Science and Technology Facilities Council

PSyclone

Expanding PSyclone target languages to leverage the wider HPC software ecosystem

Rupert Ford, Andy Porter, **Sergi Siso**, STFC Hartree Centre
Iva Kavcic, Chris Maynard, Andrew Coughtrie, UK Met Office
Joerg Henrichs, Australian Bureau of Meteorology

ESIWACE2 training course on Domain-specific Languages in Weather and Climate, 23rd-27th November 2020

Hartree Centre
Science & Technology Facilities Council

Met Office

Australian Government
Bureau of Meteorology

PSycloneBench

PSyclone Benchmarking suite available at https://github.com/stfc/PSycloneBench

- NemoLite2D

A vertically averaged version of the dynamical part of NEMO ported to the Gocean PSyclone DSL. It has multiple manual implementations in different programming models for comparison with the PSyclone-generated code.

- Shallow Water

Originally developed by Paul Swarztrauber of NCAR and is a 2D shallow-water model with periodic boundary conditions in both dimensions.

- LFRic SMV

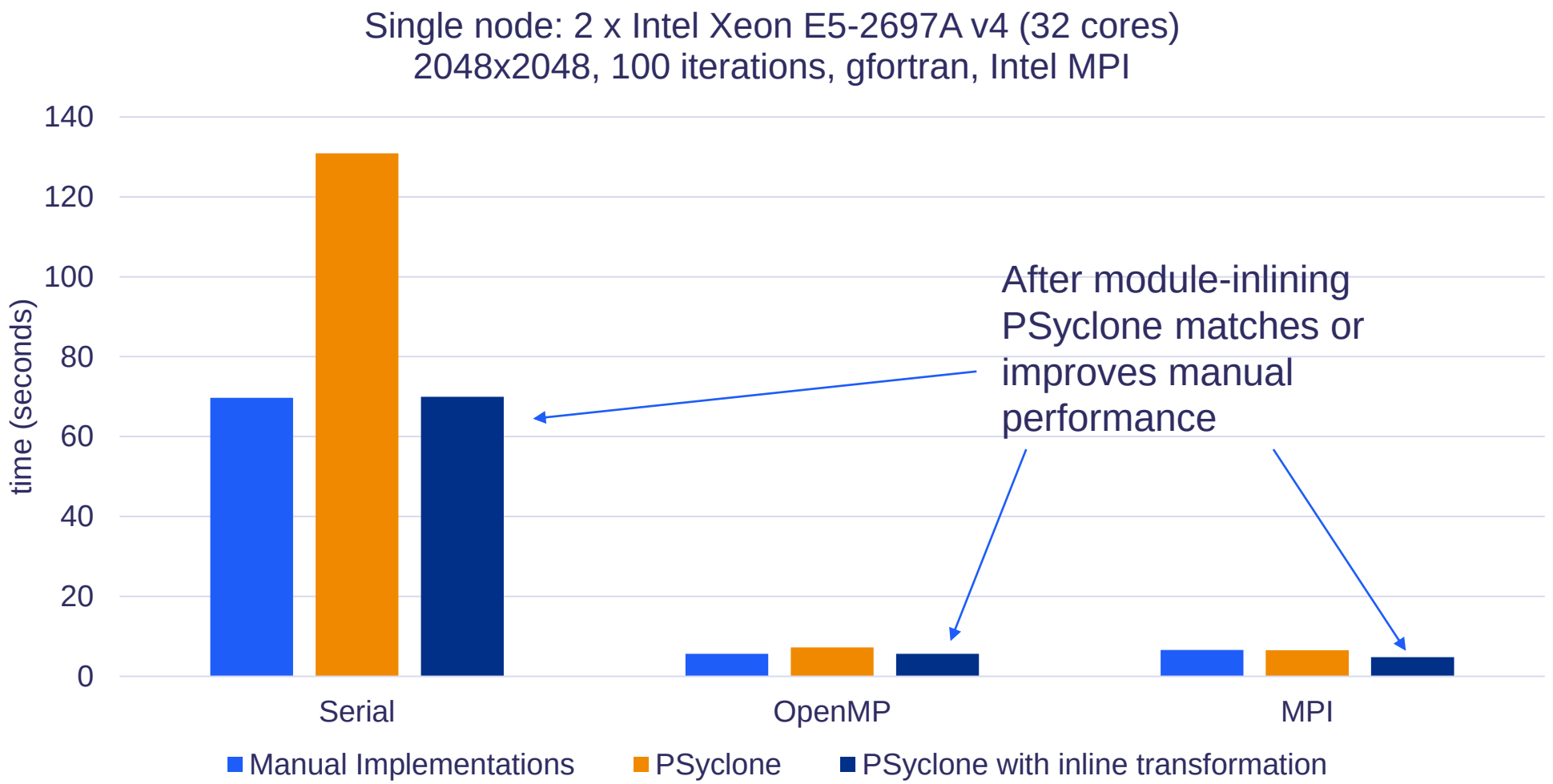
Multiple implementations of the matrix-vector multiplication operation done in the LFRic application.

