Atlas800-9000 训练服务器环境配置

一、安装 miniconda

（1）官网安装网站步骤：https://docs.conda.io/projects/miniconda/en/latest/index.html#quick-command-line-install

（2）版本选择网站：https://docs.conda.io/projects/miniconda/en/latest/miniconda-hashes.html

conda base 环境的 python 版本可以随便选择

Miniconda hash information

Name	Size	Time modified	SHA256 hash
Miniconda3-py311_23.11.0-2-MacOSX-x86_64.sh	105.5 MiB	2023-12-20 19:10:58	2b7f9e46308c28c26dd83abad3e72121ef63916eaf17b63723b5a1f728dc3032
Miniconda3-py311_23.11.0-2-MacOSX-arm64.sh	102.3 MiB	2023-12-20 19:10:58	5694c382e6056d62ed874f22692224c4f53bca22e8135b6f069111e081be07aa
Miniconda3-py311_23.11.0-2-MacOSX-arm64.pkg	101.7 MiB	2023-12-20 19:10:58	912c0b58e800f26e08d515526a8d3455755e83963b40e78597176540ea2401ca
Miniconda3-py311_23.11.0-2-MacOSX-x86_64.pkg	104.9 MiB	2023-12-20 19:10:58	74ab9e8c3e9b3c2fc7c44d710ed9bad19085d951d819c1284a46eeb0bdfe2578
Miniconda3-py311_23.11.0-2-Windows-x86_64.exe	80.5 MiB	2023-12-20 19:10:58	c9b32faa9262828702334b16bcb5b53556e630d54e5127f5c36c7ba7ed43179a
Miniconda3-py311_23.11.0-2-Linux-x86_64.sh	135.1 MiB	2023-12-20 19:10:58	c9ae82568e9665b1105117b4b1e499607d2a920f0aea6f94410e417a0eff1b9c
Miniconda3-py311_23.11.0-2-Linux-s390x.sh	131.4 MiB	2023-12-20 19:10:58	53a9e9eb97cd6e318f4f184add069436e1a46124cf864bf2d7bd67043e50e471
Miniconda3-py311_23.11.0-2-Linux-aarch64.sh	113.1 MiB	2023-12-20 19:10:58	decd447fb99dbd0fc5004481ec9bf8c04f9ba28b35a9292afe49ecefe400237f
Miniconda3-py310_23.11.0-2-MacOSX-arm64.pkg	97.7 MiB	2023-12-20 19:10:58	98f1644ddb5f9b0e0413f43489c079b08e7e85ca5bc08b50515430dfdbdd35db

（3）命令行安装

```
创建安装路径
mkdir -p ~/miniconda3

# 下载挑选的安装包，注意对应系统架构的miniconda版本
wget https://repo.anaconda.com/miniconda/Miniconda3-py311_23.11.0-2-Linux-aarch64.sh -O ~/miniconda3/miniconda.sh
# 安装
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3

# 删除安装包
rm -rf ~/miniconda3/miniconda.sh

conda初始化
~/miniconda3/bin/conda init bash

# 新建终端激活 conda 的 base  环境
```

二、conda 管理 python 环境

(1) 修改 conda 源

```
# 创建 conda 镜像管理文件
vim ~/.condarc

# 添加国内镜像源
channels:
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/menpo/
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/bioconda/
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/msys2/
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
  - https://mirrors.ustc.edu.cn/anaconda/pkgs/main/
  - https://mirrors.ustc.edu.cn/anaconda/pkgs/free/
  - https://mirrors.ustc.edu.cn/anaconda/cloud/conda-forge/
  - https://mirrors.ustc.edu.cn/anaconda/cloud/msys2/
  - https://mirrors.ustc.edu.cn/anaconda/cloud/bioconda/
  - https://mirrors.ustc.edu.cn/anaconda/cloud/menpo/
  - defaults
show_channel_urls: true

# 新建终端，激活 conda 镜像源
```

