PaddlePaddle

Release Notes

RN 10952-001_v23.07 | July 2023

Table of Contents

Chapter 1. PaddlePaddle Overview

The NVIDIA® Deep Learning SDK accelerates widely-used deep learning frameworks such as PaddlePaddle.

PaddlePaddle is the first independent R&D deep learning platform in China and has been officially open-sourced to professional communities since 2016. It is an industrial platform with advanced technologies and rich features that cover core deep learning frameworks, basic model libraries, end-to-end development kits, tools and components and service platforms. PaddlePaddle originated from industrial practices with a dedication and commitments to industrialization.

It has been widely adopted by a wide range of sectors including manufacturing, agriculture, enterprise service, and so on while serving more than 4 million developers, 157,000 companies and generating 476,000 models. With such advantages, PaddlePaddle has helped an increasing number of partners commercialize AI.

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

- PaddlePaddle tutorials
- PaddlePaddle API
- PaddlePaddle User Guide

Chapter 2. Pulling a Container

About this task

Before you can pull a container from the NGC container registry:

- Install Docker.
  - For NVIDIA DGX™ users, see Preparing to use NVIDIA Containers Getting Started Guide.
  - For non-DGX users, see NVIDIA® GPU Cloud™ (NGC) container registry installation documentation based on your platform.
- Ensure that you have access and can log in to the NGC container registry.

  Refer to NGC Getting Started Guide for more information.

The deep learning frameworks, the NGC Docker containers, and the deep learning framework containers are stored in the `nvcr.io/nvidia` repository.

Chapter 3. Running PaddlePaddle

Before you begin

Before you can run an NGC deep learning framework container, your Docker environment must support NVIDIA GPUs. To run a container, issue the appropriate command as explained in Running A Container and specify the registry, repository, and tags.

About this task

On a system with GPU support for NGC containers, when you run a container, the following occurs :

- The Docker engine loads the image into a container that runs the software.
- You define the container's runtime resources by including the additional flags and settings that are used with the command.

  These flags and settings are described in Running A Container.
- The GPUs are explicitly defined for the Docker® container, which defaults to all GPUs, but can be specified by using the `NVIDIA_VISIBLE_DEVICES` environment variable.

  For more information, refer to the nvidia-docker documentation.

Note: Starting in Docker 19.03, complete the steps below.

The method implemented in your system depends on the DGX OS version that you installed (for DGX systems), the NGC Cloud Image that was provided by a Cloud Service Provider, or the software that you installed to prepare to run NGC containers on TITAN PCs, Quadro PCs, or NVIDIA Virtual GPUs (vGPUs).

Procedure

1. Issue the command for the applicable release of the container that you want.

   The following command assumes that you want to pull the latest container.

   ```
   docker pull nvcr.io/nvidia/paddlepaddle:23.07-py3
   ```
2. Open a command prompt and paste the `pull` command.

Ensure that the pull process successfully completes before you proceed to step 3.

3. Run the container image.

   - If you have **Docker 19.03 or later**, a typical command to launch the container is:

     ```
     docker run --gpus all -it --rm nvcr.io/nvidia/paddlepaddle:xx.xx-py3
     ```

   - If you have **Docker 19.02 or earlier**, a typical command to launch the container is:

     ```
     nvidia-docker run -it --rm nvcr.io/nvidia/paddlepaddle:xx.xx-py3
     ```

   To run PaddlePaddle, import it as a Python module:

   ```
   $ python -c 'import paddle; paddle.utils.run_check()'
   Running verify PaddlePaddle program …
   W0516 06:36:54.208734   442 device_context.cc:451] Please NOTE: device: 0, GPU
    Compute Capability: 8.0, Driver API Version: 11.7, Runtime API Version: 11.7
   W0516 06:36:54.212574   442 device_context.cc:469] device: 0, cuDNN Version: 8.4.
   PaddlePaddle works well on 1 GPU.
   W0516 06:37:12.706600   442 fuse_all_reduce_op_pass.cc:76] Find all_reduce
    operators: 2. To make the speed faster, some all_reduce ops are fused during
    training, after fusion, the number of all_reduce ops is 2.
   PaddlePaddle works well on 8 GPUs.
   PaddlePaddle is installed successfully! Let's start deep learning with
    PaddlePaddle now.
   ```

   To pull data and model descriptions from locations outside the container for use by PaddlePaddle or save results to locations outside the container, mount one or more host directories as Docker data volumes.

   To share data between GPUs, NVIDIA Collective Communications Library (NCCL) might require shared system memory for IPC and pinned (page-locked) system memory resources, so the operating system's limits on these resources might need to be increased. Refer to your system's documentation for more information.

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

   ```
   --shm-size=1g --ulimit memlock=-1
   ```

   in the command line to:

   ```
   docker run --gpus all
   ```

   Similarly, on some Redhat Enterprise Linux (RHEL) systems, Docker limits the number of simultaneous PIDs in the container to 4096, which might be too small, particularly for multi-GPU training tasks.

   To increase this limit, pass the following option to docker run.

   ```
    --pids-limit=819
   ```

Chapter 4. PaddlePaddle Release 23.07

The NVIDIA container image for PaddlePaddle, release 23.07, is available on NGC.

Contents of the PaddlePaddle container

This container image includes the complete source of the NVIDIA version of PaddlePaddle in `/opt/paddlepaddle`. It is prebuilt and installed as a system Python module.

The container includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 12.1.1
- NVIDIA cuBLAS 12.1.3.1
- cuTENSOR 1.7.0.1
- NVIDIA cuDNN 8.9.3
- NVIDIA NCCL 2.18.3
- rdma-core 39.0
- OpenMPI 4.1.4+
- OpenUCX 1.15.0
- GDRCopy 2.3
- Nsight Systems 2023.2.3.1001
- Nsight Compute 2023.1.1.4
- NVIDIA HPC-X 2.15
- TensorRT 8.6.1.6 for x64 Linux
- Paddle-TRT 2.4
- SHARP 3.0.2
- DALI 1.27.0

Driver Requirements

Release 23.07 is based on CUDA 12.1.1, which requires NVIDIA Driver release 530 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), 515.65 (or later R515), 525.85 (or later R525), or 530.30 (or later R530).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.1. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This PaddlePaddle release includes the following key features and enhancements.

- The PaddlePaddle container image version 23.07 is based on v2.4.1.

Announcements

- The cuDNN frontend has been integrated into PaddlePaddle. It can be activated by turning on “fuse_resunit” and “fuse_dot_product_attention” flags in the “build_strategy”. The cuDNN frontend provides advanced fusion kernels which accelerates training speed.
- The NVIDIA/LDDL has been integrated in PaddlePaddle. The Language Datasets and Data Loaders (LDDL) is a utility library that minimizes the friction during dataset retrieval, preprocessing and loading for the language models. It successfully accelerates BERT pre-training phase 2 to 2X. See the BERT example for details. The training results with multinode are added in the BERT example, including 1 node to 32 nodes.

