The University of Texas at Austin

PyKokkos: A Performance Portability Framework for Python

SC BoF ♦ 11/20/2024 ♦ Atlanta

Presented by: Milos Gligoric

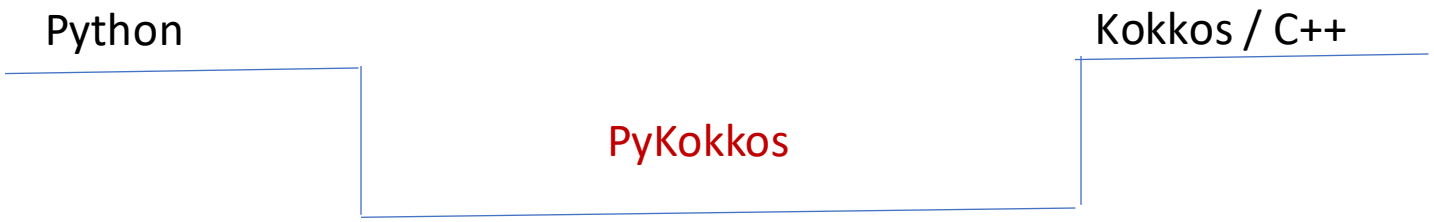
Nader Al Awar, Hannan Naeem, Umair Shahzad, Olivia Mitchell, George Biros, Milos Gligoric

Kokkos

- Kokkos is a C++ programming model for performance portability
  - It is implemented as a C++ library on top of CUDA, OpenMP, …
- Code can run on different architectures and achieve good performance
- Minimizes the amount of architecture-specific implementation details the user should know
- Provides abstractions for both parallel execution of code and data management

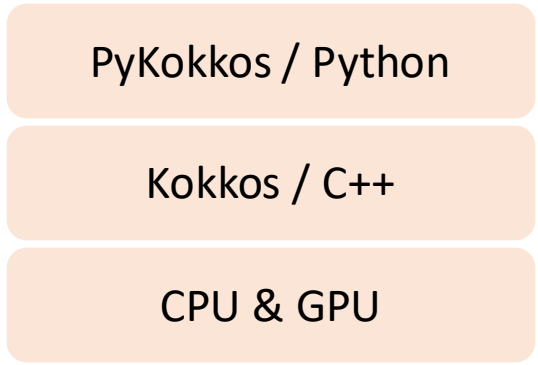
Motivation

- BUT, writing Kokkos code is not easy, especially for programmers with no C++ background
- Many scientific applications nowadays are written in Python (or people want to write them in Python)
  - Quick prototyping that sacrifice performance
- How to bridge the gap?



PyKokkos

- Enables developers to utilize a performance portability programming ecosystem (Kokkos)
- Enables developers to write Kokkos kernels in Python
- Implemented as a Python framework
  - Partially translated to C++ Kokkos
  - Partially connected via bindings