NemoLite2D

- Can be used by PSyclone with the GOcean DSL

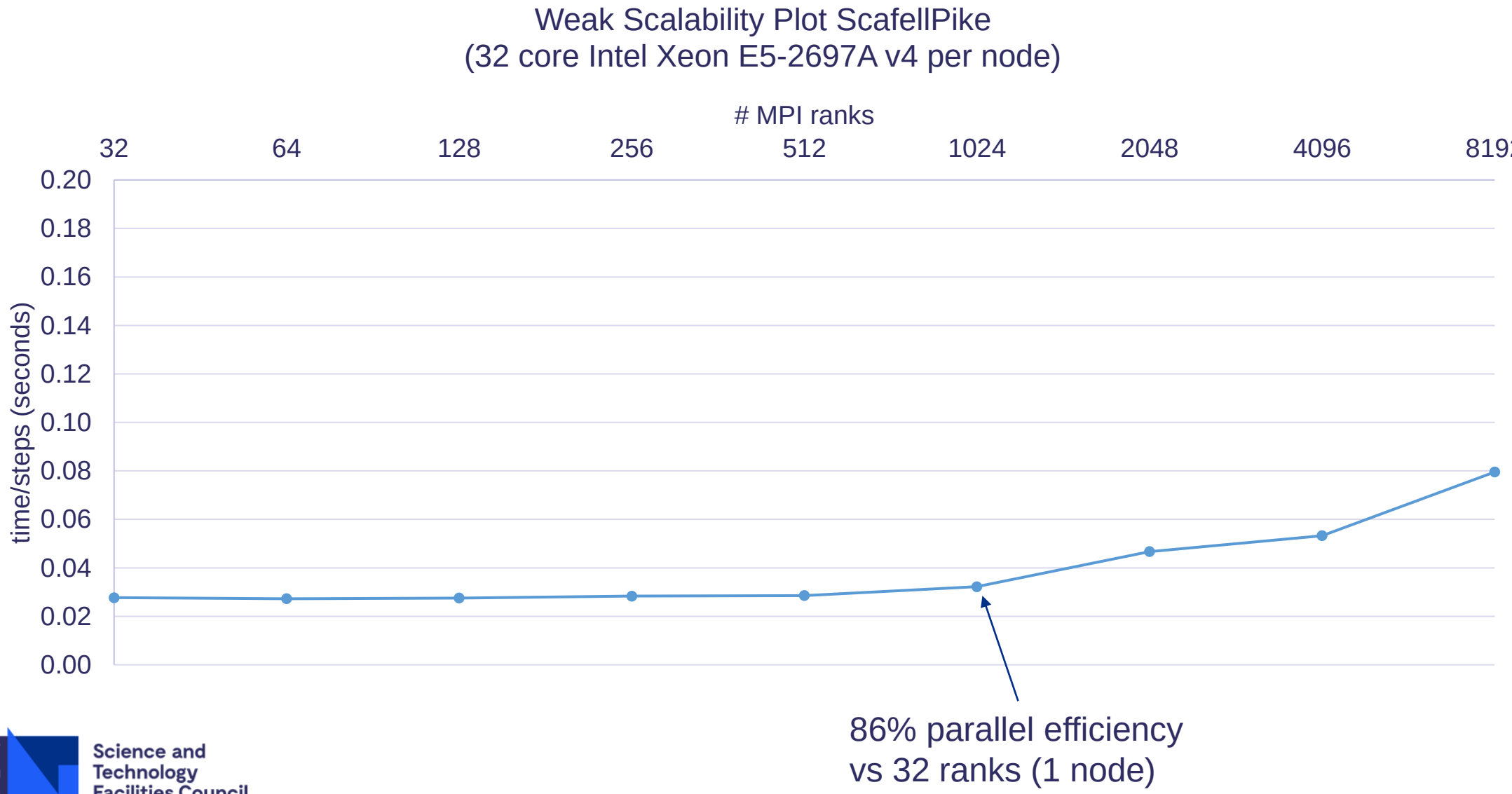
- Has multiple manual implementations:
  - Fortran: Serial, OpenMP, MPI and OpenACC
  - C++: Serial, OpenMP, OmpSs, Kokkos
  - OpenCL
  - Regent

