TencentOS Server

操作指南

腾讯云

文档目录

操作指南
使用方式

最近更新时间：2024-07-05 18:17:01

您可以在云上快速开始使用 TencentOS Server。

云上使用

您可以在腾讯云创建实例，或重装已有实例操作系统时，选择公共镜像，并选择使用 TencentOS Server 的相应版本。操作详情请参见 创建实例 及 重装系统 。

CentOS 迁移 TencentOS 指引

最近更新时间：2024-07-05 19:29:21

操作场景

CentOS 官方停止维护 CentOS 7、CentOS 8 项目，CentOS 7及 CentOS 8 停止维护时间见下表。如需了解更多信息，请参见 CentOS 官方公告 。

操作系统版本	停止维护时间	使用者影响
CentOS 8	2022年01月01日	停止维护后将无法获得包括问题修复和功能更新在内的任何软件维护和支持。
CentOS 7	2024年06月30日	

针对以上情况，若您需新购云服务器实例，建议选择使用 TencentOS Server 镜像。若您正在使用 CentOS 实例，则可参考本文替换为 TencentOS Server。

版本说明

源端主机支持操作系统版本：

- 支持 CentOS 7系列操作系统版本：
  - CentOS 7.2 64位、CentOS 7.3 64位、CentOS 7.4 64位、CentOS 7.5 64位、CentOS 7.6 64位、CentOS 7.7 64位、CentOS 7.8 64位、CentOS 7.9 64位。
- 支持 CentOS 8系列操作系统版本：
  - CentOS 8.0 64位、CentOS 8.2 64位、CentOS 8.3 64位、CentOS 8.4 64位、CentOS 8.5 64位。

目标主机建议操作系统版本：

- CentOS 7系列建议迁移至 TencentOS Server 2.4 (TK4)。
- CentOS 8系列建议迁移至 TencentOS Server 3.1 (TK4)。

注意事项

- 以下情况可能会影响业务在迁移后无法正常运行：
  - 业务程序安装且依赖了第三方的 rpm 包。
  - 迁移后的目标版本是 tkernel4，基于5.4的内核。该版本较 CentOS 7及 CentOS 8的内核版本更新，一些较旧的特性在新版本可能会发生变化。建议强依赖于内核的用户了解所依赖的特性，或可咨询 在线客服 。
  - 业务程序依赖某个固定的 gcc 版本。
    目前 TencentOS Server 2.4默认安装 gcc 4.8.5，TencentOS Server 3.1默认安装 gcc 8.5。
- 迁移结束后，需重启才能进入TencentOS Server 内核。
- 迁移不影响数据盘，仅 OS 层面的升级，不会对数据盘进行任何操作。

> ⚠ **注意：**
> 操作系统迁移会将内核升级为基于 5.4 版本的 tkernel4 内核，因此可能下列情况的系统可能受到影响：
> 1. 业务程序依赖于某个固定的内核版本，或者自行编译了内核模块，如 GPU 机型迁移后内核需要重新安装 GPU 驱动；
> 2. 原操作系统的某个模块由 rpm 包提供，在迁移后此 rpm 包可能无法为新的内核提供模块，如：
>
>   xpmem-modules-2.6.3-2.54310.kver.3.10.0_1160.108.1.el7.x86_64.x86_64 为 kernel-3.10.0-1160.108.1.el7.x86_64 提供 ko 文件，
>   但无法为迁移后的 tkernel4 内核提供。这种情况下用户可获取源码重新编译安装该模块。

资源要求

- 空闲内存大于500MB。
- 系统盘剩余空间大于10GB。
- 若/boot挂载分区，该分区空间需要大于500MB。

操作步骤

迁移准备

1. 迁移操作不可逆，为保障业务数据安全，强烈建议您在执行迁移前通过 创建快照 备份系统盘数据。
2. 操作系统迁移需要用户具有 root 权限。

执行迁移

CentOS 7系列迁移至 TencentOS Server 2.4（TK4）

1. 登录目标云服务器，详情请参见 使用标准登录方式登录 Linux 实例 。
2. 执行以下命令，获取迁移工具。

> **注意：**
> 若您的系统安装了旧版本的迁移工具，请卸载后再安装新的工具包。

```
wget https://mirrors.cloud.tencent.com/tencentos/2.4/tlinux/x86_64/RPMS/migrate2tencentos-1.07-6.tl2.x86_64.rpm
```

3. 执行以下命令，安装迁移工具。

```
rpm -ivh migrate2tencentos-1.07-6.tl2.x86_64.rpm
```

4. 执行以下命令，开始迁移。

4.1 通过下面命令之一进行迁移

4.1.1 全量迁移

将 CentOS 发行版的用户态软件包替换为 TencentOS 发行版，为系统安装 TencentOS 基于5.4的内核。

```
/usr/local/bin/EasyMigration -d remote -k
```

4.1.2 minimal 软件组迁移

将系统的核心组件包迁移成 TencentOS 发行版，为系统安装 TencentOS 基于5.4的内核。
该模式下迁移的用户态软件包规模较小，系统上其他非核心组件的软件仍然保留为 CentOS 发行版。

```
/usr/local/bin/EasyMigration -d remote -k -g minimal
```

minimal 软件组默认列表参考页面底部 附录一 。
迁移需要一定时间，请耐心等待。脚本执行完成后，输出如下图所示信息，表示已完成迁移。

```
INFO: Migration Switch complete. TencentOS recommends rebooting this system.
```

5. 重启实例，详情请参见 重启实例 。
6. 检查迁移结果。

6.1 执行以下命令，检查 os-release。

```
cat /etc/os-release
```

返回如下图所示信息：

```
[root@VM-2-43-centos ~]# cat /etc/os-release
NAME="TencentOS Server"
VERSION="2.4"
ID="tencentos"
ID_LIKE="rhel fedora centos tlinux"
VERSION_ID="2.4"
PRETTY_NAME="TencentOS Server 2.4"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:tencentos:tencentos:2"
HOME_URL="https://cloud.tencent.com/product/ts"
```

6.2 执行以下命令，检查内核。

```
uname -r
```

返回如下图所示信息：

```
[root@VM-2-43-centos ~]# uname -r
5.4.119-19-0009.1
[root@VM-2-43-centos ~]#
```

> **说明：**
> 内核默认为 yum 最新版本，请以您的实际返回结果为准，本文以图示版本为例。

6.3 执行以下命令，检查 yum。

```
yum makecache
```

返回如下图所示信息：

```
[root@VM-2-43-centos ~]# yum makecache
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * epel: mirrors.tencentyun.com
 * tlinux: mirrors.tencentyun.com
 * tlinux-extras: mirrors.tencentyun.com
 * tlinux-os: mirrors.tencentyun.com
 * tlinux-updates: mirrors.tencentyun.com
epel
tlinux
tlinux-extras
tlinux-os
tlinux-tkernel4
tlinux-updates
Metadata Cache Created
[root@VM-2-43-centos ~]#
```

