

打包，打 Python 包，在龙架构打 Python 包

xen0n

安同校园行·北京站，2025-04-19

目录

1 引子

2 背景知识

3 「龙之国度」：龙架构下的 Python 打包挑战

引子：一位普通 Python 用户的遭遇

- 折腾 Python
- 购入一台龙架构电脑
- 想在龙架构上继续折腾 Python, `pip install numpy`

引子：一位普通 Python 用户的遭遇

- 折腾 Python
- 购入一台龙架构电脑
- 想在龙架构上继续折腾 Python, `pip install numpy`

```
File "/tmp/.../cmake_.../_build_backend/backend.py", line 156, ...  
    cmake_path = _bootstrap_build(temp_path, config_settings)  
File "/tmp/.../cmake_.../_build_backend/backend.py", line 78, ...  
    raise ValueError(msg)  
ValueError: Could not find a make program. Tried ('gmake', 'make',  
'smake')
```

- 这装个 numpy 咋还要编译呢？

不想编译怎么办？

- 诶，龙芯提供了 Loongnix PyPI 源，好耶

```
; ~/.config/pip/pip.conf
```

```
[global]
```

```
extra-index-url = https://pypi.loongnix.cn/loongson/pypi
```

不想编译怎么办？

- 诶，龙芯提供了 Loongnix PyPI 源，好耶

```
; ~/.config/pip/pip.conf
```

```
[global]
```

```
extra-index-url = https://pypi.loongnix.cn/loongson/pypi
```

- `pip install numpy`

不想编译怎么办？

- 诶，龙芯提供了 Loongnix PyPI 源，好耶

```
; ~/.config/pip/pip.conf
```

```
[global]
```

```
extra-index-url = https://pypi.loongnix.cn/loongson/pypi
```

- `pip install numpy`……等等，怎么还要编译

不想编译怎么办？

- 诶，龙芯提供了 Loongnix PyPI 源，好耶

```
; ~/.config/pip/pip.conf
```

```
[global]
```

```
extra-index-url = https://pypi.loongnix.cn/loongson/pypi
```

- pip install numpy……等等，怎么还要编译
- 原来只提供了 2.1.1 版本，还只能搭配 Python 3.10
- 那我就想办法找个 Python 3.10，就用 numpy 2.1.1 好了

剧透：这样会怎么样

Importing the numpy C-extensions failed. This error can happen for many reasons, often due to issues with your setup or how NumPy was installed.

We have compiled some common reasons and troubleshooting tips at:

<https://numpy.org/devdocs/user/troubleshooting-importerror.html>

Please note and check the following:

- * The Python version is: Python3.10 from "/usr/bin/python3"
- * The NumPy version is: "2.1.1"

and make sure that they are the versions you expect.

Please carefully study the documentation linked above for further help.

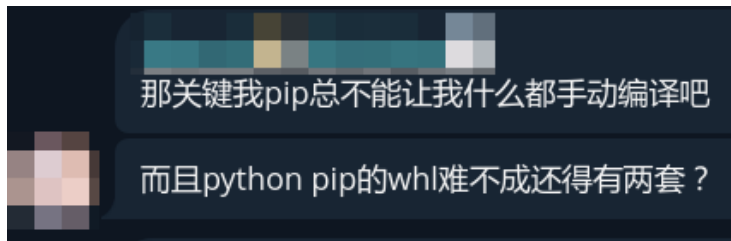
Original error was: No module named 'numpy._core._multiarray_umath'

The above exception was the direct cause of the following exception:

Traceback (most recent call last):

File "/home/███████/test.py", line 1, in <module>

用户的困惑



(很遗憾，龙架构目前是这样的)

用户的困惑

总归还是pip方便啊 11:30 AM

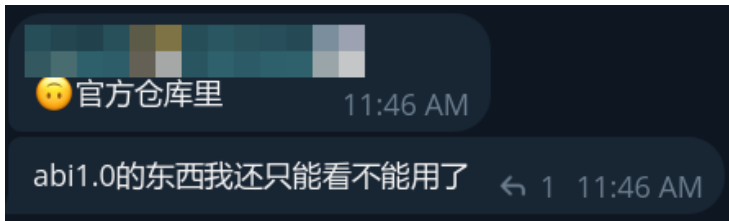
回到这个问题本身, 用, 入乡随俗, 和发行版对...

