Bug fixes and improvements for TensorFlow-TensorRT (TF-TRT) integration.
- Added an object detection example to workspace/nvidia-examples/inference/object-detection.
- Ubuntu 16.04 with October 2018 updates

Accelerating Inference In TensorFlow With TensorRT

For step-by-step instructions on how to use TF-TRT, see Accelerating Inference In TensorFlow With TensorRT User Guide. To view the key features, software enhancements and improvements, and known issues, see the Release Notes.

The TF-TRT examples were relocated from /workspace/nvidia-examples/tftrt/scripts/ to workspace/nvidia-examples/inference.

Announcements

Support for accelerating TensorFlow with TensorRT 3.x will be removed in a future release (likely TensorFlow 1.13). The generated plan files are not portable across platforms or TensorRT versions. Plans are specific to the exact GPU model they were built on (in addition to platforms and the TensorRT version) and must be retargeted to the specific GPU in case you want to run them on a different GPU. Therefore, models that were accelerated using TensorRT 3.x will no longer run. If you have a production model that was accelerated with TensorRT 3.x, you will need to convert your model with TensorRT 4.x or later again.

For more information, see the Note in Serializing A Model In C++ or Serializing A Model In Python.
Known Issues
OpenSeq2Seq is only supported in the Python 3 container.
Chapter 10.
TENSORFLOW RELEASE 18.10

The NVIDIA container image of TensorFlow, release 18.10, is available.

**Contents of TensorFlow**

This container image contains the complete source of the version of NVIDIA TensorFlow in `/opt/tensorflow`. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.4.0
- NCCL 2.3.6 (optimized for NVLink™)
- Horovod 0.13.10
- OpenMPI 3.1.2
- TensorBoard 1.10.0
- MLNX_OFED 3.4
- OpenSeq2Seq v18.10 at commit 655eb65
- TensorRT 5.0.0 RC
- DALI 0.4 Beta
**Driver Requirements**

Release 18.10 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

**Key Features and Enhancements**

This TensorFlow release includes the following key features and enhancements.

- TensorFlow container image version 18.10 is based on TensorFlow 1.10.0.
- Latest version of NCCL 2.3.6.
- Latest version of DALI 0.4 Beta
- Latest version of OpenMPI 3.1.2
- Fixed a bug in the ResNet example script when using NHWC data format.
- Fixed several issues when accelerating inference in TensorFlow with TensorRT including support for ReLu6, Identity, and dilated convolutions.
- Ubuntu 16.04 with September 2018 updates

**Accelerating Inference In TensorFlow With TensorRT**

For step-by-step instructions on how to use TF-TRT, see Accelerating Inference In TensorFlow With TensorRT User Guide. To view the key features, software enhancements and improvements, and known issues, see the Release Notes.

**Known Issues**

OpenSeq2Seq is only supported in the Python 3 container.
Chapter 11.
TENSORFLOW RELEASE 18.09

The NVIDIA container image of TensorFlow, release 18.09, is available.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- Container image 18.09–py2 contains Python 2.7; 18.09–py3 contains Python 3.5.
- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.3.0
- NCCL 2.3.4 (optimized for NVLink™)
- Horovod™ 0.13.10
- OpenMPI 3.0.0
- TensorBoard 1.10.0
- MLNX_OFED 3.4
- OpenSeq2Seq v18.09 at commit 694a230
- TensorRT 5.0.0 RC
- DALI 0.2 Beta
Driver Requirements