CentOS 8系列迁移至 TencentOS 3.1（TK4）

1. 登录目标云服务器，详情请参见 使用标准登录方式登录 Linux 实例 。
2. 执行以下命令，获取迁移工具。

> **注意：**
> 若您的系统曾经安装了旧版本的迁移工具，请卸载后再安装新的工具包。

```
wget https://mirrors.cloud.tencent.com/tlinux/3.1/extras/x86_64/os/Packages/migrate2tencentos-1.07-6.tl3.x86_64.rpm
```

3. 执行以下命令，安装迁移工具。

```
rpm -ivh migrate2tencentos-1.07-6.tl3.x86_64.rpm
```

4. 执行以下命令，开始迁移。

4.1 通过下面命令之一进行迁移

4.1.1 全量迁移

将 CentOS 发行版的用户态软件包替换为 TencentOS 发行版，为系统安装 TencentOS 基于5.4的内核。

```
/usr/local/bin/EasyMigration -d remote -k
```

4.1.2 minimal 软件组迁移

将系统的核心组件包迁移成 TencentOS 发行版，为系统安装 TencentOS 基于5.4的内核。
该模式下迁移的用户态软件包规模较小，系统上其他非核心组件的软件仍然保留为 CentOS 发行版。

```
/usr/local/bin/EasyMigration -d remote -k -g minimal
```

minimal 软件组默认列表参考页面底部 附录一 。
迁移需要一定时间，请耐心等待。脚本执行完成后，输出如下图所示信息，表示已完成迁移。

```
INFO: Migration Switch complete. TencentOS recommends rebooting this system.
```

5. 重启实例，详情请参见 重启实例 。
6. 检查迁移结果。

6.1 执行以下命令，检查 os-release。

```
cat /etc/os-release
```

返回如下图所示信息：

```
[root@VM-2-2-centos ~]# cat /etc/os-release
NAME="TencentOS Server"
VERSION="3.1 (Final)"
ID="tencentos"
ID_LIKE="rhel fedora centos"
VERSION_ID="3.1"
PLATFORM_ID="platform:el8"
PRETTY_NAME="TencentOS Server 3.1 (Final)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:tencentos:tencentos:3"
HOME_URL="https://cloud.tencent.com/product/ts"
```

6.2 执行以下命令，检查内核。

```
uname -r
```

返回如下图所示信息：

```
[root@VM-2-2-centos ~]# uname -r
5.4.119-19-0009.1
[root@VM-2-2-centos ~]#
```

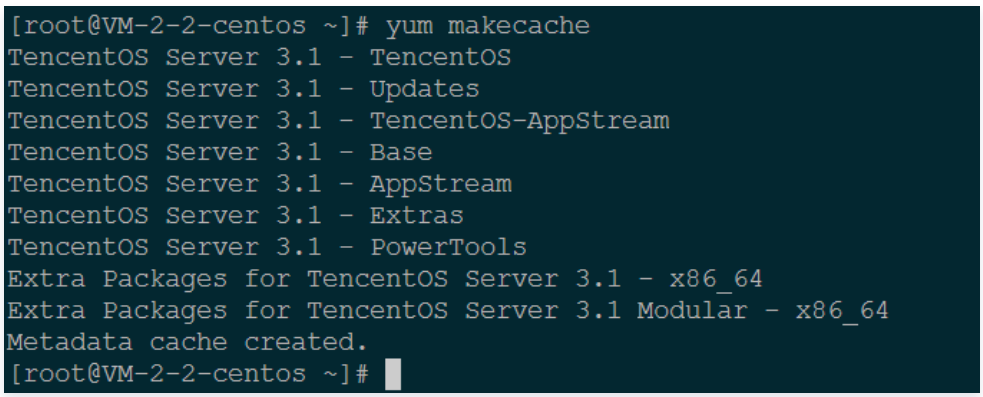
说明：

> 内核默认为 yum 最新版本，请以您的实际返回结果为准，本文以图示版本为例。

6.3 执行以下命令，检查 yum。

```
yum makecache
```

返回如下图所示信息：



若您在迁移过程中遇到问题，或对迁移有更多需求，请联系 在线客服 。

Minimal 软件组列表见下表格：

序号	名称
1	audit
2	basesystem
3	bash
4	btrfs-progs
5	coreutils
6	cronie
7	curl
8	dhclient
9	e2fsprogs
10	filesystem
11	firewalld
12	glibc
13	hostname
14	initscripts
15	iproute
16	iprutils
17	iptables
18	iputils
19	irqbalance

20	kbd
21	kexec-tools
22	less
23	man-db
24	ncurses
25	openssh-clients
26	openssh-server
27	parted
28	passwd
29	plymouth
30	policycoreutils
31	procps-ng
32	rootfiles
33	rpm
34	rsyslog
35	selinux-policy-targeted
36	setup
37	shadow-utils
38	sudo
39	systemd
40	tar
41	tuned
42	util-linux
43	vim-minimal
44	xfsprogs
45	yum
46	NetworkManager
47	NetworkManager-team
48	NetworkManager-tui
49	aic94xx-firmware
50	alsa-firmware
51	biosdevname
52	dracut-config-rescue
53	ivtv-firmware
54	iwl100-firmware

55	iwl1000-firmware
56	iwl105-firmware
57	iwl135-firmware
58	iwl2000-firmware
59	iwl2030-firmware
60	iwl3160-firmware
61	iwl3945-firmware
62	iwl4965-firmware
63	iwl5000-firmware
64	iwl5150-firmware
65	iwl6000-firmware
66	iwl6000g2a-firmware
67	iwl6000g2b-firmware
68	iwl6050-firmware
69	iwl7260-firmware
70	kernel-tools
71	libsysfs
72	linux-firmware
73	lshw
74	microcode_ctl
75	postfix
76	sg3_utils
77	sg3_utils-libs
78	dracut-config-generic
79	dracut-fips
80	dracut-fips-aesni
81	dracut-network
82	initial-setup
83	openssh-keycat
84	rdma-core
85	selinux-policy-mls
86	tboot
87	gdb
88	kexec-tools
89	latrace

90	libreport-cli
91	strace
92	systemtap-runtime
93	abrt-addon-ccpp
94	abrt-addon-python
95	abrt-cli
96	crash
97	crash-gcore-command
98	crash-ptdump-command
99	crash-trace-command
100	elfutils
101	kernel-tools
102	libreport-plugin-mailx
103	ltrace
104	memstomp
105	ps_mem
106	trace-cmd
107	valgrind
108	abrt-java-connector
109	gdb-gdbserver
110	glibc-utils
111	memtest86+
112	systemtap-client
113	systemtap-initscrip

TencentOS Server 3.1 安装 pytorch 及运行 AI 相关模型

最近更新时间：2024-03-25 14:24:41

环境准备

GPU 机型要求

购买 GPU 机型时，根据需求选择下图所示的 GPU 驱动版本、CUDA 版本、cuDNN 版本。

Python 3 版本要求

必须为 3.8 及以上版本，检查方法如下：

```
[root@VM-0-21-tencentos ~]# python3 -V
Python 3.8.16
[root@VM-0-21-tencentos ~]# pip3 -V
pip 19.3.1 from /usr/lib/python3.8/site-packages/pip (python 3.8)
[root@VM-0-21-tencentos ~]#
```

