

VMware vSphere Bitfusion 示例指南

2021 年 5 月 11 日

更新以包含 VMware vSphere Bitfusion 3.5

VMware vSphere Bitfusion 3.0

您可以从 VMware 网站下载最新的技术文档:

<https://docs.vmware.com/cn/>。

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

**威睿信息技术（中国）有
限公司**
北京办公室
北京市
朝阳区新源南路 8 号
启皓北京东塔 8 层 801
www.vmware.com/cn

上海办公室
上海市
淮海中路 333 号
瑞安大厦 804-809 室
www.vmware.com/cn

广州办公室
广州市
天河路 385 号
太古汇一座 3502 室
www.vmware.com/cn

版权所有 © 2020-2021 VMware, Inc. 保留所有权利。 [版权和商标信息](#)

目录

关于《vSphere Bitfusion 示例指南》	4
1 简介：将 TensorFlow 与 vSphere Bitfusion 配合使用	5
2 使用 vSphere Bitfusion 安装和运行 TensorFlow	6
安装 NVIDIA CUDA	6
在 Ubuntu 上安装 NVIDIA CUDA	6
在 CentOS 或 Red Hat Linux 上安装 NVIDIA CUDA	7
安装 NVIDIA cuDNN	8
在 CentOS 和 Red Hat Linux 上安装 Python	9
安装 TensorFlow	9
安装 TensorFlow 基准测试	10
运行 TensorFlow 基准测试	11

关于《vSphere Bitfusion 示例指南》

《vSphere Bitfusion 示例指南》提供了有关使用 vSphere Bitfusion 在 VMware vSphere 上运行 TensorFlow 的信息。

VMware 非常重视包容性。为了在客户、合作伙伴和内部社区中促进这一原则，我们采用包容性语言创建内容。

《vSphere Bitfusion 示例指南》介绍如何安装 TensorFlow 和开源基准测试，然后使用 vSphere Bitfusion 运行基准测试。本指南是帮助您了解如何在 vSphere Bitfusion 下使用 TensorFlow 以及其他人工智能 (AI) 和机器学习 (ML) 应用程序和框架的基础。

目标读者

本信息主要面向需要将 vSphere Bitfusion 与机器学习平台配合使用的用户。本信息的目标读者为熟悉使用 VMware vSphere 实施虚拟机技术和执行数据中心操作且具有丰富经验的 Linux 系统管理员。

简介：将 TensorFlow 与 vSphere Bitfusion 配合使用

1

要将 TensorFlow 与 vSphere Bitfusion 配合使用，必须安装并配置多个组件。

要将 TensorFlow 与 vSphere Bitfusion 配合使用，请完成以下任务。

- 1 安装 vSphere Bitfusion。请参见 VMware vSphere Bitfusion 安装指南。
- 2 安装 NVIDIA CUDA 11。
- 3 安装 NVIDIA cuDNN 8。
- 4 如果使用的是 CentOS 或 Red Hat Linux，则必须安装 Python 3。
- 5 安装 TensorFlow 2.4。
- 6 安装 TensorFlow 基准测试。
- 7 运行 TensorFlow 基准测试以衡量系统性能。

使用 vSphere Bitfusion 安装和运行 TensorFlow

2

要将 TensorFlow 与 vSphere Bitfusion 配合使用，请安装并配置多个软件包和编程框架。

本章讨论了以下主题：

- 安装 NVIDIA CUDA
- 安装 NVIDIA cuDNN
- 在 CentOS 和 Red Hat Linux 上安装 Python
- 安装 TensorFlow
- 安装 TensorFlow 基准测试
- 运行 TensorFlow 基准测试

安装 NVIDIA CUDA

CUDA 是由 NVIDIA 开发的一种并行计算平台和编程模型，用于在图形处理单元 (GPU) 上进行常规计算。CUDA 可利用 GPU 的处理能力大幅提高计算应用程序的速度。TensorFlow 基准测试将使用 CUDA。

