



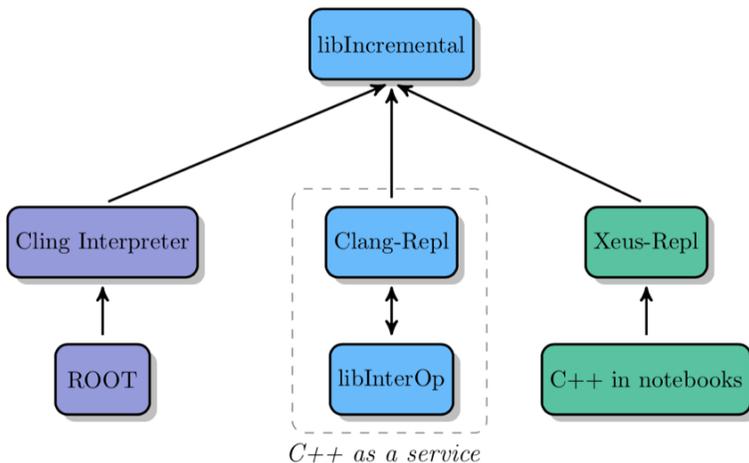
Award #: 1931408

CSSI Element: C++ as a service - rapid software development and dynamic interoperability with Python and beyond

PI: David Lange,
Institution: Princeton University

Program: OAC Office of Advanced Cyberinfrastructure

libIncremental Design



Our approach is to generalize a high-energy physics analysis tool (“Cling”) to a general-purpose and fully functional tool that becomes part of LLVM/Clang

CaaS programming model

```

In [1]: struct S { double val = 1.; };
In [2]: from libInterop import std
        python_vec = std.vector(S)(1)
In [3]: print(python_vec[0].val)
1
In [4]: class Derived(S)
        def __init__(self):
            self.val = 0
        res = Derived()
In [5]: __global__ void sum_array(int n, double *x, double *sum) {
        for (int i = 0; i < n; i++) *sum += x[i];
        }
        // Init N=1M and x[i] = 1.f. Run kernel on 1M elements on the GPU.
        sum_array<<<1, 1>>>(N, x, &res.val);
  
```

CaaS aims to provide programmers and data scientists a simple and general solution to language interoperability:

- Advance the interpretative technology to provide scientists a state-of-the-art C++ execution environment
- Enable functionality which can provide dynamic, native-like, runtime interoperability between C++ and Python
- Allow seamless utilization of heterogeneous hardware (e.g., hardware accelerators)
- Enable rapid application development even to those with a complex codebase