如果 Python 版本不满足要求，请按照如下步骤操作：

1. 安装 Python 3.8。

```
yum install -y python3.8
```

2. 配置 Python 3.8 为默认的 Python 3 版本。

```
cd /usr/bin/ && rm /usr/bin/python3 && ln -s python3.8 python3
```

3. 配置 pip 3.8 为默认的 pip 3 版本。

```
cd /usr/bin/ && rm /usr/bin/pip3 && ln -s pip3.8 pip3
```

安装 pytorch

您可以执行如下命令安装 pytorch。

```
pip3 install  torch==1.12
```

安装 pytorch 后，您也可以执行如下命令检查 pytorch 是否安装成功。

```
[root@VM-0-21-tencentos ~]# pip3 list | grep torch
torch                    1.12.0
[root@VM-0-21-tencentos ~]#
```

典型模型示例

Stable Diffusion

网站地址：https://huggingface.co/runwayml/stable-diffusion-v1-5

> **说明：**
> 该网站为国外网站，下载速度可能较慢，取决于您的网络性能。

1. 安装依赖的软件包。

```
pip3 install diffusers transformers
```

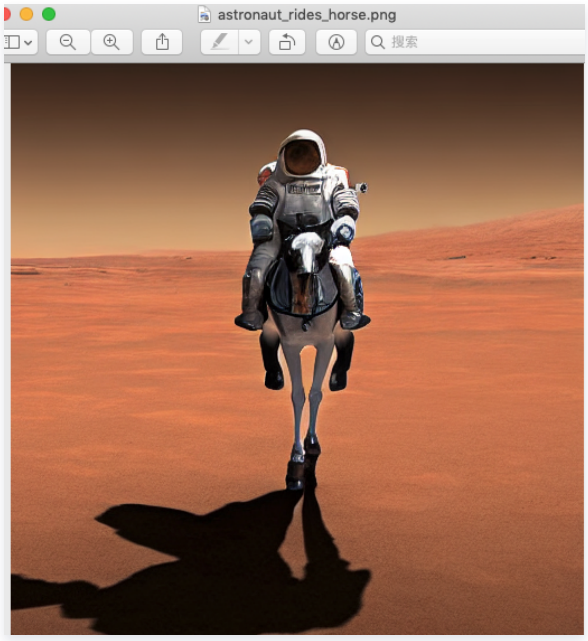
2. 将如下 Python 代码保存为 Python 脚本。假设脚本名称为：stable_diffusion.py。

```
from diffusers import StableDiffusionPipeline
import torch

model_id = "runwayml/stable-diffusion-v1-5"
pipe = StableDiffusionPipeline.from_pretrained(model_id, torch_dtype=torch.float16)
pipe = pipe.to("cuda")

prompt = "a photo of an astronaut riding a horse on mars"
image = pipe(prompt).images[0]

image.save("astronaut_rides_horse.png")
```

3. 运行上一步保存的 Python 脚本。

```
[root@VM-2-9-tencentos stable_diffusion]# python3 stable_diffusion.py
The cache for model files in Transformers v4.22.0 has been updated. Migrating your old cache. This is a one-time only operation. You can i
0it [00:00, ?it/s]
Downloading (…)ain/model_index.json: 100%|
Downloading (…)_checker/config.json: 100%|
Downloading (…)cheduler_config.json: 100%|
Downloading (…)rocessor_config.json: 100%|
Downloading (…)_encoder/config.json: 100%|
Downloading (…)cial_tokens_map.json: 100%|
Downloading (…)okenizer_config.json: 100%|
Downloading (…)7f0/unet/config.json: 100%|
Downloading (…)57f0/vae/config.json: 100%|
Downloading (…)tokenizer/merges.txt: 100%|
Downloading (…)tokenizer/vocab.json: 100%|
Downloading (…)ch_model.safetensors: 100%|
Downloading model.safetensors:  17%|
Downloading model.safetensors:  47%|
```

上图展示的是等待下载模型所需要的资源，最后会在当前目录生成一张图片：astronaut_rides_horse.png。
训练结果如下：



百川 13B 对话模型

网站地址：https://modelscope.cn/models/baichuan-inc/Baichuan-13B-Chat/summary

> **说明：**
> 该网站为国内模型网站，下载速度较快。

1. 安装依赖的软件包

```
pip3 install modelscope
pip3 install pip --upgrade      #运行过程中会失败，需要升级
pip3 install sentencepiece
```

2. 运行如下 Python 脚本。

```
import torch
from modelscope import snapshot_download, Model
model_dir = snapshot_download("baichuan-inc/Baichuan-13B-Chat", revision='v1.0.3')
model = Model.from_pretrained(model_dir, device_map="balanced", trust_remote_code=True, torch_dtype=torch.float16)
messages = []
messages.append({"role": "user", "content": "世界上第二高的山峰是哪一座？"})
response = model(messages)
print(response)
```

模型运行结果如下：

```
[root@VM-0-21-tencentos Baichuan-13B-Chat]# python3 bai.py
2023-07-12 15:20:53,287 - modelscope - INFO - PyTorch version 1.12.0 Found.
2023-07-12 15:20:53,288 - modelscope - INFO - Loading ast index from /root/.cache/modelscope/ast_indexer
2023-07-12 15:20:53,316 - modelscope - INFO - Loading done! Current index file version is 1.7.1, with md5 42c0395d32aad94a0fb1c9345805da71 and a total nu
2023-07-12 15:20:53,530 - modelscope - INFO - Use user-specified model revision: v1.0.3
2023-07-12 15:20:53,721 - modelscope - INFO - initialize model from /root/.cache/modelscope/hub/baichuan-inc/Baichuan-13B-Chat
Looking in indexes: http://mirrors.tencentyun.com/pypi/simple
Looking in links: https://modelscope.oss-cn-beijing.aliyuncs.com/releases/repo.html, https://modelscope.oss-cn-beijing.aliyuncs.com/releases/repo.html
Requirement already satisfied: cpm_kernels in /usr/local/lib/python3.8/site-packages (1.0.11)
Requirement already satisfied: transformers_stream_generator in /usr/local/lib/python3.8/site-packages (0.0.4)
Requirement already satisfied: transformers>=4.26.1 in /usr/local/lib/python3.8/site-packages (from transformers_stream_generator) (4.30.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (3.12.2)
Requirement already satisfied: huggingface-hub<1.0,>=0.14.1 in /usr/local/lib/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_gen
Requirement already satisfied: numpy>=1.17 in /usr/local/lib64/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (1.22.0
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (23.1
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib64/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (6.0)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib64/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (
Requirement already satisfied: requests in /usr/local/lib/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (2.31.0)
Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in /usr/local/lib64/python3.8/site-packages (from transformers>=4.26.1->transformers_str
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib64/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (4.65.0)
Requirement already satisfied: fsspec in /usr/local/lib/python3.8/site-packages (from huggingface-hub<1.0,>=0.14.1->transformers>=4.26.1->transformers_st
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.8/site-packages (from huggingface-hub<1.0,>=0.14.1->transformers>=4.2
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib64/python3.8/site-packages (from requests->transformers>=4.26.1->transformers_st
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.8/site-packages (from requests->transformers>=4.26.1->transformers_stream_generator
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.8/site-packages (from requests->transformers>=4.26.1->transformers_stream_gen
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/site-packages (from requests->transformers>=4.26.1->transformers_stream_gen
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to
2023-07-12 15:20:56,136 - modelscope - WARNING - ('MODELS', 'text-generation', 'Baichuan-13B-Chat') not found in ast index file
WARNING: ('MODELS', 'text-generation', 'Baichuan-13B-Chat') not found in ast index file
WARNING: The model weights are not tied. Please use the `tie_weights` method before using the `infer_auto_device` function.
Loading checkpoint shards: 100%|
{'response': '乔戈里峰。世界第二高峰——乔戈里峰西方登山者称其为k2峰，海拔高度是8611米，位于喀喇昆仑山脉的中巴边境上。', 'history': ''}
[root@VM-0-21-tencentos Baichuan-13B-Chat]# ls
```

