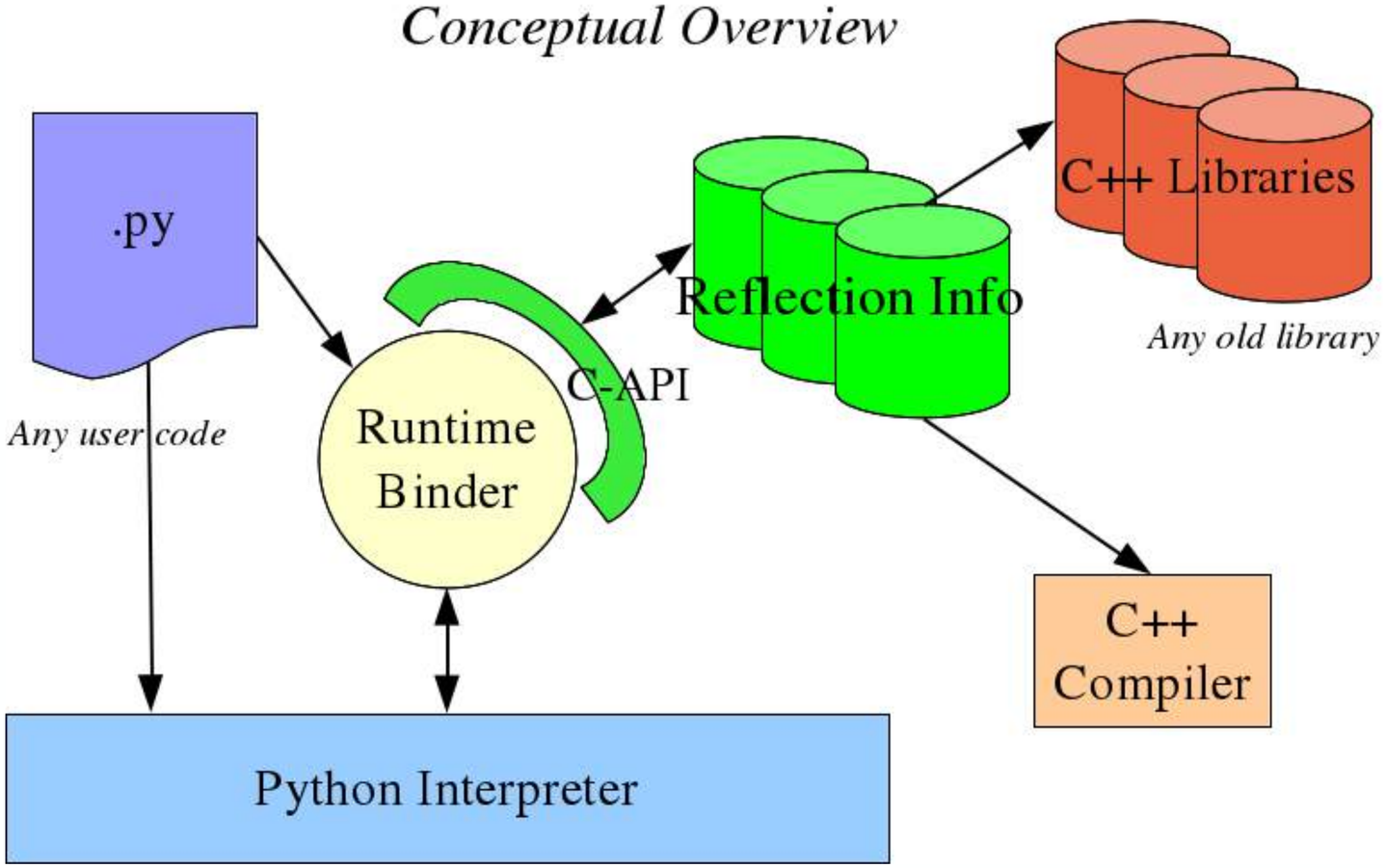
GSoC 2024 Final Presentation

Mentors - Vassil Vassilev, Jonas Rembser, Wim Lavrijsen, Aaron Jomy

Khushiyant

About cppyy

- cppyy is an automatic, run-time, Python-C++ bindings generator, for calling C++ from Python and Python from C++. Run-time generation enables detailed specialisation for higher performance, lazy loading for reduced memory use in large scale projects, Python-side cross-inheritance and callbacks for working with C++ frameworks, run-time template instantiation, automatic object down casting, exception mapping, and interactive exploration of C++ libraries

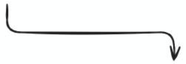

Conceptual Overview
.py
Any user code
Runtime Binder
C-API
Reflection Info
C++ Libraries
Any old library
C++ Compiler
Python Interpreter