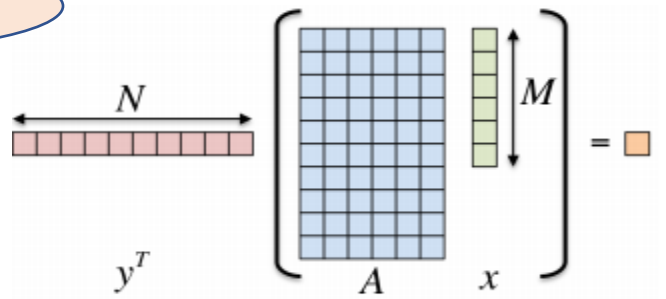


PyKokkos: Example

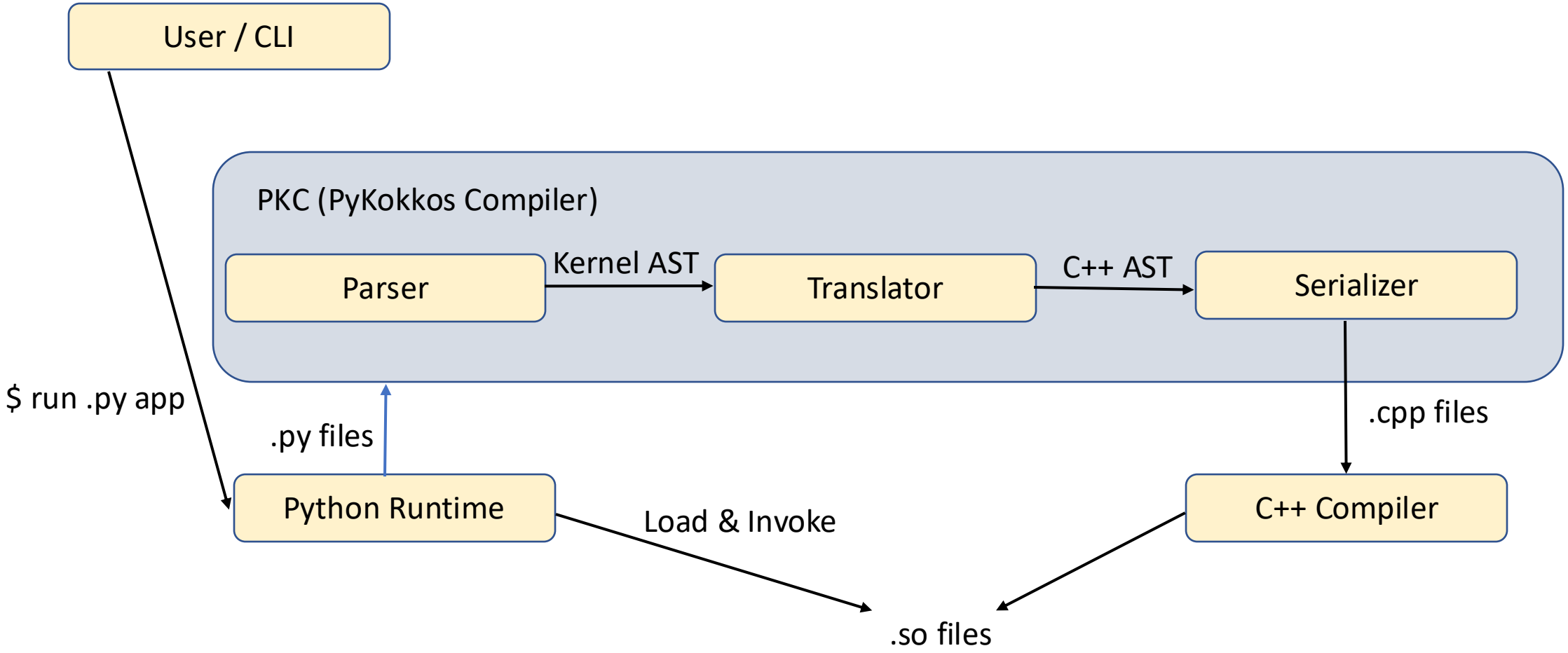
```python
@pk.functor
class Workload:
    def __init__(self, N: int, M: int):
        self.N: int = N
        self.M: int = M
        self.y: pk.View1D[pk.double] = pk.View([N], pk.double)
        self.x: pk.View1D[pk.double] = pk.View([M], pk.double)
        self.A: pk.View2D[pk.double] = pk.View([N, M], pk.double)

        self.y.fill(1)
        self.x.fill(1)
        self.A.fill(1)

    @pk.workunit
    def yAx(self, j: int, acc: pk.Acc[float]):
        temp2: float = 0
        for i in range(self.M):
            temp2 += self.A[j][i] * self.x[i]

        acc += self.y[j] * temp2


if __name__ == "__main__":
    pk.set_default_space(pk.OpenMP)
    N = 10
    M = 10
    w = Workload(N, M)
    p = pk.RangePolicy(pk.Default, 0, N)
    result = pk.parallel_reduce(p, w.yAx)
```

Bindings in use

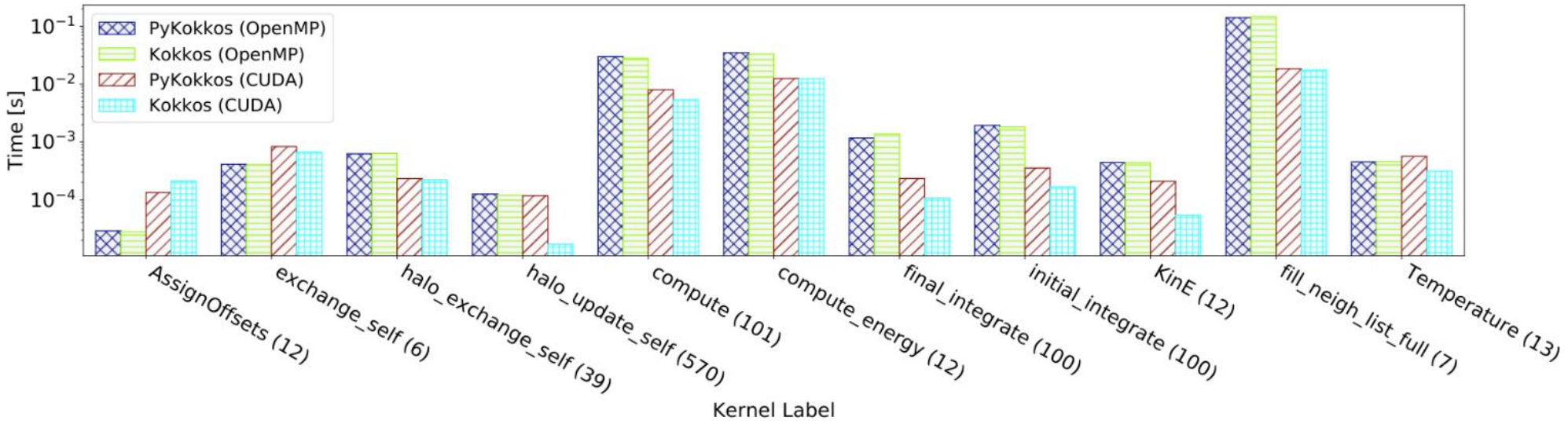
Translated to C++



PyKokkos: Workflow



PyKokkos: Results for ExaMiniMD



Highlights

- Collaboration with Sandia
  - Damien Lebrun-Grandie, Siva Rajamanickam, Jonathan R Madsen, and Christian Trott
- Part of the Kokkos organization on GitHub
  - https://github.com/kokkos/pykokkos
- Now part of Linux Foundation
- Solid performance on small/medium examples
- Used for writing kernels in various domains
  - Well known algorithms
  - Simulations of a plasma torch
  - Machine learning algorithms

https://github.com/EngineeringSoftware/python-hpc-frameworks