



天数智芯
Iluvatar CoreX

天数智芯

软件栈安装快速入门

版本：V3.1 及以上

日期：2024.11.8

适用产品：天垓 100 | 天垓 150 | 智铠 50 | 智铠 100

1 声明

1.1 版权声明

版权所有。未经天数智芯书面许可，不得以任何形式或方式将本文档的任何部分复制，传播，转录或翻译成任何语言。

1.2 免责声明

天数智芯可以随时对本文档或本文档中描述的产品进行改进和/或更改。本文档包括与天数智芯产品有关的信息，作为说明典型应用的一种方式，因此，不一定提供足以进行生产设计的完整信息。对于本文档中内容的准确性或完整性，天数智芯不做任何陈述或保证。

1.3 联系方式

地址：上海市闵行区陈行公路 2168 号 3 幢

电话：021-68886607

网址：www.iluvatar.com

Contents

1 声明	2
1.1 版权声明	2
1.2 免责声明	2
1.3 联系方式	2
2 软件栈安装快速入门	4
2.1 修订记录	4
2.2 概览	4
2.3 重要前置条件	5
2.4 安装软件栈	5
2.4.1 宿主机安装方式	5
2.4.2 Docker 安装方式	6
2.4.2.1 安装 Docker	6
2.4.2.2 生成 Docker 镜像	7
2.4.3 Conda 安装方式	8
2.5 常见问题	8
2.5.1 问题 1: 无法安装天数智算软件栈驱动	8
2.5.2 问题 2: 无法安装天数智算软件栈驱动	9
2.5.3 问题 3: 无法调用天数智算软件栈相关命令	10
2.5.4 问题 4: 哪些 Python 包必须安装天数智芯适配版本? 为什么?	10
2.5.5 问题 5: 机器无法访问网络, 无法 pip 在线安装	11
3 商标声明	12

2 软件栈安装快速入门

2.1 修订记录

- COREX-SDKQS10-IN01-04: 2024/11/8
 - 安装软件栈操作步骤，更新必要文件的相关描述
 - 安装 Docker 对使用 `--device` 参数的注意事项进行补充说明
- COREX-SDKQS10-IN01-03: 2024/8/29
 - 安装 Docker 更新 Docker 源
 - 更新问题 3: 无法调用天数智算软件栈相关命令，提供使环境变量永久生效的方法
- COREX-SDKQS10-IN01-02: 2024/8/15
 - 安装 Docker 提供 CentOS 和 Kylin 平台安装 Docker 的操作指南参考链接
 - 重要前置条件 提供离线安装天数智算软件栈的前置条件
 - 常见问题 新增问题 5: 机器无法访问网络，无法 pip 在线安装 及解决方法
- COREX-SDKQS10-IN01-01: 2024/6/30
 - 生成 Docker 镜像 提示用户可以通过设置 `--device` 参数传递指定的设备来启动容器
 - 常见问题 新增问题 4: 哪些 Python 包必须安装天数智芯适配版本？为什么？
- COREX-SDKQS10-IN01-00: 2024/1/24
文档首次对外发布

2.2 概览

本指南旨在为您提供在 x86 (CentOS 和 Ubuntu 为主) 和 ARM (Kylin 为主) 环境上快速安装天数智算软件栈的基本操作指导。

您可以选择以下方式安装软件栈：

- 宿主机安装方式：您需要安装驱动、软件栈和深度学习框架。
- Docker 安装方式：通过该镜像启动的 Docker 容器内已包含软件栈的深度学习框架，无需安装，但您需要安装驱动。
- Conda 安装方式：您需要安装驱动、软件栈和深度学习框架，多个 Conda 环境仅支持使用一套软件栈。

Note

以上安装方式均适用于 CentOS、Ubuntu 和 Kylin 环境，且不同环境的操作步骤基本一致，故下文将进行统一描述。

本指南适用于全新且未安装过天数智算软件栈的环境。更多详细内容，请参考《软件栈安装指南》。

2.3 重要前置条件

- 确认已安装与 Linux kernel 版本匹配的 GCC。对于操作系统自带的 GCC，推荐版本如下：

Linux kernel 版本	GCC 版本	需要设置环境变量
3.x	GCC	无
4.0 ~ 4.5	GCC 5	export KCPPFLAGS='-fno-pie -Wno-pointer-sign -mfentry'
4.6 ~ 5.18	GCC 7/9	无
5.19 及以上	GCC 12	无