Release 18.09 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

‣ TensorFlow container image version 18.09 is based on TensorFlow 1.10.0.
‣ Latest version of cuDNN 7.3.0.
‣ Latest version of CUDA 10.0.130 which includes support for DGX-2, Turing, and Jetson Xavier.
‣ Latest version of cuBLAS 10.0.130.
‣ Latest version of NCCL 2.3.4.
‣ Latest version of TensorRT 5.0.0 RC.
‣ Latest version of TensorBoard 1.10.0.
‣ Latest version of DALI 0.2 Beta
‣ Added support for CUDNN float32 Tensor Op Math mode, which enables float32 models to use Tensor Cores on supported hardware, at the cost of reduced precision. This is disabled by default, but can be enabled by setting the environment variables `TF_ENABLE_CUDNN_TENSOR_OP_MATH_FP32=1` (for convolutions) or `TF_ENABLE_CUDNN_RNN_TENSOR_OP_MATH_FP32=1` (for RNNs that use the `cudnn_rnn` op). This feature is currently considered experimental.
‣ Renamed the existing environment variable `TF_ENABLE_TENSOR_OP_MATH_FP32` to `TF_ENABLE_CUBLAS_TENSOR_OP_MATH_FP32`.

When using any of the `TF_ENABLE_*_TENSOR_OP_MATH_FP32` environment variables, it is recommended that models also use loss scaling to avoid numerical issues during training. For more information about loss scaling, see Training With Mixed Precision.

‣ Enhanced `tf.contrib.layers.layer_norm` by adding a `use_fused_batch_norm` parameter that improves performance. This parameter is disabled by default, but can be enabled by setting it to `True`.
‣ Ubuntu 16.04 with August 2018 updates

Accelerating Inference In TensorFlow With TensorRT

For step-by-step instructions on how to use TF-TRT, see Accelerating Inference In TensorFlow With TensorRT User Guide. To view the key features, software enhancements and improvements, and known issues, see the Release Notes.
Known Issues

- OpenSeq2Seq is only supported in the Python 3 container.
- The build_imagenet_data scripts have a missing dependency on the axel application. This can be resolved by issuing the following command:

```bash
apt-get update &&
apt-get install axel
```
Chapter 12.
TENSORFLOW RELEASE 18.08

The NVIDIA container image of TensorFlow, release 18.08, is available.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04

<table>
<thead>
<tr>
<th>Container image 18.08–py2 contains Python 2.7; 18.08–py3 contains Python 3.5.</th>
</tr>
</thead>
<tbody>
<tr>
<td>- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.425</td>
</tr>
<tr>
<td>- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.2.1</td>
</tr>
<tr>
<td>- NCCL 2.2.13 (optimized for NVLink™)</td>
</tr>
<tr>
<td>- Horovod™ 0.12.1</td>
</tr>
<tr>
<td>- OpenMPI™ 3.0.0</td>
</tr>
<tr>
<td>- TensorBoard 1.9.0</td>
</tr>
<tr>
<td>- MLNX_OFED 3.4</td>
</tr>
<tr>
<td>- OpenSeq2Seq v0.5 at commit 83e96551.</td>
</tr>
<tr>
<td>- TensorRT 4.0.1</td>
</tr>
<tr>
<td>- DALI 0.1.2 Beta</td>
</tr>
</tbody>
</table>
Driver Requirements

Release 18.08 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

‣ TensorFlow container image version 18.08 is based on TensorFlow 1.9.0.
‣ Latest version of cuDNN 7.2.1.
‣ Latest version of DALI 0.1.2 Beta.
‣ Latest version of TensorBoard 1.9.0.
‣ Added experimental support for float16 data type in Horovod, allowing functions such as `all_reduce` to accept tensors in float16 precision. (This functionality is not yet integrated into multi-GPU training examples).
‣ Ubuntu 16.04 with July 2018 updates

Announcements

Starting with the next major version of CUDA release, we will no longer provide updated Python 2 containers and will only update Python 3 containers.

Known Issues

‣ The DALI integrated ResNet-50 samples in the 18.08 NGC TensorFlow container has lower than expected accuracy and performance results. We are working to address the issue in the next release.