numpy我用发行版提供的本来就不合理 11:40 AM

pip经常有要装低版本之类的问题 11:40 AM

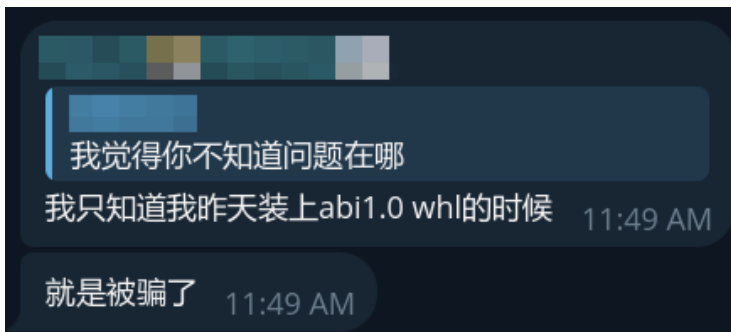
(确实)

用户的困惑



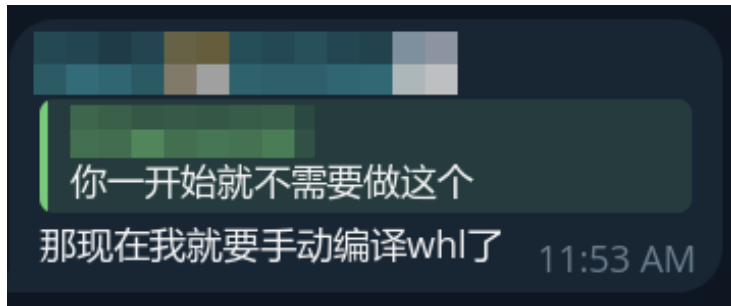
(很遗憾，是的)

用户的困惑



(我们对您的遭遇表示同情，因为我们曾经也被骗过)

用户的困惑



(很遗憾，确实如此)

怎么解决 tā 的问题呢？

- 让我们给 numpy 添加龙架构支持，上传到 PyPI 就好了！
- 我知道你很急，但你先别急.jpg

目录

1 引子

2 背景知识

3 「龙之国度」：龙架构下的 Python 打包挑战

Python Wheel 是什么

- 当前使用的 Python 软件包分发格式，前身叫 Python Egg
- 曾经由PEP 427定义，但目前定义文档维护在这里
- 是文件名与内容遵循特定格式的 Zip 压缩包，比方说
 - `requests-2.32.3-py3-none-any.whl`
这个包只要是Python 3环境，在什么Python ABI或者系统平台都能跑
 - `numpy-2.2.4-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl`
这个包有原生扩展，所以需要标明兼容的Python 版本、Python ABI和系统平台，但这一坨什么鬼

回顾：Windows 程序的兼容性问题

- 不同的 Windows 程序，要求的 MSVC 运行时库版本一般也不同
 - 与编译时所用的 MSVC 版本一致
 - 多版本的 MSVC 运行时库可以共存，用户自行或由软件安装器代为安装
- 「DLL 地狱」
 - 系统库版本繁多，排列组合爆炸，兼容性受损
 - WinSxS 机制：程序以其自带的 XML Manifest 表明其依赖的系统库具体版本

Linux 的兼容性问题

- libc: 以 glibc 为例
 - 使用 glibc X 版本编译的程序可在 glibc Y 版本上运行, 当且仅当 $Y \geq X$
 - 随发行版种类、版本更新, glibc 版本也会有不同
 - 但多个 glibc 版本无法共存
- 其他系统库: 由于一般而言 ABI 不同的库 soname 也不同, 问题较轻
- 我们将一套系统库的具体版本组合称为**基线**
 - 如果一个系统的所有相关库版本都不低于某条基线的要求, 那么声称兼容该基线的程序在该系统上运行**一般**不会有问题