Note

Kylin 推荐 GCC 7。

- 确认已安装 make。

如果您需要离线安装天数智算软件栈，您还需要做以下准备工作：

- 准备一台联网的机器 (和离线机器有同样的 OS 和内核版本) 并下载相关依赖包。

- 使用 **apt-get** 方式下载，相关的 deb 包会下载到 deb 文件夹内：

```
$ mkdir -p deb
$ apt update
$ apt install -y --download-only make gcc linux-headers-$(uname -r) -o
  ↳ Dir::Cache::archives=./deb
```

- 使用 **yum** 方式下载，相关的 rpm 包会下载到 rpm 文件夹内：

```
$ yum update
$ yum install -y --downloadonly --downloadadd=./rpm kernel-devel make gcc
```

- 将相关文件复制到离线机器上并运行以下命令即可离线安装相关依赖包：

```
$ dpkg -i *.deb # apt-get 下载方式
或
$ yum localinstall -y --nogpgcheck *.rpm # yum 下载方式
```

其他前置条件，请参考《软件栈安装指南》。

2.4 安装软件栈

2.4.1 宿主机安装方式

- 安装天数智算软件栈必要头文件，详情请联系您的应用工程师。

2. 安装驱动和软件栈：

```
$ sudo bash corex-installer-linux64-{v.r.m}-{arch}_10.2.run
```

{v.r.m} 表示软件栈版本, **{arch}** 表示适用的架构

3. 设置 PATH 和 LD_LIBRARY_PATH 环境变量, 以软件栈默认安装路径 /usr/local/corex-**{v.r.m}**/ 为例, 您需要:

- 为 PATH 环境变量加上 /usr/local/corex-**{v.r.m}**/bin
- 为 LD_LIBRARY_PATH 环境变量加上 /usr/local/corex-**{v.r.m}**/lib

4. 安装深度学习框架:

```
$ pip3 install <one_whl_file>
```

2.4.2 Docker 安装方式

2.4.2.1 安装 Docker

Note

- 本小节以在 x86_64 架构的 Ubuntu 18.04 上安装 Docker 为例进行说明。实际操作步骤会根据设备的架构和操作系统有所不同。
- 请确保您的设备可以访问外网。
- CentOS 平台, 请参考 [CentOS 平台安装 Docker 操作指南](#)。
- Kylin 平台, 请参考 [Kylin 平台安装 Docker 操作指南](#)。

参考以下步骤安装 Docker:

1. 查看当前设备的版本号:

```
$ lsb_release -c
```

2. 更新软件源列表:

```
$ sudo apt-get update
```

3. 安装软件依赖包:

```
$ sudo apt-get install apt-transport-https ca-certificates curl software-properties-common
```

4. 在系统中添加阿里云 Docker 密钥:

```
$ curl -fsSL https://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | apt-key add -
```

5. 安装 Docker 存储库, 选择 stable 长期稳定版:

```
$ sudo add-apt-repository "deb [arch={arch}] https://mirrors.aliyun.com/docker-ce/linux/  
↪ ubuntu $(lsb_release -cs) stable"  
$ sudo apt install docker-ce docker-ce-cli containerd.io  
# {arch} 表示设备的架构, 如 amd64  
# 本命令中 $(lsb_release -cs) 扫描并返回 Ubuntu 的版本号, 可用步骤 1 查询到的版本号代替
```

6. 更新存储库：

```
$ sudo apt-get update
```

7. 安装最新版本 Docker：

```
$ sudo apt-get install docker-ce
```

8. 查看安装的 Docker 版本：

```
$ docker -v
```

9. 启动 Docker 服务：

```
$ sudo systemctl start docker
```

10. (可选) 设置开机自动启动 Docker：

```
$ sudo systemctl enable docker
```

11. 查看 Docker 开启状态，显示绿点表示服务正常启动：

```
$ sudo systemctl status docker
```

2.4.2.2 生成 Docker 镜像

1. 在宿主机上安装天数智算软件栈必要头文件，详情请联系您的应用工程师。

2. 运行安装包安装驱动并生成 Docker 镜像：

```
$ sudo bash corex-docker-installer-{v.r.m}-10.2-centos7.8.2003-{py_version}-{arch}.run
# {v.r.m} 表示软件栈版本，{py_version} 表示适用的 PyTorch 版本，{arch} 表示适用的架构
# 如宿主机没有安装驱动，在运行安装包的过程中请您务必勾选 Install driver
```

3. 启动 Docker 容器：

```
$ sudo docker run --shm-size="32g" -it -v /usr/src:/usr/src \
-v /lib/modules:/lib/modules -v /dev:/dev \
--privileged --cap-add=ALL --pid=host corex:{v.r.m}
```

