Adopting CppInterOp in cppyy

Vipul Cariappa

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

Terminology and Repositories

Cppyy: (frontend) provides fully automatic, dynamic Python-C++ bindings by using the Cling C++ interpreter and LLVM.

Cppyy-Backend: Uses the cling C++ interpreter for incremental compilation and reflection.

CPyCppyy: Python facing interface using the Python C API, and consuming the Cppyy-Backend.

My Project

Adopting *CppInterOp* as an backend alternative to the cling interpreter in *Cppyy*.

CppInterOp: A Clang-based C++ Interoperability Library and Incremental Compiler.

Motivation

The main motivation is to directly use the LLVM's clang-REPL for C/C++ incremental compilation, instead of depending on cling.

The current cppyy uses lots of string manipulation. This string manipulation increases the performance overhead and affects the code readability and debuggability.

Current Status

Compiler-Research maintains a fork of *Cppyy* that uses the *CppInterOp* for C++ incremental compilation and reflection. This fork is not feature complete, and approximately half of the test cases fail.

This is due to missing feature in *CppInterOp* and incomplete implementation.

Tests Summary

My goal is to reduce the number of failing tests by 60 to 100 by the end of the internship.

Hope I am not overestimating

Pass	255
XFail	227
Skip	22

*results from my local machine

Thank You

Vipul Cariappa