在 Ubuntu 上安装 NVIDIA CUDA

可以在 Ubuntu Linux 上安装 CUDA。

确认已在 Ubuntu 操作系统上安装 vSphere Bitfusion 客户端。

步骤

- 1 导航到虚拟机上要将 NVIDIA CUDA 发行版下载到其中的目录。

```
cd <download_directory>
```

- 2 下载并移动 cuda-ubuntu2004.pin 文件。

```
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 使用 `wget` 命令下载适用于 Ubuntu 20.04 的 NVIDIA CUDA 发行版。

```
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 使用 `dpkg -i` 命令安装适用于 Ubuntu 20.04 的 CUDA 11 软件包。

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

- 5 使用 `apt-key` 命令安装密钥以对软件包进行身份验证。

`apt-key` 命令可管理 `apt` 用于对软件包进行身份验证的密钥列表。使用这些密钥进行了身份验证的软件包将视为可信。

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

- 6 更新并安装 CUDA 软件包。

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

- 7 （可选）要确认 GPU 分区大小或验证 vSphere Bitfusion 部署上可用的资源，请运行 NVIDIA System Management Interface (`nvidia-smi`) 监控应用程序。

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

- 8 导航到包含 CUDA 矩阵乘法 (`matrixMul`) 示例文件的目录。

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

- 9 针对 `matrixMul` 示例文件运行 `make` 和 `bitfusion run` 命令。

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

后续步骤

安装并配置 NVIDIA cuDNN。请参见[安装 NVIDIA cuDNN](#)。

在 CentOS 或 Red Hat Linux 上安装 NVIDIA CUDA

可以在 CentOS 8 或 Red Hat Linux 8 上安装 CUDA 11。

步骤

- 1 导航到虚拟机上要将 NVIDIA CUDA 发行版下载到其中的目录。

```
cd <download_directory>
```

- 2 要下载适用于 CentOS 8 或 Red Hat Linux 8 的 NVIDIA CUDA 11 软件包，请运行 `wget` 命令。

```
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 要安装 CUDA 软件包，请运行 `rpm -i` 命令。

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

- 4 运行如下所示的 `yum clean all` 和 `yum -y install` 命令以更新您的环境并安装 CUDA 软件包。

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

- 5 （可选）要确认 GPU 分区大小或验证 vSphere Bitfusion 部署上可用的资源，请运行 NVIDIA System Management Interface (`nvidia-smi`) 监控应用程序。

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

- 6 导航到包含 CUDA 矩阵乘法 (`matrixMul`) 示例文件的目录。

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

- 7 针对 `matrixMul` 示例文件运行 `make` 和 `bitfusion run` 命令。

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

后续步骤

安装并配置 NVIDIA cuDNN。请参见[安装 NVIDIA cuDNN](#)。

安装 NVIDIA cuDNN

cuDNN 是一个 GPU 加速的原语库，用于深度神经网络。

前提条件

创建一个 NVIDIA 开发人员帐户，以从该帐户下载与您的 NVIDIA CUDA 版本匹配并适用于您的 Linux 发行版的 cuDNN 软件包。请参见 <https://developer.nvidia.com/cudnn>。

步骤

- 1 通过运行适用于您的 Linux 发行版的命令序列安装 cuDNN 软件包。

- ◆ Ubuntu 版本 20.04

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

- ◆ CentOS 8 和 Red Hat Linux 8

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

- 2 要验证 cuDNN 是否已安装，请运行 `ldconfig -p | grep cudnn`。

后续步骤

- 如果使用的是 CentOS 或 Red Hat Linux，则必须安装 Python 3。请参见在 [CentOS 和 Red Hat Linux 上安装 Python](#)。
- 如果使用的是 Ubuntu Linux，则可以安装 TensorFlow。请参见[安装 TensorFlow](#)。

在 CentOS 和 Red Hat Linux 上安装 Python

对于 CentOS 和 Red Hat Linux，必须安装 Python 3。

如果使用 Ubuntu，则不需要执行此过程。Ubuntu 预装了 Python 3。

步骤

- 1 运行 `yum update` 命令，更新当前安装的所有软件包。

