



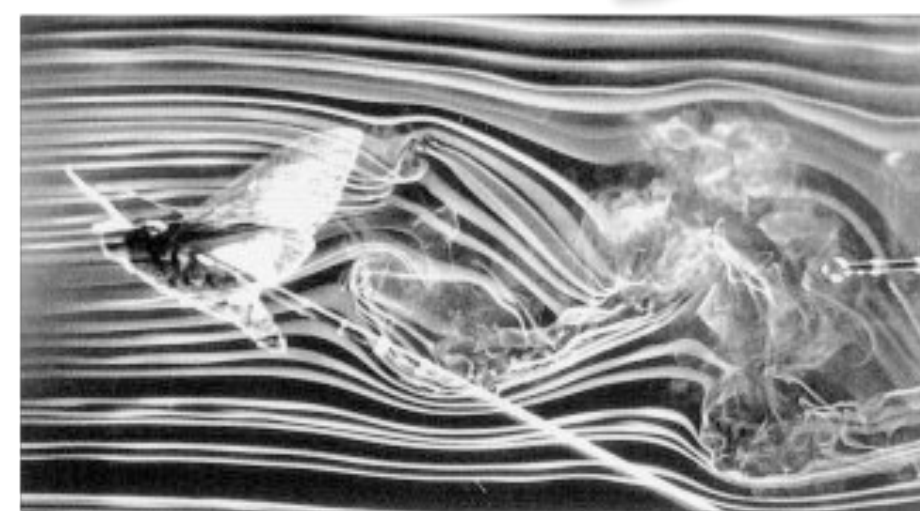
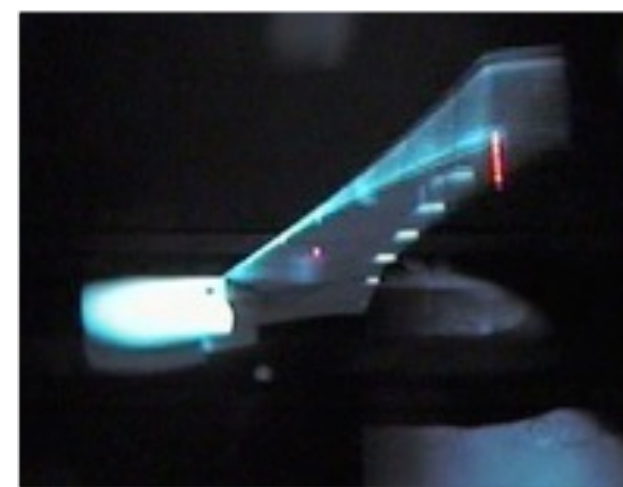
P. E. Vincent

Department of Aeronautics
Imperial College London

4th June 2016

PyFR

Flux Reconstruction
PyFR
Modern Hardware



PyFR

- Features (v1.4.0)

Governing Equations	Compressible Euler Compressible Navier Stokes
Spatial Discretisation	Arbitrary order FR on mixed unstructured grids (tris, quads, hexes, tets, prisms, pyramids)
Temporal Discretisation	Range of explicit Runge-Kutta schemes
Platforms	CPU clusters (C-OpenMP-MPI) Nvidia GPU clusters (CUDA-MPI) AMD GPU clusters (OpenCL-MPI) Xeon Phi Clusters (PyMIC-MPI)
Precision	Single Double
Input	Gmsh, CGNS
Output	Paraview

PyFR

- Features (v1.4.0)

Governing Equations	Compressible Euler Compressible Navier Stokes
Spatial Discretisation	Arbitrary order FR on mixed unstructured grids (tris, quads, hexes, tets, prisms, pyramids)
Temporal Discretisation	Range of explicit Runge-Kutta schemes
Platforms	CPU clusters (C-OpenMP-MPI) Nvidia GPU clusters (CUDA-MPI) AMD GPU clusters (OpenCL-MPI) Xeon Phi Clusters (PyMIC-MPI)
Precision	Single Double
Input	Gmsh, CGNS
Output	Paraview

PyFR

- Python Outer Layer (Hardware Independent)

Python Outer Layer
(Hardware Independent)

- Setup
- Distributed memory parallelism
- Outer 'for' loop and calls to Hardware Specific Kernels

PyFR

- Need to generate the Hardware Specific Kernels

Python Outer Layer
(Hardware Independent)

- Setup
- Distributed memory parallelism
- Outer 'for' loop and calls to Hardware Specific Kernels

PyFR

- Two **types** of kernel are required ...

Python Outer Layer (Hardware Independent)

- Setup
- Distributed memory parallelism
- Outer 'for' loop and calls to
Hardware Specific Kernels

Matrix Multiply Kernels

- Data
interpolation/
extrapolation
etc.

Point-Wise Nonlinear Kernels

- Flux functions,
Riemann
solvers etc.

PyFR

- For matrix multiply kernels it is pretty easy ...

Python Outer Layer (Hardware Independent)

- Setup
- Distributed memory parallelism
- Outer 'for' loop and calls to Hardware Specific Kernels

Matrix Multiply Kernels

- Data interpolation/extrapolation etc.

Point-Wise Nonlinear Kernels

- Flux functions, Riemann solvers etc.

Use DGEMM from
vendor supplied
BLAS

PyFR

- Harder for point-wise nonlinear kernels ...