您也可以替换脚本中的提问，例如：北京申奥成功是哪一年？返回结果如下：

```
[root@VM-0-21-tencentos Baichuan-13B-Chat]# python3 bai.py
2023-07-12 15:53:06,072 - modelscope - INFO - PyTorch version 1.12.0 Found.
2023-07-12 15:53:06,072 - modelscope - INFO - Loading ast index from /root/.cache/modelscope/ast_indexer
2023-07-12 15:53:06,102 - modelscope - INFO - Loading done! Current index file version is 1.7.1, with md5 42c0395d32aad94a0fb1c9345805da71 and a total number of 861 components indexed
2023-07-12 15:53:06,365 - modelscope - INFO - Use user-specified model revision: v1.0.3
2023-07-12 15:53:06,681 - modelscope - INFO - initialize model from /root/.cache/modelscope/hub/baichuan-inc/Baichuan-13B-Chat
Looking in indexes: http://mirrors.tencentyun.com/pypi/simple
Looking in links: https://modelscope.oss-cn-beijing.aliyuncs.com/releases/repo.html, https://modelscope.oss-cn-beijing.aliyuncs.com/releases/repo.html
Requirement already satisfied: cpm_kernels in /usr/local/lib/python3.8/site-packages (1.0.11)
Requirement already satisfied: transformers_stream_generator in /usr/local/lib/python3.8/site-packages (0.0.4)
Requirement already satisfied: transformers>=4.26.1 in /usr/local/lib/python3.8/site-packages (from transformers_stream_generator) (4.30.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (3.12.2)
Requirement already satisfied: huggingface-hub<1.0,>=0.14.1 in /usr/local/lib/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (0.16.4)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib64/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (1.22.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (23.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib64/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (6.0)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib64/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (2023.6.3)
Requirement already satisfied: requests in /usr/local/lib/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (2.31.0)
Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in /usr/local/lib64/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (0.13.3)
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib64/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (0.3.1)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.8/site-packages (from transformers>=4.26.1->transformers_stream_generator) (4.65.0)
Requirement already satisfied: fsspec in /usr/local/lib/python3.8/site-packages (from huggingface-hub<1.0,>=0.14.1->transformers>=4.26.1->transformers_stream_generator) (2023.6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.8/site-packages (from huggingface-hub<1.0,>=0.14.1->transformers>=4.26.1->transformers_stream_generator) (4.7.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib64/python3.8/site-packages (from requests->transformers>=4.26.1->transformers_stream_generator) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.8/site-packages (from requests->transformers>=4.26.1->transformers_stream_generator) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.8/site-packages (from requests->transformers>=4.26.1->transformers_stream_generator) (2.0.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/site-packages (from requests->transformers>=4.26.1->transformers_stream_generator) (2023.5.7)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
2023-07-12 15:53:09,117 - modelscope - WARNING - ('MODELS', 'text-generation', 'Baichuan-13B-Chat') not found in ast index file
WARNING: ('MODELS', 'text-generation', 'Baichuan-13B-Chat') not found in ast index file
WARNING: The model weights are not tied. Please use the `tie_weights` method before using the `infer_auto_device` function.
Loading checkpoint shards: 100%|
{'response': '2001年北京时间2001年7月13日晚上，在莫斯科的卢日尼基体育馆，当国际奥委会主席萨马兰奇宣布北京获得2008年奥运会主办权时，整个中国都沸腾了！\n从1992年的错失良机到2001年的众望所归，中国人等了整整9个春秋。', 'history': ''}
[root@VM-0-21-tencentos Baichuan-13B-Chat]#
```

openjourney

网站地址：prompthero/openjourney · Hugging Face

该例子使用源码的方式训练

1. 安装软件：

```
yum install git-lfs -y
```

2. 下载 openjourney 对应的代码。

```
# Make sure you have git-lfs installed (https://git-lfs.com)
git lfs install
git clone https://huggingface.co/prompthero/openjourney

# if you want to clone without large files – just their pointers
# prepend your git clone with the following env var:
GIT_LFS_SKIP_SMUDGE=1
```

下载过程可能较慢，其中有三个大文件，需要耐心等待。

```
[root@VM-2-9-tencentos openjourney]# ls -lh
total 4.5G
drwxr-xr-x 2 root root 4.0K Jul 12 15:57 feature_extractor
-rw-r--r-- 1 root root 2.0G Jul 12 16:16 mdjrny-v4.ckpt
-rw-r--r-- 1 root root 2.0G Jul 12 16:16 mdjrny-v4.safetensors
-rw-r--r-- 1 root root  541 Jul 12 15:57 model_index.json
-rw-r--r-- 1 root root 470M Jul 12 16:03 model.safetensors
-rw-r--r-- 1 root root 2.7K Jul 12 15:57 README.md
drwxr-xr-x 2 root root 4.0K Jul 12 16:11 safety_checker
drwxr-xr-x 2 root root 4.0K Jul 12 15:57 scheduler
drwxr-xr-x 2 root root 4.0K Jul 12 16:08 text_encoder
drwxr-xr-x 2 root root 4.0K Jul 12 15:57 tokenizer
drwxr-xr-x 2 root root 4.0K Jul 12 16:19 unet
drwxr-xr-x 2 root root 4.0K Jul 12 16:10 vae
[root@VM-2-9-tencentos openjourney]#
```

3. 运行如下脚本：

```python
from diffusers import StableDiffusionPipeline
import torch
model_id = "/root/openjourney_code/openjourney"    #修改为你本地的目录即可
pipe = StableDiffusionPipeline.from_pretrained(model_id, torch_dtype=torch.float16)
pipe = pipe.to("cuda")
prompt = "retro serie of different cars with different colors and shapes, mdjrny-v4 style"
image = pipe(prompt).images[0]
image.save("./retro_cars.png")
```