NVIDIA PaddlePaddle Container Versions

The following table shows what versions of Ubuntu, CUDA, PaddlePaddle, and TensorRT are supported in each of the NVIDIA containers for PaddlePaddle. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
23.07	22.04	NVIDIA CUDA 12.1.1	2.4.1	TensorRT 8.6.1.6
23.06				
23.04	20.04	NVIDIA CUDA 12.1.0		TensorRT 8.6.1
23.03				TensorRT 8.5.3

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
23.02		NVIDIA CUDA 12.0.1	2.4	
23.01			2.3.2	TensorRT 8.5.2.2
22.12		NVIDIA CUDA 11.8.0		TensorRT 8.5.1
22.11				TensorRT 8.5.1
22.10				TensorRT 8.5.0.12
22.09				
22.08		NVIDIA CUDA 11.7.1	2.3.1	TensorRT 8.4.2.4
22.07		NVIDIA CUDA 11.7 Update 1 Preview	2.3.0	TensorRT 8.4.1
22.06			2.2.2	TensorRT 8.2.5
22.05		NVIDIA CUDA 11.7		

Automatic Mixed Precision (AMP)

Automatic Mixed Precision (AMP) for PaddlePaddle is available in this container through the native implementation. AMP enables users to try mixed precision training by adding only 3 lines of Python to an existing FP32 (default) script. AMP will select an optimal set of operations to cast to FP16. FP16 operations require 2X reduced memory bandwidth (resulting in a 2X speedup for bandwidth-bound operations like most pointwise ops) and 2X reduced memory storage for intermediates (reducing the overall memory consumption of your model). Additionally, GEMMs and convolutions with FP16 inputs can run on Tensor Cores, which provide an 8X increase in computational throughput over FP32 arithmetic.

For more information about AMP, see the Training With Mixed Precision Guide.

Tensor Core Examples

The tensor core examples provided in GitHub and NGC focus on achieving the best performance and convergence from NVIDIA Volta™ tensor cores by using the latest deep learning example networks and model scripts for training.

Each example model trains with mixed precision Tensor Cores on Volta and NVIDIA Turing™, so 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.

- ResNet50 v1.5 model: This model is a modified version of the regular ResNet model that was introduced in the *Deep Residual Learning for Image Recognition* paper.

The v1.5 has `stride = 2` in the 3x3 convolution instead of 1x1 convolution. This model script is available on GitHub.

- BERT model:This model is a new method of pre-training language representations that obtains state-of-the-art results on a wide array of Natural Language Processing (NLP) tasks that was introduced in the *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* paper.

  This model script is available on GitHub.

Known Issues

None.

Chapter 5. PaddlePaddle Release 23.06

The NVIDIA container image for PaddlePaddle, release 23.06, is available on NGC.

Contents of the PaddlePaddle container

This container image includes the complete source of the NVIDIA version of PaddlePaddle in `/opt/paddlepaddle`. It is prebuilt and installed as a system Python module.

The container includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 12.1.1
- NVIDIA cuBLAS 12.1.3.1
- cuTENSOR 1.7
- NVIDIA cuDNN 8.9.2
- NVIDIA NCCL 2.18.1
- rdma-core 39.0
- OpenMPI 4.1.4+
- OpenUCX 1.15.0
- GDRCopy 2.3
- Nsight Systems 2023.2.3.1001
- Nsight Compute 2023.1.1.4
- NVIDIA HPC-X 2.15
- TensorRT 8.6.1.6 for x64 Linux
- Paddle-TRT 2.4
- SHARP 3.0.2
- DALI 1.26.0

Driver Requirements

Release 23.06 is based on CUDA 12.1.1, which requires NVIDIA Driver release 530 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), 515.65 (or later R515), 525.85 (or later R525), or 530.30 (or later R530).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.1. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This PaddlePaddle release includes the following key features and enhancements.

- The PaddlePaddle container image version 23.06 is based on v2.4.1.

Announcements

- The cuDNN frontend has been integrated into PaddlePaddle. It can be activated by turning on “fuse_resunit” and “fuse_dot_product_attention” flags in the “build_strategy”. The cuDNN frontend provides advanced fusion kernels which accelerates training speed.
- The NVIDIA/LDDL has been integrated in PaddlePaddle. The Language Datasets and Data Loaders (LDDL) is a utility library that minimizes the friction during dataset retrieval, preprocessing and loading for the language models. It successfully accelerates BERT pre-training phase 2 to 2X. See the BERT example for details. The training results with multinode are added in the BERT example, including 1 node to 32 nodes.

NVIDIA PaddlePaddle Container Versions

The following table shows what versions of Ubuntu, CUDA, PaddlePaddle, and TensorRT are supported in each of the NVIDIA containers for PaddlePaddle. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
23.06	22.04	NVIDIA CUDA 12.1.1	2.4.1	TensorRT 8.6.1.6
23.04	20.04	NVIDIA CUDA 12.1.0		TensorRT 8.6.1
23.03				TensorRT 8.5.3

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
23.02		NVIDIA CUDA 12.0.1	2.4	
23.01			2.3.2	TensorRT 8.5.2.2
22.12		NVIDIA CUDA 11.8.0		TensorRT 8.5.1
22.11				TensorRT 8.5.1
22.10				TensorRT 8.5.0.12
22.09				
22.08		NVIDIA CUDA 11.7.1	2.3.1	TensorRT 8.4.2.4
22.07		NVIDIA CUDA 11.7 Update 1 Preview	2.3.0	TensorRT 8.4.1
22.06			2.2.2	TensorRT 8.2.5
22.05		NVIDIA CUDA 11.7		

Automatic Mixed Precision (AMP)

Automatic Mixed Precision (AMP) for PaddlePaddle is available in this container through the native implementation. AMP enables users to try mixed precision training by adding only 3 lines of Python to an existing FP32 (default) script. AMP will select an optimal set of operations to cast to FP16. FP16 operations require 2X reduced memory bandwidth (resulting in a 2X speedup for bandwidth-bound operations like most pointwise ops) and 2X reduced memory storage for intermediates (reducing the overall memory consumption of your model). Additionally, GEMMs and convolutions with FP16 inputs can run on Tensor Cores, which provide an 8X increase in computational throughput over FP32 arithmetic.

For more information about AMP, see the Training With Mixed Precision Guide.

Tensor Core Examples

The tensor core examples provided in GitHub and NGC focus on achieving the best performance and convergence from NVIDIA Volta™ tensor cores by using the latest deep learning example networks and model scripts for training.

Each example model trains with mixed precision Tensor Cores on Volta and NVIDIA Turing™, so 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.

- ResNet50 v1.5 model: This model is a modified version of the regular ResNet model that was introduced in the *Deep Residual Learning for Image Recognition* paper.

The v1.5 has `stride = 2` in the 3x3 convolution instead of 1x1 convolution. This model script is available on GitHub.

- BERT model:This model is a new method of pre-training language representations that obtains state-of-the-art results on a wide array of Natural Language Processing (NLP) tasks that was introduced in the *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* paper.

  This model script is available on GitHub.

Known Issues

- Grouped Conv2D lacks FP16/BF16 precision due to TensorRT on H100.
- Tensor parallelism and pipeline parallelism training might hang on multiple H100.

Chapter 6. PaddlePaddle Release 23.04

The NVIDIA container image for PaddlePaddle, release 23.04, is available on NGC.

Contents of the PaddlePaddle container

This container image includes the complete source of the NVIDIA version of PaddlePaddle in `/opt/paddlepaddle`. It is prebuilt and installed as a system Python module.

The container includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 12.1.0
- NVIDIA cuBLAS 12.1.3
- cuTENSOR 1.7
- NVIDIA cuDNN 8.9.0
- NVIDIA NCCL 2.17.1
- rdma-core 36.0
- OpenMPI 4.1.4+
- GDRCopy 2.3
- Nsight Systems 2023.1.1.127
- Nsight Compute 2023.1.0.15
- NVIDIA HPC-X 2.13
- TensorRT 8.5.3 for x64 Linux
- Paddle-TRT 2.4
- SHARP 3.0.2
- DALI 1.24.0

Driver Requirements

Release 23.04 is based on CUDA 12.1.0, which requires NVIDIA Driver release 530 or later. However, if you are running on a data center GPU (for example, T4 or any other data

center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), 515.65 (or later R515), 525.85 (or later R525), or 530.30 (or later R530).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.1. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This PaddlePaddle release includes the following key features and enhancements.