Note

可以通过 **--device** 参数传递指定的设备来启动容器：

1. 需要用带冒号的格式进行映射，不能直接使用 **--device=/dev/iluvatar2**，正确的写法是 **--device=/dev/iluvatar2:/dev/iluvatar0**，冒号前面是宿主机的设备号，冒号后面是映射到容器内部的设备号。
2. 冒号后面是映射到容器内部的设备号，必须是 0-n 按照从小到大连续的顺序，例如 **device=/dev/iluvatar2:/dev/iluvatar0 --device=/dev/iluvatar5:/dev/iluvatar1**。对宿主机的卡号没有限制要求。

需要注意的是：使用 **--device** 参数时，请勿使用 **--privileged** 参数，否则会映射所有设备。

2.4.3 Conda 安装方式

1. 在宿主机上安装天数智算软件栈必要头文件，详情请联系您的应用工程师。
2. 在宿主机上安装驱动和软件栈：

```
$ sudo bash corex-installer-linux64-{v.r.m}-{arch}_10.2.run
```

{v.r.m} 表示软件栈版本，**{arch}** 表示适用的架构

3. 设置 PATH 和 LD_LIBRARY_PATH 环境变量，以软件栈默认安装路径 /usr/local/corex-**{v.r.m}**/ 为例，您需要：

- 为 PATH 环境变量加上 /usr/local/corex-**{v.r.m}**/bin
- 为 LD_LIBRARY_PATH 环境变量加上 /usr/local/corex-**{v.r.m}**/lib

4. 以 Python v3.8 为例，安装 Miniconda3：

```
$ bash Miniconda3-py38_23.9.0-0-Linux-{arch}.sh
```

{arch} 表示适用的架构

Miniconda3 获取地址：<https://repo.anaconda.com/miniconda/>

建议天垓 100 加速卡安装 Python v3.8 的 Miniconda3，智铠系列和天垓 150 加速卡安装 Python v3.10 的 Miniconda3

```
$ source ~/.bashrc
```

5. 以 Python v3.8 为例，创建 Python 环境：

```
$ conda create --name {name_for_conda} python=3.8
```

```
$ conda activate {name_for_conda}
```

{name_for_conda} 可根据软件栈版本命名，如 `corex_bi150_330`

6. 安装深度学习框架：

```
$ pip3 install <one_whl_file>
```

Note

- 不支持在 Conda 环境下安装 CUDA Toolkit 和天数智算软件栈。
- 多个 Conda 环境仅支持使用一套软件栈。
- 如下命令不支持在安装了天数智算软件栈的环境中运行：

```
$ conda install cudatoolkit
```

```
$ conda install -c conda-forge cudnn
```

2.5 常见问题

2.5.1 问题 1：无法安装天数智算软件栈驱动

问题描述

安装天数智算软件栈出现报错 Driver: Failed, error code: 54, 查询天数智芯加速卡内存地址分配情况, 发现显卡内存 32G 地址分配 Region 显示 disabled。报错示例如下:

```
$ lspci -vvv | less
$ /1e3e
af:00.0 Processing accelerators: Device 1e3e:0001
...
Region 0: Memory at 3af800000000 (64-bit, prefetchable) [disabled] [size=32G]
Region 2: Memory at e0e00000 (32-bit, non=prefetchable) [disabled] [size=256K]
```

问题分析

天数智芯加速卡对地址空间有要求, 请确保 BIOS 启用了在大于 4G 地址空间的解码。

解决方法

1. 进入主机 BIOS 设置, 确保 **Above 4G Decoding** 选项是开启状态。
2. 插卡后确认显卡内存地址分配正确, 执行 **lspci** 命令查看显卡内存地址分配情况。运行天数智芯加速卡要求显卡内存 32G 地址正确地分配在 Region 上。

例如, 您得到以下输出, 则说明内存地址分配正确。

```
$ lspci -vvv | less
$ /1e3e
af:00.0 Processing accelerators: Device 1e3e:0001
...
Region 0: Memory at 3af800000000 (64-bit, prefetchable) [size=32G]
Region 2: Memory at e0e00000 (32-bit, non=prefetchable) [size=256K]
```

2.5.2 问题 2: 无法安装天数智算软件栈驱动

问题描述

安装天数智算软件栈出现报错 Corex Driver installation failed。

解决方法

1. 确认您的 Linux kernel 版本属于《附录: 支持的 Linux kernel 版本》中天数智芯已验证的支持版本。
2. 确认已安装 Linux kernel 版本相应的头文件。

- a. 您可以通过以下命令进行检查:

```
$ uname -r
3.10.0-1160.42.2.el7.x86_64
```

- b. 检查该 kernel 版本对应的头文件是否已安装:
 - CentOS/Kylin:

```
$ ls -l /usr/src/kernels/$(uname -r)
```