Work Overview

- Support for arbitrary dimension numpy init support (template less init)
  - PR cppyy#255 (RTM, 2nd Review)
- Support for initialisation of Eigen from initisalizer_list (7 overload on linux)
  - PR CPyCppyy#41 (WIP)
  - PR cppyy#271

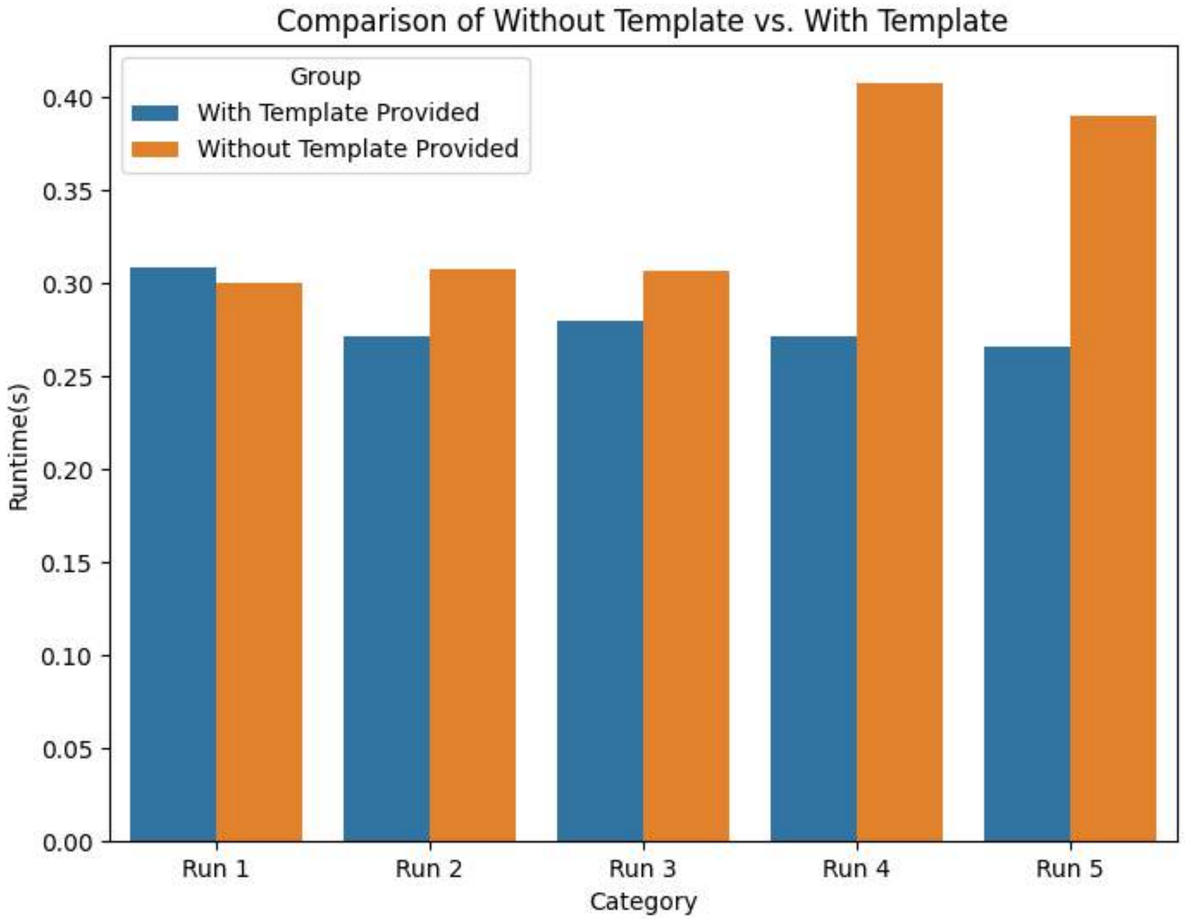
cppyy#255

```
import cppyy

v = cppyy.gbl.std.vector[
        cppyy.gbl.std.vector[
        cppyy.gbl.std.vector[
        cppyy.gbl.std.vector[
        cppyy.gbl.std.vector[float]]]]](arr) # arr is a 5-dimensonal numpy array
```



```
import cppyy

v = cppyy.gbl.std.vector(arr) # arr is a 5-dimensonal numpy array
```