```
sudo yum update
```

- 2 要安装 Python 3，请运行 `dnf` 命令。

```
sudo dnf install python3
```

- 3 运行 `python3 -V` 命令，验证使用的是否是 Python 3。

```
python3 -V  
Python 3.6.8
```

- 4 （可选）生成环境的快照。

后续步骤

安装 TensorFlow。请参见[安装 TensorFlow](#)。

安装 TensorFlow

TensorFlow 是可与 Bitfusion 配合使用的机器学习 (ML) 框架。

使用 `pip3`（即 Python 3 的软件包安装程序）安装 TensorFlow。

步骤

- 1 如果在 Ubuntu 20.04 中安装 TensorFlow，请安装其他 Python 资源。

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

- 2 通过运行适用于您的 Linux 分发版和版本的命令序列来安装 pip3。

- Ubuntu 20.04

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

- CentOS 8 和 Red Hat Linux 8

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

- 3 使用 pip3 install 命令安装 TensorFlow。

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

后续步骤

可以运行 TensorFlow 基准测试来测试 vSphere Bitfusion 部署的性能。请参见[安装 TensorFlow 基准测试](#)。

安装 TensorFlow 基准测试

TensorFlow 基准测试是用于测试 TensorFlow 框架性能的开源 ML 应用程序。

您可以针对 TensorFlow 基准测试创建分支并下载到本地环境中。在 Git 中，分支是一个单独的开发系列。

步骤

- 1 安装 Git。

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

- 2 创建 ~/bitfusion 并将其设置为工作目录。

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

- 3 将 Tensorflow 基准测试的 Git 存储库克隆到本地环境。

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

4 导航到存储库的基准目录和列表分支。

```
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 执行 Git 签出并列出 TensorFlow 基准存储库。

```
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
```

后续步骤

可以运行 TensorFlow 基准测试来测试 vSphere Bitfusion 部署的性能。请参见[运行 TensorFlow 基准测试](#)。

运行 TensorFlow 基准测试

您可以运行 TensorFlow 基准测试，以测试 vSphere Bitfusion 和 TensorFlow 部署的性能。

通过运行 TensorFlow 基准测试和使用各种配置，可以了解 vSphere Bitfusion 环境中 ML 工作负载的响应方式。

步骤

- 1 要导航到 `~/bitfusion/` 目录，请运行 `cd ~/bitfusion/`。
- 2 要使用 `tf_cnn_benchmarks.py` 基准测试脚本，请运行 `bitfusion run` 命令。

通过运行示例中的命令，可以使用单个 GPU 的全部内存和 `/data` 目录中预安装的 ML 数据。

```
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 要使用 `tf_cnn_benchmarks.py` 基准测试脚本，请使用 `bitfusion run` 参数运行 `-p 0.67` 命令。

通过运行示例中的命令，将使用单个 GPU 中 67% 的内存和 `/data` 目录中的预安装 ML 数据。使用 `-p 0.67` 参数，您可以在 GPU 剩余的 33% 内存分区中运行另一个作业。

```
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 要使用 `tf_cnn_benchmarks.py` 基准测试脚本，请使用合成数据运行 `bitfusion run` 命令。

通过运行示例中的命令，可以使用单个 GPU 的全部内存，而不使用预安装的 ML 数据。TensorFlow 可以使用一组虚构映像创建合成数据。

```
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
```

结果

现在，可以通过具有共享 GPU 的 vSphere Bitfusion 从远程服务器运行 TensorFlow 基准测试。基准测试支持许多模型和参数，可帮助您探索机器学习学科的广阔空间。有关详细信息，请参见 VMware vSphere Bitfusion 用户指南。