Contribution ID: **233**   Type: **Oral**

Efficient and Accurate Automatic Python Bindings with Cppyy and Cling

Tuesday 25 October 2022 17:40 (20 minutes)

The simplicity of Python and the power of C++ provide a hard choice for a scientific software stack. There have been multiple developments to mitigate the hard language boundaries by implementing language bindings. The static nature of C++ and the dynamic nature of Python are problematic for bindings provided by library authors and in particular features such as template instantiations with user-defined types or more advanced memory management.

The development of the C++ interpreter Cling has changed the way we can think of language bindings as it provides an incremental compilation infrastructure available at runtime. That is, Python can interrogate C++ on demand and fetch only the necessary information. This way of automatic binding provision requires no binding support by the library authors and offers better performance than Pybind11. This approach pioneered in ROOT with PyROOT and later was enhanced with its successor Cppyy. However, until now, Cppyy relied on the reflection layer of ROOT which is limited in terms of provided features and performance.

In this talk we show how basing Cppyy purely on Cling yields better correctness, performance and installation simplicity. We illustrate more advanced language interoperability of Numba-accelerated Python code capable of calling C++ functionality via Cppyy. We outline a path forward for integrating the reflection layer in LLVM upstream which will contribute to the project sustainability and will foster greater user adoption. We demonstrate usage of Cppyy through Cling's LLVM mainline version Clang-Repl.

Significance

The automatic Python bindings based on the C++ interpretation is unique, cutting-edge technology worldwide. The Cppyy project is a cornerstone for the ROOT user base and bridges the C++ and Python ecosystems and analyses. The Cppyy project has users beyond HEP. It outperforms Pybind11 and is a key motivator for starting the nanobind project by the Pybind11 community. In this talk we talk about how to bring the technology on the next level in terms of performance, robustness and sustainability. This will have a direct effect on HEP and its community should know about the project plans.

References

Experiment context, if any

Primary authors: KUNDU, Baidyanath (Princeton University (US)); VASILEV, Vassil (Princeton University (US)); LAVRIJSEN, Wim (Lawrence Berkeley National Lab. (US))

Presenter: KUNDU, Baidyanath (Princeton University (US))

Session Classification: Track 1: Computing Technology for Physics Research

Track Classification: Track 1: Computing Technology for Physics Research