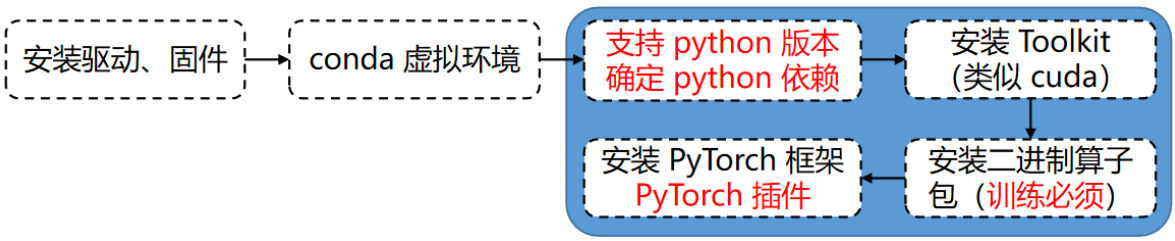
三、Atlas 800 服务器裸机安装 PyTorch1.11.0 训练环境

（1）安装前**必看**两个章节：

https://www.hiascend.com/document/detail/zh/CANNCommunityEdition/700alpha003/softwareinstall/instg/instg_0001.html

https://www.hiascend.com/document/detail/zh/ModelZoo/pytorchframework/ptes/ptes_00001.html

（2）安装流程



1. 驱动及固件版本确定（服务器系统**已安装**，无需安装）

https://support.huawei.com/enterprise/zh/doc/EDOC1100332515/51429589

```
# 查看驱动加载是否成功
npu-smi info
```

```
+-------------------------------------------------------------------------------------------+
| npu-smi 23.0.rc3                 Version: 23.0.rc3                                        |
+---------------------------+---------------+---------------------------------------------------+
| NPU   Name                | Health        | Power(W)    Temp(C)           Hugepages-Usage(page)|
| Chip                      | Bus-Id        | AICore(%)   Memory-Usage(MB)  HBM-Usage(MB)        |
+===========================+===============+===================================================+
| 0     910B                | OK            | 72.4        41                0    / 0            |
| 0                         | 0000:C1:00.0  | 0           1215 / 15137      1    / 32768        |
+===========================+===============+===================================================+
| 1     910B                | OK            | 67.6        34                0    / 0            |
| 0                         | 0000:81:00.0  | 0           2464 / 15137      0    / 32768        |
+===========================+===============+===================================================+
```

芯片型号

```
# 查看芯片固件版本号
sudo /usr/local/Ascend/driver/tools/upgrade-tool --device_index -1 --component -1
--version
```

```
Get component version(7.0.0.5.242) succeed for deviceId(0), componentType(0).
        {"device_id":0, "component":nve, "version":7.0.0.5.242}
Get component version(7.0.0.5.242) succeed for deviceId(0), componentType(3).
        {"device_id":0, "component":uefi, "version":7.0.0.5.242}
Get component version(7.0.0.5.242) succeed for deviceId(0), componentType(8).
        {"device_id":0, "component":imu, "version":7.0.0.5.242}
Get component version(7.0.0.5.242) succeed for deviceId(0), componentType(9).
        {"device_id":0, "component":imp, "version":7.0.0.5.242}
```

2. 确定 python 版本（以 **python3.8.2** 为例）

https://www.hiascend.com/document/detail/zh/ModelZoo/pytorchframework/ptes/ptes_00007.html

- Ubuntu 20.04

表2 依赖信息

类别	名称	版本要求
相关软件或工具	Python	CANN支持Python3.7.*x*（3.7.0~3.7.11）、Python3.8.*x*（3.8.0~3.8.11）、Python3.9.*x*（3.9.0~3.9.7）。 PyTorch框架支持Python3.7.x（3.7.5~3.7.11）、Python3.8.x（3.8.0~3.8.11）、Python3.9.x（3.9.0~3.9.2）。

3. conda 创建环境及安装相关依赖（非管理员**慎用 apt 等工具**直接升级或降级已有的安装）

https://www.hiascend.com/document/detail/zh/ModelZoo/pytorchframework/ptes/ptes_00007.html