- The PaddlePaddle container image version 23.04 is based on v2.4.1.

Announcements

- The NVIDIA/LDDL has been integrated in PaddlePaddle. The Language Datasets and Data Loaders (LDDL) is a utility library that minimizes the friction during dataset retrieval, preprocessing and loading for the language models. It successfully accelerates BERT pre-training phase 2 to 2X. See the BERT example for details. The training results with multinode are added in the BERT example, including 1 node to 32 nodes.

NVIDIA PaddlePaddle Container Versions

The following table shows what versions of Ubuntu, CUDA, PaddlePaddle, and TensorRT are supported in each of the NVIDIA containers for PaddlePaddle. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
23.04	22.04	NVIDIA CUDA 12.1.0	2.4.1	TensorRT 8.6.1
23.03				TensorRT 8.5.3
23.02		NVIDIA CUDA 12.0.1	2.4	
23.01			2.3.2	TensorRT 8.5.2.2
22.12		NVIDIA CUDA 11.8.0		TensorRT 8.5.1
22.11				TensorRT 8.5.1
22.10				TensorRT 8.5.0.12
22.09				
22.08		NVIDIA CUDA 11.7.1	2.3.1	TensorRT 8.4.2.4

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
22.07		NVIDIA CUDA 11.7 Update 1 Preview	2.3.0	TensorRT 8.4.1
22.06			2.2.2	TensorRT 8.2.5
22.05		NVIDIA CUDA 11.7		

Automatic Mixed Precision (AMP)

Automatic Mixed Precision (AMP) for PaddlePaddle is available in this container through the native implementation. AMP enables users to try mixed precision training by adding only 3 lines of Python to an existing FP32 (default) script. AMP will select an optimal set of operations to cast to FP16. FP16 operations require 2X reduced memory bandwidth (resulting in a 2X speedup for bandwidth-bound operations like most pointwise ops) and 2X reduced memory storage for intermediates (reducing the overall memory consumption of your model). Additionally, GEMMs and convolutions with FP16 inputs can run on Tensor Cores, which provide an 8X increase in computational throughput over FP32 arithmetic.

For more information about AMP, see the Training With Mixed Precision Guide.

Tensor Core Examples

The tensor core examples provided in GitHub and NGC focus on achieving the best performance and convergence from NVIDIA Volta™ tensor cores by using the latest deep learning example networks and model scripts for training.

Each example model trains with mixed precision Tensor Cores on Volta and NVIDIA Turing™, so 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.

- ResNet50 v1.5 model: This model is a modified version of the regular ResNet model that was introduced in the *Deep Residual Learning for Image Recognition* paper.

  The v1.5 has `stride = 2` in the 3x3 convolution instead of 1x1 convolution. This model script is available on GitHub.
- BERT model:This model is a new method of pre-training language representations that obtains state-of-the-art results on a wide array of Natural Language Processing (NLP) tasks that was introduced in the *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* paper.

  This model script is available on GitHub.

Known Issues

- Grouped Conv2D lacks FP16/BF16 precision due to TensorRT.
- Sparse kernel, CSR to COO format conversion has functional regression on P100.
- The Conv2D fusion kernel, `convBiasAct`, has functional regression due to cuDNN on P100.
- RNN of Paddle-TRT has functional regression due to TensorRT.
- Asynchronous tensor copy might malfunction in inference on P100.
- The communication kernels (i.e., NCCL) might be unstable on H100. It might take a longer time than the previous version, or hang.

Chapter 7. PaddlePaddle Release 23.03

The NVIDIA container image for PaddlePaddle, release 23.03, is available on NGC.

Contents of the PaddlePaddle container

This container image includes the complete source of the NVIDIA version of PaddlePaddle in `/opt/paddlepaddle`. It is prebuilt and installed as a system Python module.

The container includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 12.1.0
- NVIDIA cuBLAS from CUDA 12.1.0
- cuTENSOR 1.6.2.3
- NVIDIA cuDNN 8.8.1.3
- NVIDIA NCCL 2.17.1
- rdma-core 36.0
- OpenMPI 4.1.4+
- GDRCopy 2.3
- Nsight Systems 2023.1.1.127
- Nsight Compute 2023.1.0.15
- NVIDIA HPC-X 2.13
- TensorRT 8.5.3 for x64 Linux
- Paddle-TRT 2.4
- SHARP 3.0.2
- DALI 1.23.0

Driver Requirements

Release 23.03 is based on CUDA 12.1.0, which requires NVIDIA Driver release 530 or later. However, if you are running on a data center GPU (for example, T4 or any other data

center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), 515.65 (or later R515), 525.85 (or later R525), or 530.30 (or later R530).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.1. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This PaddlePaddle release includes the following key features and enhancements.

- The PaddlePaddle container image version 23.03 is based on v2.4.1.

Announcements

- The NVIDIA/LDDL has been integrated in PaddlePaddle. The Language Datasets and Data Loaders (LDDL) is a utility library that minimizes the friction during dataset retrieval, preprocessing and loading for the language models. It successfully accelerates BERT pre-training phase 2 to 2X. See the BERT example for details. The training results with multinode are added in the BERT example, including 1 node to 32 nodes.

NVIDIA PaddlePaddle Container Versions

The following table shows what versions of Ubuntu, CUDA, PaddlePaddle, and TensorRT are supported in each of the NVIDIA containers for PaddlePaddle. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
23.03	22.04	NVIDIA CUDA 12.1.0	2.4.1	TensorRT 8.5.3
23.02		NVIDIA CUDA 12.0.1	2.4	
23.01			2.3.2	TensorRT 8.5.2.2
22.12		NVIDIA CUDA 11.8.0		TensorRT 8.5.1
22.11				TensorRT 8.5.1
22.10				TensorRT 8.5.0.12
22.09				
22.08		NVIDIA CUDA 11.7.1	2.3.1	TensorRT 8.4.2.4

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
22.07		NVIDIA CUDA 11.7 Update 1 Preview	2.3.0	TensorRT 8.4.1
22.06			2.2.2	TensorRT 8.2.5
22.05		NVIDIA CUDA 11.7		

Automatic Mixed Precision (AMP)

Automatic Mixed Precision (AMP) for PaddlePaddle is available in this container through the native implementation. AMP enables users to try mixed precision training by adding only 3 lines of Python to an existing FP32 (default) script. AMP will select an optimal set of operations to cast to FP16. FP16 operations require 2X reduced memory bandwidth (resulting in a 2X speedup for bandwidth-bound operations like most pointwise ops) and 2X reduced memory storage for intermediates (reducing the overall memory consumption of your model). Additionally, GEMMs and convolutions with FP16 inputs can run on Tensor Cores, which provide an 8X increase in computational throughput over FP32 arithmetic.

For more information about AMP, see the Training With Mixed Precision Guide.

Tensor Core Examples

The tensor core examples provided in GitHub and NGC focus on achieving the best performance and convergence from NVIDIA Volta™ tensor cores by using the latest deep learning example networks and model scripts for training.

Each example model trains with mixed precision Tensor Cores on Volta and NVIDIA Turing™, so 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.