‣ There is a known performance regression in the inference benchmarks for ResNet-50. We haven't seen this regression in the inference benchmarks for VGG or training benchmarks for any network. The cause of the regression is still under investigation.
Chapter 13.
TENSORFLOW RELEASE 18.07

The NVIDIA container image of TensorFlow, release 18.07, is available.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.425
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.4
- NCCL 2.2.13 (optimized for NVLink™)
- Horovod™ 0.12.1
- OpenMPI™ 3.0.0
- TensorFlow 1.8.0
- MLNX_OFED 3.4
- OpenSeq2Seq v0.4 at commit 98ad236a.
- TensorRT 4.0.1
- DALI 0.1 Beta
Driver Requirements
Release 18.07 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements
This TensorFlow release includes the following key features and enhancements.

‣ TensorFlow container image version 18.07 is based on TensorFlow 1.8.0.
‣ Added support for DALI 0.1 Beta.
‣ Latest version of CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.425.
‣ Ubuntu 16.04 with June 2018 updates

Announcements
Starting with the next major version of CUDA release, we will no longer provide updated Python 2 containers and will only update Python 3 containers.

Known Issues
There are no known issues in this release.
Chapter 14.
TENSORFLOW RELEASE 18.06

The NVIDIA container image of TensorFlow, release 18.06, is available.

**Contents of TensorFlow**

This container image contains the complete source of the version of NVIDIA TensorFlow in `/opt/tensorflow`. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04

  Container image 18.06-py2 contains Python 2.7; 18.06-py3 contains Python 3.5.

- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.333 (see section 2.3.1)
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.4
- NCCL 2.2.13 (optimized for NVLink™)
- Horovod™ 0.12.1
- OpenMPI™ 3.0.0
- TensorFlow 1.8.0
- MLNX_OFED 3.4
- OpenSeq2Seq v0.2 at commit a4f627e
- TensorRT 4.0.1

**Driver Requirements**

Release 18.06 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.
Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- TensorFlow container image version 18.06 is based on TensorFlow 1.8.0.
- Updated scripts and README in `nvidia-examples/cnn/` to use cleaner implementation with high-level TensorFlow APIs including Datasets, Layers, and Estimators. Multi-GPU support in these scripts is now provided exclusively using Horovod/MPI.
- Fixed incorrect network definition in `resnet18` and `resnet34` models in `nvidia-examples/cnn/`.
- Updated scripts and README in `nvidia-examples/build_imagenet_data/` to improve usability and ensure that the dataset is correctly downloaded and resized.
- Added support for TensorRT 4 features to TensorFlow-TensorRT integration.
- Includes integration with TensorRT 4.0.1
- Optimized CPU bilinear image resize kernel to improve performance of input pipeline.
- Ubuntu 16.04 with May 2018 updates

Announcements

Starting with the next major version of CUDA release, we will no longer provide updated Python 2 containers and will only update Python 3 containers.

Known Issues

There are no known issues in this release.
Chapter 15.
TENSORFLOW RELEASE 18.05

The NVIDIA container image of TensorFlow, release 18.05, is available.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04

.Container image 18.05-py2 contains Python 2.7; 18.05-py3 contains Python 3.5.

- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.333 (see section 2.3.1)
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.2
- NCCL 2.1.15 (optimized for NVLink™)
- Horovod™ 0.12.1
- OpenMPI™ 3.0.0
- TensorFlow 1.7.0
- MLNX_OFED 3.4
- OpenSeq2Seq v0.2

Driver Requirements

Release 18.05 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.
Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- TensorFlow container image version 18.05 is based on TensorFlow 1.7.0.
- For developers needing more visibility between network layer calls and CUDA kernel calls, we've added support for basic NVTX ranges to the TensorFlow executor. Nsight Systems or the NVIDIA Visual Profiler, with NVTX ranges, are able to display each TensorFlow op demarcated by an NVTX range named by the op. NVTX ranges are enabled by default but can be disabled by setting the environment variable `TF_DISABLE_NVTX_RANGES=1`.
- Optimized input pipeline in `nvcnn.py` and `nvcnn_hvd.py` by casting back to uint8 immediately after image preprocessing.
- Added OpenSeq2Seq v0.2 to the base container.
- Includes integration with TensorRT 3.0.4
- Ubuntu 16.04 with April 2018 updates

Announcements

Starting with the next major version of CUDA release, we will no longer provide Python 2 containers and will only maintain Python 3 containers.