相关软件或工具：

一个个查看依赖版本即可，一般不用安装，**服务器管理员会安装**

OS依赖：

查看是否安即可，一般不用安装，**服务器管理员会安装(如 cmake 等已满足要求，后续无需再安装！！！)**

https://www.hiascend.com/document/detail/zh/ModelZoo/pytorchframework/ptes/ptes_00010.html

python依赖（建议使用 conda 管理）

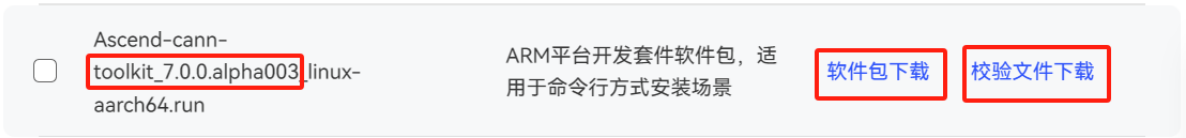
```
# 创建 python 环境（默认 conda 源可以没有 python3.8.2 版本，需要添加国内源）
conda create -n pytorch1.11.0 python=3.8.2
```

```
# 激活 conda 环境
conda activate pytorch1.11.0

# 安装 python 依赖（将相关依赖放到 requirements.txt 文件）
pip install -r requirements.txt

>>> requirements.txt 文件内容:
numpy>=1.14.3
decorator>=4.4.0
sympy>=1.4
cffi>=1.12.3
protobuf>=3.11.3
attrs
cython
pyyaml
pathlib2
scipy
requests
psutil
absl-py
```

```
Successfully built psutil
Installing collected packages: mpmath, urllib3, sympy, six, pyyaml, pycparser, psutil, protobuf, numpy, idna, decorator, cython, charset-norm
alizer, certifi, attrs, absl-py, scipy, requests, pathlib2, cffi
Successfully installed absl-py-2.0.0 attrs-23.2.0 certifi-2023.11.17 cffi-1.16.0 charset-normalizer-3.3.2 cython-3.0.8 decorator-5.1.1 idna-3
.6 mpmath-1.3.0 numpy-1.24.4 pathlib2-2.3.7.post1 protobuf-4.25.2 psutil-5.9.7 pycparser-2.21 pyyaml-6.0.1 requests-2.31.0 scipy-1.10.1 six-1
.16.0 sympy-1.12 urllib3-2.1.0
```

4. 安装 Toolkit 开发套件（社区版）

https://www.hiascend.com/developer/download/community/result?module=cann&cann=7.0.0.alpha003

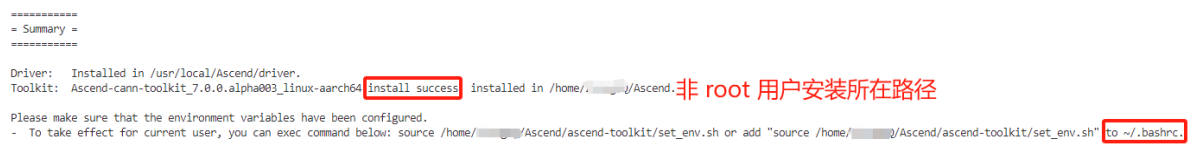


注意需要下载**校验文件**，按照步骤进行校验即可

```
# 切换目录到 PGPVerify.jar 的目录
java -jar PGPVerify.jar -f ~/download/Ascend-cann-toolkit_7.0.0.alpha003_linux-aarch64.run.asc

# 安装包校验成果结果如下
```

安装参考链接：https://www.hiascend.com/document/detail/zh/CANNCommunityEdition/700alpha003/softwareinstall/instg/instg_0038.html

```
# 添加可执行权限
chmod +x Ascend-cann-toolkit_7.0.0.alpha003_linux-aarch64.run

# 校验软件包安装文件的一致性和完整性
./Ascend-cann-toolkit_7.0.0.alpha003_linux-aarch64.run --check

# 安装
./Ascend-cann-toolkit_7.0.0.alpha003_linux-aarch64.run --install

# 根据安装成功提示信息，配置环境到 ~/.bashrc 文件，并执行以下命令生效
source ~/.bashrc
```

