Running TensorFlow on vSphere Bitfusion

22 JAN 2021

VMware vSphere Bitfusion 2.5

vmware®

You can find the most up-to-date technical documentation on the VMware website at:

https://docs.vmware.com/

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

About *Running TensorFlow on vSphere Bitfusion*

The *Running TensorFlow on vSphere Bitfusion* provides information about using vSphere Bitfusion to run TensorFlow on VMware vSphere.

Running TensorFlow on vSphere Bitfusion describes how to install TensorFlow and open-source benchmarks, and then run the benchmarks using vSphere Bitfusion. This guide serves as a basis for understanding how to use TensorFlow and other artificial intelligence (AI) and machine learning (ML) applications and frameworks under vSphere Bitfusion.

Intended Audience

This information is intended for anyone who wants to install, upgrade, or use vSphere Bitfusion. The information is written for experienced Linux system administrators who are familiar with virtual machine technology and data center operations using VMware vSphere.

Updated Information

This *Running TensorFlow on vSphere Bitfusion* guide is updated with each release of the product or when necessary.

This table provides the update history of the *Running TensorFlow on vSphere Bitfusion* guide.

Revision	Description
22 JAN 2021	This update of the *Running TensorFlow on vSphere Bitfusion* includes the latest supported operating system, driver, and software versions, which are supported with VMware vSphere Bitfusion 2.5. ■ Updated operating systems to Ubuntu 20.04, CentOS 8, and Red Hat Linux 8. ■ Updated NVIDIA CUDA version to 11. ■ Updated NVIDIA cuDNN version to 8. ■ Updated Python workflow steps. ■ Updated TensorFlow version to 2.4.
05 NOV 2020	Initial release.

Introduction to Using TensorFlow with vSphere Bitfusion

1

To use TensorFlow with vSphere Bitfusion, you must install and configure several components.

To use TensorFlow with vSphere Bitfusion, complete the following tasks.

1 Install vSphere Bitfusion. See the *VMware vSphere Bitfusion Installation Guide*.

2 Install NVIDIA CUDA 11.

3 Install NVIDIA cuDNN 8.

4 If you are using CentOS or Red Hat Linux, you must install Python 3.

5 Install TensorFlow 2.4.

6 Install TensorFlow benchmarks.

7 Run the TensorFlow benchmarks to measure your system's performance.

Installing and Running TensorFlow with vSphere Bitfusion

2

To use TensorFlow with vSphere Bitfusion, you install and configure several software packages and programming frameworks.

This chapter includes the following topics:

Installing NVIDIA CUDA

CUDA is a parallel computing platform and programming model developed by NVIDIA for general computing on graphical processing units (GPUs). CUDA dramatically speeds up computing applications by using the processing power of GPUs. CUDA is used by TensorFlow benchmarks.

Install NVIDIA CUDA on Ubuntu

You can install CUDA on Ubuntu Linux.

Verify you have installed vSphere Bitfusion client on the Ubuntu operating system.

Procedure

1. Navigate to a directory on the virtual machine in which to download the NVIDIA CUDA distribution.

   ```
   cd <download_directory>
   ```

2. Download and move the `cuda-ubuntu2004.pin` file.

   ```
   wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/cuda-ubuntu2004.pin
   sudo mv cuda-ubuntu2004.pin /etc/apt/preferences.d/cuda-repository-pin-600
   ```

3 Download the NVIDIA CUDA distribution for Ubuntu 20.04 by using the `wget` command.

```
wget <https://developer.download.nvidia.com/compute/cuda/11.0.3/local_installers/cuda-repo-ubuntu2004-11-0-local_11.0.3-450.51.06-1_amd64.deb>
```

4 Install the CUDA 11 package for Ubuntu 20.04 by using the `dpkg -i` command.

```
sudo dpkg -i cuda-repo-ubuntu2004-11-0-local_11.0.3-450.51.06-1_amd64.deb
```

5 Install the keys to authenticate the software package by using the `apt-key` command.

The `apt-key` command manages the list of keys used by `apt` to authenticate packages. Packages which have been authenticated using these keys are considered to be trusted.

```
sudo apt-key add /var/cuda-repo-ubuntu2004-11-0-local/7fa2af80.pub
```

6 Update and install the CUDA software package.

```
sudo apt-get update
sudo apt-get install cuda
```

7 (Optional) To confirm your GPU partition size or verify the resources available on your vSphere Bitfusion deployment, run the NVIDIA System Management Interface (`nvidia-smi`) monitoring application .

```
bitfusion run -n 1 nvidia-smi
```

8 Navigate to the directory that contains the CUDA Matrix Multiplication (matrixMul) sample files.

```
cd /usr/local/cuda/samples/0_Simple/matrixMul
```

9 Run the `make` and `bitfusion run` commands against the `matrixMul` sample file.

```
sudo make
bitfusion run -n 1 ./matrixMul
```

What to do next

Install and configure NVIDIA cuDNN. See Install NVIDIA cuDNN.