Known Issues

There are no known issues in this release.
The NVIDIA container image of TensorFlow, release 18.04, is available.

**Contents of TensorFlow**

This container image contains the complete source of the version of NVIDIA TensorFlow in `/opt/tensorflow`. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- **Ubuntu 16.04**
- **NVIDIA CUDA 9.0.176** (see Errata section and 2.1) including **CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.333** (see section 2.3.1)
- **NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.1**
- **NCCL 2.1.15** (optimized for **NVLink™**)
- **Horovod™ 0.11.3**
- **OpenMPI™ 3.0.0**
- **TensorBoard 0.4.0-rc1**
- **MLNX_OFED 3.4**

**Driver Requirements**

Release 18.04 is based on CUDA 9, which requires **NVIDIA Driver release 384.xx**.
Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- TensorFlow container image version 18.04 is based on TensorFlow 1.7.0.
- Added the Mellanox user-space InfiniBand driver to the container.
- Latest version of MLNX_OFED 3.4
- Added support for TensorRT integration in TensorFlow. For functionality details, see TensorRT Integration Speeds Up TensorFlow Inference and the example in the `nvidia-examples/tftrt` directory.
- Improved `nvidia_examples/nvcnn.py` and `nvcnn_hvd.py` to ensure ResNet-50 model converges correctly out of the box. See Changelog at the top of `nvidia_examples/nvcnn.py` for more details.
- Enabled Tensor Op math for cuDNN-based RNNs in FP16 precision. This is enabled by default, but can be disabled by setting the environment variable `TF_DISABLE_CUDNN_RNN_TENSOR_OP_MATH=1`.
- Includes integration with TensorRT 3.0.4
- Latest version of NCCL 2.1.15
- Ubuntu 16.04 with March 2018 updates

Announcements

Starting with the next major version of CUDA release, we will no longer provide Python 2 containers and will only maintain Python 3 containers.

Known Issues

There is a degraded performance for graph construction time of grouped convolutions. For more information, see Support for depthwise convolution by groups.
Chapter 17.
TENSORFLOW RELEASE 18.03

The NVIDIA container image of TensorFlow, release 18.03, is available.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04

  Container image 18.03–py2 contains Python 2.7; 18.03–py3 contains Python 3.5.

- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.333 (see section 2.3.1)
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.1
- NCCL 2.1.2 (optimized for NVLink™)
- Horovod™ 0.11.3
- OpenMPI™ 3.0.0
- TensorFlow 0.4.0-rc1

Driver Requirements

Release 18.03 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.
TensorFlow container image version 18.03 is based on TensorFlow 1.4.0.

- Latest updates to OpenSeq2Seq module
- Latest version of cuBLAS 9.0.333
- Latest version of cuDNN 7.1.1
- Latest version of OpenMPI 3.0.0
- Latest version of Horovod 0.11.3
- Latest version of TensorBoard 0.4.0-rc1
- Ubuntu 16.04 with February 2018 updates

**Announcements**

Starting with the next major version of CUDA release, we will no longer provide Python 2 containers and will only maintain Python 3 containers.

**Known Issues**

There are no known issues in this release.
Chapter 18.
TENSORFLOW RELEASE 18.02

The NVIDIA container image of TensorFlow, release 18.02, is available.

TensorFlow container image version 18.02 is based on TensorFlow 1.4.0.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04

Container image 18.02–py2 contains Python 2.7; 18.02–py3 contains Python 3.5.

