01 CODING SESSION | 14th APRIL 2023

pybind11 & CMake

Combining Python and C++ using pybind11
Introduction to CMake

Agenda

Combi ni ng Pyt hon and C++ usi ng pybi nd11

- Why and how?
- Case study: MurTree project
- Tutorial
  - Building a Python wrapper for C++ code

I nt r oducti on to CMake

- What is CMake and how is it used?
- Tutorial
  - Using CMake to build an application

Combining Python and C++

Motivation:

- Make C++ libraries available in Python (wrappers)
- Increase the performance of Python

Combining Python and C++

Motivation:

- Make C++ libraries available in Python (wrappers)
- Increase the performance of Python code

fast but unfriendly

Example copied from the CppCon 2017 talk
“Pybind11 - Python on Steroids” by Jan Plhak

Combining Python and C++

Motivation:

- Make C++ libraries available in Python (wrappers)
- Increase the performance of Python code

fast but unfriendly

friendly but slow

Example copied from the CppCon 2017 talk
“Pybind11 - Python on Steroids” by Jan Plhak

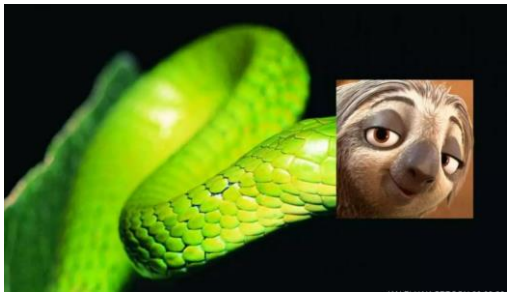
Combining Python and C++

Motivation:

- Make C++ libraries available in Python (wrappers)
- Increase the performance of Python

fast but unfriendly

friendly but slow

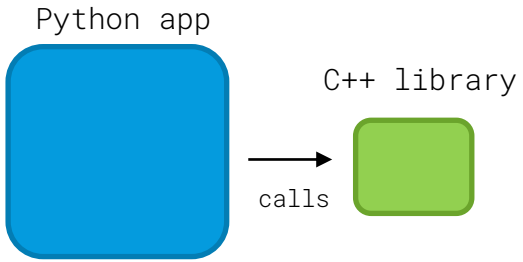


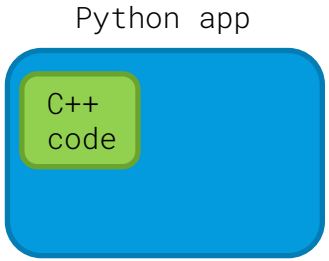
fast and friendly

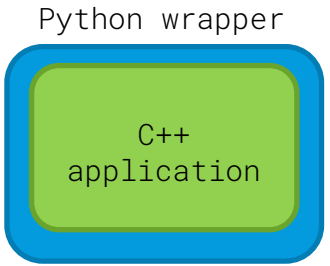
Example copied from the CppCon 2017 talk
“Pybind11 - Python on Steroids” by Jan Plhak