Python Outer Layer (Hardware Independent)

- Setup
- Distributed memory parallelism
- Outer 'for' loop and calls to Hardware Specific Kernels

Matrix Multiply Kernels

- Data interpolation/extrapolation etc.

Point-Wise Nonlinear Kernels

- Flux functions, Riemann solvers etc.

Use DGEMM from
vendor supplied
BLAS

Pass Mako
derived kernel
templates through
Mako derived
templating engine

PyFR

- These can now be called

Python Outer Layer (Hardware Independent)

- Setup
- Distributed memory parallelism
- Outer 'for' loop and calls to Hardware Specific Kernels

Matrix Multiply Kernels

- Data interpolation/extrapolation etc.

Point-Wise Nonlinear Kernels

- Flux functions, Riemann solvers etc.

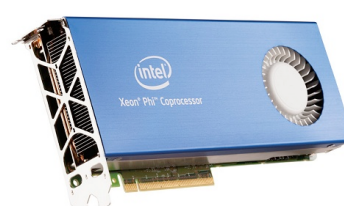
C/OpenMP Hardware Specific Kernels



CUDA Hardware Specific Kernels



PyMIC Hardware Specific Kernels



OpenCL Hardware Specific Kernels

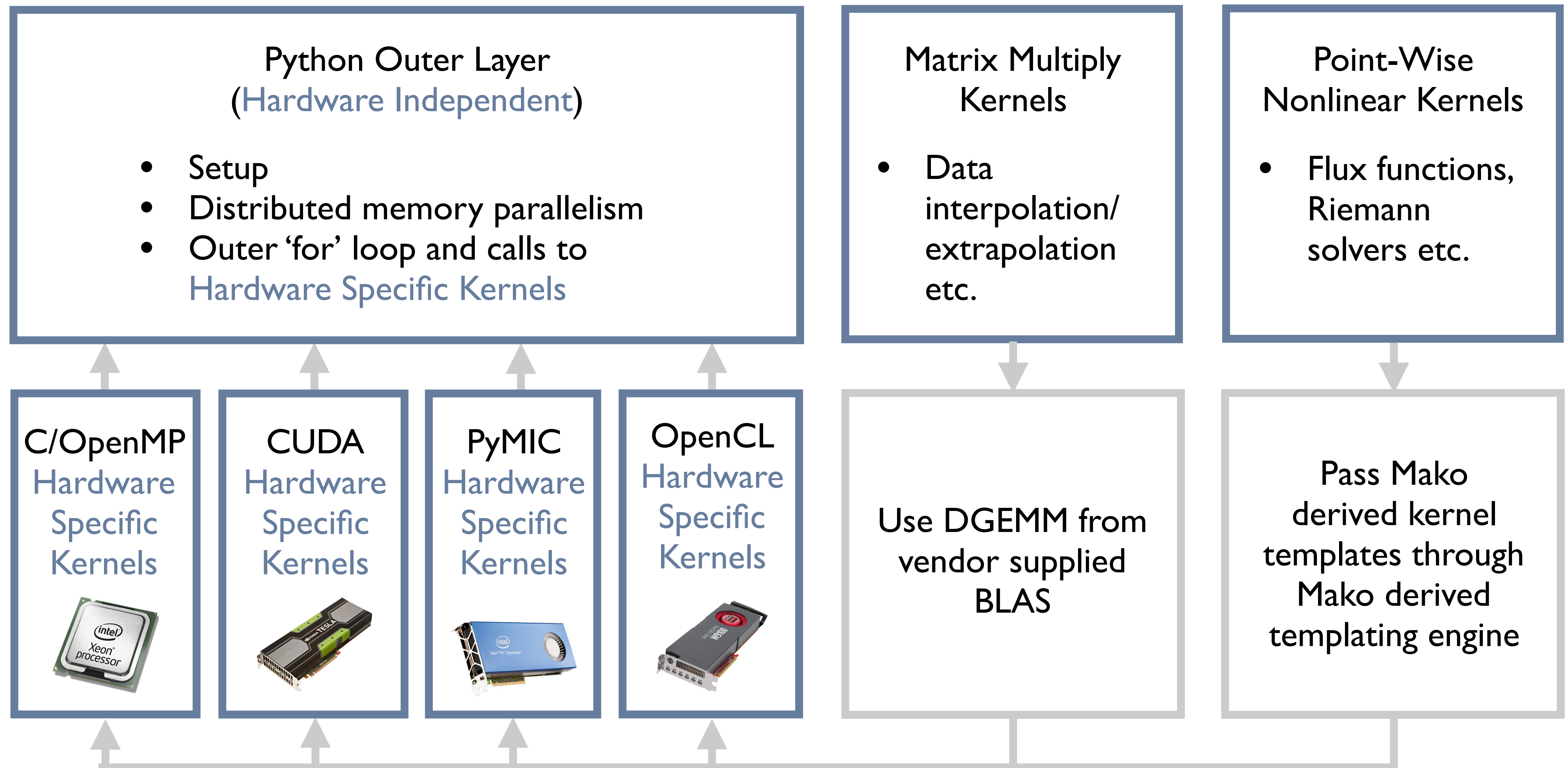


Use DGEMM from
vendor supplied
BLAS

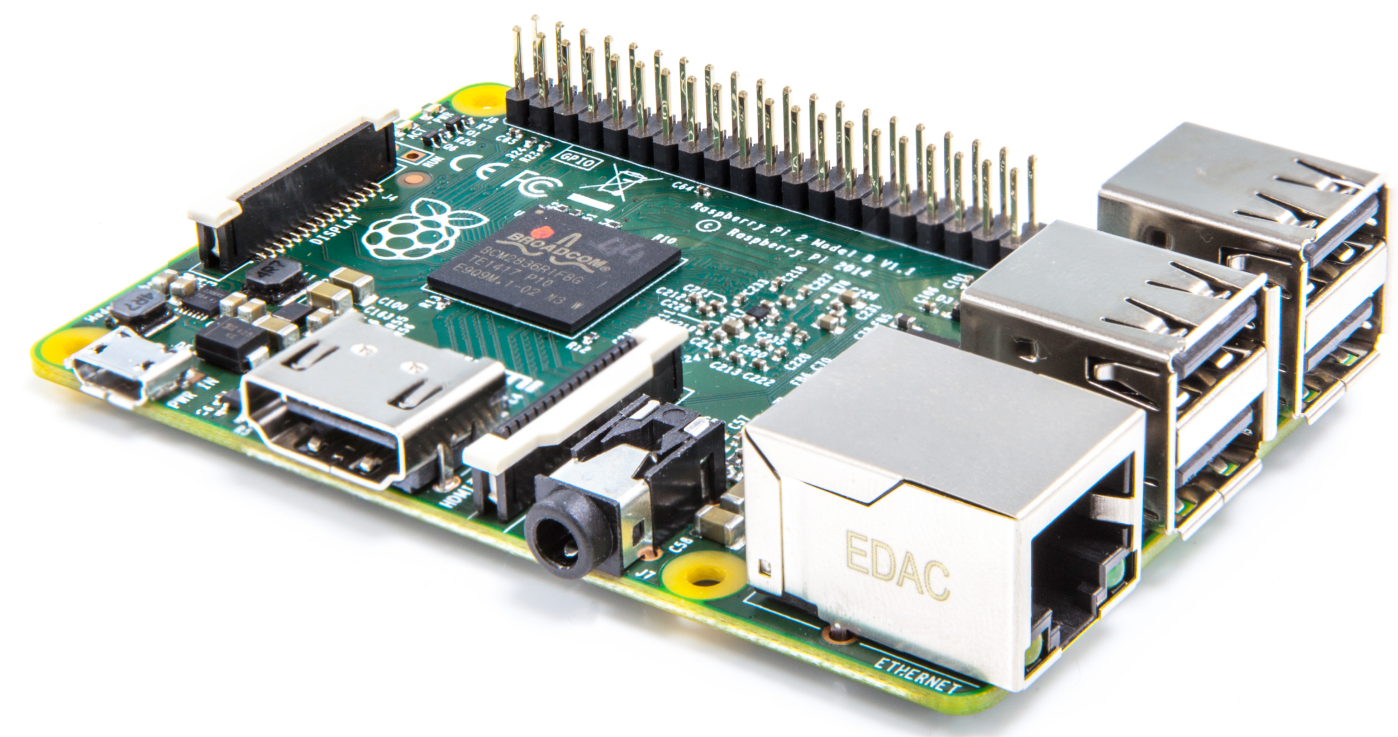
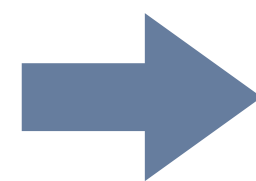
Pass Mako
derived kernel
templates through
Mako derived
templating engine

PyFR

- These can now be called

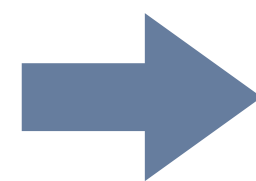


PyFR



Raspberry Pi

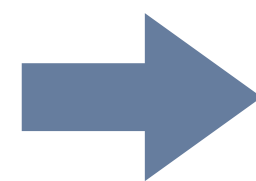
PyFR



Macbook Pro

PyFR

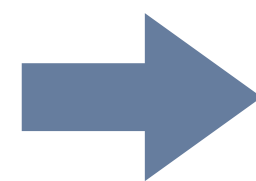
 PyFR



Heterogeneous
Workstation

PyFR

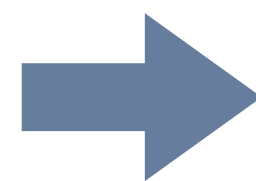
 PyFR



Wilkes (UK)

PyFR

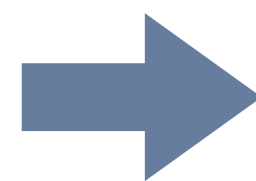
 PyFR



Piz Daint
(Switzerland)