- ResNet50 v1.5 model: This model is a modified version of the regular ResNet model that was introduced in the *Deep Residual Learning for Image Recognition* paper.

  The v1.5 has `stride = 2` in the 3x3 convolution instead of 1x1 convolution. This model script is available on GitHub.
- BERT model:This model is a new method of pre-training language representations that obtains state-of-the-art results on a wide array of Natural Language Processing (NLP) tasks that was introduced in the *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* paper.

  This model script is available on GitHub.

Known Issues

- None.

Chapter 8. PaddlePaddle Release 23.02

The NVIDIA container image for PaddlePaddle, release 23.02, is available on NGC.

Contents of the PaddlePaddle container

This container image includes the complete source of the NVIDIA version of PaddlePaddle in `/opt/paddlepaddle`. It is prebuilt and installed as a system Python module.

The container includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 12.0.1
- cuTENSOR 1.6.2.3
- NVIDIA cuDNN 8.7.0
- NVIDIA NCCL 2.16.5
- rdma-core 36.0
- OpenMPI 4.1.4+
- GDRCopy 2.3
- Nsight Systems 2022.5.1
- Nsight Compute 2022.4.1.6
- NVIDIA HPC-X 2.13
- TensorRT 8.5.3 for x64 Linux
- Paddle-TRT 2.4
- SHARP 3.0.2
- DALI 1.22.0

Driver Requirements

Release 23.02 is based on CUDA 12.0.1, which requires NVIDIA Driver release 525 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), 515.65 (or later R515), or 525.85 (or later R525).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.0. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This PaddlePaddle release includes the following key features and enhancements.

- The PaddlePaddle container image version 23.02 is based on v2.4.

Announcements

- The NVIDIA/LDDL has been integrated in PaddlePaddle. The Language Datasets and Data Loaders (LDDL) is a utility library that minimizes the friction during dataset retrieval, preprocessing and loading for the language models. It successfully accelerates BERT pre-training phase 2 to 2X. See the BERT example for details. The training results with multinode are added in the BERT example, including 1 node to 32 nodes.

NVIDIA PaddlePaddle Container Versions

The following table shows what versions of Ubuntu, CUDA, PaddlePaddle, and TensorRT are supported in each of the NVIDIA containers for PaddlePaddle. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
23.02	22.04	NVIDIA CUDA 12.0.1	2.4	TensorRT 8.5.3
23.01			2.3.2	TensorRT 8.5.2.2
22.12		NVIDIA CUDA 11.8.0		TensorRT 8.5.1
22.11				TensorRT 8.5.1
22.10				TensorRT 8.5.0.12
22.09				
22.08		NVIDIA CUDA 11.7.1	2.3.1	TensorRT 8.4.2.4
22.07		NVIDIA CUDA 11.7 Update 1 Preview	2.3.0	TensorRT 8.4.1
22.06			2.2.2	TensorRT 8.2.5
22.05		NVIDIA CUDA 11.7		

Automatic Mixed Precision (AMP)

Automatic Mixed Precision (AMP) for PaddlePaddle is available in this container through the native implementation. AMP enables users to try mixed precision training by adding only 3 lines of Python to an existing FP32 (default) script. AMP will select an optimal set of operations to cast to FP16. FP16 operations require 2X reduced memory bandwidth (resulting in a 2X speedup for bandwidth-bound operations like most pointwise ops) and 2X reduced memory storage for intermediates (reducing the overall memory consumption of your model). Additionally, GEMMs and convolutions with FP16 inputs can run on Tensor Cores, which provide an 8X increase in computational throughput over FP32 arithmetic.

For more information about AMP, see the Training With Mixed Precision Guide.

Tensor Core Examples

The tensor core examples provided in GitHub and NGC focus on achieving the best performance and convergence from NVIDIA Volta™ tensor cores by using the latest deep learning example networks and model scripts for training.

Each example model trains with mixed precision Tensor Cores on Volta and NVIDIA Turing™, so 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.

- ResNet50 v1.5 model: This model is a modified version of the regular ResNet model that was introduced in the *Deep Residual Learning for Image Recognition* paper.

  The v1.5 has `stride = 2` in the 3x3 convolution instead of 1x1 convolution. This model script is available on GitHub.
- BERT model:This model is a new method of pre-training language representations that obtains state-of-the-art results on a wide array of Natural Language Processing (NLP) tasks that was introduced in the *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* paper.

  This model script is available on GitHub.

Known Issues

- None.

Chapter 9. PaddlePaddle Release 23.01

The NVIDIA container image for PaddlePaddle, release 23.01, is available on NGC.

Contents of the PaddlePaddle container

This container image includes the complete source of the NVIDIA version of PaddlePaddle in `/opt/paddlepaddle`. It is prebuilt and installed as a system Python module.

The container includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 12.0.1
- cuTENSOR 1.6.2.3
- NVIDIA cuDNN 8.7.0
- NVIDIA NCCL 2.16.5
- rdma-core 36.0
- OpenMPI 4.1.4+
- GDRCopy 2.3
- Nsight Systems 2022.5.1
- Nsight Compute 2022.4.1.6
- NVIDIA HPC-X 2.13
- TensorRT 8.5.2.2 for x64 Linux
- Paddle-TRT 2.3.2
- SHARP 3.0.2
- DALI 1.21.0

Driver Requirements

Release 23.01 is based on CUDA 12.0.1, which requires NVIDIA Driver release 525 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), 515.65 (or later R515), or 525.85 (or later R525).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 12.0. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This PaddlePaddle release includes the following key features and enhancements.

- The PaddlePaddle container image version 23.01 is based on v2.3.2.

Announcements

- Paddle-TRT is now included.

  Paddle-TRT is the TensorRT integration for PaddlePaddle and brings the capabilities of TensorRT to PadddlePaddle in a few lines in the Python and C++ APIs.

NVIDIA PaddlePaddle Container Versions

The following table shows what versions of Ubuntu, CUDA, PaddlePaddle, and TensorRT are supported in each of the NVIDIA containers for PaddlePaddle. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
23.01	22.04	NVIDIA CUDA 12.0.1	2.3.2	TensorRT 8.5.2.2
22.12		NVIDIA CUDA 11.8.0		TensorRT 8.5.1
22.11				TensorRT 8.5.1
22.10				TensorRT 8.5.0.12
22.09				
22.08		NVIDIA CUDA 11.7.1	2.3.1	TensorRT 8.4.2.4
22.07		NVIDIA CUDA 11.7 Update 1 Preview	2.3.0	TensorRT 8.4.1
22.06			2.2.2	TensorRT 8.2.5
22.05		NVIDIA CUDA 11.7		

Automatic Mixed Precision (AMP)

Automatic Mixed Precision (AMP) for PaddlePaddle is available in this container through the native implementation. AMP enables users to try mixed precision training by adding only 3 lines of Python to an existing FP32 (default) script. AMP will select an optimal set

of operations to cast to FP16. FP16 operations require 2X reduced memory bandwidth (resulting in a 2X speedup for bandwidth-bound operations like most pointwise ops) and 2X reduced memory storage for intermediates (reducing the overall memory consumption of your model). Additionally, GEMMs and convolutions with FP16 inputs can run on Tensor Cores, which provide an 8X increase in computational throughput over FP32 arithmetic.

For more information about AMP, see the Training With Mixed Precision Guide.

Tensor Core Examples

The tensor core examples provided in GitHub and NGC focus on achieving the best performance and convergence from NVIDIA Volta™ tensor cores by using the latest deep learning example networks and model scripts for training.