Template Less Initialisation Time



Without template initialisation is slower by 18% on average

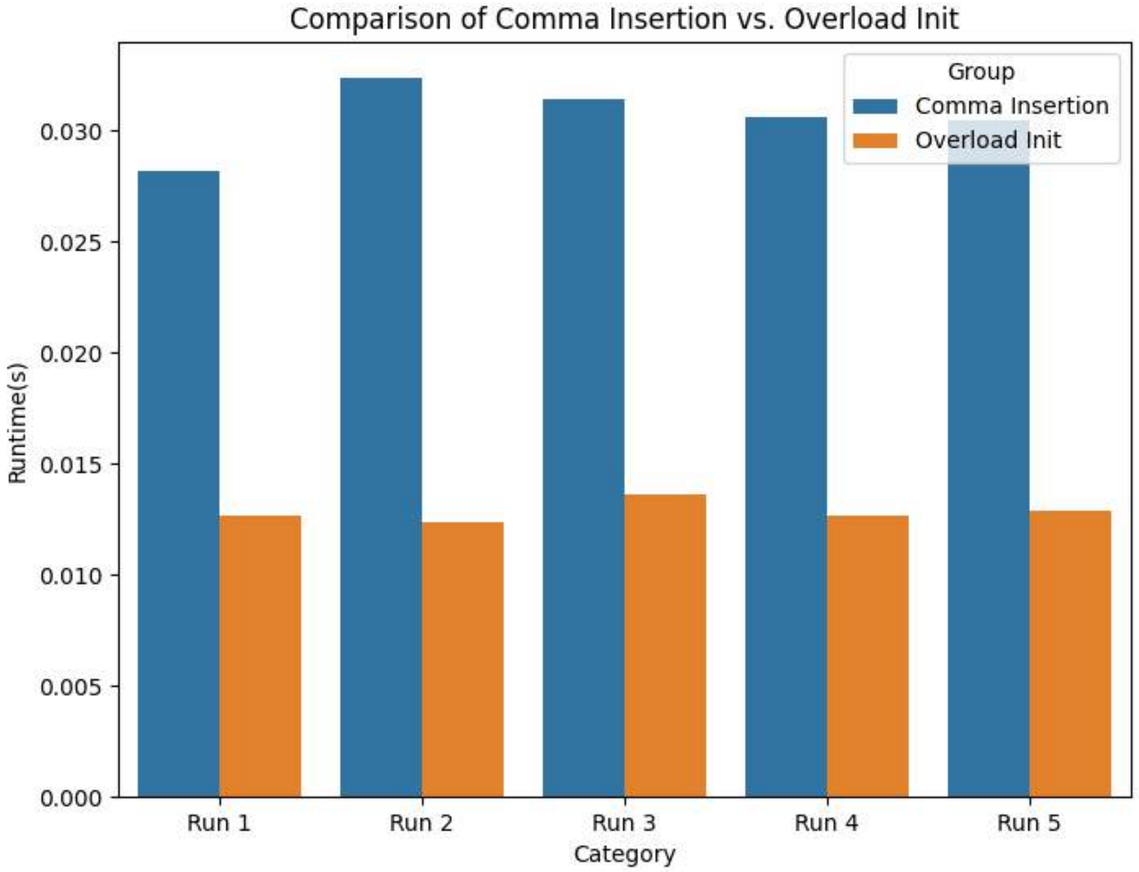
Note: *Calculated for randomly initialised* ***10x10x10*** *numpy array with on average of 5 run cycles*

CPyCppyy#41

```python
import cppyy

inc_paths = [
    "/opt/homebrew/Cellar/eigen/3.4.0_1/include/"
]

eigen_path = None
for p in inc_paths:
    p = os.path.join(p, "eigen3")
    if os.path.exists(p):
        eigen_path = p

cppyy.add_include_path(eigen_path)
with warnings.catch_warnings():
    warnings.simplefilter('ignore')
    cppyy.include('Eigen/Dense')

m = cppyy.gbl.Eigen.MatrixXd(rows, cols)
c = (m << arr[0])
for i in range(len(arr)):
  c.__comma__(arr[i])
```

```python
import cppyy

inc_paths = [
    "/opt/homebrew/Cellar/eigen/3.4.0_1/include/"
]

eigen_path = None
for p in inc_paths:
    p = os.path.join(p, "eigen3")
    if os.path.exists(p):
        eigen_path = p

cppyy.add_include_path(eigen_path)
with warnings.catch_warnings():
    warnings.simplefilter('ignore')
    cppyy.include('Eigen/Dense')

m = cppyy.gbl.Eigen.MatrixXd([arr])
```

Eigen Initialisation Comparison



Overload initialisation is faster by 58% on average

Note: *Calculated for randomly initialised* ***10000x1*** *array with on average of 5 run cycles*

What’s Next

- Documentation for the added support
- Tests and review for CPyCppyy#41
- Explore rest of the project scope

Thank You