NemoLite2D Fortran Performance



Science and Technology Facilities Council

NemoLite2D Fortran MPI Performance



Science and Technology Facilities Council

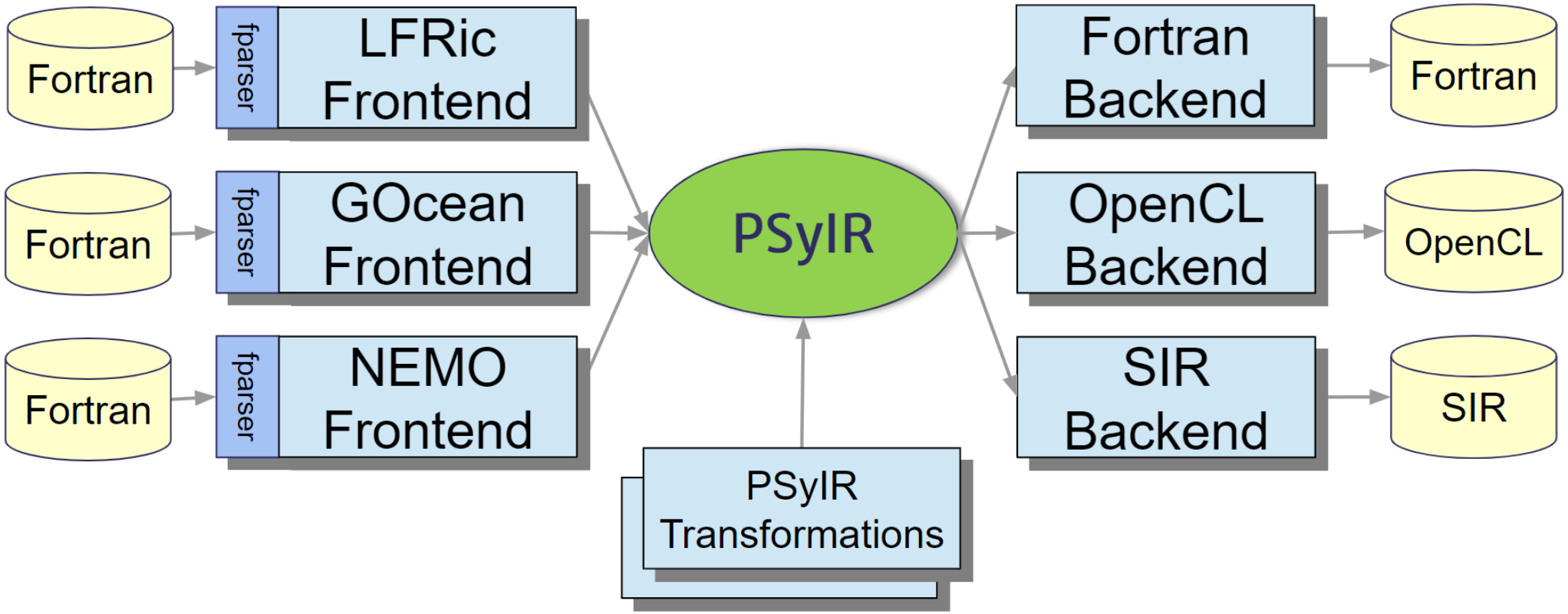
Heterogeneous hardware landscape

Position	Name	Processor	Linpack (TFlop/s)
#1	Fugaku	Fujitsu A64FX 48C	415,530
#2	Summit	IBM POWER9 22C Nvidia V100	148,600
#3	Sierra	IBM POWER9 22C Nvidia V100	94,640
#4	Taihulight	Sunway SW26010 260C	93,014
#5	Tianhe-2A	Intel Xeon E5-2692v2 12C MATRIX-2000	61,444

Upcoming systems also with Intel and AMD processors and accelerator architectures. Alternative architectures: FPGA, RISCV, …

- Is Fortran with Directives alone going to be enough?