- NVIDIA CUDA 9.0.176 including:
  - CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.282 Patch 2 which is installed by default
  - cuBLAS 9.0.234 Patch 1 as a debian file. Installing Patch 1 by issuing the `dpkg -i /opt/cuda-cublas-9-0_9.0.234-1_amd64.deb` command is the workaround for the known issue described below.
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.5
- NVIDIA® Collective Communications Library™ (NCCL) 2.1.2 (optimized for NVLink™)
- Horovod™ 0.11.2
Driver Requirements

Release 18.02 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- Updated OpenSeq2Seq example to include latest bug fixes
- Latest version of cuBLAS
- Ubuntu 16.04 with January 2018 updates

Known Issues

- cuBLAS 9.0.282 regresses RNN seq2seq FP16 performance for a small subset of input sizes. This issue should be fixed in the next update. As a workaround, install cuBLAS 9.0.234 Patch 1 by issuing the `dpkg -i /opt/cuda-cublas-9-0_9.0.234-1_amd64.deb` command.
- The broadcast and reduce (but not all_reduce) functions in the `tf.contrib.nccl` module cause an error when executed as part of a graph. This issue should be fixed in the next update. The multi-GPU training example script `nvidia-examples/cnn/nvcnn.py` includes a workaround for the `nccl.broadcast` function so that the script still runs correctly. The Horovod example script `nvidia-examples/cnn/nvcnn_hvd.py` is not affected by this issue.
- Some Python 3 codes may encounter errors when handling text strings containing non-Latin characters. This can be fixed by setting an environment variable with the following command:

  ```bash
  $ export LC_ALL=C.UTF-8
  ```

  This issue should be fixed in the next update.
Chapter 19.
TENSORFLOW RELEASE 18.01

The NVIDIA container image of TensorFlow, release 18.01, is available.

TensorFlow container image version 18.01 is based on TensorFlow 1.4.0.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed as a system Python module.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 9.0.176 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.282
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.5
- NVIDIA® Collective Communications Library™ (NCCL) 2.1.2 (optimized for NVLink™)
- Horovod™ 0.11.2

Driver Requirements

Release 18.01 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.
- Addition of Python 3 package
- **Horovod** is now pre-installed in the container
- Updated OpenSeq2Seq example to include latest bug fixes
- Latest version of cuBLAS
- Latest version of cuDNN
- Latest version of NCCL
- Ubuntu 16.04 with December 2017 updates

**Known Issues**

cuBLAS 9.0.282 regresses RNN seq2seq FP16 performance for a small subset of input sizes. As a workaround, revert back to the 11.12 container.
The NVIDIA container image of TensorFlow, release 17.12, is available. TensorFlow container image version 17.12 is based on TensorFlow 1.4.0.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed into the /usr/local/[bin,lib] directories in the container image.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 9.0.176 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.234
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.5
- NVIDIA® Collective Communications Library™ (NCCL) 2.1.2 (optimized for NVLink™)

Driver Requirements

Release 17.12 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- Latest version of CUDA
- Latest version of cuDNN
- Latest version of NCCL
- Ubuntu 16.04 with November 2017 updates

**Known Issues**

A corner case of float16 reductions is known to give the wrong result of Maxwell and earlier architectures. This will be fixed in a future release.
Chapter 21.
TENSORFLOW RELEASE 17.11

The NVIDIA container image of TensorFlow, release 17.11, is available. TensorFlow container image version 17.11 is based on TensorFlow 1.3.0.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in `/opt/tensorflow`. It is pre-built and installed into the `/usr/local/[bin,lib]` directories in the container image.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 9.0.176 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.234
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.4
- NVIDIA® Collective Communications Library™ (NCCL) 2.1.2 (optimized for NVLink™)

Driver Requirements

Release 17.11 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- Added support for float16 data type and Tensor Core math in batched matrix multiply operations.
• Added example script `nvidia-examples/cnn/nvcnn_hvd.py`, which demonstrates use of the Horovod library for multi-node training.
• Added `Dockerfile.horovod` demonstrating how to build a Docker container with the Horovod library and MPI support.
• Added OpenSeq2Seq example demonstrating sequence-to-sequence model training in `nvidia-examples/OpenSeq2Seq`.
• Latest version of CUDA
• Latest version of cuDNN
• Latest version of NCCL
• Ubuntu 16.04 with October 2017 updates

**Known Issues**

There are no known issues in this release.
Chapter 22. 
TENSORFLOW RELEASE 17.10

The NVIDIA container image of TensorFlow, release 17.10, is available.