Each example model trains with mixed precision Tensor Cores on Volta and NVIDIA Turing™, so 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.

- ResNet50 v1.5 model: This model is a modified version of the regular ResNet model that was introduced in the *Deep Residual Learning for Image Recognition* paper.

  The v1.5 has `stride = 2` in the 3x3 convolution instead of 1x1 convolution. This model script is available on GitHub.
- BERT model:This model is a new method of pre-training language representations that obtains state-of-the-art results on a wide array of Natural Language Processing (NLP) tasks that was introduced in the *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* paper.

  This model script is available on GitHub.

Known Issues

- In rare cases, using the adam optimizer with multi-threading might cause segmentation fault. Setting the environment variable `FLAGS_inner_op_parallelism` to 1 can disable the multi-threading feature and resolve this issue.

Chapter 10. PaddlePaddle Release 22.12

The NVIDIA container image for PaddlePaddle, release 22.12, is available on NGC.

Contents of the PaddlePaddle container

This container image includes the complete source of the NVIDIA version of PaddlePaddle in `/opt/paddlepaddle`. It is prebuilt and installed as a system Python module.

The container includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.8.0
- cuTENSOR 1.6.1.5
- NVIDIA cuDNN 8.7.0
- NVIDIA NCCL 2.15.5 (optimized for NVLink™)
- rdma-core 36.0
- OpenMPI 4.1.4+
- GDRCopy 2.3
- Nsight Systems 2022.4.2.1
- Nsight Compute 2022.3.0.22
- NVIDIA HPC-X 2.13
- TensorRT 8.5.1 for x64 Linux
- Paddle-TRT 2.3.2
- SHARP 3.0.2
- DALI 1.20.0

Driver Requirements

Release 22.12 is based on CUDA 11.8.0, which requires NVIDIA Driver release 520 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), or 515.65 (or later R515).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.8. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This PaddlePaddle release includes the following key features and enhancements.

- The PaddlePaddle container image version 22.12 is based on v2.3.2.

Announcements

- Paddle-TRT is now included.

  Paddle-TRT is the TensorRT integration for PaddlePaddle and brings the capabilities of TensorRT to PadddlePaddle in a few lines in the Python and C++ APIs.

NVIDIA PaddlePaddle Container Versions

The following table shows what versions of Ubuntu, CUDA, PaddlePaddle, and TensorRT are supported in each of the NVIDIA containers for PaddlePaddle. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
22.12	22.04	NVIDIA CUDA 11.8.0	2.3.2	TensorRT 8.5.1
22.11				TensorRT 8.5.1
22.10				TensorRT 8.5.0.12
22.09				
22.08		NVIDIA CUDA 11.7.1	2.3.1	TensorRT 8.4.2.4
22.07		NVIDIA CUDA 11.7 Update 1 Preview	2.3.0	TensorRT 8.4.1
22.06			2.2.2	TensorRT 8.2.5
22.05		NVIDIA CUDA 11.7		

Automatic Mixed Precision (AMP)

Automatic Mixed Precision (AMP) for PaddlePaddle is available in this container through the native implementation. AMP enables users to try mixed precision training by adding only 3 lines of Python to an existing FP32 (default) script. AMP will select an optimal set of operations to cast to FP16. FP16 operations require 2X reduced memory bandwidth (resulting in a 2X speedup for bandwidth-bound operations like most pointwise ops)

and 2X reduced memory storage for intermediates (reducing the overall memory consumption of your model). Additionally, GEMMs and convolutions with FP16 inputs can run on Tensor Cores, which provide an 8X increase in computational throughput over FP32 arithmetic.

For more information about AMP, see the Training With Mixed Precision Guide.

Tensor Core Examples

The tensor core examples provided in GitHub and NGC focus on achieving the best performance and convergence from NVIDIA Volta™ tensor cores by using the latest deep learning example networks and model scripts for training.

Each example model trains with mixed precision Tensor Cores on Volta and NVIDIA Turing™, so 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.

- ResNet50 v1.5 model: This model is a modified version of the regular ResNet model that was introduced in the *Deep Residual Learning for Image Recognition* paper.

  The v1.5 has `stride = 2` in the 3x3 convolution instead of 1x1 convolution. This model script is available on GitHub.
- BERT model:This model is a new method of pre-training language representations that obtains state-of-the-art results on a wide array of Natural Language Processing (NLP) tasks that was introduced in the *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* paper.

  This model script is available on GitHub.

Known Issues

- In rare cases, using the adam optimizer with multi-threading might cause segmentation fault. Setting the environment variable `FLAGS_inner_op_parallelism` to 1 can disable the multi-threading feature and resolve this issue.
- On H100 NVLink systems using 2 GPUs for training, certain communication patterns can trigger a corner-case bug that manifests either as a hang or as an "illegal instruction" exception. A workaround for this case is to set the environment variable `NCCL_PROTO=^LL128`. This issue will be addressed in an upcoming release.

Chapter 11. PaddlePaddle Release 22.11

The NVIDIA container image for PaddlePaddle, release 22.11, is available on NGC.

Contents of the PaddlePaddle container

This container image includes the complete source of the NVIDIA version of PaddlePaddle in `/opt/paddlepaddle`. It is prebuilt and installed as a system Python module.

The container includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.8.0
- cuTENSOR 1.6.1.5
- NVIDIA cuDNN 8.7.0
- NVIDIA NCCL 2.15.5 (optimized for NVLink™)
- rdma-core 36.0
- OpenMPI 4.1.4+
- GDRCopy 2.3
- Nsight Systems 2022.4.2.1
- Nsight Compute 2022.3.0.22
- NVIDIA HPC-X 2.12.2tp1
- TensorRT 8.5.1 for x64 Linux
- Paddle-TRT 2.3.2
- SHARP 3.0.2
- DALI 1.18.0

Driver Requirements

Release 22.11 is based on CUDA 11.8.0, which requires NVIDIA Driver release 520 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), or 515.65 (or later R515).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.8. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This PaddlePaddle release includes the following key features and enhancements.

- The PaddlePaddle container image version 22.11 is based on v2.3.2.

Announcements

- Paddle-TRT is now included.

  Paddle-TRT is the TensorRT integration for PaddlePaddle and brings the capabilities of TensorRT to PadddlePaddle in a few lines in the Python and C++ APIs.

NVIDIA PaddlePaddle Container Versions

The following table shows what versions of Ubuntu, CUDA, PaddlePaddle, and TensorRT are supported in each of the NVIDIA containers for PaddlePaddle. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
22.11	22.04	NVIDIA CUDA 11.8.0	2.3.2	TensorRT 8.5.1
22.10				TensorRT 8.5.0.12
22.09				
22.08		NVIDIA CUDA 11.7.1	2.3.1	TensorRT 8.4.2.4
22.07		NVIDIA CUDA 11.7 Update 1 Preview	2.3.0	TensorRT 8.4.1
22.06			2.2.2	TensorRT 8.2.5
22.05		NVIDIA CUDA 11.7		

Automatic Mixed Precision (AMP)

Automatic Mixed Precision (AMP) for PaddlePaddle is available in this container through the native implementation. AMP enables users to try mixed precision training by adding only 3 lines of Python to an existing FP32 (default) script. AMP will select an optimal set of operations to cast to FP16. FP16 operations require 2X reduced memory bandwidth (resulting in a 2X speedup for bandwidth-bound operations like most pointwise ops) and 2X reduced memory storage for intermediates (reducing the overall memory

consumption of your model). Additionally, GEMMs and convolutions with FP16 inputs can run on Tensor Cores, which provide an 8X increase in computational throughput over FP32 arithmetic.