运行结果如下：

```
[root@VM-2-9-tencentos openjourney_code]# python3 open.py
Cannot initialize model with low cpu memory usage because `accelerate` was not found in the environment. Defaulting to `low_cpu
_mem_usage=False`. It is strongly recommended to install `accelerate` for faster and less memory-intense model loading. You can
 do so with:
```
pip install accelerate
```
.
`text_config_dict` is provided which will be used to initialize `CLIPTextConfig`. The value `text_config["id2label"]` will be o
verriden.
100%|██████████| 50/50 [00:03<00:00, 14.03it/s]
[root@VM-2-9-tencentos openjourney_code]#
[root@VM-2-9-tencentos openjourney_code]# ls
openjourney  open.py  retro_cars.png
[root@VM-2-9-tencentos openjourney_code]#
```

运行完毕会生成一张图片，如下图所示：

⚠ 注意：

如果出现类似如下报错，说明 GPU 显存不够，需要更高配置的 GPU 机型。

```
RuntimeError: CUDA out of memory. Tried to allocate 1.25 GiB (GPU 0; 14.76 GiB total capacity; 12.96 GiB already
allocated; 993.75 MiB free; 12.96 GiB reserved in total by
PyTorch) If reserved memory is >> allocated memory try setting max_split_size_mb to avoid fragmentation.  See
documentation for Memory Management
```

TencentOS Server 3.1 安装主要深度学习框架及示例

最近更新时间：2024-03-25 14:24:41

环境准备

GPU 机型要求

购买 GPU 机型时，根据需求选择下图所示的 GPU 驱动版本、CUDA 版本、cuDNN 版本。

Python 3 版本要求

必须为 3.8 及以上版本，检查方法如下：

```
[root@VM-0-21-tencentos ~]# python3 -V
Python 3.8.16
[root@VM-0-21-tencentos ~]# pip3 -V
pip 19.3.1 from /usr/lib/python3.8/site-packages/pip (python 3.8)
[root@VM-0-21-tencentos ~]#
```

如果 Python 版本不满足要求，请按照如下步骤操作：

1. 安装 Python 3.8。

```
yum install -y python38 python38-devel
```

2. 配置 Python 3.8 为默认的 Python 3 版本。

```
cd /usr/bin/ && rm /usr/bin/python3 && ln -s python3.8 python3
```

3. 配置 pip 3.8 为默认的 pip 3 版本。

```
cd /usr/bin/ && rm /usr/bin/pip3 && ln -s pip3.8 pip3
```

升级 pip 及 setuptools

```
pip3 install --upgrade pip
pip3 install --upgrade setuptools
```

配置 python3-config

```
ln -s /usr/bin/python3.8-config /usr/bin/python3-config
```

典型模型示例

Tensorflow 训练示例

1. 安装 TensorFlow。

   1.1 您可以执行以下命令安装 TensorFlow。

```
pip3 install tensorflow==2.6
```

   1.2 安装 TensorFlow 后，您也可以执行以下命令检查 TensorFlow 是否安装成功。

```
[root@VM-16-12-tencentos ~]# pip3 list |  grep tensorflow
tensorflow            2.6.0
tensorflow-estimator    2.13.0
```

2. 安装依赖。

```
pip3 install keras==2.6
pip3 uninstall protobuf      #卸载默认的 4.23.4高版本
pip3 install protobuf==3.20.0
```

3. 将如下 Python 代码保存为 Python 脚本，例如：demo.py。

```
import tensorflow as tf


mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10)
])

predictions = model(x_train[:1]).numpy()
loss_fn = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
loss_fn(y_train[:1], predictions).numpy()

model.compile(optimizer='adam',
              loss=loss_fn,
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)

model.evaluate(x_test,  y_test, verbose=2)
```

4. 运行保存好的 Python 脚本。

```
[root@VM-0-10-tencentos tf]# python3 demo.py
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11493376/11490434 [==============================] - 53s 5us/step
11501568/11490434 [==============================] - 53s 5us/step
2023-07-19 20:01:40.952882: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neu
ral Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-07-19 20:01:43.950482: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/devic
e:GPU:0 with 20812 MB memory:  -> device: 0, name: NVIDIA A10, pci bus id: 0000:0b:01.0, compute capability: 8.6
2023-07-19 20:01:43.952451: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/devic
e:GPU:1 with 20812 MB memory:  -> device: 1, name: NVIDIA A10, pci bus id: 0000:0b:02.0, compute capability: 8.6
2023-07-19 20:01:43.954224: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/devic
e:GPU:2 with 20812 MB memory:  -> device: 2, name: NVIDIA A10, pci bus id: 0000:0b:03.0, compute capability: 8.6
2023-07-19 20:01:43.955987: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/devic
e:GPU:3 with 20812 MB memory:  -> device: 3, name: NVIDIA A10, pci bus id: 0000:0b:04.0, compute capability: 8.6
2023-07-19 20:01:43.957774: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/devic
e:GPU:4 with 22338 MB memory:  -> device: 4, name: NVIDIA A10, pci bus id: 0000:41:01.0, compute capability: 8.6
2023-07-19 20:01:43.959711: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/devic
e:GPU:5 with 22338 MB memory:  -> device: 5, name: NVIDIA A10, pci bus id: 0000:41:02.0, compute capability: 8.6
2023-07-19 20:01:43.961455: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/devic
e:GPU:6 with 20812 MB memory:  -> device: 6, name: NVIDIA A10, pci bus id: 0000:41:03.0, compute capability: 8.6
2023-07-19 20:01:43.963299: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/devic
e:GPU:7 with 20812 MB memory:  -> device: 7, name: NVIDIA A10, pci bus id: 0000:41:04.0, compute capability: 8.6
2023-07-19 20:01:44.638762: I tensorflow/stream_executor/cuda/cuda_blas.cc:1760] TensorFloat-32 will be used for the matrix multiplicatio
n. This will only be logged once.
2023-07-19 20:01:44.772794: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimization Passes are enab
led (registered 2)
Epoch 1/5
1875/1875 [==============================] - 2s 827us/step - loss: 0.2927 - accuracy: 0.9144
Epoch 2/5
1875/1875 [==============================] - 2s 820us/step - loss: 0.1415 - accuracy: 0.9577
Epoch 3/5
1875/1875 [==============================] - 2s 816us/step - loss: 0.1054 - accuracy: 0.9681
Epoch 4/5
1875/1875 [==============================] - 2s 816us/step - loss: 0.0865 - accuracy: 0.9733
Epoch 5/5
1875/1875 [==============================] - 2s 815us/step - loss: 0.0742 - accuracy: 0.9765
313/313 - 0s - loss: 0.0709 - accuracy: 0.9783
```

