Python 中 ctypes 库调用 C 函数功能的实现

葛家韬

2024 年 12 月 12 日

摘要

在 python 中实现对 C 函数的调用有诸多优势，如提高程序运行速度，调用 C 库函数，提高程序扩展性等。python 的标准库 ctypes 可以实现调用 C 函数的功能，本文介绍 ctypes 库的实现，并讨论此技术的实际应用。

1 引言

python 是世界范围内使用最广泛的语言之一，其具有较高的灵活性与扩展性，拥有丰富的第三方库，可以适配如深度学习，科学计算等诸多应用场景。然而作为动态类型的解释型语言，其性能通常弱于编译型语言（如 C/C++）。

在 Python 中调用 C 函数可以显著提高程序的性能表现，同时增强其扩展性，并使其能够访问底层系统属性。如果调用的是经过优化的 C 库函数，Python 的性能提升会更加显著。

ctypes 库实现了 python 对 C 函数的调用，使开发者可以在开发 python 程序过程中利用 C 的高性能。ctypes 库是 python 的标准库，它通过访问本地动态链接库实现 python 对 C 函数的调用，同时通过对函数参数与返回值的管理使 python 能识别 C 的数据类型。

本文将分析 ctypes 的内部工作原理，涵盖其加载动态链接库的功能，对函数参数与返回值的管理。本文还将讨论其实际应用与局限性。

2 加载动态链接库

实现 python 对 C 函数的调用要经由以下过程（见图 1）：1.C 编译器将 C 函数转化为动态链接库（蓝色区域）。2.ctypes 库调用该动态链接库（红

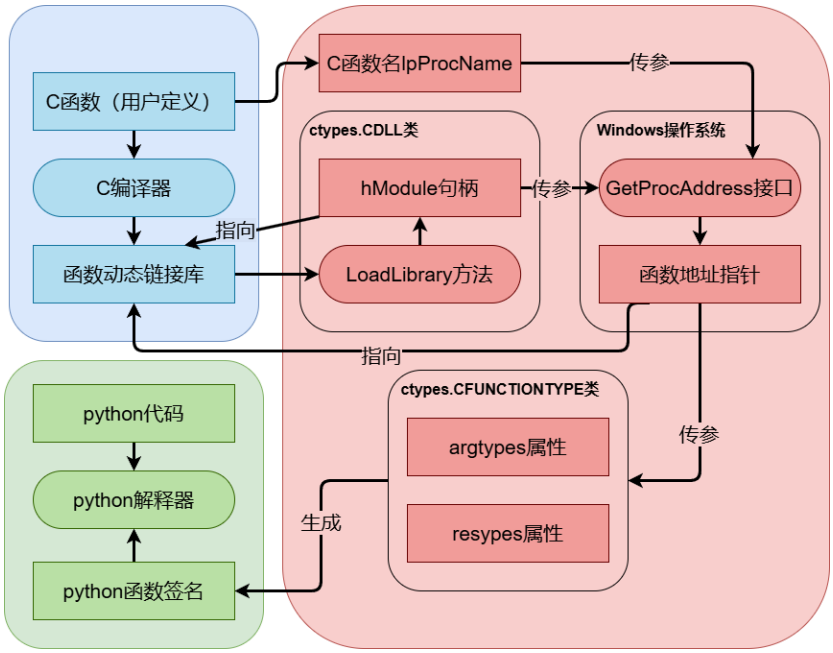

图 1: C 编译器生成 DLL 及 ctypes 通过访问 DLL 调用 C 函数过程示意图

色、绿色区域。其中红色区域由 ctypes 库执行, 绿色区域由 python 解释器执行)。

动态链接库是一种可以被不同程序调用的模块化代码。相较于传统 C 程序，其加载发生在被调用时而非主程序启动时。动态链接库常被用于实现抽象，复用代码与节省内存空间，可提高开发效率与维护效率。动态链接库在 Windows 平台下以.dll 格式的文件存储。C 中用户定义的函数添加有标识 `__declspec(dllexport)` 后可被编译器导出为动态链接库。