For more information about AMP, see the Training With Mixed Precision Guide.

Tensor Core Examples

The tensor core examples provided in GitHub and NGC focus on achieving the best performance and convergence from NVIDIA Volta™ tensor cores by using the latest deep learning example networks and model scripts for training.

Each example model trains with mixed precision Tensor Cores on Volta and NVIDIA Turing™, so 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.

- ResNet50 v1.5 model: This model is a modified version of the regular ResNet model that was introduced in the *Deep Residual Learning for Image Recognition* paper.

  The v1.5 has `stride = 2` in the 3x3 convolution instead of 1x1 convolution. This model script is available on GitHub.

Known Issues

- In rare cases, using the adam optimizer with multi-threading might cause segmentation fault. Setting the environment variable `FLAGS_inner_op_parallelism` to 1 can disable the multi-threading feature and resolve this issue.
- On H100 NVLink systems using 2 GPUs for training, certain communication patterns can trigger a corner-case bug that manifests either as a hang or as an "illegal instruction" exception. A workaround for this case is to set the environment variable `NCCL_PROTO=^LL128`. This issue will be addressed in an upcoming release.

Chapter 12. PaddlePaddle Release 22.10

The NVIDIA container image for PaddlePaddle, release 22.10, is available on NGC.

Contents of the PaddlePaddle container

This container image includes the complete source of the NVIDIA version of PaddlePaddle in `/opt/paddlepaddle`. It is prebuilt and installed as a system Python module.

The container includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.8.0
- cuTENSOR 1.6.1.5
- NVIDIA cuDNN 8.6.0.163
- NVIDIA NCCL 2.15.5 (optimized for NVLink™)
- rdma-core 36.0
- OpenMPI 4.1.4+
- GDRCopy 2.3
- Nsight Systems 2022.4.2.1
- Nsight Compute 2022.3.0.22
- NVIDIA HPC-X 2.12.2tp1
- TensorRT 8.5.0.12 for x64 Linux
- Paddle-TRT 2.3.2
- SHARP 3.0.2
- DALI 1.18.0

Driver Requirements

Release 22.10 is based on CUDA 11.8.0, which requires NVIDIA Driver release 520 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), or 515.65 (or later R515).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.8. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This PaddlePaddle release includes the following key features and enhancements.

- The PaddlePaddle container image version 22.10 is based on v2.3.2.

Announcements

- Paddle-TRT is now included.

  Paddle-TRT is the TensorRT integration for PaddlePaddle and brings the capabilities of TensorRT to PadddlePaddle in a few lines in the Python and C++ APIs.

NVIDIA PaddlePaddle Container Versions

The following table shows what versions of Ubuntu, CUDA, PaddlePaddle, and TensorRT are supported in each of the NVIDIA containers for PaddlePaddle. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
22.10	22.04	NVIDIA CUDA 11.8.0	2.3.2	TensorRT 8.5.0.12
22.09				
22.08		NVIDIA CUDA 11.7.1	2.3.1	TensorRT 8.4.2.4
22.07		NVIDIA CUDA 11.7 Update 1 Preview	2.3.0	TensorRT 8.4.1
22.06			2.2.2	TensorRT 8.2.5
22.05		NVIDIA CUDA 11.7		

Automatic Mixed Precision (AMP)

Automatic Mixed Precision (AMP) for PaddlePaddle is available in this container through the native implementation. AMP enables users to try mixed precision training by adding only 3 lines of Python to an existing FP32 (default) script. AMP will select an optimal set of operations to cast to FP16. FP16 operations require 2X reduced memory bandwidth (resulting in a 2X speedup for bandwidth-bound operations like most pointwise ops) and 2X reduced memory storage for intermediates (reducing the overall memory consumption of your model). Additionally, GEMMs and convolutions with FP16 inputs

can run on Tensor Cores, which provide an 8X increase in computational throughput over FP32 arithmetic.

For more information about AMP, see the Training With Mixed Precision Guide.

Tensor Core Examples

The tensor core examples provided in GitHub and NGC focus on achieving the best performance and convergence from NVIDIA Volta™ tensor cores by using the latest deep learning example networks and model scripts for training.

Each example model trains with mixed precision Tensor Cores on Volta and NVIDIA Turing™, so 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.

- ResNet50 v1.5 model: This model is a modified version of the regular ResNet model that was introduced in the *Deep Residual Learning for Image Recognition* paper.

  The v1.5 has `stride = 2` in the 3x3 convolution instead of 1x1 convolution. This model script is available on GitHub.

Known Issues

- In rare cases, using the adam optimizer with multi-threading might cause segmentation fault. Setting the environment variable `FLAGS_inner_op_parallelism` to 1 can disable the multi-threading feature and resolve this issue.
- On H100 NVLink systems using 2 GPUs for training, certain communication patterns can trigger a corner-case bug that manifests either as a hang or as an "illegal instruction" exception. A workaround for this case is to set the environment variable `NCCL_PROTO=^LL128`. This issue will be addressed in an upcoming release.
- The `phi::funcs::ReduceAnyKernel` function has a data-racing issue. It will be fixed in an upcoming release.

Chapter 13. PaddlePaddle Release 22.09

The NVIDIA container image for PaddlePaddle, release 22.09, is available on NGC.

Contents of the PaddlePaddle container

This container image includes the complete source of the NVIDIA version of PaddlePaddle in `/opt/paddlepaddle`. It is prebuilt and installed as a system Python module.

The container includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.8.0
- cuTENSOR 1.6.1.5
- NVIDIA cuDNN 8.6.0.163
- NVIDIA NCCL 2.15.1 (optimized for NVLink™)
- rdma-core 36.0
- OpenMPI 4.1.2rc4+
- GDRCopy 2.3
- Nsight Systems 2022.3.1.43
- Nsight Compute 2022.3.0.22
- NVIDIA HPC-X 2.12.1a0
- TensorRT 8.5.0.12 for x64 Linux
- Paddle-TRT 2.3.2
- SHARP 2.6.0
- DALI 1.17.0

Driver Requirements

Release 22.09 is based on CUDA 11.8.0, which requires NVIDIA Driver release 520 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), or 515.65 (or later R515).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.8. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This PaddlePaddle release includes the following key features and enhancements.

- The PaddlePaddle container image version 22.09 is based on v2.3.2.

Announcements

- Paddle-TRT is now included.

  Paddle-TRT is the TensorRT integration for PaddlePaddle and brings the capabilities of TensorRT to PadddlePaddle in a few lines in the Python and C++ APIs.

NVIDIA PaddlePaddle Container Versions

The following table shows what versions of Ubuntu, CUDA, PaddlePaddle, and TensorRT are supported in each of the NVIDIA containers for PaddlePaddle. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
22.09	22.04	NVIDIA CUDA 11.8.0	2.3.2	TensorRT 8.5.0.12
22.08		NVIDIA CUDA 11.7.1	2.3.1	TensorRT 8.4.2.4
22.07		NVIDIA CUDA 11.7 Update 1 Preview	2.3.0	TensorRT 8.4.1
22.06			2.2.2	TensorRT 8.2.5
22.05		NVIDIA CUDA 11.7		

Automatic Mixed Precision (AMP)

Automatic Mixed Precision (AMP) for PaddlePaddle is available in this container through the native implementation. AMP enables users to try mixed precision training by adding only 3 lines of Python to an existing FP32 (default) script. AMP will select an optimal set of operations to cast to FP16. FP16 operations require 2X reduced memory bandwidth (resulting in a 2X speedup for bandwidth-bound operations like most pointwise ops) and 2X reduced memory storage for intermediates (reducing the overall memory consumption of your model). Additionally, GEMMs and convolutions with FP16 inputs

can run on Tensor Cores, which provide an 8X increase in computational throughput over FP32 arithmetic.