Combining Python and C++

Multiple ways to do it

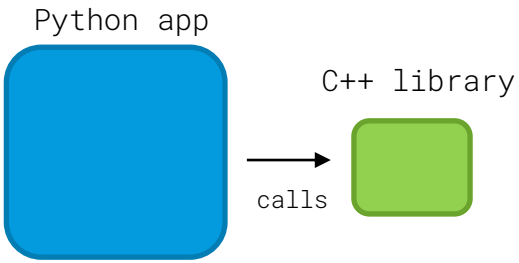


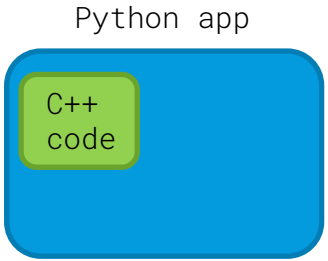


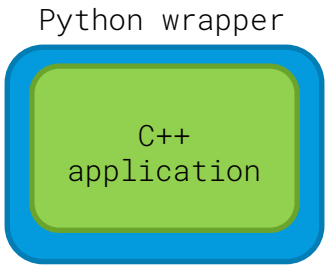


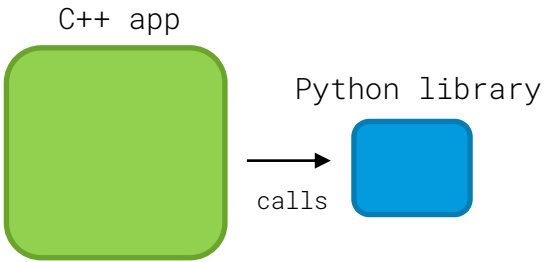
Combining Python and C++

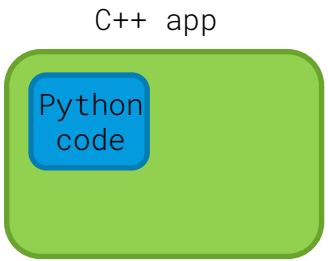
Multiple ways to do it

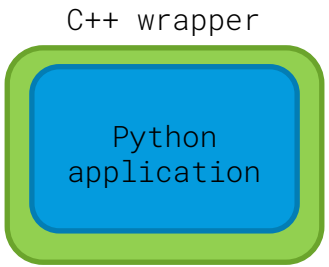






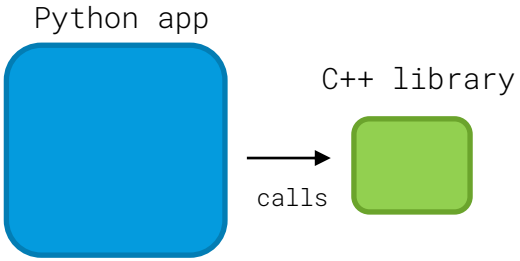




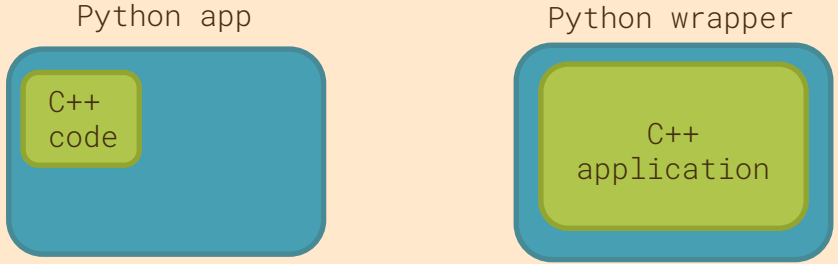


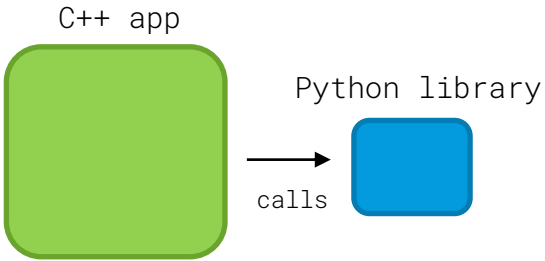
Combining Python and C++

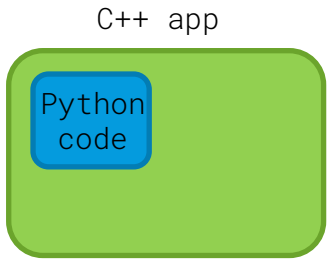
Multiple ways to do it

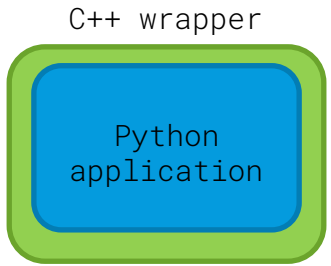


Extending Python with C++





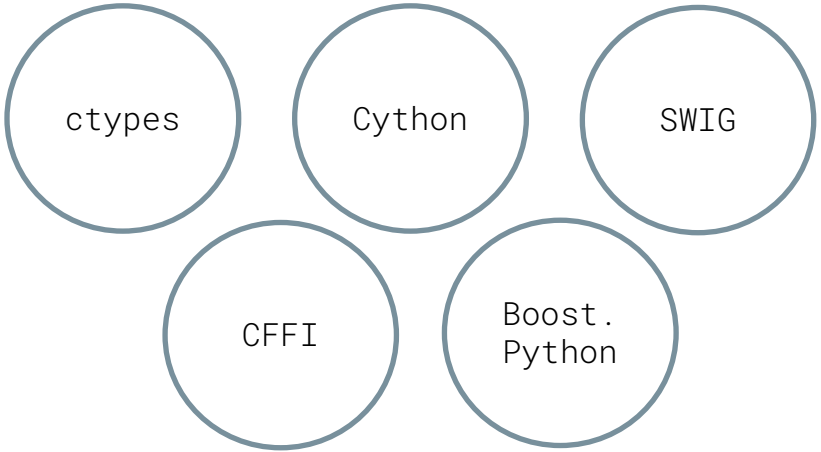




Extending Python with C++

Various tools to do the job

https://stromberg.dnsalias.org/~strombrg/speeding-python/



pybind11

https://pybind11.readthedocs.io/

Extending Python with C++

pybind11

- Lightweight library that creates Python bindings for C++ code
- Exposes C++ types in Python and viceversa
- Compatible with modern C++ language features
- Project example: KpLib
  https://gitlab.com/muellergroup/kplib

https://tech.blueyonder.com/python-calling-c++/

Case Study: MurTree

Project description

- Researcher developed a C++ application. He wants his application to be available for Python users. The DCC builds a Python wrapper for MurTree using pybind11.
- Things in our favour:
  - codebase is small
  - simple architecture
  - no dependencies

https://github.com/MurTree/murtree

Case Study: MurTree

Overview

MurTree GitHub organization repositories

- murtree – original c++ code
- pymurtree – bindings only (uses murtree repo)
- murtree-data – input datasets

pymurtree

- pyproject.toml – registers dependency on pybind11, setuptools and gitpython
- setup.py – clones murtree repo using gitpython and adds bindings using pybind function
- src/main.cpp – declares python bindings for cpp code

Let’s t ake a l ook at t he code

Pybind11 Mini Tutorial

Creating a Python module from existing C++ code

Goal: run the following python code where the add function is implemented in C++

```
>>import pymymath
>>pymymath.add(1,3)
>>4.0
```

https://github.com/yiquintero/dcc-codingsession-pybind11

Pybind11 Mini Tutorial

Adding a new feature to mymath and pymymath

Goal: Add a new function to the mymath library and expose it to Python users by adding it to the pymymath module

👉 Don't forget to uninstall the previous pymymath module using "pip uninstall pymymath"

```
>>import pymymath
>>pymymath.subtract(3,1)
>>2.0
```

https://github.com/yiquintero/dcc-codingsession-pybind11