TensorFlow container image version 17.10 is based on TensorFlow 1.3.0.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed into the /usr/local/[bin,lib] directories in the container image.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA® 9.0
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.3
- NVIDIA® Collective Communications Library™ (NCCL) 2.0.5 (optimized for NVLink™)

Driver Requirements

Release 17.10 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- Added PNG image support to nvcnn.py.
- Fixed issue with batchnorm op that broke backwards compatibility in the previous release.
Renamed the `TF_ENABLE_TENSOR_OP_MATH (default=1)` environment variable to `TF_DISABLE_TENSOR_OP_MATH (default=0)`.

- Upgraded Bazel to version 0.5.4.
- Worked around hash mismatches in third-party source downloads.
- Enabled compilation flags `-march=sandybridge -mtune=broadwell`.
- Updated Eigen to the top of the tree and removed custom patches.
- Latest version of CUDA
- Latest version of cuDNN
- Latest version of NCCL
- Ubuntu 16.04 with September 2017 updates

**Known Issues**

There are no known issues in this release.
Chapter 23. TENSORFLOW RELEASE 17.09

The NVIDIA container image of TensorFlow, release 17.09, is available. TensorFlow container image version 17.09 is based on TensorFlow 1.3.0.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed into the /usr/local/[bin,lib] directories in the container image.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- **Ubuntu 16.04**
- **NVIDIA CUDA® 9.0**
- **NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.2**
- **NVIDIA® Collective Communications Library™ (NCCL) 2.0.5 (optimized for NVLink™)**

Driver Requirements

Release 17.09 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- Tensor Core operation support in TensorFlow is enabled by default on Volta for FP16 convolutions and matrix multiplies, which should give a speedup for FP16 models.
- Added experimental support for:
FP16 training in `nvidia-examples/cnn/nvcnn.py`
FP16 input/output in the fused batch normalization operation 
`tfe.nn.fused_batch_norm`
Tensor Core operation in FP16 convolutions and matrix multiplications
  - Added the `TF_ENABLE_TENSOR_OP_MATH` parameter which enables and disables Tensor Core operation (defaults to enabled).
Tensor Core operation in FP32 matrix multiplications
  - Added the `TF_ENABLE_TENSOR_OP_MATH_FP32` parameter which enables and disables Tensor Core operation for float32 matrix multiplications (defaults to disabled because it reduces precision).
Increased the `TF_AUTOTUNE_THRESHOLD` parameter which improves auto-tune stability.
Increased the `CUDA_DEVICE_MAX_CONNECTIONS` parameter which solves performance issues related to streams on Tesla K80 GPUs.
Enhancements to `nvidia-examples/cnn/nvcnn.py`
  - Fixed a bug where the final layer was wrong when running in evaluation mode.
  - Changed `is_training` to a constant instead of a placeholder for better performance and reduced memory use.
  - Merged gradients for all layers into a single NCCL call for better performance.
  - Disabled use of XLA by default for better performance.
  - Disabled `zero_debias_moving_mean` in batch normalization operation.
Latest version of CUDA
Latest version of cuDNN
Latest version of NCCL
Ubuntu 16.04 with August 2017 updates

**Known Issues**

There are no known issues in this release.
Chapter 24.
TENSORFLOW RELEASE 17.07

The NVIDIA container image of TensorFlow, release 17.07, is available. TensorFlow container image version 17.07 is based on TensorFlow 1.2.1.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed into the /usr/local/[bin,lib] directories in the container image.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA® 8.0.61.2 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) Patch 2
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 6.0.21
- NVIDIA® Collective Communications Library™ (NCCL) 2.0.3 (optimized for NVLink™)

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- Small bug-fixes in evaluation mode of nvidia-examples/cnn/nvcnn.py
- Ubuntu 16.04 with June 2017 updates

Known Issues

There are no known issues in this release.
The NVIDIA container image of TensorFlow, release 17.06, is available.