- 如何让一个 Python wheel 在尽量多种类的系统上都能正常工作？
 - 合理范围内 glibc 越旧越好！
- manylinux 最早是 Python 社区定义的单条具体的基线
 - 基于 CentOS 5，兼容当时所有流行 Linux 发行版，故名
 - 后续由于外界软件版本逐渐更新、基础发行版停止维护等原因，又推出了人工定义、年份命名的数条基线
- 当前 manylinux 的定义基于 glibc 版本
 - 使用 musl 作为 libc 的系列基线叫 musllinux
 - 它们的权威定义位于 pypa/auditwheel 项目

目录

1 引子

2 背景知识

3 「龙之国度」：龙架构下的 Python 打包挑战

这有啥挑战的

- 现在是 PEP 517 时代了，直接 `pip install` 不就好了
 - 但这样一来还是让用户编译了
 - 不带原生扩展的包压根不需要特殊处理，而我们此刻关心的包全部都带
- 那我们就在项目 CI 流水线里自动化编译、上传呗
 - 基于哪个发行版的哪个版本？
 - PyPI 不接受 loongarch64 架构的 wheel 上传，怎么办？

龙架构目前的发行版支持状况

- Alpine Linux

- 良好，已经正式发布了（3.21），这就是 musllinux 基线了
- 那 manylinux 呢？

龙架构目前的发行版支持状况

- Alpine Linux
 - 良好，已经正式发布了 (3.21)，这就是 musllinux 基线了
 - 那 manylinux 呢？
- Arch Linux for Loong64、Gentoo
 - 良好，可惜是滚动的
- Debian
 - 良好，可惜是 sid，sid 也是滚动的
 - loong64 port 是否能进入 Debian 13 不是板上钉钉
- 安同 OS
 - 良好，不是滚动发行版，但并非为容器计算场景优化
- 那我们将 Arch/Debian/Gentoo 做个快照如何

龙架构目前的发行版支持状况

- Alpine Linux
 - 良好，已经正式发布了 (3.21)，这就是 musllinux 基线了
 - 那 manylinux 呢？
- Arch Linux for Loong64、Gentoo
 - 良好，可惜是滚动的
- Debian
 - 良好，可惜是 sid，sid 也是滚动的
 - loong64 port 是否能进入 Debian 13 不是板上钉钉
- 安同 OS
 - 良好，不是滚动发行版，但并非为容器计算场景优化
- 那我们将 Arch/Debian/Gentoo 做个快照如何
 - 想法是好的，但版本太高了
 - 五家国内商业发行版以《开源生态发展合作倡议》之名，cos 三体人，将各自的系统组件版本锁死了
 - 划重点：glibc 2.38 (2023.07.31)

是两难境地……吗？

- 直接基于《倡议》发行版做不就好了吗
 - 但是维护水平参差不齐
 - 各家对老版本系统库的 backport 补丁集也不完全相同
 - 近年的一些重要 ABI 演进未被包含。如果在 2025 年标准化一条 2023 基线，长远看是亏的
- 那我们基于社区发行版，自己锁版本
 - 旧版本有安全漏洞、功能 bug 之类，仍然需要 backport，工作量巨大
 - 在不受《倡议》约束的社区，有些组件的版本被锁死是没有道理，甚至不可接受的
- 破局：是否有办法「既要又要」？
 - 锁住 glibc（这个很有必要），但尽量保持其他组件——尤其工具链的新鲜
 - WIP: 本人正在进行此项实验

目前的工作进展

- auditwheel
 - 基线定义
 - manylinux done by @wxpppp
 - musllinux also done
- manylinux 基础镜像
 - 与预期基线完全平齐的基础系统容器镜像
 - WIP: github.com/loong64 by @wojiushixiaobai
- cibuildwheel
 - CI 环境自动化构建支持
 - TODO or done: 可能无 README 更新之外的工作量
- pypi/warehouse
 - 上传支持
 - TODO: 还涉及沟通工作

感谢!

提问时间