Pytorch 训练示例

1. 安装 Pytorch。

   1.1 您可以执行以下命令安装 Pytorch。

   ```
   pip3 install torch==1.12.1+cu113 torchvision==0.13.1+cu113 torchaudio==0.12.1 --extra-index-url
   https://download.pytorch.org/whl/cu113
   ```

   1.2 安装 Pytorch 后，您也可以执行以下命令检查 Pytorch 是否安装成功。

   ```
   [root@VM-16-12-tencentos ~]# pip3 list | grep torch
   torch            1.12.1+cu113
   torchaudio       0.12.1+cu113
   torchvision      0.13.1+cu113
   ```

2. 将如下 Python 代码保存为 Python 脚本，例如：demo.py。

```
import torch
from torch import nn
from torch.utils.data import DataLoader
from torchvision import datasets
from torchvision.transforms import ToTensor


# Download training data from open datasets.
training_data = datasets.FashionMNIST(
    root="data",
    train=True,
    download=True,
    transform=ToTensor(),
)

# Download test data from open datasets.
test_data = datasets.FashionMNIST(
    root="data",
```

```
    train=False,
    download=True,
    transform=ToTensor(),
)

batch_size = 64

# Create data loaders.
train_dataloader = DataLoader(training_data, batch_size=batch_size)
test_dataloader = DataLoader(test_data, batch_size=batch_size)

# Get cpu, gpu or mps device for training.
device = (
    "cuda"
    if torch.cuda.is_available()
    else "mps"
    if torch.backends.mps.is_available()
    else "cpu"
)
print(f"Using {device} device")

# Define model
class NeuralNetwork(nn.Module):
    def __init__(self):
        super().__init__()
        self.flatten = nn.Flatten()
        self.linear_relu_stack = nn.Sequential(
            nn.Linear(28*28, 512),
            nn.ReLU(),
            nn.Linear(512, 512),
            nn.ReLU(),
            nn.Linear(512, 10)
        )

    def forward(self, x):
        x = self.flatten(x)
        logits = self.linear_relu_stack(x)
        return logits

model = NeuralNetwork().to(device)

loss_fn = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(model.parameters(), lr=1e-3)


def train(dataloader, model, loss_fn, optimizer):
    size = len(dataloader.dataset)
    model.train()
    for batch, (X, y) in enumerate(dataloader):
        X, y = X.to(device), y.to(device)

        # Compute prediction error
        pred = model(X)
        loss = loss_fn(pred, y)

        # Backpropagation
        loss.backward()
        optimizer.step()
        optimizer.zero_grad()

        if batch % 100 == 0:
            loss, current = loss.item(), (batch + 1) * len(X)
            print(f"loss: {loss:>7f}  [{current:>5d}/{size:>5d}]")
```

```
def test(dataloader, model, loss_fn):
    size = len(dataloader.dataset)
    num_batches = len(dataloader)
    model.eval()
    test_loss, correct = 0, 0
    with torch.no_grad():
        for X, y in dataloader:
            X, y = X.to(device), y.to(device)
            pred = model(X)
            test_loss += loss_fn(pred, y).item()
            correct += (pred.argmax(1) == y).type(torch.float).sum().item()
    test_loss /= num_batches
    correct /= size
    print(f"Test Error: \n Accuracy: {(100*correct):>0.1f}%, Avg loss: {test_loss:>8f} \n")


epochs = 5
for t in range(epochs):
    print(f"Epoch {t+1}\n-------------------------------")
    train(train_dataloader, model, loss_fn, optimizer)
    test(test_dataloader, model, loss_fn)
print("Done!")
```

3. 运行保存好的 Python 脚本。

```
[root@VM-0-10-tencentos py]# python3 demo.py
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-images-idx3-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-images-idx3-ubyte.gz to data/FashionMNIST/raw/train-images-i
dx3-ubyte.gz
100.0%
Extracting data/FashionMNIST/raw/train-images-idx3-ubyte.gz to data/FashionMNIST/raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-labels-idx1-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-labels-idx1-ubyte.gz to data/FashionMNIST/raw/train-labels-i
dx1-ubyte.gz
100.0%
Extracting data/FashionMNIST/raw/train-labels-idx1-ubyte.gz to data/FashionMNIST/raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-images-idx3-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-images-idx3-ubyte.gz to data/FashionMNIST/raw/t10k-images-idx
3-ubyte.gz
100.0%
Extracting data/FashionMNIST/raw/t10k-images-idx3-ubyte.gz to data/FashionMNIST/raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-labels-idx1-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-labels-idx1-ubyte.gz to data/FashionMNIST/raw/t10k-labels-idx
1-ubyte.gz
100.0%
Extracting data/FashionMNIST/raw/t10k-labels-idx1-ubyte.gz to data/FashionMNIST/raw

Using cuda device
Epoch 1
-------------------------------
loss: 2.305151  [   64/60000]
loss: 2.294250  [ 6464/60000]
loss: 2.282794  [12864/60000]
loss: 2.274063  [19264/60000]
loss: 2.257649  [25664/60000]
loss: 2.238950  [32064/60000]
loss: 2.237539  [38464/60000]
loss: 2.214913  [44864/60000]
loss: 2.212914  [51264/60000]
loss: 2.172773  [57664/60000]
Test Error:
 Accuracy: 42.0%, Avg loss: 2.171635
```

DeepSpeed 训练示例

1. 安装 DeepSpeed。

   1.1 您可以执行以下命令安装 DeepSpeed。

   ```
   pip3 install deepspeed
   ```

   1.2 安装 DeepSpeed 后，您也可以执行以下命令检查 DeepSpeed 是否安装成功。

```
[root@VM-16-12-tencentos ~]# pip3 list | grep deepspeed
deepspeed         0.9.5
```

2. 安装依赖。

```
# NCCL
sudo yum-config-manager --add-repo https://developer.download.nvidia.com/compute/cuda/repos/rhel8/x86_64/cuda-rhel8.repo
sudo yum install libnccl libnccl-devel libnccl-static -y

pip3 install datasets evaluate accelerate sentencepiece transformers==4.28.1 pydantic==1.10.7

pip3 uninstall numpy    #卸载numpy的高版本
pip3 install numpy==1.22.0
```

3. 下载示例代码，并做相应修改。

```
wget https://taco-1251783334.cos.ap-shanghai.myqcloud.com/demo/LLM/llama.tar.gz
tar xzf llama.tar.gz && cd llama
```