Install NVIDIA CUDA on CentOS or Red Hat Linux

You can install CUDA 11 on CentOS 8 or Red Hat Linux 8.

Procedure

1. Navigate to directory on the virtual machine in which to download the NVIDIA CUDA distribution.

   ```
   cd <download_directory>
   ```

2. To download the NVIDIA CUDA 11 package for CentOS 8 or Red Hat Linux 8, run the `wget` command.

   ```
   wget https://developer.download.nvidia.com/compute/cuda/11.0.3/local_installers/cuda-repo-rhel8-11-0-local-11.0.3_450.51.06-1.x86_64.rpm
   ```

3. To install the CUDA package, run the `rpm -i` command.

   ```
   sudo rpm -i cuda-repo-rhel8-11-0-local-11.0.3_450.51.06-1.x86_64.rpm
   ```

4. Run the `yum clean all` and `yum -y install` commands as shown to update your environment and install the CUDA software package.

   ```
   sudo yum clean all
   sudo yum -y install cuda
   ```

5. (Optional) To confirm your GPU partition size or verify the resources available on your vSphere Bitfusion deployment, run the NVIDIA System Management Interface (`nvidia-smi`) monitoring application .

   ```
   bitfusion run -n 1 nvidia-smi
   ```

6. Navigate to the directory containing the CUDA Matrix Multiplication (matrixMul) sample files.

   ```
   cd /usr/local/cuda/samples/0_Simple/matrixMul
   ```

7. Run the `make` and `bitfusion run` commands against the matrixMul sample file.

   ```
   sudo make
   bitfusion run -n 1 ./matrixMul
   ```

What to do next

Install and configure NVIDIA cuDNN. See Install NVIDIA cuDNN.

Install NVIDIA cuDNN

cuDNN is a GPU-accelerated library of primitives for use with deep neural networks.

Prerequisites

Create an NVIDIA developer account from which to download the cuDNN package matching your NVIDIA CUDA version, and appropriate for your Linux distribution. See https://developer.nvidia.com/cudnn.

Procedure

1 Install the cuDNN package by running the command sequence for your Linux distribution.

- Ubuntu version 20.04

```
sudo dpkg -i libcudnn8_8.0.5.39-1+cuda11.0_amd64.deb
```

- CentOS 8 and Red Hat Linux 8

```
sudo rpm -ivh libcudnn8-8.0.5.39-1.cuda11.0.x86_64.rpm
```

2 To verify that cuDNN is installed, run `ldconfig -p | grep cudnn`.

What to do next

- If you are using CentOS or Red Hat Linux, you must install Python 3. See Install Python on CentOS and Red Hat Linux.
- If you are using Ubuntu Linux, you can install TensorFlow. See Install TensorFlow.

Install Python on CentOS and Red Hat Linux

For CentOS and Red Hat Linux, you must install Python 3.

If you are using Ubuntu you do not have to perform this procedure. Ubuntu comes preinstalled with Python 3.

Procedure

1 Update all currently installed packages by running the `yum update` command.

```
sudo yum update
```

2 To install Python 3, run the `dnf` command.

```
sudo dnf install python3
```

3 Verify that you are using Python 3 with the `python3 -V` command.

```
python3 -V
Python 3.6.8
```

4 (Optional) Take a snapshot of your environment.

What to do next

Install TensorFlow. See Install TensorFlow.

Install TensorFlow

TensorFlow is the machine learning (ML) framework you use with Bitfusion.

Install TensorFlow by using `pip3`, which is the package installer for Python 3.

Procedure

1 If you install TensorFlow on Ubuntu 20.04, install additional Python resources.

```
sudo apt-get -y install python3-testresources
```

2 Install `pip3` by running the command sequence for your Linux distribution and version.

- Ubuntu 20.04

```
sudo apt-get install -y python3-pip
```

- CentOS 8 and Red Hat Linux 8

```
sudo yum install -y python36-devel
sudo pip3 install -U pip setuptools
```

3 Install TensorFlow by using the `pip3 install` command.

```
sudo pip3 install tensorflow-gpu==2.4
```

What to do next

You can run TensorFlow benchmarks to test your vSphere Bitfusion deployment's performance. See Install TensorFlow BenchMarks.