TensorFlow container image version 17.06 is based on TensorFlow 1.1.0.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed into the /usr/local/[bin,lib] directories in the container image.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA® 8.0.61
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 6.0.21
- NVIDIA® Collective Communications Library™ (NCCL) 1.6.1 (optimized for NVLink™)

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- Ubuntu 16.04 with May 2017 updates

Known Issues

The inception_v4 model, with a batch size of 64 per GPU, and with large input images or resolution (for example, 480 pixels on the shortest side), are seen to run out of memory. To work around this in TensorFlow 17.06, reduce the resolution or reduce the batch size to allow the model to fit.
Chapter 26. 
TENSORFLOW RELEASE 17.05

The NVIDIA container image of TensorFlow, release 17.05, is available. TensorFlow container image version 17.05 is based on TensorFlow 1.0.1.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed into the /usr/local/[bin,lib] directories in the container image.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA® 8.0.61
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 6.0.21
- NVIDIA® Collective Communications Library™ (NCCL) 1.6.1 (optimized for NVLink™)

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- Latest cuDNN release
- Ubuntu 16.04 with April 2017 updates

Known Issues

The inception_v4 model, with a batch size of 64 per GPU, and with large input images or resolution (for example, 480 pixels on the shortest side), are seen to run out of
memory. To work around this in TensorFlow 17.05, reduce the resolution or reduce the batch size to allow the model to fit.
Chapter 27.
TENSORFLOW RELEASE 17.04

The NVIDIA container image of TensorFlow, release 17.04, is available. TensorFlow container image version 17.04 is based on TensorFlow 1.0.1.

Contents of TensorFlow
This container image contains the complete source of the version of NVIDIA TensorFlow in `/opt/tensorflow`. It is pre-built and installed into the `/usr/local/[bin,lib]` directories in the container image.

To achieve optimum TensorFlow performance, for image based training, the container includes a sample script that demonstrates efficient training of convolutional neural networks (CNNs). The sample script may need to be modified to fit your application.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA® 8.0.61
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 6.0.20
- NVIDIA® Collective Communications Library™ (NCCL) 1.6.1 (optimized for NVLink™)

Key Features and Enhancements
This TensorFlow release includes the following key features and enhancements.

- 2x improvement on 8GPUs; 1.5X on 4 GPUs
- Updated NCCL integration with support for NVLink
- Multi-GPU CNN examples that demonstrates efficient training of CNNs using NCCL
- XLA (Accelerated Linear Algebra) support enabled, allowing users to offload operations to TensorFlow experimental XLA back-end
- Ubuntu 16.04 with March 2017 updates
Known Issues

There are no known issues in this release.
Chapter 28.
TENSORFLOW RELEASE 17.03

The NVIDIA container image of TensorFlow, release 17.03, is available. TensorFlow container image version 17.03 is based on TensorFlow 1.0.0.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in `/opt/tensorflow`. It is pre-built and installed into the `/usr/local/[bin,lib]` directories in the container image.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA® 8.0.61
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 6.0.20
- NVIDIA® Collective Communications Library™ (NCCL) 1.6.1

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- Multi-GPU BigLSTM example that trains a recurrent neural network (RNN) to learn a language model
- Ubuntu 16.04 with February 2017 updates

Known Issues

There are no known issues in this release.
The NVIDIA container image of TensorFlow, release 17.02, is available. TensorFlow container image version 17.02 is based on TensorFlow 0.12.0.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed into the /usr/local/[bin,lib] directories in the container image.