- Ubuntu:

```
$ ls -l /usr/src/linux-headers-$(uname -r)
```

- c. 若对应的 Linux kernel 头文件未安装，请查阅以下参考文档：

- CentOS/Kylin: <https://www.tecmint.com/install-kernel-headers-in-centos-7>
- Ubuntu: <https://www.tecmint.com/install-kernel-headers-in-ubuntu-and-debian>

3. 确认已安装与 Linux kernel 版本匹配的 GCC，详见[重要前置条件](#)。

2.5.3 问题 3：无法调用天数智算软件栈相关命令

问题描述

调用天数智算软件栈相关命令，如 **ixsmi**，提示 Command 'ixsmi' not found。

问题分析

当前用户未设置环境变量。

解决方法

在宿主机上安装天数智算软件栈后，您需要设置 PATH 和 LD_LIBRARY_PATH 环境变量。以软件栈默认安装路径 /usr/local/corex-{v.r.m}/ 为例，您需要：

- 为 PATH 环境变量加上 /usr/local/corex-{v.r.m}/bin
- 为 LD_LIBRARY_PATH 环境变量加上 /usr/local/corex-{v.r.m}/lib

参考如下方法设置环境变量永久生效：

1. 将以下 export 写入 ~/.bashrc 文件：

```
$ export LD_LIBRARY_PATH=/usr/local/corex-{v.r.m}/lib
$ export PATH=/usr/local/corex-{v.r.m}/bin:$PATH
```

2. 保存后执行 **source ~/.bashrc** 命令即可生效。

Note

- 无论是 root 用户还是非 root 用户使用软件栈，都需要设置环境变量。
- 如果在使用软件栈中途切换用户 (如使用 root 用户安装软件栈后，切换 user 用户使用测试工具)，必须要为切换后的用户重新设置环境变量。

2.5.4 问题 4：哪些 Python 包必须安装天数智芯适配版本？为什么？

依赖 CUDA 的 Python 包，都需要安装天数智芯适配的版本，详细列表可参考《版本发布说明》中的“发布 whl 包说明”小节。不依赖 CUDA 的 Python 包，您都可以直接使用 **pip** 命令在线安装，比如 **pip install transformers**，因为 Transformers 依赖 PyTorch，并不直接依赖 CUDA。

针对天数智芯硬件，您必须使用天数智芯适配版框架，不应使用官方开源版本，以避免编译和运行报错。例如，使用 **pip install torch** 会采用在线方式下载并安装 PyTorch 官方开源版本，编译和运行可能会报错。天数智芯适配版的 .whl 包是针对天数智芯加速卡和天数智算软件栈优化的。使用这些定制包可以确保深度学习框架和其它相关库能够适配适配天数智芯的硬件指令，充分利用天数智芯加速卡的性能，提供更好的计算效率和优化的运行时体验。

由于天数智芯适配版 .whl 包目前没有在线发布，故不支持在线安装。您需要联系您的应用工程师获取所需 .whl 包并使用 **pip3** 命令进行安装。

2.5.5 问题 5：机器无法访问网络，无法 pip 在线安装

解决方法

以 vLLM 离线安装为例：

1. 在可以联网的机器上运行以下命令：

```
$ pip3 download vllm-0.3.3+corex.{v.r.m}-py3-none-any.whl -d "./whl" # 将 {v.r.m} 替换为对
  ↳ 应的天数智算软件栈版本号
```

此时，将会下载 vLLM 依赖的相关 whl 包到 whl 文件夹中。

2. 将 whl 文件夹拷贝至离线的机器上并运行以下命令即可离线安装 vLLM：

```
$ pip3 install --no-index --find-links="./whl" vllm-0.3.3+corex.{v.r.m}-py3-none-any.whl #
  ↳ 将 {v.r.m} 替换为对应的天数智算软件栈版本号
```

3 商标声明

- 天数智芯、天数智芯 logo、Iluvatar CoreX 等商标、标识、组合商标为上海天数智芯半导体有限公司之注册商标或商标，受法律保护。
- 除了天数智芯的注册商标外，本内容中使用的其他产品名称及标志仅用于识别目的，该等名称及标志可能是归属于其各自公司的商标。我们否认对该等名称及标志的所有权利。
- CentOS 标识为 Red Hat 公司的商标。
- Docker 为 Docker 公司在美国和其他国家的商标或注册商标。
- Linux 为 Linus Torvalds 在美国和其它国家的注册商标。
- NVIDIA 和 CUDA 为 NVIDIA 公司在美国和/或其它国家的商标和/或注册商标。
- PyTorch 为 Facebook 公司的商标。
- TensorFlow 为 Google 公司的商标。
- Ubuntu 为 Canonical 公司的注册商标。