For more information about AMP, see the Training With Mixed Precision Guide.

Tensor Core Examples

The tensor core examples provided in GitHub and NGC focus on achieving the best performance and convergence from NVIDIA Volta™ tensor cores by using the latest deep learning example networks and model scripts for training.

Each example model trains with mixed precision Tensor Cores on Volta and NVIDIA Turing™, so 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.

- ResNet50 v1.5 model: This model is a modified version of the regular ResNet model that was introduced in the *Deep Residual Learning for Image Recognition* paper.

  The v1.5 has `stride = 2` in the 3x3 convolution instead of 1x1 convolution. This model script is available on GitHub.

Known Issues

- In rare cases, using the adam optimizer with multi-threading might cause segmentation fault. Setting the environment variable `FLAGS_inner_op_parallelism` to 1 can disable the multi-threading feature and resolve this issue.
- On H100 NVLink systems using 2 GPUs for training, certain communication patterns can trigger a corner-case bug that manifests either as a hang or as an "illegal instruction" exception. A workaround for this case is to set the environment variable `NCCL_PROTO=^LL128`. This issue will be addressed in an upcoming release.

Chapter 14. PaddlePaddle Release 22.08

The NVIDIA container image for PaddlePaddle, release 22.08, is available on NGC.

Contents of the PaddlePaddle container

This container image includes the complete source of the NVIDIA version of PaddlePaddle in `/opt/paddlepaddle`. It is prebuilt and installed as a system Python module.

The container includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.7.1 with NVIDIA cuBLAS 11.10.3.66
- cuTENSOR 1.6.0.2
- NVIDIA cuDNN 8.5.0.96
- NVIDIA NCCL 2.12.12 (optimized for NVLink™)
- rdma-core 36.0
- OpenMPI 4.1.2rc4+
- GDRCopy 2.3
- Nsight Systems 2022.1.3.18
- Nsight Compute 2022.2.1.0
- NVIDIA HPC-X 2.10 with OpenUCX 1.12.0
- TensorRT 8.4.2.4 for x64 Linux
- Paddle-TRT 2.3.1
- SHARP 2.6.0
- DALI 1.16.0

Driver Requirements

Release 22.08 is based on CUDA 11.7.1, which requires NVIDIA Driver release 515 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), or 510.47 (or later R510).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.7. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This PaddlePaddle release includes the following key features and enhancements.

- The PaddlePaddle container image version 22.08 is based on v2.3.1.

Announcements

- Paddle-TRT is now included.

  Paddle-TRT is the TensorRT integration for PaddlePaddle and brings the capabilities of TensorRT to PadddlePaddle in a few lines in the Python and C++ APIs.

NVIDIA PaddlePaddle Container Versions

The following table shows what versions of Ubuntu, CUDA, PaddlePaddle, and TensorRT are supported in each of the NVIDIA containers for PaddlePaddle. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
22.08	22.04	NVIDIA CUDA 11.7.1	2.3.1	TensorRT 8.4.2.4
22.07		NVIDIA CUDA 11.7 Update 1 Preview	2.3.0	TensorRT 8.4.1
22.06			2.2.2	TensorRT 8.2.5
22.05		NVIDIA CUDA 11.7		

Automatic Mixed Precision (AMP)

Automatic Mixed Precision (AMP) for PaddlePaddle is available in this container through the native implementation. AMP enables users to try mixed precision training by adding only 3 lines of Python to an existing FP32 (default) script. AMP will select an optimal set of operations to cast to FP16. FP16 operations require 2X reduced memory bandwidth (resulting in a 2X speedup for bandwidth-bound operations like most pointwise ops) and 2X reduced memory storage for intermediates (reducing the overall memory consumption of your model). Additionally, GEMMs and convolutions with FP16 inputs can run on Tensor Cores, which provide an 8X increase in computational throughput over FP32 arithmetic.

For more information about AMP, see the Training With Mixed Precision Guide.

Tensor Core Examples

The tensor core examples provided in GitHub and NGC focus on achieving the best performance and convergence from NVIDIA Volta™ tensor cores by using the latest deep learning example networks and model scripts for training.

Each example model trains with mixed precision Tensor Cores on Volta and NVIDIA Turing™, so 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.

- ResNet50 v1.5 model: This model is a modified version of the regular ResNet model that was introduced in the *Deep Residual Learning for Image Recognition* paper.

  The v1.5 has `stride = 2` in the 3x3 convolution instead of 1x1 convolution. This model script is available on GitHub.

Known Issues

None.

Chapter 15. PaddlePaddle Release 22.07

The NVIDIA container image for PaddlePaddle, release 22.07, is available on NGC.

Contents of the PaddlePaddle container

This container image includes the complete source of the NVIDIA version of PaddlePaddle in `/opt/paddlepaddle`. It is prebuilt and installed as a system Python module.

The container includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.7 Update 1 Preview with NVIDIA cuBLAS 11.10.1.25
- cuTENSOR 1.5.0.3
- NVIDIA cuDNN 8.4.1
- NVIDIA NCCL 2.12.12 (optimized for NVLink™)
- rdma-core 36.0
- OpenMPI 4.1.2rc4+
- GDRCopy 2.3
- Nsight Systems 2022.1.3.3
- Nsight Compute 2022.2.1.0
- NVIDIA HPC-X 2.10 with OpenUCX 1.12.0
- TensorRT 8.4.1 for x64 Linux
- Paddle-TRT 2.3.0
- SHARP 2.6.0
- DALI 1.15.0

Driver Requirements

Release 22.07 is based on CUDA 11.7 Update 1 Preview, which requires NVIDIA Driver release 515 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), or 510.47 (or later R510).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.7. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This PaddlePaddle release includes the following key features and enhancements.

- The PaddlePaddle container image version 22.07 is based on v2.3.0.

Announcements

- Paddle-TRT is now included.

  Paddle-TRT is the TensorRT integration for PaddlePaddle and brings the capabilities of TensorRT to PadddlePaddle in a few lines in the Python and C++ APIs.

NVIDIA PaddlePaddle Container Versions

The following table shows what versions of Ubuntu, CUDA, PaddlePaddle, and TensorRT are supported in each of the NVIDIA containers for PaddlePaddle. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
22.07	22.04	NVIDIA CUDA 11.7 Update 1 Preview	2.3.0	TensorRT 8.4.1
22.06			2.2.2	TensorRT 8.2.5
22.05		NVIDIA CUDA 11.7		

Automatic Mixed Precision (AMP)

Automatic Mixed Precision (AMP) for PaddlePaddle is available in this container through the native implementation. AMP enables users to try mixed precision training by adding only 3 lines of Python to an existing FP32 (default) script. AMP will select an optimal set of operations to cast to FP16. FP16 operations require 2X reduced memory bandwidth (resulting in a 2X speedup for bandwidth-bound operations like most pointwise ops) and 2X reduced memory storage for intermediates (reducing the overall memory consumption of your model). Additionally, GEMMs and convolutions with FP16 inputs can run on Tensor Cores, which provide an 8X increase in computational throughput over FP32 arithmetic.

For more information about AMP, see the Training With Mixed Precision Guide.