Vision

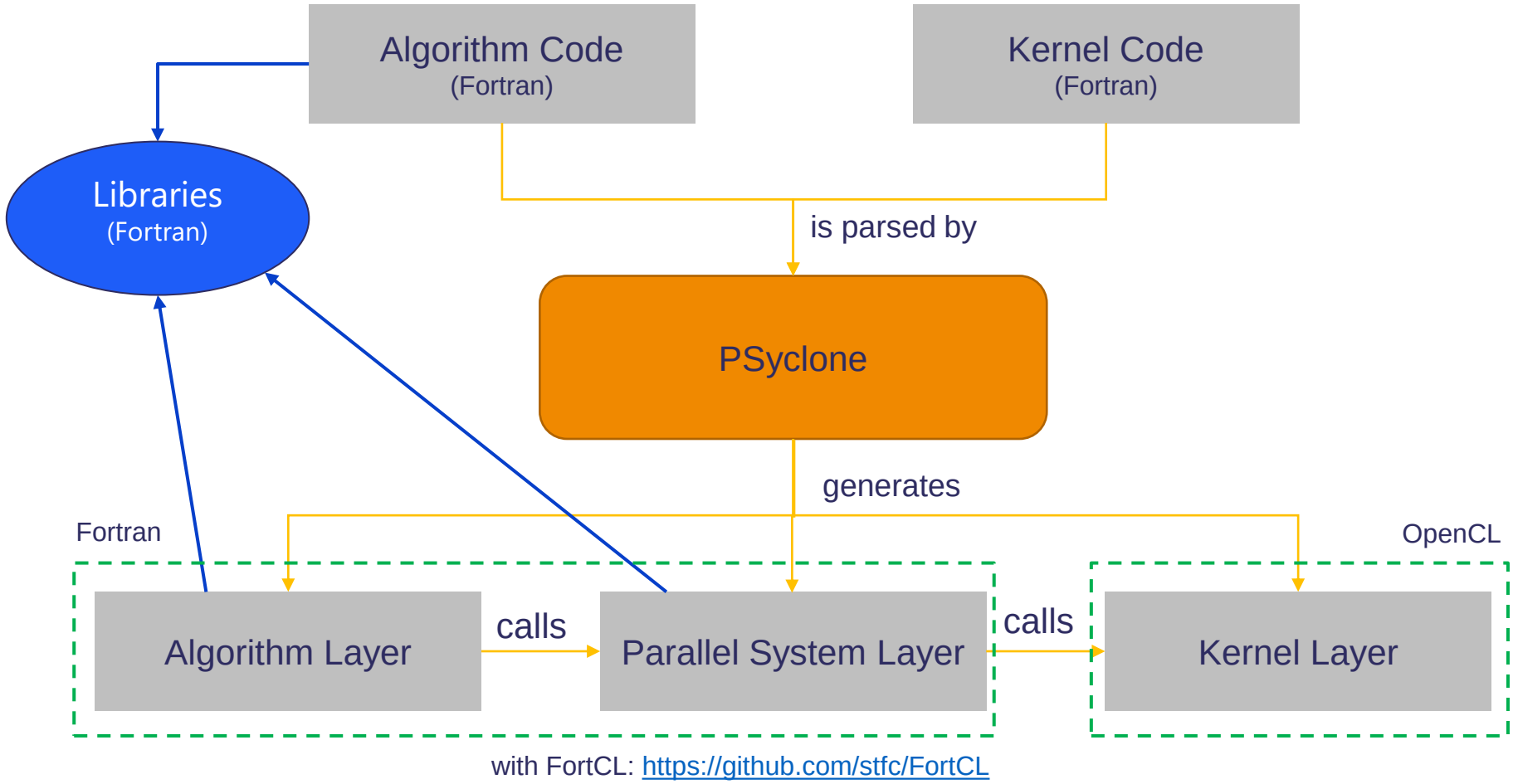


Hartree Centre – US Exascale Computing Project collaboration funded by STFC (UKRI)

Science and Technology Facilities Council

EUROEXA has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 754337
ESiWACE2 has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 823988

Fortran-to-OpenCL example



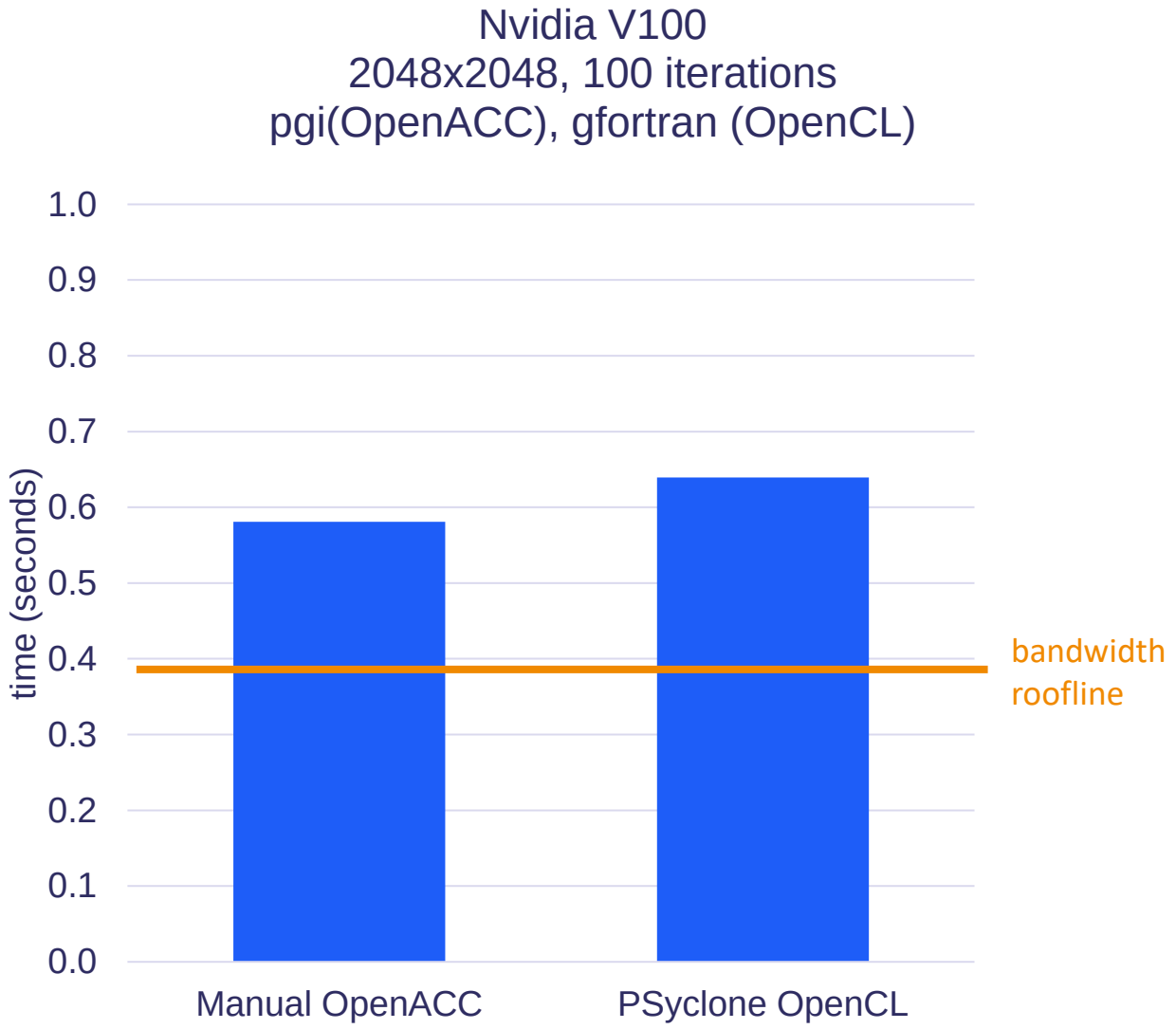
with FortCL: https://github.com/stfc/FortCL