`ctypes.CDLL` 类 [1] 可以不依赖编译器的加载动态链接库。该类中的 `LoadLibrary` 方法调用 Windows 操作系统提供的 `LoadLibraryW` 接口函数。`LoadLibraryW` 返回一个句柄（指向被调用动态链接库的指针），该句柄（`hModule`）作为参数传入 Windows 操作系统的 `GetProcAddress` 接口函数，同时传入的还有被调 C 语言函数的函数名（`lpProcName`）。该函数将返回被调函数的地址。

ctypes type	C type	Python type
c_bool	_Bool	bool (1)
c_char	char	1-character bytes object
c_wchar	wchar_t	1-character string
c_byte	char	int
c_ubyte	unsigned char	int
c_short	short	int
c_ushort	unsigned short	int
c_int	int	int
c_uint	unsigned int	int
c_long	long	int
c_ulong	unsigned long	int
c_longlong	__int64 or long long	int
c_ulonglong	unsigned __int64 or unsigned long long	int
c_size_t	size_t	int
c_ssize_t	ssize_t or Py_ssize_t	int
c_float	float	float
c_double	double	float
c_longdouble	long double	float
c_char_p	char * (NUL terminated)	bytes object or None
c_wchar_p	wchar_t * (NUL terminated)	string or None
c_void_p	void *	int or None

图 2: ctypes 数据类型，C 数据类型与 python 数据类型的对应关系。ctypes 根据该对应关系将 C 数据类型推断为 ctypes 数据类型

3 函数参数与返回值的管理

`ctypes.CFUNCTYPE` 类 [3] 可以将被调函数的地址转换为 python 可以识别的函数签名（将形参和返回值数据类型转化为 python 可以识别的数据类型）。除函数地址外，还需要设置属性 `argtypes` 和 `restype` 的值（合适的 ctypes 数据类型）使 python 能传入与接收正确的参数类型。若不显式设置 `argtypes` 和 `restype` 的值，ctypes 会根据一定规则将 C 的数据类型推断为 ctypes 数据类型（见图 2），如 `int` 推断为 `ctypes.c_int`。

进而，python 可以通过该函数签名传入 python 数据类型的参数，并得到解释器可以识别的返回值。

4 讨论 ctypes 的应用与局限性

ctypes 在开发过程中有广泛的应用，这得益于其调用高性能 C 函数与访问底层属性的功能。如在计算机视觉与图像处理工程中，使用 C 编写的程序可以更高效的进行处理数据，而 ctypes 可以为这些高效的 C 函数提供易于调用的 python 接口，提高开发效率。又如在自动化测试中，对一些硬件的控制常需要借助 C 接口，而 ctypes 可以将其转换为 python 接口，使测试过程更高效。然而 ctypes 具有一定的局限性 [2]，如对复杂 C 数据类型的支持能力有限，频繁的类型转换增大时空开销等。

5 结论

ctypes 库功能强大，应用广泛，可以使 python 快捷调用高效且经过优化的 C 库，同时可以通过其类型系统实现 python 与 C 的数据交换，应用与复杂的数据处理场景。然而 ctypes 库的使用具有一定局限性，如反复调用过程中频繁的类型转换带来较大的时空开销，影响性能。同时其对复杂的 C 数据类型支持能力有限，且存在跨平台兼容性问题。尽管具有局限性，ctypes 作为一款高效的工具，为开发者提供便利。

参考文献

[1] Python Software Foundation. ctypes —a foreign function library for python. https://docs.python.org/3/library/ctypes.html#module-ctypes, 2024.

[2] 佚名. Python ctypes：揭秘高级 python 与底层交互秘籍. https://zhuanlan.zhihu.com/p/669892902, 2023.

[3] 佚名. ctypes 封装 c 语言. https://blog.csdn.net/daizikui/article/details/137386375, 2024.