Install TensorFlow BenchMarks

The TensorFlow benchmarks are open-source ML applications designed to test the performance of the TensorFlow framework.

You branch and download the TensorFlow benchmarks to your local environment. In Git, a branch is a separate line of development.

Procedure

1 Install Git.

```
sudo yum -y update
sudo yum install git
```

2 Create and make `~/bitfusion` your working directory.

```
mkdir bitfusion
cd ~/bitfusion
```

3 Clone the Git repository of Tensorflow benchmarks to your local environment.

```
git clone https://github.com/tensorflow/benchmarks.git
```

4 Navigate to the benchmarks directory and list branches of the repository.

```
cd benchmarks
git branch -a
```

```
master
remotes/origin/HEAD -> origin/master
...
remotes/origin/cnn_tf_v1.13_compatible
...
remotes/origin/cnn_tf_v2.1_compatible
...
```

5 Do a Git checkout and list the TensorFlow benchmarks repository.

```
git checkout cnn_tf_v2.1_compatible
```

```
Branch cnn_tf_v2.1_compatible set up to track remote branch cnn_tf_v2.1_compatible
from origin.
Switched to a new branch 'cnn_tf_v2.1_compatible'
```

```
git branch
```

```
cnn_tf_tf_v2.1_compatible
master
```

What to do next

You can run TensorFlow benchmarks to test your vSphere Bitfusion deployment's performance. See Run TensorFlow Benchmarks.

Run TensorFlow Benchmarks

You can run the TensorFlow benchmarks to test the performance of your vSphere Bitfusion and TensorFlow deployment.

By running the TensorFlow benchmarks and using various configurations, you can understand how ML workloads respond in your vSphere Bitfusion environment.

Procedure

1. To navigate to the `~/bitfusion/` directory, run `cd ~/bitfusion/`.
2. To use the `tf_cnn_benchmarks.py` benchmark script, run the `bitfusion run` command.

   By running the commands in the example, you use the entire memory of a single GPU and pre-installed ML data in the `/data` directory.

   ```
   bitfusion run -n 1 -- python3 \
   ./benchmarks/scripts/tf_cnn_benchmarks/tf_cnn_benchmarks.py \
   --data_format=NCHW \
   --batch_size=64 \
   --model=resnet50 \
   --variable_update=replicated \
   --local_parameter_device=gpu \
   --nodistortions \
   --num_gpus=1 \
   --num_batches=100 \
   --data_dir=/data \
   --data_name=imagenet \
   --use_fp16=False
   ```

3. To use the `tf_cnn_benchmarks.py` benchmark script, run the `bitfusion run` command with the `-p 0.67` parameter.

   By running the commands in the example, you use 67% of the memory of a single GPU and pre-installed ML data in the `/data` directory. The `-p 0.67` parameter lets you run another job in the remaining 33% of the GPU's memory partition.

   ```
   bitfusion run -n 1 -p 0.67 -- python3 \
   ./benchmarks/scripts/tf_cnn_benchmarks/tf_cnn_benchmarks.py \
   --data_format=NCHW \
   --batch_size=64 \
   --model=resnet50 \
   --variable_update=replicated \
   --local_parameter_device=gpu \
   --nodistortions \
   --num_gpus=1 \
   --num_batches=100 \
   --data_dir=/data \
   --data_name=imagenet \
   --use_fp16=False
   ```

4. To use the `tf_cnn_benchmarks.py` benchmark script, run the `bitfusion run` command with synthesized data.

   By running the commands in the example, you use the entire memory of a single GPU and no pre-installed ML data. TensorFlow can create synthesized data with a pretend set of images.

   ```
   bitfusion run -n 1 -- python3 \
   ./benchmarks/scripts/tf_cnn_benchmarks/tf_cnn_benchmarks.py \
   --data_format=NCHW \
   --batch_size=64 \
   ```

```
--model=resnet50 \
--variable_update=replicated \
--local_parameter_device=gpu \
--nodistortions \
--num_gpus=1 \
--num_batches=100 \
--use_fp16=False
```

Results

You can now run TensorFlow benchmarks with vSphere Bitfusion with shared GPUs from a remote server. The benchmarks support many models and parameters to help you explore a large space within the machine learning discipline. For more information, see *VMware vSphere Bitfusion User Guide*.