Science and Technology Facilities Council

OpenCL backend results

```
global_trans = KernelGlobalsToArguments()
ocl_trans = OCLTrans()

# Get the code we want to transform
schedule = psy.invokes.get('invoke_0').schedule

# Remove global variables from kernels
for kern in schedule.kernels():
    global_trans.apply(kern)

# Transform the code to OpenCL
ocl_trans.apply(schedule)

# Specify OpenCL local size to improve
# performance
for kern in schedule.kernels():
    kern.set_opencl_options({'local_size': 64})
```

```
$ psyclone -s ./ocl_trans.py nemolite2d.f90
```



Science and Technology Facilities Council

Ongoing work on C/C++ based backends

- EuroEXA: OpenCL can target FPGAs, but the generated code is substantially different than for other architectures, e.g: OpenCL tasks, HLS/Xilinx's pragmas, …

- Collaboration with ECP: New C++ frameworks provide good opportunities to ease the performance portability challenge. Is a PSyclone Kokkos backend a good solution?

EuroEXA

- PSyclone can already target FPGAs but current performance is not competitive with CPU/GPUs:
  - Only 10-20% LUTs and FFs utilization on a Xilinx U200 FPGA. More parallelism need to be exposed in the OpenCL code.
  - Next steps include using OpenCL Tasks instead of NDRanges.
  - Multi-FPGA executions coming soon.

Collaboration with ECP

Objectives:

- Explore integration of ECP technology stack, particularly the Kokkos programming model.
- PSyclone is available on the Spack package manager.
- Share knowledge and best practices with the ECP partners.