PyFR

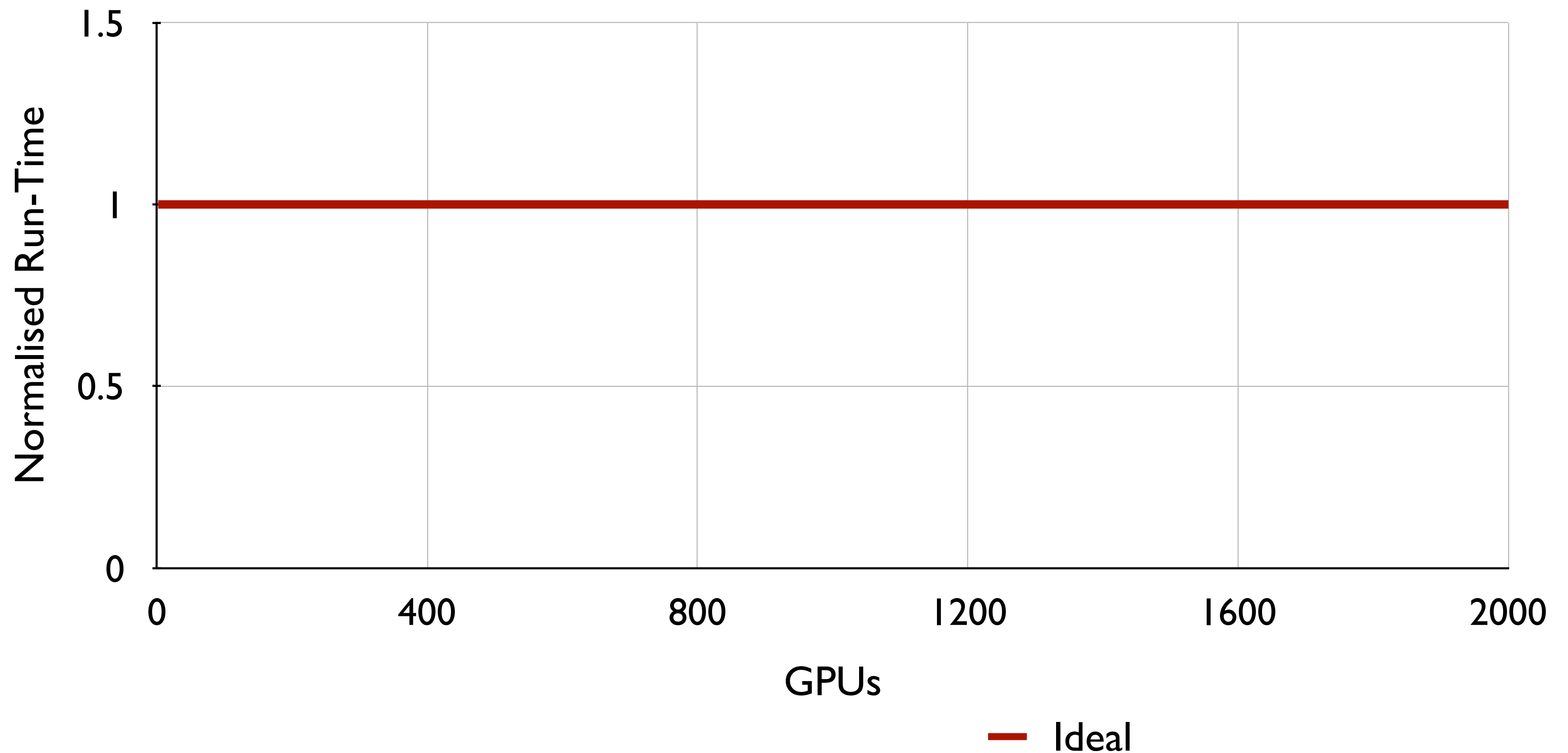
 PyFR



Titan (USA)