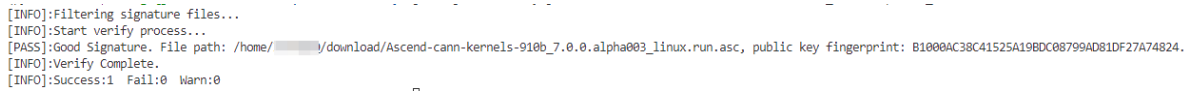


5. 安装二进制算子包（**训练必须**）

安装步骤和安装 Toolkit 开发套件一样，只是下载的安装包不一样（注意版本的一致性）

☐	Ascend-cann-kernels-910b_7.0.0.alpha003_linux.run	CANN算子二进制安装包，适用于命令行方式安装场景	软件包下载	校验文件下载

```
# 切换目录到 PGPVerify.jar 的目录
java -jar PGPVerify.jar -f ~/download/Ascend-cann-kernels-910b_7.0.0.alpha003_linux.run.asc

# 安装包校验成果结果如下
```



```
# 添加可执行权限
chmod +x Ascend-cann-kernels-910b_7.0.0.alpha003_linux.run

# 校验软件包安装文件的一致性和完整性
./Ascend-cann-kernels-910b_7.0.0.alpha003_linux.run --check

# 安装
./Ascend-cann-kernels-910b_7.0.0.alpha003_linux.run --install

# 检查路径，是否安装成功（安装成功有对应路径）
ls ~/Ascend/ascend-toolkit/7.0.0.alpha003/opp/built-
in/op_impl/ai_core/tbe/kernel
```

```
Do you accept the EULA to install CANN?[Y/N]Y
[kernels_910b] [20240113-11:13:51] [INFO] install start
[kernels_910b] [20240113-11:13:51] [INFO] The installation path is /home/[redacted]/Ascend/ascend-toolkit.
[kernels_910b] [20240113-11:13:51] [INFO] install package Ascend910B-opp_kernel-7.1.t8.0.b205.run start
[kernels_910b] [20240113-11:14:43] [INFO] Ascend910B-opp_kernel-7.1.t8.0.b205.run --full --quiet --nox11 install success
[kernels_910b] [20240113-11:14:43] [INFO] Ascend-cann-kernels-910b_7.0.0.alpha003_linux install success. The installation path is /home/[redacted]/Ascend/ascend-toolkit.
```

6. 安装 PyTorch 训练框架（**！！！注意软件的配套使用**）

参考链接：https://www.hiascend.com/document/detail/zh/CANNCommunityEdition/700alpha003/softwareinstall/instg/instg_0046.html

版本配套关系

表1 PyTorch版本配套关系　务必对应

CANN版本	PyTorch版本	PyTorch Ascend Adapter插件版本	PyTorch Ascend Adapter插件代码分支名称
7.0.0.alpha003	PyTorch1.11.0	5.0.RC3	v1.11.0-5.0.rc3
	PyTorch2.0.1		v2.0.1-5.0.rc3
	PyTorch2.1.0		v2.1.0-5.0.rc3

版本配套关系

表1 PyTorch版本配套关系　以安装 PyTorch1.11.0、python3.8.2 为例

PyTorch Ascend Adapter插件安装包版本	Python版本
1.11.0.post4　重点	Python3.7.x（3.7.5及以上）、Python3.8.x、Python3.9.x、Python3.10.x
2.0.1	Python3.8.x、Python3.9.x、Python3.10.x
2.1.0.rc1	Python3.8.x、Python3.9.x、Python3.10.x

a.安装 PyTorch1.11.0

```
# 获取对应版本的 .whl 包
wget https://download.pytorch.org/whl/torch-1.11.0-cp38-cp38-
manylinux2014_aarch64.whl

# 在对应 conda 环境下安装
pip install torch-1.11.0-cp38-cp38-manylinux2014_aarch64.whl
```

```
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Processing ./torch-1.11.0-cp38-cp38-manylinux2014_aarch64.whl
Collecting typing-extensions (from torch==1.11.0)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/b7/f4/6a90020cd2d93349b442bfcb657d0dc91eee65491600b2cb1d388bc98e6b/typing_extensions-4.9.0-py3-none-any.whl (32 kB)
Installing collected packages: typing-extensions, torch
Successfully installed torch-1.11.0 typing-extensions-4.9.0
```