Tensor Core Examples

The tensor core examples provided in GitHub and NGC focus on achieving the best performance and convergence from NVIDIA Volta™ tensor cores by using the latest deep learning example networks and model scripts for training.

Each example model trains with mixed precision Tensor Cores on Volta and NVIDIA Turing™, so 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.

- ResNet50 v1.5 model: This model is a modified version of the regular ResNet model that was introduced in the *Deep Residual Learning for Image Recognition* paper.

  The v1.5 has `stride = 2` in the 3x3 convolution instead of 1x1 convolution. This model script is available on GitHub.

Known Issues

None.

Chapter 16. PaddlePaddle Release 22.06

The NVIDIA container image for PaddlePaddle, release 22.06, is available on NGC.

Contents of the PaddlePaddle container

This container image includes the complete source of the NVIDIA version of PaddlePaddle in `/opt/paddlepaddle`. It is prebuilt and installed as a system Python module.

The container includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.7 Update 1 Preview with NVIDIA cuBLAS 11.10.1.25
- cuTENSOR 1.5.0.3
- NVIDIA cuDNN 8.4.1
- NVIDIA NCCL 2.12.12 (optimized for NVLink™)
- rdma-core 36.0
- OpenMPI 4.1.2rc4+
- GDRCopy 2.3
- Nsight Systems 2022.1.3.3
- Nsight Compute 2022.2.0.0
- NVIDIA HPC-X 2.10 with OpenUCX 1.12.0
- TensorRT 8.2.5 for x64 Linux
- PaddlePaddle-TRT 2.2.2
- SHARP 2.6.0
- DALI 1.14.0

Driver Requirements

Release 22.06 is based on CUDA 11.7 Update 1 Preview, which requires NVIDIA Driver release 515 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), or 510.47 (or later R510).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.7. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This PaddlePaddle release includes the following key features and enhancements.

- The PaddlePaddle container image version 22.06 is based on v2.2.2.

Announcements

- The Paddle-TRT is now included.

  Paddle-TRT is the TensorRT integration for PaddlePaddle and brings the capabilities of TensorRT to PadddlePaddle in a few lines in the Python and C++ APIs.

NVIDIA PaddlePaddle Container Versions

The following table shows what versions of Ubuntu, CUDA, PaddlePaddle, and TensorRT are supported in each of the NVIDIA containers for PaddlePaddle. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
22.06	22.04	NVIDIA CUDA 11.7 Update 1 Preview	2.2.2	TensorRT 8.2.5
22.05		NVIDIA CUDA 11.7		

Automatic Mixed Precision (AMP)

Automatic Mixed Precision (AMP) for PaddlePaddle is available in this container through the native implementation. AMP enables users to try mixed precision training by adding only 3 lines of Python to an existing FP32 (default) script. AMP will select an optimal set of operations to cast to FP16. FP16 operations require 2X reduced memory bandwidth (resulting in a 2X speedup for bandwidth-bound operations like most pointwise ops) and 2X reduced memory storage for intermediates (reducing the overall memory consumption of your model). Additionally, GEMMs and convolutions with FP16 inputs can run on Tensor Cores, which provide an 8X increase in computational throughput over FP32 arithmetic.

For more information about AMP, see the Training With Mixed Precision Guide.

Tensor Core Examples

The tensor core examples provided in GitHub and NGC focus on achieving the best performance and convergence from NVIDIA Volta™ tensor cores by using the latest deep learning example networks and model scripts for training.

Each example model trains with mixed precision Tensor Cores on Volta and NVIDIA Turing™, so 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.

- ResNet50 v1.5 model: This model is a modified version of the regular ResNet model that was introduced in the *Deep Residual Learning for Image Recognition* paper.

  The v1.5 has `stride = 2` in the 3x3 convolution instead of 1x1 convolution. This model script is available on GitHub.

Known Issues

None.

Chapter 17. PaddlePaddle Release 22.05

The NVIDIA container image for PaddlePaddle, release 22.05, is available on NGC.

Contents of the PaddlePaddle container

This container image includes the complete source of the NVIDIA version of PaddlePaddle in `/opt/paddlepaddle`. It is prebuilt and installed as a system Python module.

The container includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.7 with NVIDIA cuBLAS 11.10.1.25
- cuTensor 1.5.0.3
- NVIDIA cuDNN 8.4.0.27
- NVIDIA NCCL 2.12.10 (optimized for NVLink™)
- rdma-core 36.0
- OpenMPI 4.1.2rc4+
- GDRCopy 2.3
- Nsight Systems 2022.1.3.3
- Nsight Compute 2022.2.0.0
- NVIDIA HPC-X 2.10 with OpenUCX 1.12.0
- TensorRT 8.2.5.1 for x64 Linux
- PaddlePaddle-TRT 2.2.2
- SHARP 2.6.0
- DALI 1.13.0

Driver Requirements

Release 22.05 is based on CUDA 11.7, which requires NVIDIA Driver release 515 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), or 510.47 (or later R510).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.7. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This PaddlePaddle release includes the following key features and enhancements.

- The PaddlePaddle container image version 22.05 is based on v2.2.2.

Announcements

- The Paddle-TRT is now included.

  Paddle-TRT is the TensorRT integration for PaddlePaddle and brings the capabilities of TensorRT to PadddlePaddle in a few lines in the Python and C++ APIs.

NVIDIA PaddlePaddle Container Versions

The following table shows what versions of Ubuntu, CUDA, PaddlePaddle, and TensorRT are supported in each of the NVIDIA containers for PaddlePaddle. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	PaddlePaddle	TensorRT
22.05	22.04	NVIDIA CUDA 11.7	2.2.2	TensorRT 8.2.5.1

Automatic Mixed Precision (AMP)

Automatic Mixed Precision (AMP) for PaddlePaddle is available in this container through the native implementation. AMP enables users to try mixed precision training by adding only 3 lines of Python to an existing FP32 (default) script. AMP will select an optimal set of operations to cast to FP16. FP16 operations require 2X reduced memory bandwidth (resulting in a 2X speedup for bandwidth-bound operations like most pointwise ops) and 2X reduced memory storage for intermediates (reducing the overall memory consumption of your model). Additionally, GEMMs and convolutions with FP16 inputs can run on Tensor Cores, which provide an 8X increase in computational throughput over FP32 arithmetic.

For more information about AMP, see the Training With Mixed Precision Guide.

Tensor Core Examples

The tensor core examples provided in GitHub and NGC focus on achieving the best performance and convergence from NVIDIA Volta™ tensor cores by using the latest deep learning example networks and model scripts for training.

Each example model trains with mixed precision Tensor Cores on Volta and NVIDIA Turing™, so 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.

- ResNet50 v1.5 model: This model is a modified version of the regular ResNet model that was introduced in the *Deep Residual Learning for Image Recognition* paper.

  The v1.5 has `stride = 2` in the 3x3 convolution instead of 1x1 convolution. This model script is available on GitHub.

Known Issues

None.

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform 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 document. NVIDIA accepts no 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 document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
https://www.nvidia.com

Trademarks

NVIDIA, the NVIDIA logo, and cuBLAS, CUDA, DALI, DGX, DGX-1, DGX-2, DGX Station, DLProf, Jetson, Kepler, Maxwell, NCCL, Nsight Compute, Nsight Systems, NvCaffe, PerfWorks, Pascal, SDK Manager, Tegra, TensorRT, Triton Inference Server, Tesla, TF-TRT, and Volta are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
https://www.nvidia.com