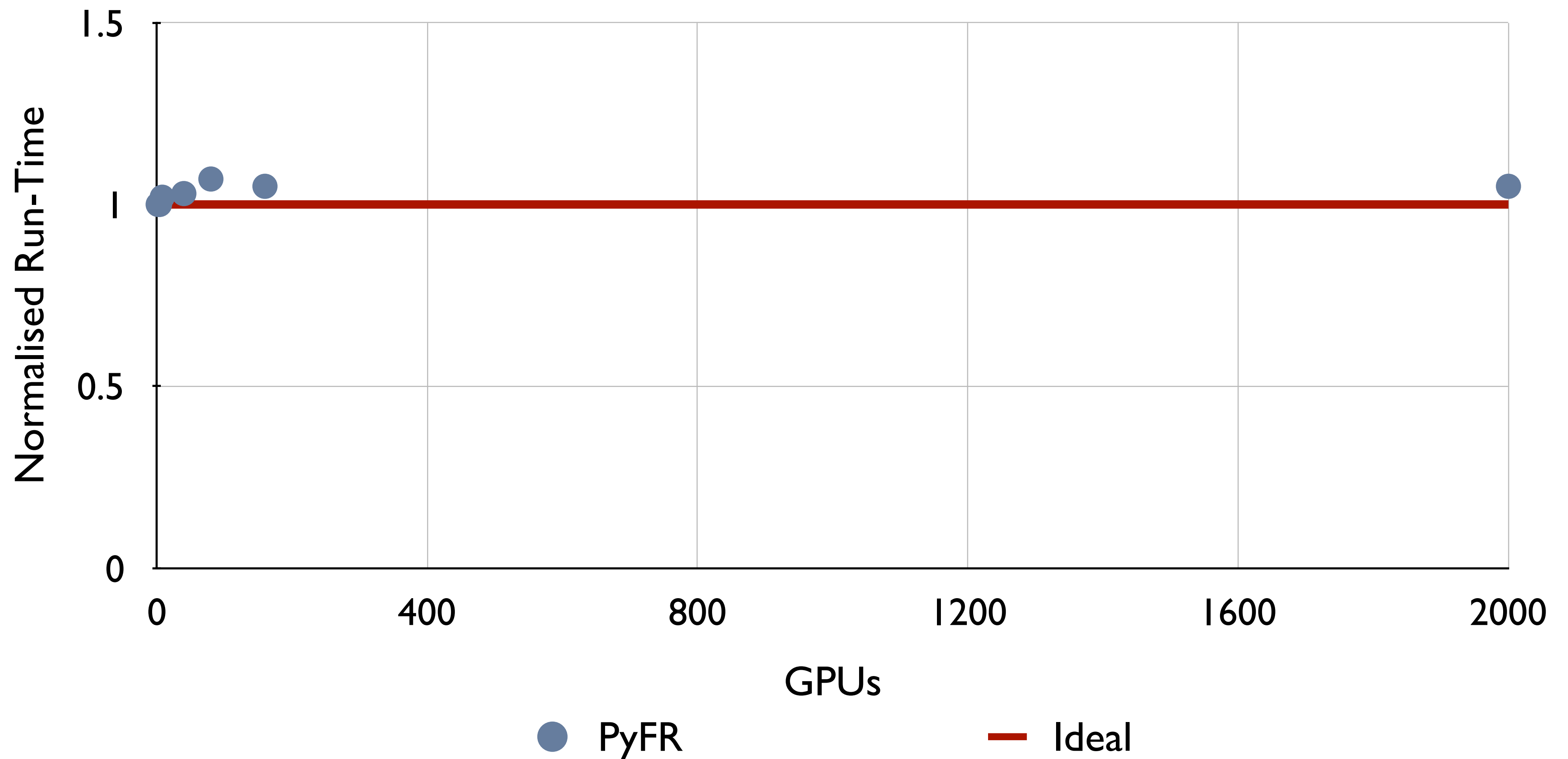
PyFR

- 3D Navier-Stokes **weak** scaling on Piz Daint



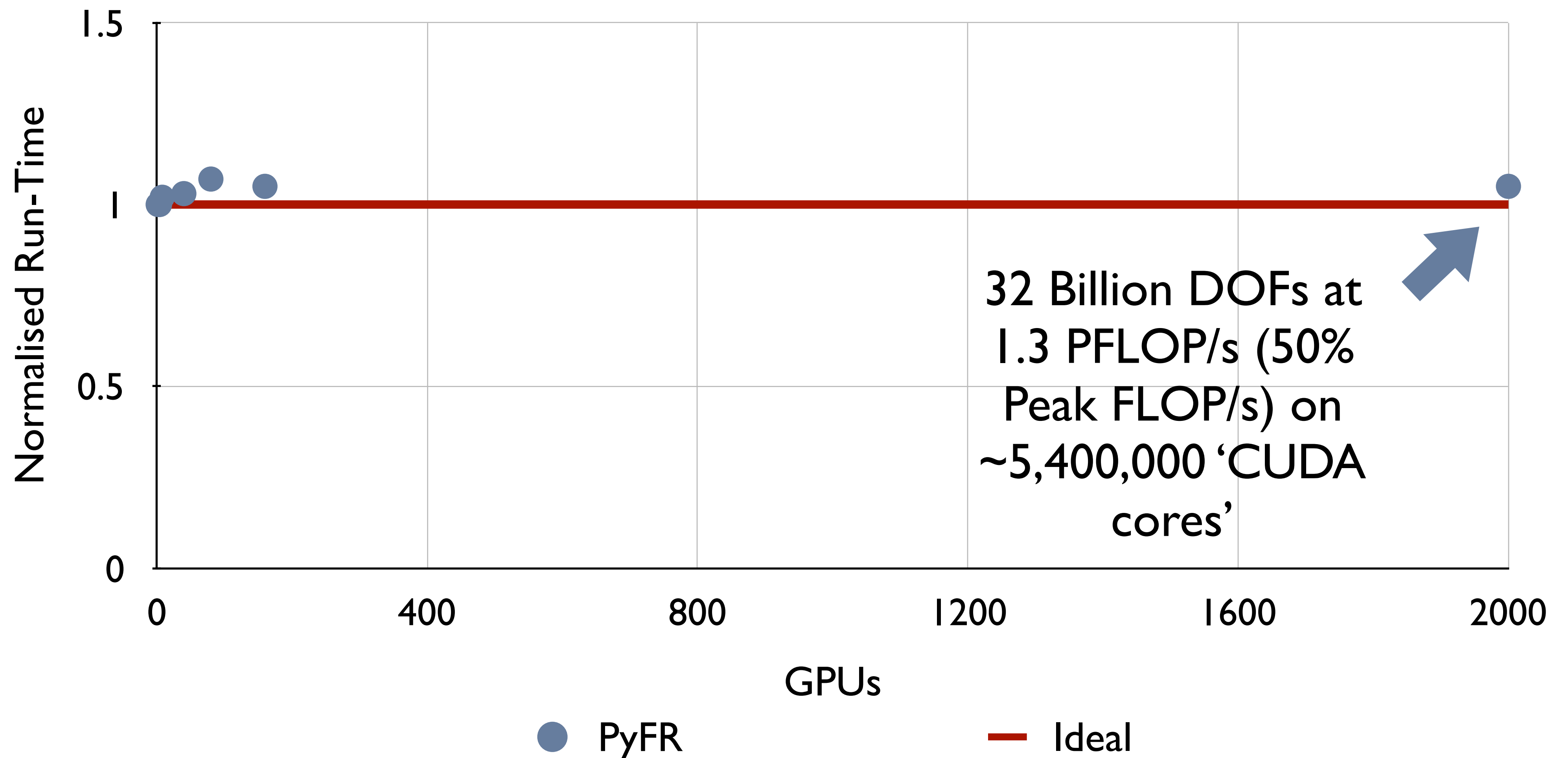
PyFR

- 3D Navier-Stokes **weak** scaling on Piz Daint



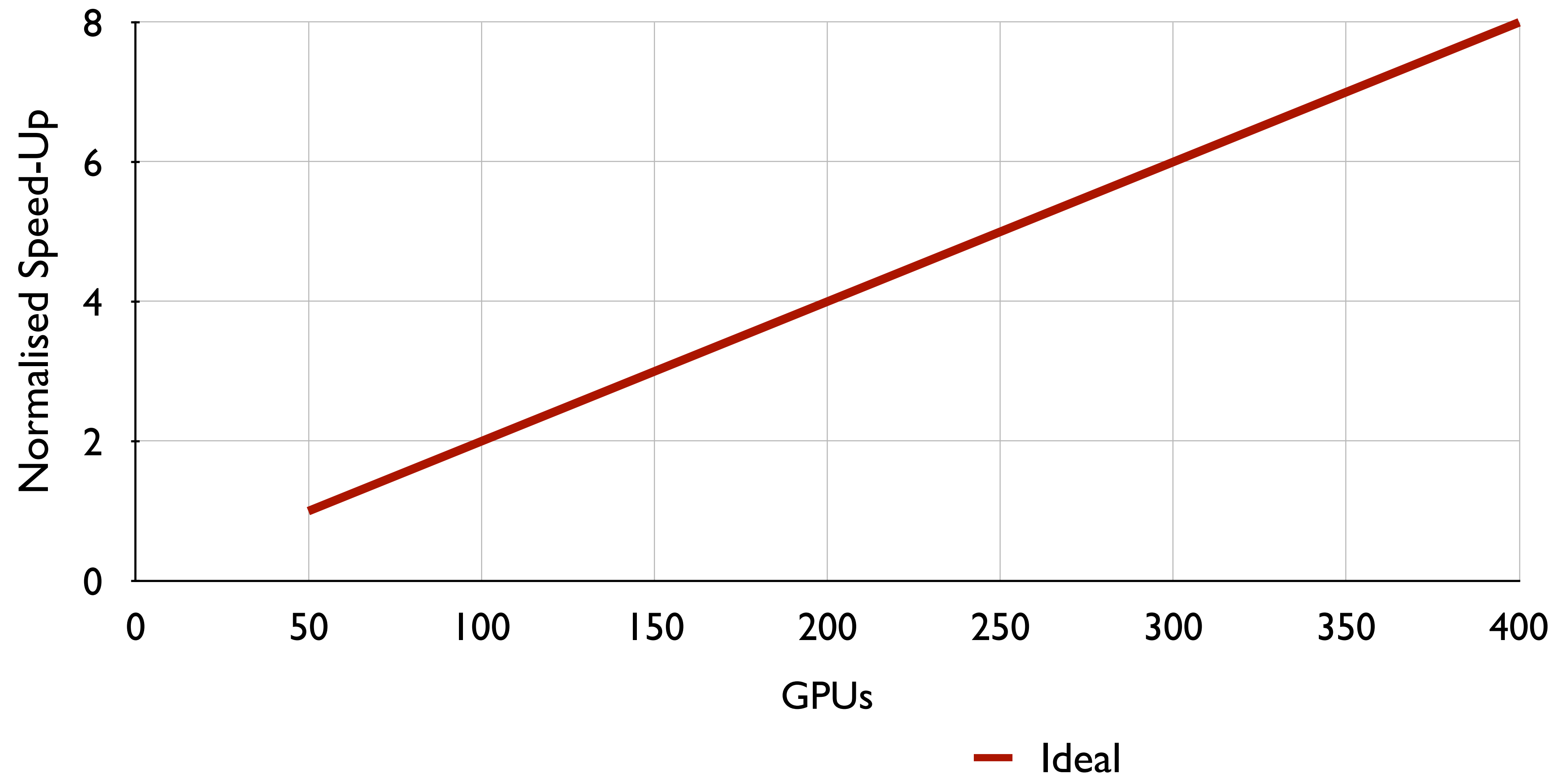
PyFR

- 3D Navier-Stokes **weak** scaling on Piz Daint



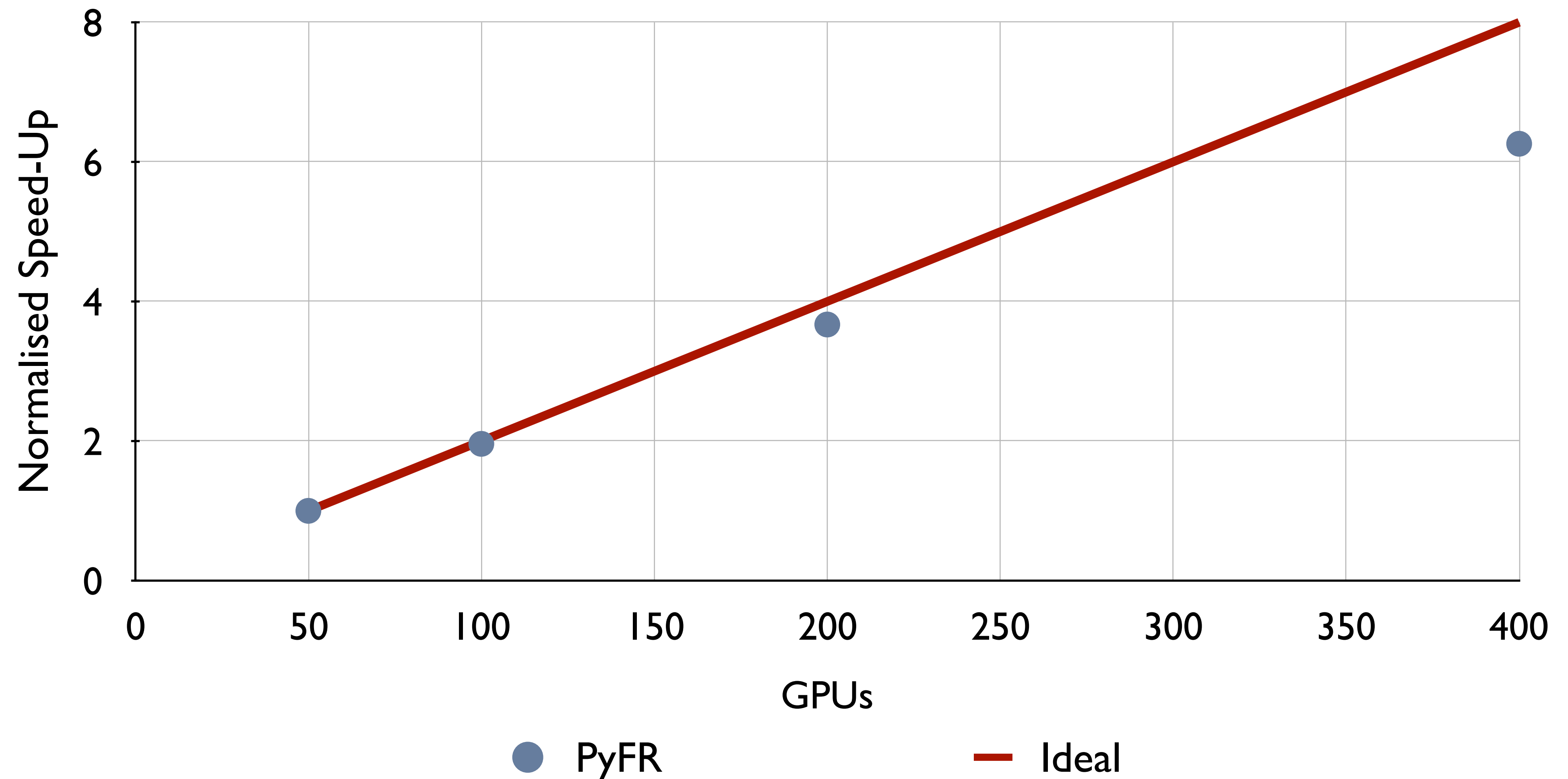