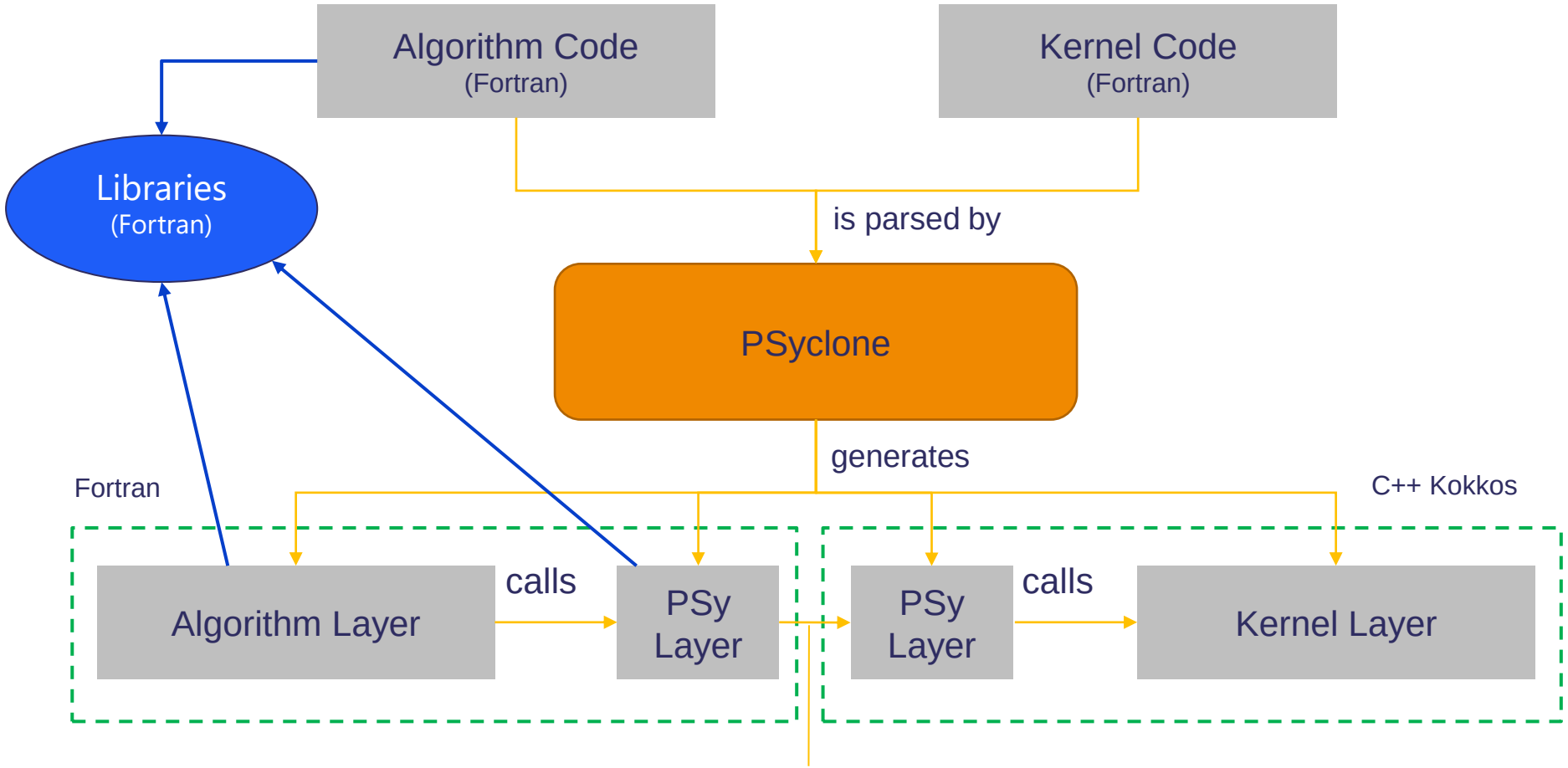
Initial exploration with Kokkos

- Fully integrate with Kokkos (adopt View containers)
  - Assume data duplication (Fortran Array <-> Views)
  - Use all Kokkos functionality.

- Use UnmanagedViews and Kokkos Fortran interface