3.1 通过以下命令获取本机 eth0 ip，写到 hostfile 中。

```
ifconfig eth0
```

3.2 根据本机 GPU 数量修改 start.sh 文件中的 GPUS_PER_NODE 变量，默认为1。

```
GPUS_PER_NODE=1
```

4. 执行 start.sh 开始训练。

```
[root@VM-16-12-tencentos llama]# bash start.sh
torchrun --nproc_per_node 1 --nnodes 1 --node_rank 0 --master_addr 172.26.16.12 --master_port 6000 run_clm.py --deepspeed ds_zero3_no_of
r_name ./model/ --train_file ./data/pt_sample_data.txt --per_device_train_batch_size 1 --per_device_eval_batch_size 1 --do_train --seed
pe cosine --learning_rate 2e-4 --warmup_ratio 0.05 --weight_decay 0.01 --logging_strategy steps --logging_steps 10 --save_strategy steps
dient_accumulation_steps 1 --preprocessing_num_workers 8 --block_size 1024 --output_dir output --overwrite_output_dir --ddp_timeout 3000
oat16 --gradient_checkpointing --ddp_find_unused_parameters False --report_to none
[2023-07-19 17:38:05,166] [INFO] [real_accelerator.py:110:get_accelerator] Setting ds_accelerator to cuda (auto detect)
[2023-07-19 17:38:08,539] [WARNING] [comm.py:152:init_deepspeed_backend] NCCL backend in DeepSpeed not yet implemented
[2023-07-19 17:38:08,539] [INFO] [comm.py:594:init_distributed] cdb=None
[2023-07-19 17:38:08,539] [INFO] [comm.py:625:init_distributed] Initializing TorchBackend in DeepSpeed with backend nccl
07/19/2023 17:38:08 - WARNING - __main__ - Process rank: 0, device: cuda:0, n_gpu: 1distributed training: True, 16-bits training: True
07/19/2023 17:38:08 - INFO - __main__ - Training/evaluation parameters TrainingArguments(
_n_gpu=1,
adafactor=False,
adam_beta1=0.9,
adam_beta2=0.999,
adam_epsilon=1e-08,
auto_find_batch_size=False,
bf16=False,
bf16_full_eval=False,
data_seed=None,
dataloader_drop_last=False,
dataloader_num_workers=0,
dataloader_pin_memory=True,
ddp_bucket_cap_mb=None,
ddp_find_unused_parameters=False,
ddp_timeout=30000,
```

```
: 277.89 | backward_allreduce: 10.20 | step: 61.14
 95%|▉| 19/20 [00:11<00:00,  2.12it/s][2023-07-19 17:42:02,512] [INFO] [logging.py:96:log_dist] [Rank 0] rank=0 time (ms) | opti
mizer_step: 33.99
[2023-07-19 17:42:02,512] [INFO] [logging.py:96:log_dist] [Rank 0] step=20, skipped=2, lr=[5.418275829936537e-06], mom=[[0.9, 0.999]]
[2023-07-19 17:42:02,513] [INFO] [timer.py:215:stop] epoch=0/micro_step=20/global_step=20, RunningAvgSamplesPerSec=2.1502114540330632, Cu
rrSamplesPerSec=2.151743894343386, MemAllocated=8.85GB, MaxMemAllocated=11.47GB
[2023-07-19 17:42:02,513] [INFO] [logging.py:96:log_dist] [Rank 0] rank=0 time (ms) | forward_microstep: 113.69 | backward_microstep: 288
.21 | backward_inner_microstep: 277.95 | backward_allreduce_microstep: 10.18 | step_microstep: 61.74
[2023-07-19 17:42:02,513] [INFO] [logging.py:96:log_dist] [Rank 0] rank=0 time (ms) | forward: 113.68 | backward: 288.22 | backward_inner
: 277.97 | backward_allreduce: 10.18 | step: 61.74
{'loss': 7.6801, 'learning_rate': 5.418275829936537e-06, 'epoch': 0.0}
100%|▉| 20/20 [00:12<00:00,  2.12it/s][INFO|trainer.py:2039] 2023-07-19 17:42:02,515 >>

Training completed. Do not forget to share your model on huggingface.co/models =)


{'train_runtime': 12.1298, 'train_samples_per_second': 1.649, 'train_steps_per_second': 1.649, 'train_loss': 8.6630859375, 'epoch': 0.0}
100%|▉| 20/20 [00:12<00:00,  1.65it/s]
***** train metrics *****
  epoch                    =        0.0
  train_loss               =     8.6631
  train_runtime            = 0:00:12.12
  train_samples            =       7131
  train_samples_per_second =      1.649
  train_steps_per_second   =      1.649
[INFO|modelcard.py:451] 2023-07-19 17:42:02,517 >> Dropping the following result as it does not have all the necessary fields:
{'task': {'name': 'Causal Language Modeling', 'type': 'text-generation'}}
[root@VM-16-12-tencentos llama]#
```

Megatron-LM 训练示例

1. 安装 Megatron-LM。

   您可以执行以下命令安装 Megatron-LM。

```
yum install git -y
git clone https://github.com/NVIDIA/Megatron-LM.git && cd Megatron-LM && git checkout -b v3.0.2 v3.0.2
```

2. 安装依赖。

```
# NCCL
sudo yum-config-manager --add-repo https://developer.download.nvidia.com/compute/cuda/repos/rhel8/x86_64/cuda-
rhel8.repo
sudo yum install libnccl libnccl-devel libnccl-static



# apex
git clone https://github.com/NVIDIA/apex.git && cd apex
pip3 install -v --disable-pip-version-check --no-cache-dir --no-build-isolation --config-settings "--build-option=--cpp_ext" --
config-settings "--build-option=--cuda_ext" ./
```

3. （可选）上一步如果编译错误，需要注释掉如下代码。

```
diff --git a/setup.py b/setup.py
index b156cfa..150e548 100644
--- a/setup.py
+++ b/setup.py
@@ -174,7 +174,7 @@ if "--distributed_lamb" in sys.argv:
 if "--cuda_ext" in sys.argv:
     sys.argv.remove("--cuda_ext")
     raise_if_cuda_home_none("--cuda_ext")
-    check_cuda_torch_binary_vs_bare_metal(CUDA_HOME)
+#    check_cuda_torch_binary_vs_bare_metal(CUDA_HOME)
```

4. 安装 pybind11。

```
# pybind11
pip3 install pybind11
```

5. 下载示例代码，并做相应修改。

```
wget https://taco-1251783334.cos.ap-shanghai.myqcloud.com/demo/LLM/gpt.tar.gz
tar xzf gpt.tar.gz && cd gpt

cp hostfile download_data.sh start.sh path/to/Megatron-LM && cd path/to/Megatron-LM      #path是Megatorn-LM所在的位置
```

5.1 通过以下命令获取本机 eth0 ip，写到 hostfile 中。

```
ifconfig eth0
```

5.2 根据本机 GPU 数量修改 start.sh 文件中的 GPUS_PER_NODE 变量，默认为1。

```
GPUS_PER_NODE=1
```

6. 执行 download_data.sh 下载数据集。

```
[root@VM-16-12-tencentos Megatron-LM]# bash download_dataset.sh
gpt2-merges.txt                 100%[=============================================================>] 445.62K   547KB/s    in 0.8s
gpt2-vocab.json                 100%[=============================================================>]   1018K  1.20MB/s    in 0.8s
my-gpt2_text_document.bin       100%[=============================================================>] 446.97M  9.81MB/s    in 45s
my-gpt2_text_document.idx       100%[=============================================================>]   1.51M   180KB/s    in 9.2s
[root@VM-16-12-tencentos Megatron-LM]#
```