PyFR

- 3D Navier-Stokes **strong** scaling on Piz Daint



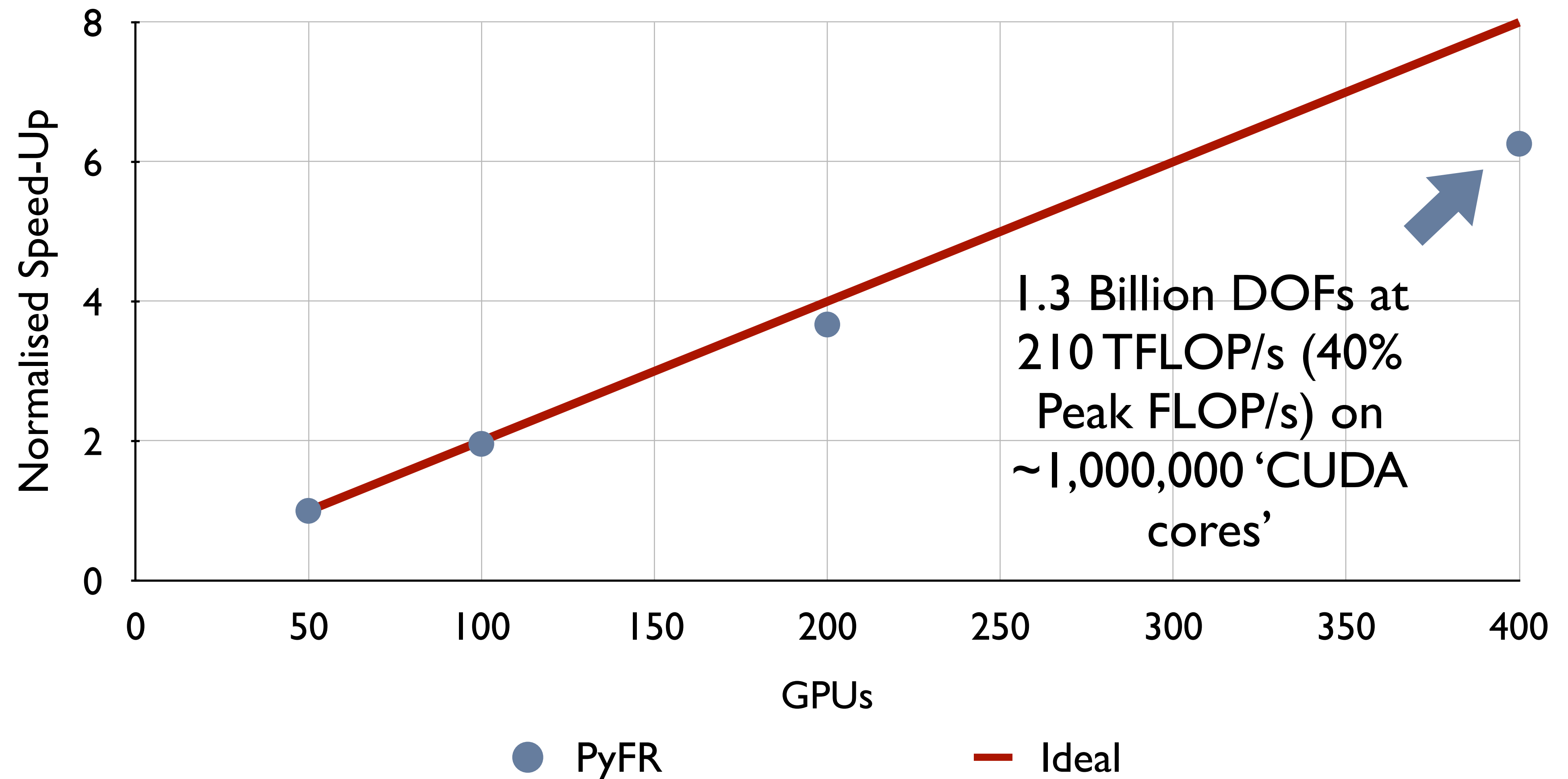
PyFR

- 3D Navier-Stokes **strong** scaling on Piz Daint



PyFR

- 3D Navier-Stokes **strong** scaling on Piz Daint

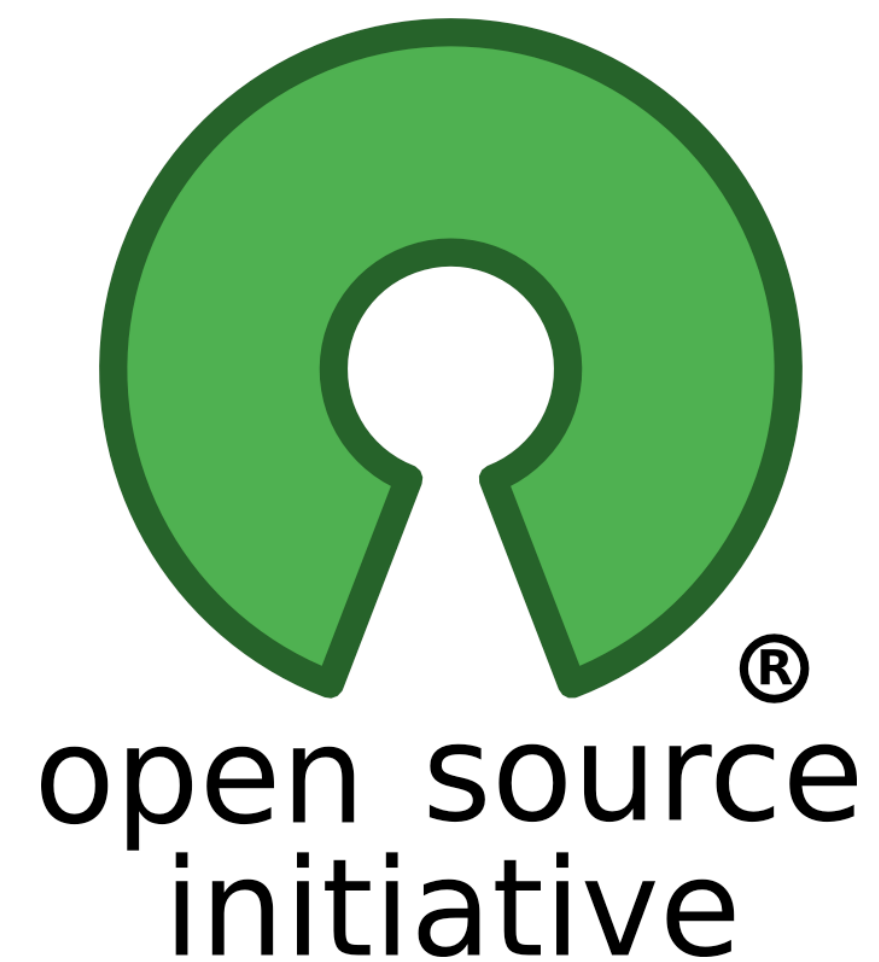


PyFR

- ~8.0k lines of code

PyFR

- Open source '3 Clause New Style BSD License'



PyFR



- Website: www.pyfr.org
- Twitter: [@PyFR_Solver](https://twitter.com/PyFR_Solver)
- Paper: [Computer Physics Communications](#) [8]

[8] F. D. Witherden et al. PyFR: An Open Source Framework for Solving Advection-Diffusion Type Problems on Streaming Architectures using the Flux Reconstruction Approach. *Computer Physics Communications*. 2014



Team



Brian Vermeire



Jin Seok Park



Lorenza Grechy



Freddie Witherden



Arvind Iyer



George Ntemos



Francesco Iori



Antony Farrington



Niki Loppi

Funding

EPSRC

Pioneering research
and skills

Innovate UK
Technology Strategy Board



BAE SYSTEMS