  - Some Kokkos functionality like acceleration available but PSyclone still in charge of the data-layout.

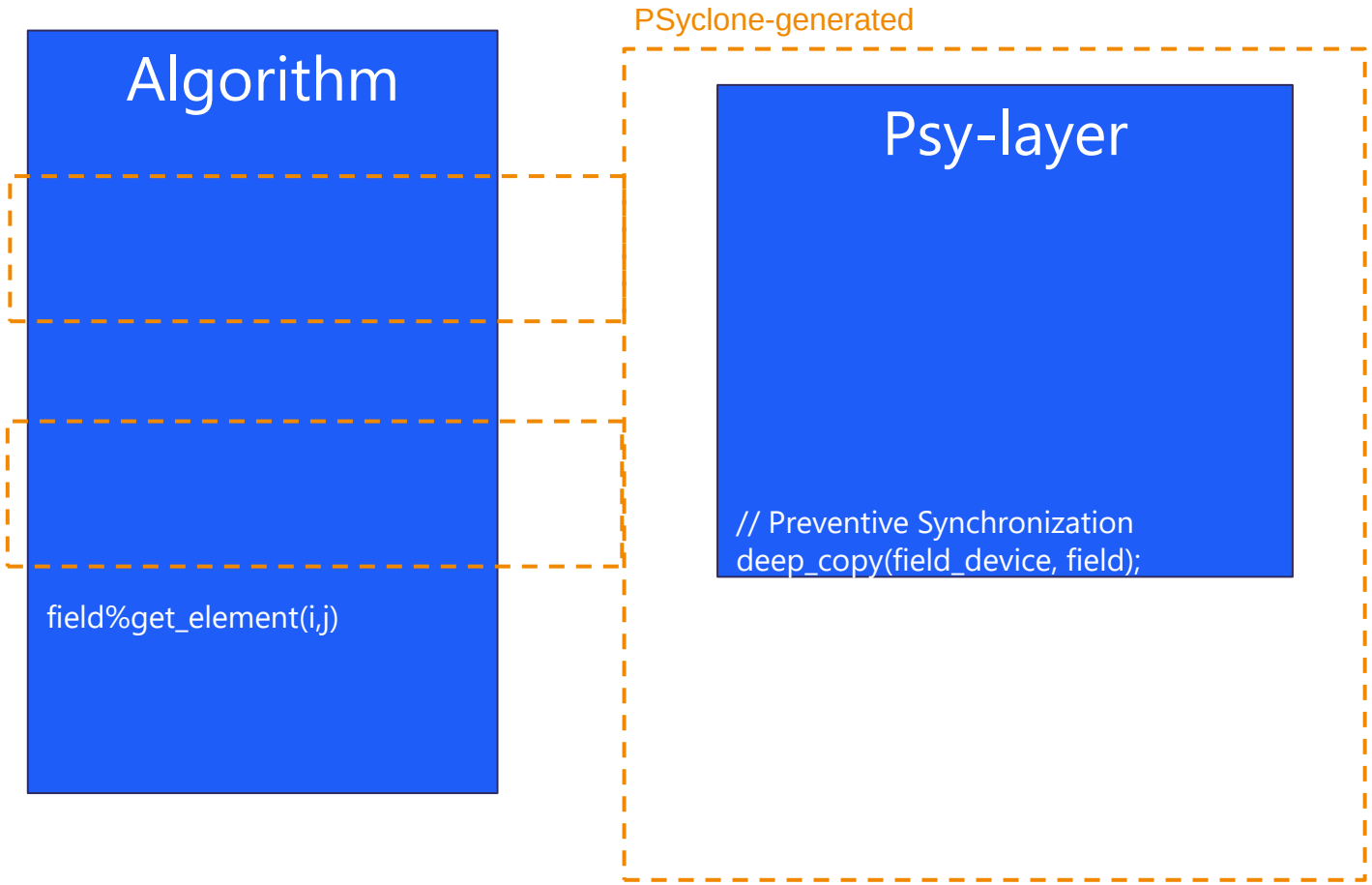
- Use Kokkos with rawpointers from Fortran Arrays
  - No automatic device acceleration or data-layout abstraction.
  - We can still use Kokkos parallel execution model.