7. 执行 start.sh 开始训练。

```
[root@VM-16-12-tencentos Megatron-LM]# bash start.sh
torchrun --nproc_per_node 1 --nnodes 1 --node_rank 0 --master_addr 172.26.16.12 --master_port 6000 pretrain_gpt.py --tensor-model-paralle
l-size 1 --pipeline-model-parallel-size 1 --sequence-parallel --num-layers 24 --hidden-size 1024 --num-attention-heads 16 --micro-batch-s
ize 1 --global-batch-size 10 --seq-length 2048 --max-position-embeddings 2048 --train-iters 500000 --lr-decay-iters 320000 --save /root/M
egatron-LM/checkpoint --data-path /root/Megatron-LM/data/my-gpt2_text_document --vocab-file /root/Megatron-LM/data/gpt2-vocab.json --merg
e-file /root/Megatron-LM/data/gpt2-merges.txt --data-impl mmap --split 949,50,1 --distributed-backend nccl --lr 0.00015 --lr-decay-style
cosine --min-lr 1.0e-5 --weight-decay 1e-2 --clip-grad 1.0 --lr-warmup-fraction .01 --log-interval 1 --save-interval 10000 --eval-interva
l 10000 --exit-interval 10000 --eval-iters 1000 --bf16 2>&1 | tee gpt_300MB_tp1_pp{1}_dp1_bs{GLOBAL_BATCH_SIZE}.log
using world size: 1, data-parallel-size: 1, tensor-model-parallel size: 1, pipeline-model-parallel size: 1
accumulate and all-reduce gradients in fp32 for bfloat16 data type.
using torch.bfloat16 for parameters ...
------------------------ arguments ------------------------
  accumulate_allreduce_grads_in_fp32 ..... True
  adam_beta1 ..................................... 0.9
  adam_beta2 ..................................... 0.999
  adam_eps ....................................... 1e-08
  adlr_autoresume ................................ False
  adlr_autoresume_interval ....................... 1000
  apply_query_key_layer_scaling .................. True
  apply_residual_connection_post_layernorm ....... False
  async_tensor_model_parallel_allreduce .......... True
  attention_dropout .............................. 0.1
  attention_softmax_in_fp32 ...................... False
  bert_binary_head ............................... True
  bert_load ...................................... None
  bf16 ........................................... True
  bias_dropout_fusion ............................ True
  bias_gelu_fusion ............................... True
  biencoder_projection_dim ....................... 0
  biencoder_shared_query_context_model ........... False
  block_data_path ................................ None
  classes_fraction ............................... 1.0
```

```
tch-generator: 16.56
 iteration       95/     100 | consumed samples:          950 | elapsed time per iteration (ms): 6586.8 | learning rate: 4.453E-06 | glob
al batch size:    10 | lm loss: 9.037986E+00 | loss scale: 1.0 | grad norm: 2.467 | number of skipped iterations:   0 | number of nan ite
rations:   0 |
time (ms) | forward-compute: 2307.76 | backward-compute: 4236.42 | backward-params-all-reduce: 3.59 | backward-embedding-all-reduce: 0.03
 | optimizer-copy-to-main-grad: 0.87 | optimizer-clip-main-grad: 7.95 | optimizer-copy-main-to-model-params: 4.96 | optimizer: 28.36 | ba
tch-generator: 16.94
 iteration       96/     100 | consumed samples:          960 | elapsed time per iteration (ms): 6591.2 | learning rate: 4.500E-06 | glob
al batch size:    10 | lm loss: 9.025996E+00 | loss scale: 1.0 | grad norm: 2.624 | number of skipped iterations:   0 | number of nan ite
rations:   0 |
time (ms) | forward-compute: 2312.53 | backward-compute: 4236.26 | backward-params-all-reduce: 3.53 | backward-embedding-all-reduce: 0.02
 | optimizer-copy-to-main-grad: 0.70 | optimizer-clip-main-grad: 7.61 | optimizer-copy-main-to-model-params: 4.88 | optimizer: 27.74 | ba
tch-generator: 19.96
 iteration       97/     100 | consumed samples:          970 | elapsed time per iteration (ms): 6589.2 | learning rate: 4.547E-06 | glob
al batch size:    10 | lm loss: 8.975855E+00 | loss scale: 1.0 | grad norm: 3.045 | number of skipped iterations:   0 | number of nan ite
rations:   0 |
time (ms) | forward-compute: 2308.23 | backward-compute: 4238.96 | backward-params-all-reduce: 3.54 | backward-embedding-all-reduce: 0.02
 | optimizer-copy-to-main-grad: 0.74 | optimizer-clip-main-grad: 7.63 | optimizer-copy-main-to-model-params: 4.91 | optimizer: 27.84 | ba
tch-generator: 17.53
 iteration       98/     100 | consumed samples:          980 | elapsed time per iteration (ms): 6581.4 | learning rate: 4.594E-06 | glob
al batch size:    10 | lm loss: 9.115416E+00 | loss scale: 1.0 | grad norm: 2.372 | number of skipped iterations:   0 | number of nan ite
rations:   0 |
time (ms) | forward-compute: 2307.72 | backward-compute: 4231.55 | backward-params-all-reduce: 3.55 | backward-embedding-all-reduce: 0.03
 | optimizer-copy-to-main-grad: 0.77 | optimizer-clip-main-grad: 7.73 | optimizer-copy-main-to-model-params: 4.89 | optimizer: 27.92 | ba
tch-generator: 17.61
 iteration       99/     100 | consumed samples:          990 | elapsed time per iteration (ms): 6593.7 | learning rate: 4.641E-06 | glob
al batch size:    10 | lm loss: 9.044435E+00 | loss scale: 1.0 | grad norm: 2.570 | number of skipped iterations:   0 | number of nan ite
rations:   0 |
time (ms) | forward-compute: 2312.31 | backward-compute: 4238.56 | backward-params-all-reduce: 3.54 | backward-embedding-all-reduce: 0.03
 | optimizer-copy-to-main-grad: 0.81 | optimizer-clip-main-grad: 7.73 | optimizer-copy-main-to-model-params: 4.91 | optimizer: 28.02 | ba
tch-generator: 18.13
 iteration      100/     100 | consumed samples:         1000 | elapsed time per iteration (ms): 6585.2 | learning rate: 4.687E-06 | glob
al batch size:    10 | lm loss: 9.020534E+00 | loss scale: 1.0 | grad norm: 2.889 | number of skipped iterations:   0 | number of nan ite
rations:   0 |
time (ms) | forward-compute: 2310.96 | backward-compute: 4230.31 | backward-params-all-reduce: 3.57 | backward-embedding-all-reduce: 0.03
 | optimizer-copy-to-main-grad: 0.85 | optimizer-clip-main-grad: 8.04 | optimizer-copy-main-to-model-params: 5.00 | optimizer: 28.54 | ba
tch-generator: 20.49
[after training is done] datetime: 2023-07-19 20:06:36
```