b.安装 torch_npu 插件（对应插件安装包为 1.11.0.post4）

版本下载链接：https://www.hiascend.com/developer/download/community/result?module=pt+cann&pt=5.0.0.alpha003&cann=7.0.0.alpha003


```
# 克隆匹配的版本（无对应 python 版本，则需要源码编译）
wget https://gitee.com/ascend/pytorch/releases/download/v5.0.rc3-
pytorch1.11.0/torch_npu-1.11.0.post4-cp38-cp38-linux_aarch64.whl

# 安装
pip3 install torch_npu-1.11.0.post4-cp38-cp38-linux_aarch64.whl

# 安装对应版本的 torchvision
pip3 install torchvision==0.12.0

# 验证安装是否成功
python3 -c "import torch;import torch_npu; a = torch.randn(3, 4).npu(); print(a +
a);"
```

```
tensor([[-1.0093,  1.2539, -2.8704, -0.4322],
        [ 1.7491, -2.3016, -1.7202, -1.8529],
        [-1.5210,  0.5712, -3.1471, -1.1727]], device='npu:0')
```

c.安装 APEX 模块

参考链接：https://www.hiascend.com/document/detail/zh/CANNCommunityEdition/700alpha003/softwareinstall/instg/instg_0050.html

源码分支确定（master 分支查看）：https://gitee.com/ascend/apex/tree/master/

五、Apex-patch配套软件

AscendPyTorch版本	支持PyTorch版本	Pytorch Gitee分支名称	Apex Gitee分支名称
5.0.rc2	1.8.1.post2, 1.11.0, 2.0.1.rc1	v1.8.1-5.0.rc2, v1.11.0-5.0.rc2	5.0.rc2
5.0.rc3	1.11.0, 2.0.1.rc1	v1.11.1-5.0.rc3, v2.0.1-5.0.rc3	5.0.rc3

应该是 v1.11.0-5.0.rc3

```
# 降级 setuptools 版本
pip3 install setuptools==65.7.0

# 获取 APEX 源码（PyTorch1.11.0-5.0.rc3）
git clone -b 5.0.rc3 https://gitee.com/ascend/apex.git

# 进入克隆下来的分支查看 README.zh.md 进行安装
cd apex/
bash scripts/build.sh --python=3.8
cd apex/dist/
pip3 uninstall apex
pip3 install --upgrade apex-0.1_ascend-cp38-cp38-linux_aarch64.whl
```

```
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Processing ./apex-0.1_ascend-cp38-cp38-linux_aarch64.whl
DEPRECATION: apex 0.1-ascend has a non-standard version number. pip 24.0 will enforce this behaviour char
uggest that they release a version with a conforming version number. Discussion can be found at https://g
Installing collected packages: apex
Successfully installed apex-0.1-ascend
```

d.环境安装验证

参考链接：https://www.hiascend.com/document/detail/zh/CANNCommunityEdition/700alpha003/ptmoddevg/ptmigr/AImpug_0002.html

```
# 执行程序验证是否可用
python test.py
```

```
Gradient overflow.  Skipping step, loss scaler 0 reducing loss scale to 32768.0

start training epoch: 2
start training epoch: 3
start training epoch: 4
start training epoch: 5
start training epoch: 6
start training epoch: 7
start training epoch: 8
start training epoch: 9
start training epoch: 10
```

conda 管理环境 Q&A

（1）pip 安装的依赖在当前 python 环境不生效

```
(pytorch1.11.0)  :~$ pip3 -version
bash: /home/ZhangRQ/.local/bin/pip3: No such file or directory
```

原因是当前用户在没有安装 conda 前本地安装了 python 环境，在 ~/.local 路径下，可以将相关的 python 目录和 ~/.pip 删除，然后新建终端初始化 pip 环境。（切忌不能参考网络上彻底卸载 python 的方法，尤其是涉及到 sudo 权限的卸载，**/usr/local下的python环境不能卸载**）