The container also includes the following:

- Ubuntu 14.04
- NVIDIA CUDA® 8.0.61
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 6.0.13
- NVIDIA® Collective Communications Library™ (NCCL) 1.6.1 (optimized for NVLink™)

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- Fused image color adjustment kernels for improved preprocessing performance
- Ubuntu 14.04 with January 2017 updates

Known Issues

There are no known issues in this release.
The NVIDIA container image of TensorFlow, release 17.01, is available. TensorFlow container image version 17.01 is based on TensorFlow 0.12.0.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in /opt/tensorflow. It is pre-built and installed into the /usr/local/[bin,lib] directories in the container image.

The container also includes the following:

- Ubuntu 14.04
- NVIDIA CUDA® 8.0.54
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 6.0.10
- NVIDIA® Collective Communications Library™ (NCCL) 1.6.1 (optimized for NVLink™)

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- Ubuntu 14.04 with December 2016 updates

Known Issues

There are no known issues in this release.
Chapter 31. 
TENSORFLOW RELEASE 16.12

The NVIDIA container image of TensorFlow, release 16.12, is available. 
TensorFlow container image version 16.12 is based on TensorFlow 0.12.0.

Contents of TensorFlow

This container image contains the complete source of the version of NVIDIA TensorFlow in 
/_opt/tensorflow_. It is pre-built and installed into the 
/_usr/local/[bin,lib]_ directories in the container image.

The container also includes the following:

- Ubuntu 14.04
- NVIDIA CUDA® 8.0.54
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 6.0.5
- NVIDIA® Collective Communications Library™ (NCCL) 1.6.1 (optimized for NVLink™)

Key Features and Enhancements

This TensorFlow release includes the following key features and enhancements.

- Supports multi-GPU training
  - [BETA] NCCL integration for improved multi-GPU scaling

  Requires explicit use by the model script.

- Supports recurrent neural networks
  - Support for cuDNN recurrent neural networks (RNN) layers

  Requires explicit use by the model script.
- Better I/O throughput via `libjpeg-turbo`, fast iDCT decoding
- Support for the non-fused Winograd algorithm for improved convolution performance.
- TensorBoard; a data visualization toolkit
- Several built-in TensorFlow examples
- Ubuntu 14.04 with November 2016 updates

**Known Issues**

There are no known issues in this release.
Notice

THE INFORMATION IN THIS GUIDE AND ALL OTHER INFORMATION CONTAINED IN NVIDIA DOCUMENTATION REFERENCED IN THIS GUIDE IS PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE INFORMATION FOR THE PRODUCT, 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 product described in this guide shall be limited in accordance with the NVIDIA terms and conditions of sale for the product.

THE NVIDIA PRODUCT DESCRIBED IN THIS GUIDE IS NOT FAULT TOLERANT AND IS NOT DESIGNED, MANUFACTURED OR INTENDED FOR USE IN CONNECTION WITH THE DESIGN, CONSTRUCTION, MAINTENANCE, AND/OR OPERATION OF ANY SYSTEM WHERE THE USE OR A FAILURE OF SUCH SYSTEM COULD RESULT IN A SITUATION THAT THREATENS THE SAFETY OF HUMAN LIFE OR SEVERE PHYSICAL HARM OR PROPERTY DAMAGE (INCLUDING, FOR EXAMPLE, USE IN CONNECTION WITH ANY NUCLEAR, AVIONICS, LIFE SUPPORT OR OTHER LIFE CRITICAL APPLICATION). NVIDIA EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR SUCH HIGH RISK USES. NVIDIA SHALL NOT BE LIABLE TO CUSTOMER OR ANY THIRD PARTY, IN WHOLE OR IN PART, FOR ANY CLAIMS OR DAMAGES ARISING FROM SUCH HIGH RISK USES.

NVIDIA makes no representation or warranty that the product described in this guide 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 guide. 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 guide, or (ii) customer product designs.

Other than the right for customer to use the information in this guide with the product, no other license, either expressed or implied, is hereby granted by NVIDIA under this guide. Reproduction of information in this guide 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.

Trademarks

NVIDIA, the NVIDIA logo, DGX, DGX-1, DGX-2, and DGX Station are trademarks and/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

© 2019 NVIDIA Corporation. All rights reserved.

www.nvidia.com