Plans for C++ frameworks

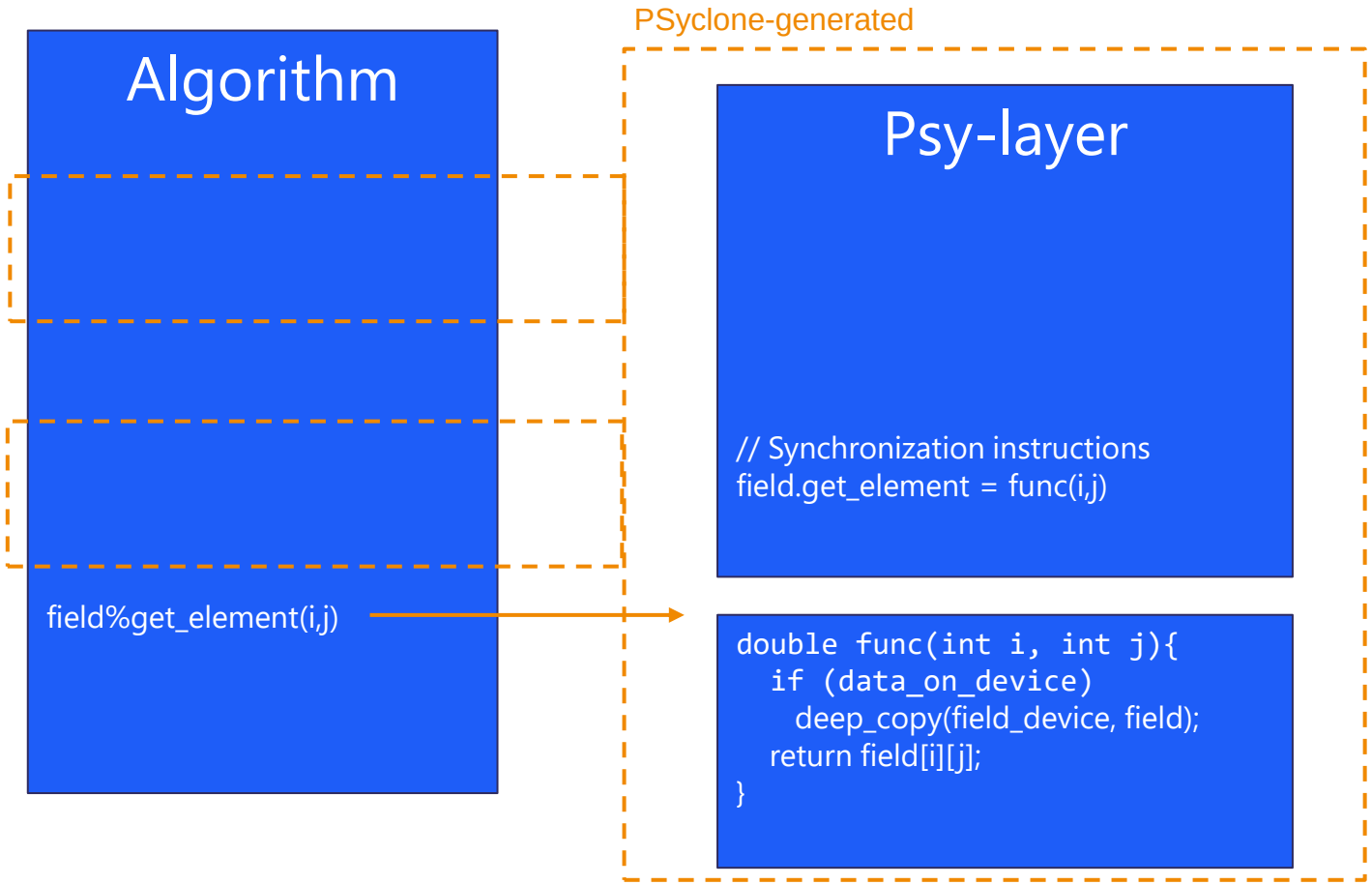


Prepare all needed functionality with raw pointers and function wrappers and use a single Fortran C_ISO_BINDING call.

Issue: Unnecessary preventive data copies

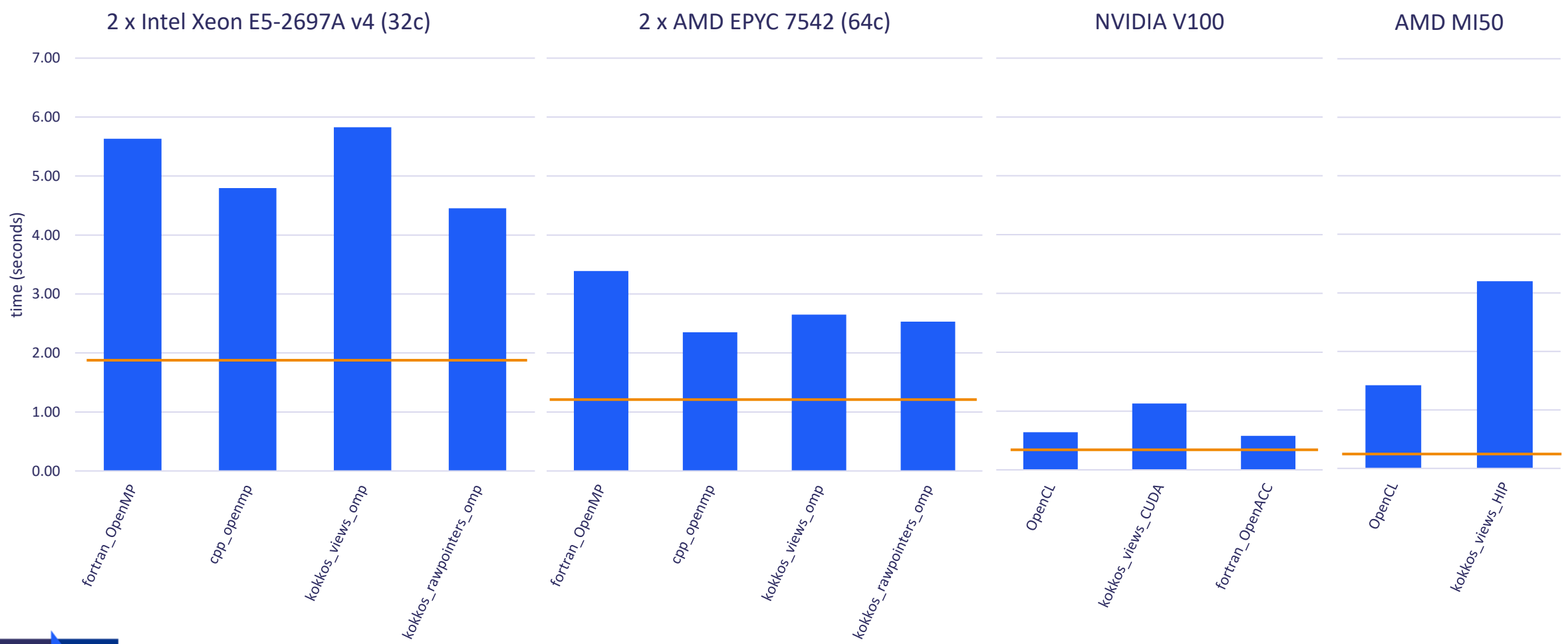


Lazy synchronisation to reduce data copies



Science and Technology Facilities Council

NemoLite2D manual implementations



Science and Technology Facilities Council

bandwidth
roofline

Take away

- Increasingly heterogenous hardware and software ecosystem in HPC.
- PSyclone currently supports CPU and GPU computing, we are developing basic support for FPGA computing.
- Initial work on supporting performance portable C++ programming models like Kokkos or SYCL.

Questions?

Read more about PSyclone and PSycloneBench at:
https://github.com/stfc/PSyclone
https://github.com/stfc/PSycloneBench

Contact: sergi.siso@stfc.ac.uk